

NVIDIA Performance Primitives (NPP)

Version 7.5

July 17, 2015

Contents

1	NVIDIA Performance Primitives	1
1.1	What is NPP?	2
1.2	Documentation	2
1.3	Technical Specifications	2
1.4	Files	3
1.4.1	Header Files	3
1.4.2	Library Files	3
1.5	Supported NVIDIA Hardware	4
2	General API Conventions	5
2.1	Memory Management	6
2.1.1	Scratch Buffer and Host Pointer	6
2.2	Function Naming	7
2.3	Integer Result Scaling	7
2.4	Rounding Modes	8
2.4.1	Rounding Mode Parameter	8
3	Signal-Processing Specific API Conventions	9
3.1	Signal Data	10
3.1.1	Parameter Names for Signal Data	10
3.1.1.1	Source Signal Pointer	10
3.1.1.2	Destination Signal Pointer	10
3.1.1.3	In-Place Signal Pointer	10
3.1.2	Signal Data Alignment Requirements	11
3.1.3	Signal Data Related Error Codes	11
3.2	Signal Length	11
3.2.1	Length Related Error Codes	11
4	Imaging-Processing Specific API Conventions	13

4.1	Function Naming	14
4.2	Image Data	14
4.2.1	Line Step	15
4.2.2	Parameter Names for Image Data	15
4.2.2.1	Passing Source-Image Data	15
4.2.2.2	Passing Destination-Image Data	16
4.2.2.3	Passing In-Place Image Data	18
4.2.2.4	Passing Mask-Image Data	18
4.2.2.5	Passing Channel-of-Interest Data	18
4.2.3	Image Data Alignment Requirements	18
4.2.4	Image Data Related Error Codes	19
4.3	Region-of-Interest (ROI)	19
4.3.1	ROI Related Error Codes	19
4.4	Masked Operation	20
4.5	Channel-of-Interest API	20
4.5.1	Select-Channel Source-Image Pointer	20
4.5.2	Select-Channel Source-Image	20
4.5.3	Select-Channel Destination-Image Pointer	20
4.6	Source-Image Sampling	21
4.6.1	Point-Wise Operations	21
4.6.2	Neighborhood Operations	21
4.6.2.1	Mask-Size Parameter	21
4.6.2.2	Anchor-Point Parameter	22
4.6.2.3	Sampling Beyond Image Boundaries	22
5	Module Index	23
5.1	Modules	23
6	Data Structure Index	29
6.1	Data Structures	29
7	Module Documentation	31
7.1	NPP Core	31
7.1.1	Detailed Description	31
7.1.2	Function Documentation	32
7.1.2.1	nppGetGpuComputeCapability	32
7.1.2.2	nppGetGpuName	32
7.1.2.3	nppGetGpuNumSMs	32

7.1.2.4	nppGetLibVersion	32
7.1.2.5	nppGetMaxThreadsPerBlock	32
7.1.2.6	nppGetMaxThreadsPerSM	33
7.1.2.7	nppGetStream	33
7.1.2.8	nppSetStream	33
7.2	NPP Type Definitions and Constants	34
7.2.1	Define Documentation	39
7.2.1.1	NPP_MAX_16S	39
7.2.1.2	NPP_MAX_16U	39
7.2.1.3	NPP_MAX_32S	40
7.2.1.4	NPP_MAX_32U	40
7.2.1.5	NPP_MAX_64S	40
7.2.1.6	NPP_MAX_64U	40
7.2.1.7	NPP_MAX_8S	40
7.2.1.8	NPP_MAX_8U	40
7.2.1.9	NPP_MAXABS_32F	40
7.2.1.10	NPP_MAXABS_64F	40
7.2.1.11	NPP_MIN_16S	40
7.2.1.12	NPP_MIN_16U	40
7.2.1.13	NPP_MIN_32S	40
7.2.1.14	NPP_MIN_32U	41
7.2.1.15	NPP_MIN_64S	41
7.2.1.16	NPP_MIN_64U	41
7.2.1.17	NPP_MIN_8S	41
7.2.1.18	NPP_MIN_8U	41
7.2.1.19	NPP_MINABS_32F	41
7.2.1.20	NPP_MINABS_64F	41
7.2.2	Enumeration Type Documentation	41
7.2.2.1	NppCmpOp	41
7.2.2.2	NppGpuComputeCapability	41
7.2.2.3	NppHintAlgorithm	42
7.2.2.4	NppiAlphaOp	42
7.2.2.5	NppiAxis	43
7.2.2.6	NppiBayerGridPosition	43
7.2.2.7	NppiBorderType	43
7.2.2.8	NppiHuffmanTableType	43

7.2.2.9	NppiInterpolationMode	43
7.2.2.10	NppiMaskSize	44
7.2.2.11	NppRoundMode	44
7.2.2.12	NppStatus	45
7.2.2.13	NppsZCType	47
7.3	Basic NPP Data Types	48
7.3.1	Typedef Documentation	49
7.3.1.1	Npp16s	49
7.3.1.2	Npp16u	49
7.3.1.3	Npp32f	49
7.3.1.4	Npp32fc	49
7.3.1.5	Npp32s	49
7.3.1.6	Npp32sc	50
7.3.1.7	Npp32u	50
7.3.1.8	Npp32uc	50
7.3.1.9	Npp64f	50
7.3.1.10	Npp64fc	50
7.3.1.11	Npp64s	50
7.3.1.12	Npp64sc	50
7.3.1.13	Npp64u	50
7.3.1.14	Npp8s	50
7.3.1.15	Npp8u	50
7.3.2	Function Documentation	50
7.3.2.1	__align__	50
7.3.2.2	__align__	51
7.3.3	Variable Documentation	51
7.3.3.1	Npp16sc	51
7.3.3.2	Npp16uc	51
7.3.3.3	Npp8uc	51
7.4	NPP Image Processing	52
7.5	Arithmetic and Logical Operations	53
7.6	Arithmetic Operations	54
7.7	AddC	56
7.7.1	Detailed Description	61
7.7.2	Function Documentation	61
7.7.2.1	nppiAddC_16s_AC4IRSfs	61

7.7.2.2	nppiAddC_16s_AC4RSfs	61
7.7.2.3	nppiAddC_16s_C1IRSfs	61
7.7.2.4	nppiAddC_16s_C1RSfs	62
7.7.2.5	nppiAddC_16s_C3IRSfs	62
7.7.2.6	nppiAddC_16s_C3RSfs	63
7.7.2.7	nppiAddC_16s_C4IRSfs	63
7.7.2.8	nppiAddC_16s_C4RSfs	63
7.7.2.9	nppiAddC_16sc_AC4IRSfs	64
7.7.2.10	nppiAddC_16sc_AC4RSfs	64
7.7.2.11	nppiAddC_16sc_C1IRSfs	65
7.7.2.12	nppiAddC_16sc_C1RSfs	65
7.7.2.13	nppiAddC_16sc_C3IRSfs	65
7.7.2.14	nppiAddC_16sc_C3RSfs	66
7.7.2.15	nppiAddC_16u_AC4IRSfs	66
7.7.2.16	nppiAddC_16u_AC4RSfs	67
7.7.2.17	nppiAddC_16u_C1IRSfs	67
7.7.2.18	nppiAddC_16u_C1RSfs	67
7.7.2.19	nppiAddC_16u_C3IRSfs	68
7.7.2.20	nppiAddC_16u_C3RSfs	68
7.7.2.21	nppiAddC_16u_C4IRSfs	69
7.7.2.22	nppiAddC_16u_C4RSfs	69
7.7.2.23	nppiAddC_32f_AC4IR	69
7.7.2.24	nppiAddC_32f_AC4R	70
7.7.2.25	nppiAddC_32f_C1IR	70
7.7.2.26	nppiAddC_32f_C1R	70
7.7.2.27	nppiAddC_32f_C3IR	71
7.7.2.28	nppiAddC_32f_C3R	71
7.7.2.29	nppiAddC_32f_C4IR	71
7.7.2.30	nppiAddC_32f_C4R	72
7.7.2.31	nppiAddC_32fc_AC4IR	72
7.7.2.32	nppiAddC_32fc_AC4R	72
7.7.2.33	nppiAddC_32fc_C1IR	73
7.7.2.34	nppiAddC_32fc_C1R	73
7.7.2.35	nppiAddC_32fc_C3IR	73
7.7.2.36	nppiAddC_32fc_C3R	74
7.7.2.37	nppiAddC_32fc_C4IR	74

7.7.2.38	<code>nppiAddC_32fc_C4R</code>	74
7.7.2.39	<code>nppiAddC_32s_C1IRSfs</code>	75
7.7.2.40	<code>nppiAddC_32s_C1RSfs</code>	75
7.7.2.41	<code>nppiAddC_32s_C3IRSfs</code>	75
7.7.2.42	<code>nppiAddC_32s_C3RSfs</code>	76
7.7.2.43	<code>nppiAddC_32sc_AC4IRSfs</code>	76
7.7.2.44	<code>nppiAddC_32sc_AC4RSfs</code>	77
7.7.2.45	<code>nppiAddC_32sc_C1IRSfs</code>	77
7.7.2.46	<code>nppiAddC_32sc_C1RSfs</code>	77
7.7.2.47	<code>nppiAddC_32sc_C3IRSfs</code>	78
7.7.2.48	<code>nppiAddC_32sc_C3RSfs</code>	78
7.7.2.49	<code>nppiAddC_8u_AC4IRSfs</code>	79
7.7.2.50	<code>nppiAddC_8u_AC4RSfs</code>	79
7.7.2.51	<code>nppiAddC_8u_C1IRSfs</code>	79
7.7.2.52	<code>nppiAddC_8u_C1RSfs</code>	80
7.7.2.53	<code>nppiAddC_8u_C3IRSfs</code>	80
7.7.2.54	<code>nppiAddC_8u_C3RSfs</code>	80
7.7.2.55	<code>nppiAddC_8u_C4IRSfs</code>	81
7.7.2.56	<code>nppiAddC_8u_C4RSfs</code>	81
7.8	MulC	82
7.8.1	Detailed Description	87
7.8.2	Function Documentation	87
7.8.2.1	<code>nppiMulC_16s_AC4IRSfs</code>	87
7.8.2.2	<code>nppiMulC_16s_AC4RSfs</code>	87
7.8.2.3	<code>nppiMulC_16s_C1IRSfs</code>	88
7.8.2.4	<code>nppiMulC_16s_C1RSfs</code>	88
7.8.2.5	<code>nppiMulC_16s_C3IRSfs</code>	88
7.8.2.6	<code>nppiMulC_16s_C3RSfs</code>	89
7.8.2.7	<code>nppiMulC_16s_C4IRSfs</code>	89
7.8.2.8	<code>nppiMulC_16s_C4RSfs</code>	89
7.8.2.9	<code>nppiMulC_16sc_AC4IRSfs</code>	90
7.8.2.10	<code>nppiMulC_16sc_AC4RSfs</code>	90
7.8.2.11	<code>nppiMulC_16sc_C1IRSfs</code>	91
7.8.2.12	<code>nppiMulC_16sc_C1RSfs</code>	91
7.8.2.13	<code>nppiMulC_16sc_C3IRSfs</code>	91
7.8.2.14	<code>nppiMulC_16sc_C3RSfs</code>	92

7.8.2.15	nppiMulC_16u_AC4IRSfs	92
7.8.2.16	nppiMulC_16u_AC4RSfs	93
7.8.2.17	nppiMulC_16u_C1IRSfs	93
7.8.2.18	nppiMulC_16u_C1RSfs	93
7.8.2.19	nppiMulC_16u_C3IRSfs	94
7.8.2.20	nppiMulC_16u_C3RSfs	94
7.8.2.21	nppiMulC_16u_C4IRSfs	95
7.8.2.22	nppiMulC_16u_C4RSfs	95
7.8.2.23	nppiMulC_32f_AC4IR	95
7.8.2.24	nppiMulC_32f_AC4R	96
7.8.2.25	nppiMulC_32f_C1IR	96
7.8.2.26	nppiMulC_32f_C1R	96
7.8.2.27	nppiMulC_32f_C3IR	97
7.8.2.28	nppiMulC_32f_C3R	97
7.8.2.29	nppiMulC_32f_C4IR	97
7.8.2.30	nppiMulC_32f_C4R	98
7.8.2.31	nppiMulC_32fc_AC4IR	98
7.8.2.32	nppiMulC_32fc_AC4R	98
7.8.2.33	nppiMulC_32fc_C1IR	99
7.8.2.34	nppiMulC_32fc_C1R	99
7.8.2.35	nppiMulC_32fc_C3IR	99
7.8.2.36	nppiMulC_32fc_C3R	100
7.8.2.37	nppiMulC_32fc_C4IR	100
7.8.2.38	nppiMulC_32fc_C4R	100
7.8.2.39	nppiMulC_32s_C1IRSfs	101
7.8.2.40	nppiMulC_32s_C1RSfs	101
7.8.2.41	nppiMulC_32s_C3IRSfs	101
7.8.2.42	nppiMulC_32s_C3RSfs	102
7.8.2.43	nppiMulC_32sc_AC4IRSfs	102
7.8.2.44	nppiMulC_32sc_AC4RSfs	103
7.8.2.45	nppiMulC_32sc_C1IRSfs	103
7.8.2.46	nppiMulC_32sc_C1RSfs	103
7.8.2.47	nppiMulC_32sc_C3IRSfs	104
7.8.2.48	nppiMulC_32sc_C3RSfs	104
7.8.2.49	nppiMulC_8u_AC4IRSfs	105
7.8.2.50	nppiMulC_8u_AC4RSfs	105

7.8.2.51	<code>nppiMulC_8u_C1IRSfs</code>	105
7.8.2.52	<code>nppiMulC_8u_C1RSfs</code>	106
7.8.2.53	<code>nppiMulC_8u_C3IRSfs</code>	106
7.8.2.54	<code>nppiMulC_8u_C3RSfs</code>	106
7.8.2.55	<code>nppiMulC_8u_C4IRSfs</code>	107
7.8.2.56	<code>nppiMulC_8u_C4RSfs</code>	107
7.9	<code>MulCScale</code>	108
7.9.1	Detailed Description	109
7.9.2	Function Documentation	109
7.9.2.1	<code>nppiMulCScale_16u_AC4IR</code>	109
7.9.2.2	<code>nppiMulCScale_16u_AC4R</code>	110
7.9.2.3	<code>nppiMulCScale_16u_C1IR</code>	110
7.9.2.4	<code>nppiMulCScale_16u_C1R</code>	110
7.9.2.5	<code>nppiMulCScale_16u_C3IR</code>	111
7.9.2.6	<code>nppiMulCScale_16u_C3R</code>	111
7.9.2.7	<code>nppiMulCScale_16u_C4IR</code>	111
7.9.2.8	<code>nppiMulCScale_16u_C4R</code>	112
7.9.2.9	<code>nppiMulCScale_8u_AC4IR</code>	112
7.9.2.10	<code>nppiMulCScale_8u_AC4R</code>	112
7.9.2.11	<code>nppiMulCScale_8u_C1IR</code>	113
7.9.2.12	<code>nppiMulCScale_8u_C1R</code>	113
7.9.2.13	<code>nppiMulCScale_8u_C3IR</code>	113
7.9.2.14	<code>nppiMulCScale_8u_C3R</code>	114
7.9.2.15	<code>nppiMulCScale_8u_C4IR</code>	114
7.9.2.16	<code>nppiMulCScale_8u_C4R</code>	114
7.10	<code>SubC</code>	115
7.10.1	Detailed Description	120
7.10.2	Function Documentation	120
7.10.2.1	<code>nppiSubC_16s_AC4IRSfs</code>	120
7.10.2.2	<code>nppiSubC_16s_AC4RSfs</code>	120
7.10.2.3	<code>nppiSubC_16s_C1IRSfs</code>	120
7.10.2.4	<code>nppiSubC_16s_C1RSfs</code>	121
7.10.2.5	<code>nppiSubC_16s_C3IRSfs</code>	121
7.10.2.6	<code>nppiSubC_16s_C3RSfs</code>	122
7.10.2.7	<code>nppiSubC_16s_C4IRSfs</code>	122
7.10.2.8	<code>nppiSubC_16s_C4RSfs</code>	122

7.10.2.9	nppiSubC_16sc_AC4IRSfs	123
7.10.2.10	nppiSubC_16sc_AC4RSfs	123
7.10.2.11	nppiSubC_16sc_C1IRSfs	124
7.10.2.12	nppiSubC_16sc_C1RSfs	124
7.10.2.13	nppiSubC_16sc_C3IRSfs	124
7.10.2.14	nppiSubC_16sc_C3RSfs	125
7.10.2.15	nppiSubC_16u_AC4IRSfs	125
7.10.2.16	nppiSubC_16u_AC4RSfs	126
7.10.2.17	nppiSubC_16u_C1IRSfs	126
7.10.2.18	nppiSubC_16u_C1RSfs	126
7.10.2.19	nppiSubC_16u_C3IRSfs	127
7.10.2.20	nppiSubC_16u_C3RSfs	127
7.10.2.21	nppiSubC_16u_C4IRSfs	128
7.10.2.22	nppiSubC_16u_C4RSfs	128
7.10.2.23	nppiSubC_32f_AC4IR	128
7.10.2.24	nppiSubC_32f_AC4R	129
7.10.2.25	nppiSubC_32f_C1IR	129
7.10.2.26	nppiSubC_32f_C1R	129
7.10.2.27	nppiSubC_32f_C3IR	130
7.10.2.28	nppiSubC_32f_C3R	130
7.10.2.29	nppiSubC_32f_C4IR	130
7.10.2.30	nppiSubC_32f_C4R	131
7.10.2.31	nppiSubC_32fc_AC4IR	131
7.10.2.32	nppiSubC_32fc_AC4R	131
7.10.2.33	nppiSubC_32fc_C1IR	132
7.10.2.34	nppiSubC_32fc_C1R	132
7.10.2.35	nppiSubC_32fc_C3IR	132
7.10.2.36	nppiSubC_32fc_C3R	133
7.10.2.37	nppiSubC_32fc_C4IR	133
7.10.2.38	nppiSubC_32fc_C4R	133
7.10.2.39	nppiSubC_32s_C1IRSfs	134
7.10.2.40	nppiSubC_32s_C1RSfs	134
7.10.2.41	nppiSubC_32s_C3IRSfs	134
7.10.2.42	nppiSubC_32s_C3RSfs	135
7.10.2.43	nppiSubC_32sc_AC4IRSfs	135
7.10.2.44	nppiSubC_32sc_AC4RSfs	136

7.10.2.45	<code>nppiSubC_32sc_C1IRSfs</code>	136
7.10.2.46	<code>nppiSubC_32sc_C1RSfs</code>	136
7.10.2.47	<code>nppiSubC_32sc_C3IRSfs</code>	137
7.10.2.48	<code>nppiSubC_32sc_C3RSfs</code>	137
7.10.2.49	<code>nppiSubC_8u_AC4IRSfs</code>	138
7.10.2.50	<code>nppiSubC_8u_AC4RSfs</code>	138
7.10.2.51	<code>nppiSubC_8u_C1IRSfs</code>	138
7.10.2.52	<code>nppiSubC_8u_C1RSfs</code>	139
7.10.2.53	<code>nppiSubC_8u_C3IRSfs</code>	139
7.10.2.54	<code>nppiSubC_8u_C3RSfs</code>	139
7.10.2.55	<code>nppiSubC_8u_C4IRSfs</code>	140
7.10.2.56	<code>nppiSubC_8u_C4RSfs</code>	140
7.11	<code>DivC</code>	141
7.11.1	Detailed Description	146
7.11.2	Function Documentation	146
7.11.2.1	<code>nppiDivC_16s_AC4IRSfs</code>	146
7.11.2.2	<code>nppiDivC_16s_AC4RSfs</code>	146
7.11.2.3	<code>nppiDivC_16s_C1IRSfs</code>	147
7.11.2.4	<code>nppiDivC_16s_C1RSfs</code>	147
7.11.2.5	<code>nppiDivC_16s_C3IRSfs</code>	147
7.11.2.6	<code>nppiDivC_16s_C3RSfs</code>	148
7.11.2.7	<code>nppiDivC_16s_C4IRSfs</code>	148
7.11.2.8	<code>nppiDivC_16s_C4RSfs</code>	148
7.11.2.9	<code>nppiDivC_16sc_AC4IRSfs</code>	149
7.11.2.10	<code>nppiDivC_16sc_AC4RSfs</code>	149
7.11.2.11	<code>nppiDivC_16sc_C1IRSfs</code>	150
7.11.2.12	<code>nppiDivC_16sc_C1RSfs</code>	150
7.11.2.13	<code>nppiDivC_16sc_C3IRSfs</code>	150
7.11.2.14	<code>nppiDivC_16sc_C3RSfs</code>	151
7.11.2.15	<code>nppiDivC_16u_AC4IRSfs</code>	151
7.11.2.16	<code>nppiDivC_16u_AC4RSfs</code>	152
7.11.2.17	<code>nppiDivC_16u_C1IRSfs</code>	152
7.11.2.18	<code>nppiDivC_16u_C1RSfs</code>	152
7.11.2.19	<code>nppiDivC_16u_C3IRSfs</code>	153
7.11.2.20	<code>nppiDivC_16u_C3RSfs</code>	153
7.11.2.21	<code>nppiDivC_16u_C4IRSfs</code>	154

7.11.2.22 nppiDivC_16u_C4RSfs	154
7.11.2.23 nppiDivC_32f_AC4IR	154
7.11.2.24 nppiDivC_32f_AC4R	155
7.11.2.25 nppiDivC_32f_C1IR	155
7.11.2.26 nppiDivC_32f_C1R	155
7.11.2.27 nppiDivC_32f_C3IR	156
7.11.2.28 nppiDivC_32f_C3R	156
7.11.2.29 nppiDivC_32f_C4IR	156
7.11.2.30 nppiDivC_32f_C4R	157
7.11.2.31 nppiDivC_32fc_AC4IR	157
7.11.2.32 nppiDivC_32fc_AC4R	157
7.11.2.33 nppiDivC_32fc_C1IR	158
7.11.2.34 nppiDivC_32fc_C1R	158
7.11.2.35 nppiDivC_32fc_C3IR	158
7.11.2.36 nppiDivC_32fc_C3R	159
7.11.2.37 nppiDivC_32fc_C4IR	159
7.11.2.38 nppiDivC_32fc_C4R	159
7.11.2.39 nppiDivC_32s_C1IRSfs	160
7.11.2.40 nppiDivC_32s_C1RSfs	160
7.11.2.41 nppiDivC_32s_C3IRSfs	160
7.11.2.42 nppiDivC_32s_C3RSfs	161
7.11.2.43 nppiDivC_32sc_AC4IRSfs	161
7.11.2.44 nppiDivC_32sc_AC4RSfs	162
7.11.2.45 nppiDivC_32sc_C1IRSfs	162
7.11.2.46 nppiDivC_32sc_C1RSfs	162
7.11.2.47 nppiDivC_32sc_C3IRSfs	163
7.11.2.48 nppiDivC_32sc_C3RSfs	163
7.11.2.49 nppiDivC_8u_AC4IRSfs	164
7.11.2.50 nppiDivC_8u_AC4RSfs	164
7.11.2.51 nppiDivC_8u_C1IRSfs	164
7.11.2.52 nppiDivC_8u_C1RSfs	165
7.11.2.53 nppiDivC_8u_C3IRSfs	165
7.11.2.54 nppiDivC_8u_C3RSfs	165
7.11.2.55 nppiDivC_8u_C4IRSfs	166
7.11.2.56 nppiDivC_8u_C4RSfs	166
7.12 AbsDiffC	167

7.12.1	Detailed Description	167
7.12.2	Function Documentation	167
7.12.2.1	npplAbsDiffC_16u_C1R	167
7.12.2.2	npplAbsDiffC_32f_C1R	167
7.12.2.3	npplAbsDiffC_8u_C1R	168
7.13	Add	169
7.13.1	Detailed Description	174
7.13.2	Function Documentation	174
7.13.2.1	npplAdd_16s_AC4IRSfs	174
7.13.2.2	npplAdd_16s_AC4RSfs	174
7.13.2.3	npplAdd_16s_C1IRSfs	175
7.13.2.4	npplAdd_16s_C1RSfs	175
7.13.2.5	npplAdd_16s_C3IRSfs	176
7.13.2.6	npplAdd_16s_C3RSfs	176
7.13.2.7	npplAdd_16s_C4IRSfs	177
7.13.2.8	npplAdd_16s_C4RSfs	177
7.13.2.9	npplAdd_16sc_AC4IRSfs	177
7.13.2.10	npplAdd_16sc_AC4RSfs	178
7.13.2.11	npplAdd_16sc_C1IRSfs	178
7.13.2.12	npplAdd_16sc_C1RSfs	179
7.13.2.13	npplAdd_16sc_C3IRSfs	179
7.13.2.14	npplAdd_16sc_C3RSfs	179
7.13.2.15	npplAdd_16u_AC4IRSfs	180
7.13.2.16	npplAdd_16u_AC4RSfs	180
7.13.2.17	npplAdd_16u_C1IRSfs	181
7.13.2.18	npplAdd_16u_C1RSfs	181
7.13.2.19	npplAdd_16u_C3IRSfs	182
7.13.2.20	npplAdd_16u_C3RSfs	182
7.13.2.21	npplAdd_16u_C4IRSfs	182
7.13.2.22	npplAdd_16u_C4RSfs	183
7.13.2.23	npplAdd_32f_AC4IR	183
7.13.2.24	npplAdd_32f_AC4R	184
7.13.2.25	npplAdd_32f_C1IR	184
7.13.2.26	npplAdd_32f_C1R	184
7.13.2.27	npplAdd_32f_C3IR	185
7.13.2.28	npplAdd_32f_C3R	185

7.13.2.29	<code>npplAdd_32f_C4IR</code>	186
7.13.2.30	<code>npplAdd_32f_C4R</code>	186
7.13.2.31	<code>npplAdd_32fc_AC4IR</code>	186
7.13.2.32	<code>npplAdd_32fc_AC4R</code>	187
7.13.2.33	<code>npplAdd_32fc_C1IR</code>	187
7.13.2.34	<code>npplAdd_32fc_C1R</code>	187
7.13.2.35	<code>npplAdd_32fc_C3IR</code>	188
7.13.2.36	<code>npplAdd_32fc_C3R</code>	188
7.13.2.37	<code>npplAdd_32fc_C4IR</code>	189
7.13.2.38	<code>npplAdd_32fc_C4R</code>	189
7.13.2.39	<code>npplAdd_32s_C1IRSfs</code>	189
7.13.2.40	<code>npplAdd_32s_C1R</code>	190
7.13.2.41	<code>npplAdd_32s_C1RSfs</code>	190
7.13.2.42	<code>npplAdd_32s_C3IRSfs</code>	191
7.13.2.43	<code>npplAdd_32s_C3RSfs</code>	191
7.13.2.44	<code>npplAdd_32sc_AC4IRSfs</code>	191
7.13.2.45	<code>npplAdd_32sc_AC4RSfs</code>	192
7.13.2.46	<code>npplAdd_32sc_C1IRSfs</code>	192
7.13.2.47	<code>npplAdd_32sc_C1RSfs</code>	193
7.13.2.48	<code>npplAdd_32sc_C3IRSfs</code>	193
7.13.2.49	<code>npplAdd_32sc_C3RSfs</code>	193
7.13.2.50	<code>npplAdd_8u_AC4IRSfs</code>	194
7.13.2.51	<code>npplAdd_8u_AC4RSfs</code>	194
7.13.2.52	<code>npplAdd_8u_C1IRSfs</code>	195
7.13.2.53	<code>npplAdd_8u_C1RSfs</code>	195
7.13.2.54	<code>npplAdd_8u_C3IRSfs</code>	196
7.13.2.55	<code>npplAdd_8u_C3RSfs</code>	196
7.13.2.56	<code>npplAdd_8u_C4IRSfs</code>	196
7.13.2.57	<code>npplAdd_8u_C4RSfs</code>	197
7.14	<code>AddSquare</code>	198
7.14.1	Detailed Description	198
7.14.2	Function Documentation	198
7.14.2.1	<code>npplAddSquare_16u32f_C1IMR</code>	198
7.14.2.2	<code>npplAddSquare_16u32f_C1IR</code>	199
7.14.2.3	<code>npplAddSquare_32f_C1IMR</code>	199
7.14.2.4	<code>npplAddSquare_32f_C1IR</code>	200

7.14.2.5	nppiAddSquare_8u32f_C1IMR	200
7.14.2.6	nppiAddSquare_8u32f_C1IR	200
7.15	AddProduct	201
7.15.1	Detailed Description	201
7.15.2	Function Documentation	201
7.15.2.1	nppiAddProduct_16u32f_C1IMR	201
7.15.2.2	nppiAddProduct_16u32f_C1IR	202
7.15.2.3	nppiAddProduct_32f_C1IMR	202
7.15.2.4	nppiAddProduct_32f_C1IR	203
7.15.2.5	nppiAddProduct_8u32f_C1IMR	203
7.15.2.6	nppiAddProduct_8u32f_C1IR	204
7.16	AddWeighted	205
7.16.1	Detailed Description	205
7.16.2	Function Documentation	205
7.16.2.1	nppiAddWeighted_16u32f_C1IMR	205
7.16.2.2	nppiAddWeighted_16u32f_C1IR	206
7.16.2.3	nppiAddWeighted_32f_C1IMR	206
7.16.2.4	nppiAddWeighted_32f_C1IR	207
7.16.2.5	nppiAddWeighted_8u32f_C1IMR	207
7.16.2.6	nppiAddWeighted_8u32f_C1IR	208
7.17	Mul	209
7.17.1	Detailed Description	214
7.17.2	Function Documentation	214
7.17.2.1	nppiMul_16s_AC4IRSfs	214
7.17.2.2	nppiMul_16s_AC4RSfs	215
7.17.2.3	nppiMul_16s_C1IRSfs	215
7.17.2.4	nppiMul_16s_C1RSfs	215
7.17.2.5	nppiMul_16s_C3IRSfs	216
7.17.2.6	nppiMul_16s_C3RSfs	216
7.17.2.7	nppiMul_16s_C4IRSfs	217
7.17.2.8	nppiMul_16s_C4RSfs	217
7.17.2.9	nppiMul_16sc_AC4IRSfs	217
7.17.2.10	nppiMul_16sc_AC4RSfs	218
7.17.2.11	nppiMul_16sc_C1IRSfs	218
7.17.2.12	nppiMul_16sc_C1RSfs	219
7.17.2.13	nppiMul_16sc_C3IRSfs	219

7.17.2.14 nppiMul_16sc_C3RSfs	219
7.17.2.15 nppiMul_16u_AC4IRSfs	220
7.17.2.16 nppiMul_16u_AC4RSfs	220
7.17.2.17 nppiMul_16u_C1IRSfs	221
7.17.2.18 nppiMul_16u_C1RSfs	221
7.17.2.19 nppiMul_16u_C3IRSfs	222
7.17.2.20 nppiMul_16u_C3RSfs	222
7.17.2.21 nppiMul_16u_C4IRSfs	222
7.17.2.22 nppiMul_16u_C4RSfs	223
7.17.2.23 nppiMul_32f_AC4IR	223
7.17.2.24 nppiMul_32f_AC4R	224
7.17.2.25 nppiMul_32f_C1IR	224
7.17.2.26 nppiMul_32f_C1R	224
7.17.2.27 nppiMul_32f_C3IR	225
7.17.2.28 nppiMul_32f_C3R	225
7.17.2.29 nppiMul_32f_C4IR	226
7.17.2.30 nppiMul_32f_C4R	226
7.17.2.31 nppiMul_32fc_AC4IR	226
7.17.2.32 nppiMul_32fc_AC4R	227
7.17.2.33 nppiMul_32fc_C1IR	227
7.17.2.34 nppiMul_32fc_C1R	227
7.17.2.35 nppiMul_32fc_C3IR	228
7.17.2.36 nppiMul_32fc_C3R	228
7.17.2.37 nppiMul_32fc_C4IR	229
7.17.2.38 nppiMul_32fc_C4R	229
7.17.2.39 nppiMul_32s_C1IRSfs	229
7.17.2.40 nppiMul_32s_C1R	230
7.17.2.41 nppiMul_32s_C1RSfs	230
7.17.2.42 nppiMul_32s_C3IRSfs	231
7.17.2.43 nppiMul_32s_C3RSfs	231
7.17.2.44 nppiMul_32sc_AC4IRSfs	231
7.17.2.45 nppiMul_32sc_AC4RSfs	232
7.17.2.46 nppiMul_32sc_C1IRSfs	232
7.17.2.47 nppiMul_32sc_C1RSfs	233
7.17.2.48 nppiMul_32sc_C3IRSfs	233
7.17.2.49 nppiMul_32sc_C3RSfs	233

7.17.2.50	<code>nppiMul_8u_AC4IRSfs</code>	234
7.17.2.51	<code>nppiMul_8u_AC4RSfs</code>	234
7.17.2.52	<code>nppiMul_8u_C1IRSfs</code>	235
7.17.2.53	<code>nppiMul_8u_C1RSfs</code>	235
7.17.2.54	<code>nppiMul_8u_C3IRSfs</code>	236
7.17.2.55	<code>nppiMul_8u_C3RSfs</code>	236
7.17.2.56	<code>nppiMul_8u_C4IRSfs</code>	236
7.17.2.57	<code>nppiMul_8u_C4RSfs</code>	237
7.18	MulScale	238
7.18.1	Detailed Description	239
7.18.2	Function Documentation	239
7.18.2.1	<code>nppiMulScale_16u_AC4IR</code>	239
7.18.2.2	<code>nppiMulScale_16u_AC4R</code>	240
7.18.2.3	<code>nppiMulScale_16u_C1IR</code>	240
7.18.2.4	<code>nppiMulScale_16u_C1R</code>	241
7.18.2.5	<code>nppiMulScale_16u_C3IR</code>	241
7.18.2.6	<code>nppiMulScale_16u_C3R</code>	241
7.18.2.7	<code>nppiMulScale_16u_C4IR</code>	242
7.18.2.8	<code>nppiMulScale_16u_C4R</code>	242
7.18.2.9	<code>nppiMulScale_8u_AC4IR</code>	243
7.18.2.10	<code>nppiMulScale_8u_AC4R</code>	243
7.18.2.11	<code>nppiMulScale_8u_C1IR</code>	243
7.18.2.12	<code>nppiMulScale_8u_C1R</code>	244
7.18.2.13	<code>nppiMulScale_8u_C3IR</code>	244
7.18.2.14	<code>nppiMulScale_8u_C3R</code>	245
7.18.2.15	<code>nppiMulScale_8u_C4IR</code>	245
7.18.2.16	<code>nppiMulScale_8u_C4R</code>	245
7.19	Sub	247
7.19.1	Detailed Description	252
7.19.2	Function Documentation	252
7.19.2.1	<code>nppiSub_16s_AC4IRSfs</code>	252
7.19.2.2	<code>nppiSub_16s_AC4RSfs</code>	253
7.19.2.3	<code>nppiSub_16s_C1IRSfs</code>	253
7.19.2.4	<code>nppiSub_16s_C1RSfs</code>	254
7.19.2.5	<code>nppiSub_16s_C3IRSfs</code>	254
7.19.2.6	<code>nppiSub_16s_C3RSfs</code>	254

7.19.2.7	nppiSub_16s_C4IRSfs	255
7.19.2.8	nppiSub_16s_C4RSfs	255
7.19.2.9	nppiSub_16sc_AC4IRSfs	256
7.19.2.10	nppiSub_16sc_AC4RSfs	256
7.19.2.11	nppiSub_16sc_C1IRSfs	256
7.19.2.12	nppiSub_16sc_C1RSfs	257
7.19.2.13	nppiSub_16sc_C3IRSfs	257
7.19.2.14	nppiSub_16sc_C3RSfs	258
7.19.2.15	nppiSub_16u_AC4IRSfs	258
7.19.2.16	nppiSub_16u_AC4RSfs	258
7.19.2.17	nppiSub_16u_C1IRSfs	259
7.19.2.18	nppiSub_16u_C1RSfs	259
7.19.2.19	nppiSub_16u_C3IRSfs	260
7.19.2.20	nppiSub_16u_C3RSfs	260
7.19.2.21	nppiSub_16u_C4IRSfs	261
7.19.2.22	nppiSub_16u_C4RSfs	261
7.19.2.23	nppiSub_32f_AC4IR	261
7.19.2.24	nppiSub_32f_AC4R	262
7.19.2.25	nppiSub_32f_C1IR	262
7.19.2.26	nppiSub_32f_C1R	263
7.19.2.27	nppiSub_32f_C3IR	263
7.19.2.28	nppiSub_32f_C3R	263
7.19.2.29	nppiSub_32f_C4IR	264
7.19.2.30	nppiSub_32f_C4R	264
7.19.2.31	nppiSub_32fc_AC4IR	265
7.19.2.32	nppiSub_32fc_AC4R	265
7.19.2.33	nppiSub_32fc_C1IR	265
7.19.2.34	nppiSub_32fc_C1R	266
7.19.2.35	nppiSub_32fc_C3IR	266
7.19.2.36	nppiSub_32fc_C3R	267
7.19.2.37	nppiSub_32fc_C4IR	267
7.19.2.38	nppiSub_32fc_C4R	267
7.19.2.39	nppiSub_32s_C1IRSfs	268
7.19.2.40	nppiSub_32s_C1R	268
7.19.2.41	nppiSub_32s_C1RSfs	269
7.19.2.42	nppiSub_32s_C3IRSfs	269

7.19.2.43	nppiSub_32s_C3RSfs	269
7.19.2.44	nppiSub_32s_C4IRSfs	270
7.19.2.45	nppiSub_32s_C4RSfs	270
7.19.2.46	nppiSub_32sc_AC4IRSfs	271
7.19.2.47	nppiSub_32sc_AC4RSfs	271
7.19.2.48	nppiSub_32sc_C1IRSfs	272
7.19.2.49	nppiSub_32sc_C1RSfs	272
7.19.2.50	nppiSub_32sc_C3IRSfs	272
7.19.2.51	nppiSub_32sc_C3RSfs	273
7.19.2.52	nppiSub_8u_AC4IRSfs	273
7.19.2.53	nppiSub_8u_AC4RSfs	274
7.19.2.54	nppiSub_8u_C1IRSfs	274
7.19.2.55	nppiSub_8u_C1RSfs	274
7.19.2.56	nppiSub_8u_C3IRSfs	275
7.19.2.57	nppiSub_8u_C3RSfs	275
7.19.2.58	nppiSub_8u_C4IRSfs	276
7.19.2.59	nppiSub_8u_C4RSfs	276
7.20	Div	277
7.20.1	Detailed Description	282
7.20.2	Function Documentation	282
7.20.2.1	nppiDiv_16s_AC4IRSfs	282
7.20.2.2	nppiDiv_16s_AC4RSfs	282
7.20.2.3	nppiDiv_16s_C1IRSfs	283
7.20.2.4	nppiDiv_16s_C1RSfs	283
7.20.2.5	nppiDiv_16s_C3IRSfs	284
7.20.2.6	nppiDiv_16s_C3RSfs	284
7.20.2.7	nppiDiv_16s_C4IRSfs	284
7.20.2.8	nppiDiv_16s_C4RSfs	285
7.20.2.9	nppiDiv_16sc_AC4IRSfs	285
7.20.2.10	nppiDiv_16sc_AC4RSfs	286
7.20.2.11	nppiDiv_16sc_C1IRSfs	286
7.20.2.12	nppiDiv_16sc_C1RSfs	286
7.20.2.13	nppiDiv_16sc_C3IRSfs	287
7.20.2.14	nppiDiv_16sc_C3RSfs	287
7.20.2.15	nppiDiv_16u_AC4IRSfs	288
7.20.2.16	nppiDiv_16u_AC4RSfs	288

7.20.2.17 nppiDiv_16u_C1IRSfs	289
7.20.2.18 nppiDiv_16u_C1RSfs	289
7.20.2.19 nppiDiv_16u_C3IRSfs	289
7.20.2.20 nppiDiv_16u_C3RSfs	290
7.20.2.21 nppiDiv_16u_C4IRSfs	290
7.20.2.22 nppiDiv_16u_C4RSfs	291
7.20.2.23 nppiDiv_32f_AC4IR	291
7.20.2.24 nppiDiv_32f_AC4R	291
7.20.2.25 nppiDiv_32f_C1IR	292
7.20.2.26 nppiDiv_32f_C1R	292
7.20.2.27 nppiDiv_32f_C3IR	293
7.20.2.28 nppiDiv_32f_C3R	293
7.20.2.29 nppiDiv_32f_C4IR	293
7.20.2.30 nppiDiv_32f_C4R	294
7.20.2.31 nppiDiv_32fc_AC4IR	294
7.20.2.32 nppiDiv_32fc_AC4R	294
7.20.2.33 nppiDiv_32fc_C1IR	295
7.20.2.34 nppiDiv_32fc_C1R	295
7.20.2.35 nppiDiv_32fc_C3IR	296
7.20.2.36 nppiDiv_32fc_C3R	296
7.20.2.37 nppiDiv_32fc_C4IR	296
7.20.2.38 nppiDiv_32fc_C4R	297
7.20.2.39 nppiDiv_32s_C1IRSfs	297
7.20.2.40 nppiDiv_32s_C1R	297
7.20.2.41 nppiDiv_32s_C1RSfs	298
7.20.2.42 nppiDiv_32s_C3IRSfs	298
7.20.2.43 nppiDiv_32s_C3RSfs	299
7.20.2.44 nppiDiv_32sc_AC4IRSfs	299
7.20.2.45 nppiDiv_32sc_AC4RSfs	299
7.20.2.46 nppiDiv_32sc_C1IRSfs	300
7.20.2.47 nppiDiv_32sc_C1RSfs	300
7.20.2.48 nppiDiv_32sc_C3IRSfs	301
7.20.2.49 nppiDiv_32sc_C3RSfs	301
7.20.2.50 nppiDiv_8u_AC4IRSfs	302
7.20.2.51 nppiDiv_8u_AC4RSfs	302
7.20.2.52 nppiDiv_8u_C1IRSfs	302

7.20.2.53	<code>nppiDiv_8u_C1RSfs</code>	303
7.20.2.54	<code>nppiDiv_8u_C3RSfs</code>	303
7.20.2.55	<code>nppiDiv_8u_C3RSfs</code>	304
7.20.2.56	<code>nppiDiv_8u_C4RSfs</code>	304
7.20.2.57	<code>nppiDiv_8u_C4RSfs</code>	304
7.21	<code>Div_Round</code>	306
7.21.1	Detailed Description	308
7.21.2	Function Documentation	308
7.21.2.1	<code>nppiDiv_Round_16s_AC4RSfs</code>	308
7.21.2.2	<code>nppiDiv_Round_16s_AC4RSfs</code>	309
7.21.2.3	<code>nppiDiv_Round_16s_C1RSfs</code>	309
7.21.2.4	<code>nppiDiv_Round_16s_C1RSfs</code>	310
7.21.2.5	<code>nppiDiv_Round_16s_C3RSfs</code>	310
7.21.2.6	<code>nppiDiv_Round_16s_C3RSfs</code>	311
7.21.2.7	<code>nppiDiv_Round_16s_C4RSfs</code>	311
7.21.2.8	<code>nppiDiv_Round_16s_C4RSfs</code>	312
7.21.2.9	<code>nppiDiv_Round_16u_AC4RSfs</code>	312
7.21.2.10	<code>nppiDiv_Round_16u_AC4RSfs</code>	313
7.21.2.11	<code>nppiDiv_Round_16u_C1RSfs</code>	313
7.21.2.12	<code>nppiDiv_Round_16u_C1RSfs</code>	314
7.21.2.13	<code>nppiDiv_Round_16u_C3RSfs</code>	314
7.21.2.14	<code>nppiDiv_Round_16u_C3RSfs</code>	315
7.21.2.15	<code>nppiDiv_Round_16u_C4RSfs</code>	315
7.21.2.16	<code>nppiDiv_Round_16u_C4RSfs</code>	316
7.21.2.17	<code>nppiDiv_Round_8u_AC4RSfs</code>	316
7.21.2.18	<code>nppiDiv_Round_8u_AC4RSfs</code>	317
7.21.2.19	<code>nppiDiv_Round_8u_C1RSfs</code>	317
7.21.2.20	<code>nppiDiv_Round_8u_C1RSfs</code>	318
7.21.2.21	<code>nppiDiv_Round_8u_C3RSfs</code>	318
7.21.2.22	<code>nppiDiv_Round_8u_C3RSfs</code>	319
7.21.2.23	<code>nppiDiv_Round_8u_C4RSfs</code>	319
7.21.2.24	<code>nppiDiv_Round_8u_C4RSfs</code>	320
7.22	<code>Abs</code>	321
7.22.1	Detailed Description	322
7.22.2	Function Documentation	322
7.22.2.1	<code>nppiAbs_16s_AC4IR</code>	322

7.22.2.2	nppiAbs_16s_AC4R	322
7.22.2.3	nppiAbs_16s_C1IR	323
7.22.2.4	nppiAbs_16s_C1R	323
7.22.2.5	nppiAbs_16s_C3IR	323
7.22.2.6	nppiAbs_16s_C3R	324
7.22.2.7	nppiAbs_16s_C4IR	324
7.22.2.8	nppiAbs_16s_C4R	324
7.22.2.9	nppiAbs_32f_AC4IR	325
7.22.2.10	nppiAbs_32f_AC4R	325
7.22.2.11	nppiAbs_32f_C1IR	325
7.22.2.12	nppiAbs_32f_C1R	326
7.22.2.13	nppiAbs_32f_C3IR	326
7.22.2.14	nppiAbs_32f_C3R	326
7.22.2.15	nppiAbs_32f_C4IR	327
7.22.2.16	nppiAbs_32f_C4R	327
7.23	AbsDiff	328
7.23.1	Detailed Description	328
7.23.2	Function Documentation	328
7.23.2.1	nppiAbsDiff_16u_C1R	328
7.23.2.2	nppiAbsDiff_32f_C1R	329
7.23.2.3	nppiAbsDiff_8u_C1R	329
7.23.2.4	nppiAbsDiff_8u_C3R	329
7.23.2.5	nppiAbsDiff_8u_C4R	330
7.24	Sqr	331
7.24.1	Detailed Description	333
7.24.2	Function Documentation	334
7.24.2.1	nppiSqr_16s_AC4IRSfs	334
7.24.2.2	nppiSqr_16s_AC4RSfs	334
7.24.2.3	nppiSqr_16s_C1IRSfs	334
7.24.2.4	nppiSqr_16s_C1RSfs	335
7.24.2.5	nppiSqr_16s_C3IRSfs	335
7.24.2.6	nppiSqr_16s_C3RSfs	335
7.24.2.7	nppiSqr_16s_C4IRSfs	336
7.24.2.8	nppiSqr_16s_C4RSfs	336
7.24.2.9	nppiSqr_16u_AC4IRSfs	336
7.24.2.10	nppiSqr_16u_AC4RSfs	337

7.24.2.11	nppiSqr_16u_C1RSfs	337
7.24.2.12	nppiSqr_16u_C1RSfs	337
7.24.2.13	nppiSqr_16u_C3RSfs	338
7.24.2.14	nppiSqr_16u_C3RSfs	338
7.24.2.15	nppiSqr_16u_C4RSfs	338
7.24.2.16	nppiSqr_16u_C4RSfs	339
7.24.2.17	nppiSqr_32f_AC4IR	339
7.24.2.18	nppiSqr_32f_AC4R	339
7.24.2.19	nppiSqr_32f_C1IR	340
7.24.2.20	nppiSqr_32f_C1R	340
7.24.2.21	nppiSqr_32f_C3IR	340
7.24.2.22	nppiSqr_32f_C3R	341
7.24.2.23	nppiSqr_32f_C4IR	341
7.24.2.24	nppiSqr_32f_C4R	341
7.24.2.25	nppiSqr_8u_AC4RSfs	342
7.24.2.26	nppiSqr_8u_AC4RSfs	342
7.24.2.27	nppiSqr_8u_C1RSfs	342
7.24.2.28	nppiSqr_8u_C1RSfs	343
7.24.2.29	nppiSqr_8u_C3RSfs	343
7.24.2.30	nppiSqr_8u_C3RSfs	343
7.24.2.31	nppiSqr_8u_C4RSfs	344
7.24.2.32	nppiSqr_8u_C4RSfs	344
7.25	Sqrt	345
7.25.1	Detailed Description	347
7.25.2	Function Documentation	347
7.25.2.1	nppiSqrt_16s_AC4RSfs	347
7.25.2.2	nppiSqrt_16s_AC4RSfs	348
7.25.2.3	nppiSqrt_16s_C1RSfs	348
7.25.2.4	nppiSqrt_16s_C1RSfs	348
7.25.2.5	nppiSqrt_16s_C3RSfs	349
7.25.2.6	nppiSqrt_16s_C3RSfs	349
7.25.2.7	nppiSqrt_16u_AC4RSfs	349
7.25.2.8	nppiSqrt_16u_AC4RSfs	350
7.25.2.9	nppiSqrt_16u_C1RSfs	350
7.25.2.10	nppiSqrt_16u_C1RSfs	351
7.25.2.11	nppiSqrt_16u_C3RSfs	351

7.25.2.12	nppiSqrt_16u_C3RSfs	351
7.25.2.13	nppiSqrt_32f_AC4IR	352
7.25.2.14	nppiSqrt_32f_AC4R	352
7.25.2.15	nppiSqrt_32f_C1IR	352
7.25.2.16	nppiSqrt_32f_C1R	353
7.25.2.17	nppiSqrt_32f_C3IR	353
7.25.2.18	nppiSqrt_32f_C3R	353
7.25.2.19	nppiSqrt_32f_C4IR	354
7.25.2.20	nppiSqrt_32f_C4R	354
7.25.2.21	nppiSqrt_8u_AC4IRSfs	354
7.25.2.22	nppiSqrt_8u_AC4RSfs	355
7.25.2.23	nppiSqrt_8u_C1IRSfs	355
7.25.2.24	nppiSqrt_8u_C1RSfs	355
7.25.2.25	nppiSqrt_8u_C3IRSfs	356
7.25.2.26	nppiSqrt_8u_C3RSfs	356
7.26	Ln	357
7.26.1	Detailed Description	358
7.26.2	Function Documentation	358
7.26.2.1	nppiLn_16s_C1IRSfs	358
7.26.2.2	nppiLn_16s_C1RSfs	359
7.26.2.3	nppiLn_16s_C3IRSfs	359
7.26.2.4	nppiLn_16s_C3RSfs	359
7.26.2.5	nppiLn_16u_C1IRSfs	360
7.26.2.6	nppiLn_16u_C1RSfs	360
7.26.2.7	nppiLn_16u_C3IRSfs	360
7.26.2.8	nppiLn_16u_C3RSfs	361
7.26.2.9	nppiLn_32f_C1IR	361
7.26.2.10	nppiLn_32f_C1R	361
7.26.2.11	nppiLn_32f_C3IR	362
7.26.2.12	nppiLn_32f_C3R	362
7.26.2.13	nppiLn_8u_C1IRSfs	362
7.26.2.14	nppiLn_8u_C1RSfs	363
7.26.2.15	nppiLn_8u_C3IRSfs	363
7.26.2.16	nppiLn_8u_C3RSfs	363
7.27	Exp	364
7.27.1	Detailed Description	365

7.27.2	Function Documentation	365
7.27.2.1	npplExp_16s_C1IRSfs	365
7.27.2.2	npplExp_16s_C1RSfs	366
7.27.2.3	npplExp_16s_C3IRSfs	366
7.27.2.4	npplExp_16s_C3RSfs	366
7.27.2.5	npplExp_16u_C1IRSfs	367
7.27.2.6	npplExp_16u_C1RSfs	367
7.27.2.7	npplExp_16u_C3IRSfs	367
7.27.2.8	npplExp_16u_C3RSfs	368
7.27.2.9	npplExp_32f_C1IR	368
7.27.2.10	npplExp_32f_C1R	368
7.27.2.11	npplExp_32f_C3IR	369
7.27.2.12	npplExp_32f_C3R	369
7.27.2.13	npplExp_8u_C1RSfs	369
7.27.2.14	npplExp_8u_C1RSfs	370
7.27.2.15	npplExp_8u_C3RSfs	370
7.27.2.16	npplExp_8u_C3RSfs	370
7.28	Logical Operations	371
7.29	AndC	372
7.29.1	Detailed Description	374
7.29.2	Function Documentation	374
7.29.2.1	npplAndC_16u_AC4IR	374
7.29.2.2	npplAndC_16u_AC4R	374
7.29.2.3	npplAndC_16u_C1IR	374
7.29.2.4	npplAndC_16u_C1R	375
7.29.2.5	npplAndC_16u_C3IR	375
7.29.2.6	npplAndC_16u_C3R	375
7.29.2.7	npplAndC_16u_C4IR	376
7.29.2.8	npplAndC_16u_C4R	376
7.29.2.9	npplAndC_32s_AC4IR	377
7.29.2.10	npplAndC_32s_AC4R	377
7.29.2.11	npplAndC_32s_C1IR	377
7.29.2.12	npplAndC_32s_C1R	378
7.29.2.13	npplAndC_32s_C3IR	378
7.29.2.14	npplAndC_32s_C3R	378
7.29.2.15	npplAndC_32s_C4IR	379

7.29.2.16	nppiAndC_32s_C4R	379
7.29.2.17	nppiAndC_8u_AC4IR	379
7.29.2.18	nppiAndC_8u_AC4R	380
7.29.2.19	nppiAndC_8u_C1IR	380
7.29.2.20	nppiAndC_8u_C1R	380
7.29.2.21	nppiAndC_8u_C3IR	381
7.29.2.22	nppiAndC_8u_C3R	381
7.29.2.23	nppiAndC_8u_C4IR	381
7.29.2.24	nppiAndC_8u_C4R	382
7.30	OrC	383
7.30.1	Detailed Description	385
7.30.2	Function Documentation	385
7.30.2.1	nppiOrC_16u_AC4IR	385
7.30.2.2	nppiOrC_16u_AC4R	385
7.30.2.3	nppiOrC_16u_C1IR	385
7.30.2.4	nppiOrC_16u_C1R	386
7.30.2.5	nppiOrC_16u_C3IR	386
7.30.2.6	nppiOrC_16u_C3R	386
7.30.2.7	nppiOrC_16u_C4IR	387
7.30.2.8	nppiOrC_16u_C4R	387
7.30.2.9	nppiOrC_32s_AC4IR	388
7.30.2.10	nppiOrC_32s_AC4R	388
7.30.2.11	nppiOrC_32s_C1IR	388
7.30.2.12	nppiOrC_32s_C1R	389
7.30.2.13	nppiOrC_32s_C3IR	389
7.30.2.14	nppiOrC_32s_C3R	389
7.30.2.15	nppiOrC_32s_C4IR	390
7.30.2.16	nppiOrC_32s_C4R	390
7.30.2.17	nppiOrC_8u_AC4IR	390
7.30.2.18	nppiOrC_8u_AC4R	391
7.30.2.19	nppiOrC_8u_C1IR	391
7.30.2.20	nppiOrC_8u_C1R	391
7.30.2.21	nppiOrC_8u_C3IR	392
7.30.2.22	nppiOrC_8u_C3R	392
7.30.2.23	nppiOrC_8u_C4IR	392
7.30.2.24	nppiOrC_8u_C4R	393

7.31	XorC	394
7.31.1	Detailed Description	396
7.31.2	Function Documentation	396
7.31.2.1	nppiXorC_16u_AC4IR	396
7.31.2.2	nppiXorC_16u_AC4R	396
7.31.2.3	nppiXorC_16u_C1IR	396
7.31.2.4	nppiXorC_16u_C1R	397
7.31.2.5	nppiXorC_16u_C3IR	397
7.31.2.6	nppiXorC_16u_C3R	397
7.31.2.7	nppiXorC_16u_C4IR	398
7.31.2.8	nppiXorC_16u_C4R	398
7.31.2.9	nppiXorC_32s_AC4IR	399
7.31.2.10	nppiXorC_32s_AC4R	399
7.31.2.11	nppiXorC_32s_C1IR	399
7.31.2.12	nppiXorC_32s_C1R	400
7.31.2.13	nppiXorC_32s_C3IR	400
7.31.2.14	nppiXorC_32s_C3R	400
7.31.2.15	nppiXorC_32s_C4IR	401
7.31.2.16	nppiXorC_32s_C4R	401
7.31.2.17	nppiXorC_8u_AC4IR	401
7.31.2.18	nppiXorC_8u_AC4R	402
7.31.2.19	nppiXorC_8u_C1IR	402
7.31.2.20	nppiXorC_8u_C1R	402
7.31.2.21	nppiXorC_8u_C3IR	403
7.31.2.22	nppiXorC_8u_C3R	403
7.31.2.23	nppiXorC_8u_C4IR	403
7.31.2.24	nppiXorC_8u_C4R	404
7.32	RShiftC	405
7.32.1	Detailed Description	408
7.32.2	Function Documentation	408
7.32.2.1	nppiRShiftC_16s_AC4IR	408
7.32.2.2	nppiRShiftC_16s_AC4R	408
7.32.2.3	nppiRShiftC_16s_C1IR	409
7.32.2.4	nppiRShiftC_16s_C1R	409
7.32.2.5	nppiRShiftC_16s_C3IR	409
7.32.2.6	nppiRShiftC_16s_C3R	410

7.32.2.7	<code>nppiRShiftC_16s_C4IR</code>	410
7.32.2.8	<code>nppiRShiftC_16s_C4R</code>	410
7.32.2.9	<code>nppiRShiftC_16u_AC4IR</code>	411
7.32.2.10	<code>nppiRShiftC_16u_AC4R</code>	411
7.32.2.11	<code>nppiRShiftC_16u_C1IR</code>	412
7.32.2.12	<code>nppiRShiftC_16u_C1R</code>	412
7.32.2.13	<code>nppiRShiftC_16u_C3IR</code>	412
7.32.2.14	<code>nppiRShiftC_16u_C3R</code>	413
7.32.2.15	<code>nppiRShiftC_16u_C4IR</code>	413
7.32.2.16	<code>nppiRShiftC_16u_C4R</code>	413
7.32.2.17	<code>nppiRShiftC_32s_AC4IR</code>	414
7.32.2.18	<code>nppiRShiftC_32s_AC4R</code>	414
7.32.2.19	<code>nppiRShiftC_32s_C1IR</code>	414
7.32.2.20	<code>nppiRShiftC_32s_C1R</code>	415
7.32.2.21	<code>nppiRShiftC_32s_C3IR</code>	415
7.32.2.22	<code>nppiRShiftC_32s_C3R</code>	415
7.32.2.23	<code>nppiRShiftC_32s_C4IR</code>	416
7.32.2.24	<code>nppiRShiftC_32s_C4R</code>	416
7.32.2.25	<code>nppiRShiftC_8s_AC4IR</code>	416
7.32.2.26	<code>nppiRShiftC_8s_AC4R</code>	417
7.32.2.27	<code>nppiRShiftC_8s_C1IR</code>	417
7.32.2.28	<code>nppiRShiftC_8s_C1R</code>	417
7.32.2.29	<code>nppiRShiftC_8s_C3IR</code>	418
7.32.2.30	<code>nppiRShiftC_8s_C3R</code>	418
7.32.2.31	<code>nppiRShiftC_8s_C4IR</code>	418
7.32.2.32	<code>nppiRShiftC_8s_C4R</code>	419
7.32.2.33	<code>nppiRShiftC_8u_AC4IR</code>	419
7.32.2.34	<code>nppiRShiftC_8u_AC4R</code>	419
7.32.2.35	<code>nppiRShiftC_8u_C1IR</code>	420
7.32.2.36	<code>nppiRShiftC_8u_C1R</code>	420
7.32.2.37	<code>nppiRShiftC_8u_C3IR</code>	420
7.32.2.38	<code>nppiRShiftC_8u_C3R</code>	421
7.32.2.39	<code>nppiRShiftC_8u_C4IR</code>	421
7.32.2.40	<code>nppiRShiftC_8u_C4R</code>	421
7.33	<code>LShiftC</code>	422
7.33.1	Detailed Description	424

7.33.2	Function Documentation	424
7.33.2.1	nppiLShiftC_16u_AC4IR	424
7.33.2.2	nppiLShiftC_16u_AC4R	424
7.33.2.3	nppiLShiftC_16u_C1IR	424
7.33.2.4	nppiLShiftC_16u_C1R	425
7.33.2.5	nppiLShiftC_16u_C3IR	425
7.33.2.6	nppiLShiftC_16u_C3R	425
7.33.2.7	nppiLShiftC_16u_C4IR	426
7.33.2.8	nppiLShiftC_16u_C4R	426
7.33.2.9	nppiLShiftC_32s_AC4IR	427
7.33.2.10	nppiLShiftC_32s_AC4R	427
7.33.2.11	nppiLShiftC_32s_C1IR	427
7.33.2.12	nppiLShiftC_32s_C1R	428
7.33.2.13	nppiLShiftC_32s_C3IR	428
7.33.2.14	nppiLShiftC_32s_C3R	428
7.33.2.15	nppiLShiftC_32s_C4IR	429
7.33.2.16	nppiLShiftC_32s_C4R	429
7.33.2.17	nppiLShiftC_8u_AC4IR	429
7.33.2.18	nppiLShiftC_8u_AC4R	430
7.33.2.19	nppiLShiftC_8u_C1IR	430
7.33.2.20	nppiLShiftC_8u_C1R	430
7.33.2.21	nppiLShiftC_8u_C3IR	431
7.33.2.22	nppiLShiftC_8u_C3R	431
7.33.2.23	nppiLShiftC_8u_C4IR	431
7.33.2.24	nppiLShiftC_8u_C4R	432
7.34	And	433
7.34.1	Detailed Description	435
7.34.2	Function Documentation	435
7.34.2.1	nppiAnd_16u_AC4IR	435
7.34.2.2	nppiAnd_16u_AC4R	435
7.34.2.3	nppiAnd_16u_C1IR	435
7.34.2.4	nppiAnd_16u_C1R	436
7.34.2.5	nppiAnd_16u_C3IR	436
7.34.2.6	nppiAnd_16u_C3R	437
7.34.2.7	nppiAnd_16u_C4IR	437
7.34.2.8	nppiAnd_16u_C4R	437

7.34.2.9	nppiAnd_32s_AC4IR	438
7.34.2.10	nppiAnd_32s_AC4R	438
7.34.2.11	nppiAnd_32s_C1IR	439
7.34.2.12	nppiAnd_32s_C1R	439
7.34.2.13	nppiAnd_32s_C3IR	439
7.34.2.14	nppiAnd_32s_C3R	440
7.34.2.15	nppiAnd_32s_C4IR	440
7.34.2.16	nppiAnd_32s_C4R	440
7.34.2.17	nppiAnd_8u_AC4IR	441
7.34.2.18	nppiAnd_8u_AC4R	441
7.34.2.19	nppiAnd_8u_C1IR	442
7.34.2.20	nppiAnd_8u_C1R	442
7.34.2.21	nppiAnd_8u_C3IR	442
7.34.2.22	nppiAnd_8u_C3R	443
7.34.2.23	nppiAnd_8u_C4IR	443
7.34.2.24	nppiAnd_8u_C4R	443
7.35	Or	445
7.35.1	Detailed Description	447
7.35.2	Function Documentation	447
7.35.2.1	nppiOr_16u_AC4IR	447
7.35.2.2	nppiOr_16u_AC4R	447
7.35.2.3	nppiOr_16u_C1IR	447
7.35.2.4	nppiOr_16u_C1R	448
7.35.2.5	nppiOr_16u_C3IR	448
7.35.2.6	nppiOr_16u_C3R	449
7.35.2.7	nppiOr_16u_C4IR	449
7.35.2.8	nppiOr_16u_C4R	449
7.35.2.9	nppiOr_32s_AC4IR	450
7.35.2.10	nppiOr_32s_AC4R	450
7.35.2.11	nppiOr_32s_C1IR	451
7.35.2.12	nppiOr_32s_C1R	451
7.35.2.13	nppiOr_32s_C3IR	451
7.35.2.14	nppiOr_32s_C3R	452
7.35.2.15	nppiOr_32s_C4IR	452
7.35.2.16	nppiOr_32s_C4R	452
7.35.2.17	nppiOr_8u_AC4IR	453

7.35.2.18	nppiOr_8u_AC4R	453
7.35.2.19	nppiOr_8u_C1IR	454
7.35.2.20	nppiOr_8u_C1R	454
7.35.2.21	nppiOr_8u_C3IR	454
7.35.2.22	nppiOr_8u_C3R	455
7.35.2.23	nppiOr_8u_C4IR	455
7.35.2.24	nppiOr_8u_C4R	455
7.36	Xor	457
7.36.1	Detailed Description	459
7.36.2	Function Documentation	459
7.36.2.1	nppiXor_16u_AC4IR	459
7.36.2.2	nppiXor_16u_AC4R	459
7.36.2.3	nppiXor_16u_C1IR	459
7.36.2.4	nppiXor_16u_C1R	460
7.36.2.5	nppiXor_16u_C3IR	460
7.36.2.6	nppiXor_16u_C3R	461
7.36.2.7	nppiXor_16u_C4IR	461
7.36.2.8	nppiXor_16u_C4R	461
7.36.2.9	nppiXor_32s_AC4IR	462
7.36.2.10	nppiXor_32s_AC4R	462
7.36.2.11	nppiXor_32s_C1IR	463
7.36.2.12	nppiXor_32s_C1R	463
7.36.2.13	nppiXor_32s_C3IR	463
7.36.2.14	nppiXor_32s_C3R	464
7.36.2.15	nppiXor_32s_C4IR	464
7.36.2.16	nppiXor_32s_C4R	464
7.36.2.17	nppiXor_8u_AC4IR	465
7.36.2.18	nppiXor_8u_AC4R	465
7.36.2.19	nppiXor_8u_C1IR	466
7.36.2.20	nppiXor_8u_C1R	466
7.36.2.21	nppiXor_8u_C3IR	466
7.36.2.22	nppiXor_8u_C3R	467
7.36.2.23	nppiXor_8u_C4IR	467
7.36.2.24	nppiXor_8u_C4R	467
7.37	Not	469
7.37.1	Detailed Description	469

7.37.2	Function Documentation	469
7.37.2.1	nppiNot_8u_AC4IR	469
7.37.2.2	nppiNot_8u_AC4R	470
7.37.2.3	nppiNot_8u_C1IR	470
7.37.2.4	nppiNot_8u_C1R	470
7.37.2.5	nppiNot_8u_C3IR	471
7.37.2.6	nppiNot_8u_C3R	471
7.37.2.7	nppiNot_8u_C4IR	471
7.37.2.8	nppiNot_8u_C4R	472
7.38	Alpha Composition	473
7.39	AlphaCompC	474
7.39.1	Detailed Description	475
7.39.2	Function Documentation	475
7.39.2.1	nppiAlphaCompC_16s_C1R	475
7.39.2.2	nppiAlphaCompC_16u_AC4R	476
7.39.2.3	nppiAlphaCompC_16u_C1R	476
7.39.2.4	nppiAlphaCompC_16u_C3R	477
7.39.2.5	nppiAlphaCompC_16u_C4R	477
7.39.2.6	nppiAlphaCompC_32f_C1R	478
7.39.2.7	nppiAlphaCompC_32s_C1R	478
7.39.2.8	nppiAlphaCompC_32u_C1R	479
7.39.2.9	nppiAlphaCompC_8s_C1R	479
7.39.2.10	nppiAlphaCompC_8u_AC4R	480
7.39.2.11	nppiAlphaCompC_8u_C1R	480
7.39.2.12	nppiAlphaCompC_8u_C3R	481
7.39.2.13	nppiAlphaCompC_8u_C4R	481
7.40	AlphaPremulC	482
7.40.1	Detailed Description	483
7.40.2	Function Documentation	483
7.40.2.1	nppiAlphaPremulC_16u_AC4IR	483
7.40.2.2	nppiAlphaPremulC_16u_AC4R	483
7.40.2.3	nppiAlphaPremulC_16u_C1IR	484
7.40.2.4	nppiAlphaPremulC_16u_C1R	484
7.40.2.5	nppiAlphaPremulC_16u_C3IR	485
7.40.2.6	nppiAlphaPremulC_16u_C3R	485
7.40.2.7	nppiAlphaPremulC_16u_C4IR	485

7.40.2.8	nppiAlphaPremulC_16u_C4R	486
7.40.2.9	nppiAlphaPremulC_8u_AC4IR	486
7.40.2.10	nppiAlphaPremulC_8u_AC4R	486
7.40.2.11	nppiAlphaPremulC_8u_C1IR	487
7.40.2.12	nppiAlphaPremulC_8u_C1R	487
7.40.2.13	nppiAlphaPremulC_8u_C3IR	487
7.40.2.14	nppiAlphaPremulC_8u_C3R	488
7.40.2.15	nppiAlphaPremulC_8u_C4IR	488
7.40.2.16	nppiAlphaPremulC_8u_C4R	488
7.41	AlphaComp	489
7.41.1	Detailed Description	490
7.41.2	Function Documentation	490
7.41.2.1	nppiAlphaComp_16s_AC1R	490
7.41.2.2	nppiAlphaComp_16u_AC1R	490
7.41.2.3	nppiAlphaComp_16u_AC4R	491
7.41.2.4	nppiAlphaComp_32f_AC1R	491
7.41.2.5	nppiAlphaComp_32f_AC4R	492
7.41.2.6	nppiAlphaComp_32s_AC1R	492
7.41.2.7	nppiAlphaComp_32s_AC4R	493
7.41.2.8	nppiAlphaComp_32u_AC1R	493
7.41.2.9	nppiAlphaComp_32u_AC4R	494
7.41.2.10	nppiAlphaComp_8s_AC1R	494
7.41.2.11	nppiAlphaComp_8u_AC1R	494
7.41.2.12	nppiAlphaComp_8u_AC4R	495
7.42	AlphaPremul	496
7.42.1	Detailed Description	496
7.42.2	Function Documentation	496
7.42.2.1	nppiAlphaPremul_16u_AC4IR	496
7.42.2.2	nppiAlphaPremul_16u_AC4R	497
7.42.2.3	nppiAlphaPremul_8u_AC4IR	497
7.42.2.4	nppiAlphaPremul_8u_AC4R	497
7.43	Color and Sampling Conversion	498
7.43.1	Detailed Description	498
7.44	Color Model Conversion	499
7.44.1	Detailed Description	527
7.44.2	Function Documentation	527

7.44.2.1	nppiBGRTToCbYCr422_709HDTV_8u_AC4C2R	527
7.44.2.2	nppiBGRTToCbYCr422_709HDTV_8u_C3C2R	528
7.44.2.3	nppiBGRTToCbYCr422_8u_AC4C2R	528
7.44.2.4	nppiBGRTToHLS_8u_AC4P4R	529
7.44.2.5	nppiBGRTToHLS_8u_AC4R	529
7.44.2.6	nppiBGRTToHLS_8u_AP4C4R	529
7.44.2.7	nppiBGRTToHLS_8u_AP4R	530
7.44.2.8	nppiBGRTToHLS_8u_C3P3R	530
7.44.2.9	nppiBGRTToHLS_8u_P3C3R	530
7.44.2.10	nppiBGRTToHLS_8u_P3R	531
7.44.2.11	nppiBGRTToLab_8u_C3R	531
7.44.2.12	nppiBGRTToYCbCr411_8u_AC4P3R	531
7.44.2.13	nppiBGRTToYCbCr411_8u_C3P3R	532
7.44.2.14	nppiBGRTToYCbCr420_709CSC_8u_AC4P3R	532
7.44.2.15	nppiBGRTToYCbCr420_709CSC_8u_C3P3R	532
7.44.2.16	nppiBGRTToYCbCr420_709HDTV_8u_AC4P3R	533
7.44.2.17	nppiBGRTToYCbCr420_8u_AC4P3R	533
7.44.2.18	nppiBGRTToYCbCr420_8u_C3P3R	534
7.44.2.19	nppiBGRTToYCbCr422_8u_AC4C2R	534
7.44.2.20	nppiBGRTToYCbCr422_8u_AC4P3R	534
7.44.2.21	nppiBGRTToYCbCr422_8u_C3C2R	535
7.44.2.22	nppiBGRTToYCbCr422_8u_C3P3R	535
7.44.2.23	nppiBGRTToYCbCr_8u_AC4P3R	536
7.44.2.24	nppiBGRTToYCbCr_8u_AC4P4R	536
7.44.2.25	nppiBGRTToYCbCr_8u_C3P3R	536
7.44.2.26	nppiBGRTToYCrCb420_709CSC_8u_AC4P3R	537
7.44.2.27	nppiBGRTToYCrCb420_709CSC_8u_C3P3R	537
7.44.2.28	nppiBGRTToYCrCb420_8u_AC4P3R	538
7.44.2.29	nppiBGRTToYCrCb420_8u_C3P3R	538
7.44.2.30	nppiBGRTToYUV420_8u_AC4P3R	538
7.44.2.31	nppiBGRTToYUV_8u_AC4P4R	539
7.44.2.32	nppiBGRTToYUV_8u_AC4R	539
7.44.2.33	nppiBGRTToYUV_8u_C3P3R	540
7.44.2.34	nppiBGRTToYUV_8u_C3R	540
7.44.2.35	nppiBGRTToYUV_8u_P3R	540
7.44.2.36	nppiCbYCr422ToBGR_709HDTV_8u_C2C3R	541

7.44.2.37 nppiCbYCr422ToBGR_709HDTV_8u_C2C4R	541
7.44.2.38 nppiCbYCr422ToBGR_8u_C2C4R	541
7.44.2.39 nppiCbYCr422ToRGB_8u_C2C3R	542
7.44.2.40 nppiCFAToRGB_16u_C1C3R	542
7.44.2.41 nppiCFAToRGB_8u_C1C3R	543
7.44.2.42 nppiCFAToRGBA_16u_C1AC4R	543
7.44.2.43 nppiCFAToRGBA_8u_C1AC4R	544
7.44.2.44 nppiColorToGray_16s_AC4C1R	544
7.44.2.45 nppiColorToGray_16s_C3C1R	544
7.44.2.46 nppiColorToGray_16s_C4C1R	545
7.44.2.47 nppiColorToGray_16u_AC4C1R	545
7.44.2.48 nppiColorToGray_16u_C3C1R	546
7.44.2.49 nppiColorToGray_16u_C4C1R	546
7.44.2.50 nppiColorToGray_32f_AC4C1R	546
7.44.2.51 nppiColorToGray_32f_C3C1R	547
7.44.2.52 nppiColorToGray_32f_C4C1R	547
7.44.2.53 nppiColorToGray_8u_AC4C1R	548
7.44.2.54 nppiColorToGray_8u_C3C1R	548
7.44.2.55 nppiColorToGray_8u_C4C1R	548
7.44.2.56 nppiHLSToBGR_8u_AC4P4R	549
7.44.2.57 nppiHLSToBGR_8u_AC4R	549
7.44.2.58 nppiHLSToBGR_8u_AP4C4R	549
7.44.2.59 nppiHLSToBGR_8u_AP4R	550
7.44.2.60 nppiHLSToBGR_8u_C3P3R	550
7.44.2.61 nppiHLSToBGR_8u_P3C3R	550
7.44.2.62 nppiHLSToBGR_8u_P3R	551
7.44.2.63 nppiHLSToRGB_8u_AC4R	551
7.44.2.64 nppiHLSToRGB_8u_C3R	551
7.44.2.65 nppiHSVToRGB_8u_AC4R	552
7.44.2.66 nppiHSVToRGB_8u_C3R	552
7.44.2.67 nppiLabToBGR_8u_C3R	552
7.44.2.68 nppiLUVToRGB_8u_AC4R	553
7.44.2.69 nppiLUVToRGB_8u_C3R	553
7.44.2.70 nppiNV21ToBGR_8u_P2C4R	553
7.44.2.71 nppiNV21ToRGB_8u_P2C4R	554
7.44.2.72 nppiRGBToCbYCr422_8u_C3C2R	554

7.44.2.73	nppiRGBToCbYCr422Gamma_8u_C3C2R	554
7.44.2.74	nppiRGBToGray_16s_AC4C1R	555
7.44.2.75	nppiRGBToGray_16s_C3C1R	555
7.44.2.76	nppiRGBToGray_16u_AC4C1R	555
7.44.2.77	nppiRGBToGray_16u_C3C1R	556
7.44.2.78	nppiRGBToGray_32f_AC4C1R	556
7.44.2.79	nppiRGBToGray_32f_C3C1R	557
7.44.2.80	nppiRGBToGray_8u_AC4C1R	557
7.44.2.81	nppiRGBToGray_8u_C3C1R	557
7.44.2.82	nppiRGBToHLS_8u_AC4R	558
7.44.2.83	nppiRGBToHLS_8u_C3R	558
7.44.2.84	nppiRGBToHSV_8u_AC4R	558
7.44.2.85	nppiRGBToHSV_8u_C3R	559
7.44.2.86	nppiRGBToLUV_8u_AC4R	559
7.44.2.87	nppiRGBToLUV_8u_C3R	559
7.44.2.88	nppiRGBToXYZ_8u_AC4R	560
7.44.2.89	nppiRGBToXYZ_8u_C3R	560
7.44.2.90	nppiRGBToYCbCr420_8u_C3P3R	560
7.44.2.91	nppiRGBToYCbCr422_8u_C3C2R	561
7.44.2.92	nppiRGBToYCbCr422_8u_C3P3R	561
7.44.2.93	nppiRGBToYCbCr422_8u_P3C2R	561
7.44.2.94	nppiRGBToYCbCr_8u_AC4P3R	562
7.44.2.95	nppiRGBToYCbCr_8u_AC4R	562
7.44.2.96	nppiRGBToYCbCr_8u_C3P3R	562
7.44.2.97	nppiRGBToYCbCr_8u_C3R	563
7.44.2.98	nppiRGBToYCbCr_8u_P3R	563
7.44.2.99	nppiRGBToYCC_8u_AC4R	564
7.44.2.100	nppiRGBToYCC_8u_C3R	564
7.44.2.101	nppiRGBToYCrCb420_8u_AC4P3R	564
7.44.2.102	nppiRGBToYCrCb422_8u_C3C2R	565
7.44.2.103	nppiRGBToYCrCb422_8u_P3C2R	565
7.44.2.104	nppiRGBToYUV420_8u_C3P3R	565
7.44.2.105	nppiRGBToYUV420_8u_P3R	566
7.44.2.106	nppiRGBToYUV422_8u_C3C2R	566
7.44.2.107	nppiRGBToYUV422_8u_C3P3R	566
7.44.2.108	nppiRGBToYUV422_8u_P3R	567

7.44.2.109nppiRGBToYUV_8u_AC4P4R	567
7.44.2.110nppiRGBToYUV_8u_AC4R	567
7.44.2.111nppiRGBToYUV_8u_C3P3R	568
7.44.2.112nppiRGBToYUV_8u_C3R	568
7.44.2.113nppiRGBToYUV_8u_P3R	568
7.44.2.114nppiXYZToRGB_8u_AC4R	569
7.44.2.115nppiXYZToRGB_8u_C3R	569
7.44.2.116nppiYCbCr411ToBGR_8u_P3C3R	570
7.44.2.117nppiYCbCr411ToBGR_8u_P3C4R	570
7.44.2.118nppiYCbCr420ToBGR_709CSC_8u_P3C3R	570
7.44.2.119nppiYCbCr420ToBGR_709HDTV_8u_P3C4R	571
7.44.2.120nppiYCbCr420ToBGR_8u_P3C3R	571
7.44.2.121nppiYCbCr420ToBGR_8u_P3C4R	571
7.44.2.122nppiYCbCr420ToRGB_8u_P3C3R	572
7.44.2.123nppiYCbCr422ToBGR_8u_C2C3R	572
7.44.2.124nppiYCbCr422ToBGR_8u_C2C4R	572
7.44.2.125nppiYCbCr422ToBGR_8u_P3C3R	573
7.44.2.126nppiYCbCr422ToRGB_8u_C2C3R	573
7.44.2.127nppiYCbCr422ToRGB_8u_C2P3R	574
7.44.2.128nppiYCbCr422ToRGB_8u_P3C3R	574
7.44.2.129nppiYCbCrToBGR_709CSC_8u_P3C3R	574
7.44.2.130nppiYCbCrToBGR_709CSC_8u_P3C4R	575
7.44.2.131nppiYCbCrToBGR_8u_P3C3R	575
7.44.2.132nppiYCbCrToBGR_8u_P3C4R	575
7.44.2.133nppiYCbCrToRGB_8u_AC4R	576
7.44.2.134nppiYCbCrToRGB_8u_C3R	576
7.44.2.135nppiYCbCrToRGB_8u_P3C3R	576
7.44.2.136nppiYCbCrToRGB_8u_P3C4R	577
7.44.2.137nppiYCbCrToRGB_8u_P3R	577
7.44.2.138nppiYCCToRGB_8u_AC4R	578
7.44.2.139nppiYCCToRGB_8u_C3R	578
7.44.2.140nppiYCrCb420ToRGB_8u_P3C4R	578
7.44.2.141nppiYCrCb422ToRGB_8u_C2C3R	579
7.44.2.142nppiYCrCb422ToRGB_8u_C2P3R	579
7.44.2.143nppiYUV420ToBGR_8u_P3C3R	579
7.44.2.144nppiYUV420ToBGR_8u_P3C4R	580

7.44.2.145	nppiYUV420ToRGB_8u_P3AC4R	580
7.44.2.146	nppiYUV420ToRGB_8u_P3C3R	580
7.44.2.147	nppiYUV420ToRGB_8u_P3C4R	581
7.44.2.148	nppiYUV420ToRGB_8u_P3R	581
7.44.2.149	nppiYUV422ToRGB_8u_C2C3R	581
7.44.2.150	nppiYUV422ToRGB_8u_P3AC4R	582
7.44.2.151	nppiYUV422ToRGB_8u_P3C3R	582
7.44.2.152	nppiYUV422ToRGB_8u_P3R	582
7.44.2.153	nppiYUVToBGR_8u_AC4R	583
7.44.2.154	nppiYUVToBGR_8u_C3R	583
7.44.2.155	nppiYUVToBGR_8u_P3C3R	583
7.44.2.156	nppiYUVToBGR_8u_P3R	584
7.44.2.157	nppiYUVToRGB_8u_AC4R	584
7.44.2.158	nppiYUVToRGB_8u_C3R	584
7.44.2.159	nppiYUVToRGB_8u_P3C3R	585
7.44.2.160	nppiYUVToRGB_8u_P3R	585
7.45	Color Sampling Format Conversion	586
7.45.1	Detailed Description	593
7.45.2	Function Documentation	593
7.45.2.1	nppiCbYCr422ToYCbCr411_8u_C2P3R	593
7.45.2.2	nppiCbYCr422ToYCbCr420_8u_C2P2R	594
7.45.2.3	nppiCbYCr422ToYCbCr420_8u_C2P3R	594
7.45.2.4	nppiCbYCr422ToYCbCr422_8u_C2P3R	595
7.45.2.5	nppiCbYCr422ToYCbCr422_8u_C2R	595
7.45.2.6	nppiCbYCr422ToYCbCr420_8u_C2P3R	595
7.45.2.7	nppiYCbCr411_8u_P2P3R	596
7.45.2.8	nppiYCbCr411_8u_P3P2R	596
7.45.2.9	nppiYCbCr411ToYCbCr420_8u_P2P3R	597
7.45.2.10	nppiYCbCr411ToYCbCr420_8u_P3P2R	597
7.45.2.11	nppiYCbCr411ToYCbCr420_8u_P3R	597
7.45.2.12	nppiYCbCr411ToYCbCr422_8u_P2C2R	598
7.45.2.13	nppiYCbCr411ToYCbCr422_8u_P2P3R	598
7.45.2.14	nppiYCbCr411ToYCbCr422_8u_P3C2R	599
7.45.2.15	nppiYCbCr411ToYCbCr422_8u_P3R	599
7.45.2.16	nppiYCbCr411ToYCbCr420_8u_P2P3R	599
7.45.2.17	nppiYCbCr411ToYCbCr422_8u_P3C2R	600

7.45.2.18	nppiYCbCr411ToYCrCb422_8u_P3R	600
7.45.2.19	nppiYCbCr420_8u_P2P3R	601
7.45.2.20	nppiYCbCr420_8u_P3P2R	601
7.45.2.21	nppiYCbCr420ToCbYCr422_8u_P2C2R	601
7.45.2.22	nppiYCbCr420ToYCbCr411_8u_P2P3R	602
7.45.2.23	nppiYCbCr420ToYCbCr411_8u_P3P2R	602
7.45.2.24	nppiYCbCr420ToYCbCr422_8u_P2C2R	603
7.45.2.25	nppiYCbCr420ToYCbCr422_8u_P2P3R	603
7.45.2.26	nppiYCbCr420ToYCbCr422_8u_P3R	604
7.45.2.27	nppiYCbCr420ToYCrCb420_8u_P2P3R	604
7.45.2.28	nppiYCbCr422_8u_C2P3R	604
7.45.2.29	nppiYCbCr422_8u_P3C2R	605
7.45.2.30	nppiYCbCr422ToCbYCr422_8u_C2R	605
7.45.2.31	nppiYCbCr422ToYCbCr411_8u_C2P2R	606
7.45.2.32	nppiYCbCr422ToYCbCr411_8u_C2P3R	606
7.45.2.33	nppiYCbCr422ToYCbCr411_8u_P3P2R	606
7.45.2.34	nppiYCbCr422ToYCbCr411_8u_P3R	607
7.45.2.35	nppiYCbCr422ToYCbCr420_8u_C2P2R	607
7.45.2.36	nppiYCbCr422ToYCbCr420_8u_C2P3R	608
7.45.2.37	nppiYCbCr422ToYCbCr420_8u_P3P2R	608
7.45.2.38	nppiYCbCr422ToYCbCr420_8u_P3R	609
7.45.2.39	nppiYCbCr422ToYCrCb420_8u_C2P3R	609
7.45.2.40	nppiYCbCr422ToYCrCb422_8u_C2R	609
7.45.2.41	nppiYCbCr422ToYCrCb422_8u_P3C2R	610
7.45.2.42	nppiYCrCb420ToCbYCr422_8u_P3C2R	610
7.45.2.43	nppiYCrCb420ToYCbCr411_8u_P3P2R	611
7.45.2.44	nppiYCrCb420ToYCbCr420_8u_P3P2R	611
7.45.2.45	nppiYCrCb420ToYCbCr422_8u_P3C2R	612
7.45.2.46	nppiYCrCb420ToYCbCr422_8u_P3R	612
7.45.2.47	nppiYCrCb422ToYCbCr411_8u_C2P3R	612
7.45.2.48	nppiYCrCb422ToYCbCr420_8u_C2P3R	613
7.45.2.49	nppiYCrCb422ToYCbCr422_8u_C2P3R	613
7.46	Color Gamma Correction	614
7.46.1	Detailed Description	615
7.46.2	Function Documentation	615
7.46.2.1	nppiGammaFwd_8u_AC4IR	615

7.46.2.2	nppiGammaFwd_8u_AC4R	615
7.46.2.3	nppiGammaFwd_8u_C3IR	616
7.46.2.4	nppiGammaFwd_8u_C3R	616
7.46.2.5	nppiGammaFwd_8u_IP3R	616
7.46.2.6	nppiGammaFwd_8u_P3R	617
7.46.2.7	nppiGammaInv_8u_AC4IR	617
7.46.2.8	nppiGammaInv_8u_AC4R	617
7.46.2.9	nppiGammaInv_8u_C3IR	618
7.46.2.10	nppiGammaInv_8u_C3R	618
7.46.2.11	nppiGammaInv_8u_IP3R	618
7.46.2.12	nppiGammaInv_8u_P3R	619
7.47	Complement Color Key	620
7.47.1	Detailed Description	620
7.47.2	Function Documentation	620
7.47.2.1	nppiAlphaCompColorKey_8u_AC4R	620
7.47.2.2	nppiCompColorKey_8u_C1R	621
7.47.2.3	nppiCompColorKey_8u_C3R	621
7.47.2.4	nppiCompColorKey_8u_C4R	622
7.48	Color Processing	623
7.48.1	Detailed Description	637
7.48.2	Function Documentation	637
7.48.2.1	nppiColorTwist32f_16s_AC4IR	637
7.48.2.2	nppiColorTwist32f_16s_AC4R	638
7.48.2.3	nppiColorTwist32f_16s_C1IR	638
7.48.2.4	nppiColorTwist32f_16s_C1R	639
7.48.2.5	nppiColorTwist32f_16s_C2IR	639
7.48.2.6	nppiColorTwist32f_16s_C2R	639
7.48.2.7	nppiColorTwist32f_16s_C3IR	640
7.48.2.8	nppiColorTwist32f_16s_C3R	640
7.48.2.9	nppiColorTwist32f_16s_IP3R	641
7.48.2.10	nppiColorTwist32f_16s_P3R	641
7.48.2.11	nppiColorTwist32f_16u_AC4IR	641
7.48.2.12	nppiColorTwist32f_16u_AC4R	642
7.48.2.13	nppiColorTwist32f_16u_C1IR	642
7.48.2.14	nppiColorTwist32f_16u_C1R	642
7.48.2.15	nppiColorTwist32f_16u_C2IR	643

7.48.2.16 nppiColorTwist32f_16u_C2R	643
7.48.2.17 nppiColorTwist32f_16u_C3IR	644
7.48.2.18 nppiColorTwist32f_16u_C3R	644
7.48.2.19 nppiColorTwist32f_16u_IP3R	644
7.48.2.20 nppiColorTwist32f_16u_P3R	645
7.48.2.21 nppiColorTwist32f_8s_AC4IR	645
7.48.2.22 nppiColorTwist32f_8s_AC4R	645
7.48.2.23 nppiColorTwist32f_8s_C1IR	646
7.48.2.24 nppiColorTwist32f_8s_C1R	646
7.48.2.25 nppiColorTwist32f_8s_C2IR	647
7.48.2.26 nppiColorTwist32f_8s_C2R	647
7.48.2.27 nppiColorTwist32f_8s_C3IR	647
7.48.2.28 nppiColorTwist32f_8s_C3R	648
7.48.2.29 nppiColorTwist32f_8s_C4IR	648
7.48.2.30 nppiColorTwist32f_8s_C4R	648
7.48.2.31 nppiColorTwist32f_8s_IP3R	649
7.48.2.32 nppiColorTwist32f_8s_P3R	649
7.48.2.33 nppiColorTwist32f_8u_AC4IR	650
7.48.2.34 nppiColorTwist32f_8u_AC4R	650
7.48.2.35 nppiColorTwist32f_8u_C1IR	650
7.48.2.36 nppiColorTwist32f_8u_C1R	651
7.48.2.37 nppiColorTwist32f_8u_C2IR	651
7.48.2.38 nppiColorTwist32f_8u_C2R	652
7.48.2.39 nppiColorTwist32f_8u_C3IR	652
7.48.2.40 nppiColorTwist32f_8u_C3R	652
7.48.2.41 nppiColorTwist32f_8u_C4IR	653
7.48.2.42 nppiColorTwist32f_8u_C4R	653
7.48.2.43 nppiColorTwist32f_8u_IP3R	654
7.48.2.44 nppiColorTwist32f_8u_P3R	654
7.48.2.45 nppiColorTwist32fC_8u_C4IR	654
7.48.2.46 nppiColorTwist32fC_8u_C4R	655
7.48.2.47 nppiColorTwist_32f_AC4IR	655
7.48.2.48 nppiColorTwist_32f_AC4R	656
7.48.2.49 nppiColorTwist_32f_C1IR	656
7.48.2.50 nppiColorTwist_32f_C1R	657
7.48.2.51 nppiColorTwist_32f_C2IR	657

7.48.2.52 nppiColorTwist_32f_C2R	657
7.48.2.53 nppiColorTwist_32f_C3IR	658
7.48.2.54 nppiColorTwist_32f_C3R	658
7.48.2.55 nppiColorTwist_32f_C4IR	659
7.48.2.56 nppiColorTwist_32f_C4R	659
7.48.2.57 nppiColorTwist_32f_IP3R	659
7.48.2.58 nppiColorTwist_32f_P3R	660
7.48.2.59 nppiColorTwist_32fC_C4IR	660
7.48.2.60 nppiColorTwist_32fC_C4R	661
7.48.2.61 nppiLUT_16s_AC4IR	661
7.48.2.62 nppiLUT_16s_AC4R	662
7.48.2.63 nppiLUT_16s_C1IR	662
7.48.2.64 nppiLUT_16s_C1R	663
7.48.2.65 nppiLUT_16s_C3IR	663
7.48.2.66 nppiLUT_16s_C3R	664
7.48.2.67 nppiLUT_16s_C4IR	664
7.48.2.68 nppiLUT_16s_C4R	665
7.48.2.69 nppiLUT_16u_AC4IR	665
7.48.2.70 nppiLUT_16u_AC4R	666
7.48.2.71 nppiLUT_16u_C1IR	666
7.48.2.72 nppiLUT_16u_C1R	667
7.48.2.73 nppiLUT_16u_C3IR	667
7.48.2.74 nppiLUT_16u_C3R	668
7.48.2.75 nppiLUT_16u_C4IR	668
7.48.2.76 nppiLUT_16u_C4R	669
7.48.2.77 nppiLUT_32f_AC4IR	669
7.48.2.78 nppiLUT_32f_AC4R	670
7.48.2.79 nppiLUT_32f_C1IR	670
7.48.2.80 nppiLUT_32f_C1R	671
7.48.2.81 nppiLUT_32f_C3IR	671
7.48.2.82 nppiLUT_32f_C3R	672
7.48.2.83 nppiLUT_32f_C4IR	672
7.48.2.84 nppiLUT_32f_C4R	673
7.48.2.85 nppiLUT_8u_AC4IR	673
7.48.2.86 nppiLUT_8u_AC4R	674
7.48.2.87 nppiLUT_8u_C1IR	674

7.48.2.88 nppiLUT_8u_C1R	675
7.48.2.89 nppiLUT_8u_C3IR	675
7.48.2.90 nppiLUT_8u_C3R	676
7.48.2.91 nppiLUT_8u_C4IR	676
7.48.2.92 nppiLUT_8u_C4R	677
7.48.2.93 nppiLUT_Cubic_16s_AC4IR	677
7.48.2.94 nppiLUT_Cubic_16s_AC4R	678
7.48.2.95 nppiLUT_Cubic_16s_C1IR	678
7.48.2.96 nppiLUT_Cubic_16s_C1R	679
7.48.2.97 nppiLUT_Cubic_16s_C3IR	679
7.48.2.98 nppiLUT_Cubic_16s_C3R	680
7.48.2.99 nppiLUT_Cubic_16s_C4IR	680
7.48.2.100 nppiLUT_Cubic_16s_C4R	681
7.48.2.101 nppiLUT_Cubic_16u_AC4IR	681
7.48.2.102 nppiLUT_Cubic_16u_AC4R	682
7.48.2.103 nppiLUT_Cubic_16u_C1IR	682
7.48.2.104 nppiLUT_Cubic_16u_C1R	683
7.48.2.105 nppiLUT_Cubic_16u_C3IR	683
7.48.2.106 nppiLUT_Cubic_16u_C3R	684
7.48.2.107 nppiLUT_Cubic_16u_C4IR	684
7.48.2.108 nppiLUT_Cubic_16u_C4R	685
7.48.2.109 nppiLUT_Cubic_32f_AC4IR	685
7.48.2.110 nppiLUT_Cubic_32f_AC4R	686
7.48.2.111 nppiLUT_Cubic_32f_C1IR	686
7.48.2.112 nppiLUT_Cubic_32f_C1R	687
7.48.2.113 nppiLUT_Cubic_32f_C3IR	687
7.48.2.114 nppiLUT_Cubic_32f_C3R	688
7.48.2.115 nppiLUT_Cubic_32f_C4IR	688
7.48.2.116 nppiLUT_Cubic_32f_C4R	689
7.48.2.117 nppiLUT_Cubic_8u_AC4IR	689
7.48.2.118 nppiLUT_Cubic_8u_AC4R	690
7.48.2.119 nppiLUT_Cubic_8u_C1IR	690
7.48.2.120 nppiLUT_Cubic_8u_C1R	691
7.48.2.121 nppiLUT_Cubic_8u_C3IR	691
7.48.2.122 nppiLUT_Cubic_8u_C3R	692
7.48.2.123 nppiLUT_Cubic_8u_C4IR	692

7.48.2.124nppiLUT_Cubic_8u_C4R	693
7.48.2.125nppiLUT_Linear_16s_AC4IR	693
7.48.2.126nppiLUT_Linear_16s_AC4R	694
7.48.2.127nppiLUT_Linear_16s_C1IR	694
7.48.2.128nppiLUT_Linear_16s_C1R	695
7.48.2.129nppiLUT_Linear_16s_C3IR	695
7.48.2.130nppiLUT_Linear_16s_C3R	696
7.48.2.131nppiLUT_Linear_16s_C4IR	696
7.48.2.132nppiLUT_Linear_16s_C4R	697
7.48.2.133nppiLUT_Linear_16u_AC4IR	697
7.48.2.134nppiLUT_Linear_16u_AC4R	698
7.48.2.135nppiLUT_Linear_16u_C1IR	698
7.48.2.136nppiLUT_Linear_16u_C1R	699
7.48.2.137nppiLUT_Linear_16u_C3IR	699
7.48.2.138nppiLUT_Linear_16u_C3R	700
7.48.2.139nppiLUT_Linear_16u_C4IR	700
7.48.2.140nppiLUT_Linear_16u_C4R	701
7.48.2.141nppiLUT_Linear_32f_AC4IR	701
7.48.2.142nppiLUT_Linear_32f_AC4R	702
7.48.2.143nppiLUT_Linear_32f_C1IR	702
7.48.2.144nppiLUT_Linear_32f_C1R	703
7.48.2.145nppiLUT_Linear_32f_C3IR	703
7.48.2.146nppiLUT_Linear_32f_C3R	704
7.48.2.147nppiLUT_Linear_32f_C4IR	704
7.48.2.148nppiLUT_Linear_32f_C4R	705
7.48.2.149nppiLUT_Linear_8u_AC4IR	705
7.48.2.150nppiLUT_Linear_8u_AC4R	706
7.48.2.151nppiLUT_Linear_8u_C1IR	706
7.48.2.152nppiLUT_Linear_8u_C1R	707
7.48.2.153nppiLUT_Linear_8u_C3IR	707
7.48.2.154nppiLUT_Linear_8u_C3R	708
7.48.2.155nppiLUT_Linear_8u_C4IR	708
7.48.2.156nppiLUT_Linear_8u_C4R	709
7.48.2.157nppiLUT_Trilinear_8u_AC4IR	710
7.48.2.158nppiLUT_Trilinear_8u_AC4R	710
7.48.2.159nppiLUT_Trilinear_8u_C4R	711

7.48.2.160	nppiLUTPalette_16u24u_C1R	711
7.48.2.161	nppiLUTPalette_16u32u_C1R	712
7.48.2.162	nppiLUTPalette_16u8u_C1R	712
7.48.2.163	nppiLUTPalette_16u_AC4R	713
7.48.2.164	nppiLUTPalette_16u_C1R	713
7.48.2.165	nppiLUTPalette_16u_C3R	714
7.48.2.166	nppiLUTPalette_16u_C4R	714
7.48.2.167	nppiLUTPalette_8u24u_C1R	715
7.48.2.168	nppiLUTPalette_8u32u_C1R	715
7.48.2.169	nppiLUTPalette_8u_AC4R	716
7.48.2.170	nppiLUTPalette_8u_C1R	716
7.48.2.171	nppiLUTPalette_8u_C3R	717
7.48.2.172	nppiLUTPalette_8u_C4R	717
7.48.2.173	nppiLUTPaletteSwap_16u_C3A0C4R	718
7.48.2.174	nppiLUTPaletteSwap_8u_C3A0C4R	719
7.49	Compression	720
7.49.1	Detailed Description	720
7.49.2	Typedef Documentation	721
7.49.2.1	NppiDecodeHuffmanSpec	721
7.49.3	Function Documentation	721
7.49.3.1	nppiDecodeHuffmanScanHost_JPEG_8u16s_P1R	721
7.49.3.2	nppiDecodeHuffmanScanHost_JPEG_8u16s_P3R	721
7.49.3.3	nppiDecodeHuffmanSpecFreeHost_JPEG	722
7.49.3.4	nppiDecodeHuffmanSpecGetBufSize_JPEG	722
7.49.3.5	nppiDecodeHuffmanSpecInitAllocHost_JPEG	723
7.49.3.6	nppiDecodeHuffmanSpecInitHost_JPEG	723
7.50	Quantization Functions	724
7.50.1	Typedef Documentation	725
7.50.1.1	NppiDCTState	725
7.50.2	Function Documentation	725
7.50.2.1	nppiDCTFree	725
7.50.2.2	nppiDCTInitAlloc	725
7.50.2.3	nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R	725
7.50.2.4	nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R_NEW	726
7.50.2.5	nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R	726
7.50.2.6	nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R_NEW	727

7.50.2.7	<code>nppiQuantFwdRawTableInit_JPEG_8u</code>	728
7.50.2.8	<code>nppiQuantFwdTableInit_JPEG_8u16u</code>	728
7.50.2.9	<code>nppiQuantInvTableInit_JPEG_8u16u</code>	728
7.51	Labeling and Segmentation	730
7.51.1	Detailed Description	730
7.51.2	Typedef Documentation	730
7.51.2.1	<code>NppiGraphcutState</code>	730
7.52	GraphCut	731
7.52.1	Function Documentation	732
7.52.1.1	<code>nppiGraphcut8_32f8u</code>	732
7.52.1.2	<code>nppiGraphcut8_32s8u</code>	733
7.52.1.3	<code>nppiGraphcut8GetSize</code>	734
7.52.1.4	<code>nppiGraphcut8InitAlloc</code>	734
7.52.1.5	<code>nppiGraphcut_32f8u</code>	734
7.52.1.6	<code>nppiGraphcut_32s8u</code>	735
7.52.1.7	<code>nppiGraphcutFree</code>	736
7.52.1.8	<code>nppiGraphcutGetSize</code>	736
7.52.1.9	<code>nppiGraphcutInitAlloc</code>	737
7.53	Data Exchange and Initialization	738
7.53.1	Detailed Description	738
7.54	Set	739
7.54.1	Detailed Description	745
7.54.2	Function Documentation	745
7.54.2.1	<code>nppiSet_16s_AC4MR</code>	745
7.54.2.2	<code>nppiSet_16s_AC4R</code>	746
7.54.2.3	<code>nppiSet_16s_C1MR</code>	746
7.54.2.4	<code>nppiSet_16s_C1R</code>	746
7.54.2.5	<code>nppiSet_16s_C2R</code>	747
7.54.2.6	<code>nppiSet_16s_C3CR</code>	747
7.54.2.7	<code>nppiSet_16s_C3MR</code>	747
7.54.2.8	<code>nppiSet_16s_C3R</code>	748
7.54.2.9	<code>nppiSet_16s_C4CR</code>	748
7.54.2.10	<code>nppiSet_16s_C4MR</code>	748
7.54.2.11	<code>nppiSet_16s_C4R</code>	749
7.54.2.12	<code>nppiSet_16sc_AC4R</code>	749
7.54.2.13	<code>nppiSet_16sc_C1R</code>	749

7.54.2.14 nppiSet_16sc_C2R	750
7.54.2.15 nppiSet_16sc_C3R	750
7.54.2.16 nppiSet_16sc_C4R	750
7.54.2.17 nppiSet_16u_AC4MR	751
7.54.2.18 nppiSet_16u_AC4R	751
7.54.2.19 nppiSet_16u_C1MR	751
7.54.2.20 nppiSet_16u_C1R	752
7.54.2.21 nppiSet_16u_C2R	752
7.54.2.22 nppiSet_16u_C3CR	752
7.54.2.23 nppiSet_16u_C3MR	753
7.54.2.24 nppiSet_16u_C3R	753
7.54.2.25 nppiSet_16u_C4CR	753
7.54.2.26 nppiSet_16u_C4MR	754
7.54.2.27 nppiSet_16u_C4R	754
7.54.2.28 nppiSet_32f_AC4MR	754
7.54.2.29 nppiSet_32f_AC4R	755
7.54.2.30 nppiSet_32f_C1MR	755
7.54.2.31 nppiSet_32f_C1R	756
7.54.2.32 nppiSet_32f_C2R	756
7.54.2.33 nppiSet_32f_C3CR	756
7.54.2.34 nppiSet_32f_C3MR	757
7.54.2.35 nppiSet_32f_C3R	757
7.54.2.36 nppiSet_32f_C4CR	757
7.54.2.37 nppiSet_32f_C4MR	758
7.54.2.38 nppiSet_32f_C4R	758
7.54.2.39 nppiSet_32fc_AC4R	758
7.54.2.40 nppiSet_32fc_C1R	759
7.54.2.41 nppiSet_32fc_C2R	759
7.54.2.42 nppiSet_32fc_C3R	759
7.54.2.43 nppiSet_32fc_C4R	760
7.54.2.44 nppiSet_32s_AC4MR	760
7.54.2.45 nppiSet_32s_AC4R	760
7.54.2.46 nppiSet_32s_C1MR	761
7.54.2.47 nppiSet_32s_C1R	761
7.54.2.48 nppiSet_32s_C2R	761
7.54.2.49 nppiSet_32s_C3CR	762

7.54.2.50	nppiSet_32s_C3MR	762
7.54.2.51	nppiSet_32s_C3R	762
7.54.2.52	nppiSet_32s_C4CR	763
7.54.2.53	nppiSet_32s_C4MR	763
7.54.2.54	nppiSet_32s_C4R	763
7.54.2.55	nppiSet_32sc_AC4R	764
7.54.2.56	nppiSet_32sc_C1R	764
7.54.2.57	nppiSet_32sc_C2R	764
7.54.2.58	nppiSet_32sc_C3R	765
7.54.2.59	nppiSet_32sc_C4R	765
7.54.2.60	nppiSet_32u_AC4R	765
7.54.2.61	nppiSet_32u_C1R	766
7.54.2.62	nppiSet_32u_C2R	766
7.54.2.63	nppiSet_32u_C3R	766
7.54.2.64	nppiSet_32u_C4R	767
7.54.2.65	nppiSet_8s_AC4R	767
7.54.2.66	nppiSet_8s_C1R	767
7.54.2.67	nppiSet_8s_C2R	768
7.54.2.68	nppiSet_8s_C3R	768
7.54.2.69	nppiSet_8s_C4R	768
7.54.2.70	nppiSet_8u_AC4MR	769
7.54.2.71	nppiSet_8u_AC4R	769
7.54.2.72	nppiSet_8u_C1MR	769
7.54.2.73	nppiSet_8u_C1R	770
7.54.2.74	nppiSet_8u_C2R	770
7.54.2.75	nppiSet_8u_C3CR	770
7.54.2.76	nppiSet_8u_C3MR	771
7.54.2.77	nppiSet_8u_C3R	771
7.54.2.78	nppiSet_8u_C4CR	771
7.54.2.79	nppiSet_8u_C4MR	772
7.54.2.80	nppiSet_8u_C4R	772
7.55	Copy	773
7.55.1	Function Documentation	782
7.55.1.1	nppiCopy_16s_AC4MR	782
7.55.1.2	nppiCopy_16s_AC4R	783
7.55.1.3	nppiCopy_16s_C1C3R	783

7.55.1.4	nppiCopy_16s_C1C4R	784
7.55.1.5	nppiCopy_16s_C1MR	784
7.55.1.6	nppiCopy_16s_C1R	784
7.55.1.7	nppiCopy_16s_C3C1R	785
7.55.1.8	nppiCopy_16s_C3CR	785
7.55.1.9	nppiCopy_16s_C3MR	785
7.55.1.10	nppiCopy_16s_C3P3R	786
7.55.1.11	nppiCopy_16s_C3R	786
7.55.1.12	nppiCopy_16s_C4C1R	786
7.55.1.13	nppiCopy_16s_C4CR	787
7.55.1.14	nppiCopy_16s_C4MR	787
7.55.1.15	nppiCopy_16s_C4P4R	787
7.55.1.16	nppiCopy_16s_C4R	788
7.55.1.17	nppiCopy_16s_P3C3R	788
7.55.1.18	nppiCopy_16s_P4C4R	788
7.55.1.19	nppiCopy_16sc_AC4R	789
7.55.1.20	nppiCopy_16sc_C1R	789
7.55.1.21	nppiCopy_16sc_C2R	789
7.55.1.22	nppiCopy_16sc_C3R	790
7.55.1.23	nppiCopy_16sc_C4R	790
7.55.1.24	nppiCopy_16u_AC4MR	790
7.55.1.25	nppiCopy_16u_AC4R	791
7.55.1.26	nppiCopy_16u_C1C3R	791
7.55.1.27	nppiCopy_16u_C1C4R	791
7.55.1.28	nppiCopy_16u_C1MR	792
7.55.1.29	nppiCopy_16u_C1R	792
7.55.1.30	nppiCopy_16u_C3C1R	792
7.55.1.31	nppiCopy_16u_C3CR	793
7.55.1.32	nppiCopy_16u_C3MR	793
7.55.1.33	nppiCopy_16u_C3P3R	793
7.55.1.34	nppiCopy_16u_C3R	794
7.55.1.35	nppiCopy_16u_C4C1R	794
7.55.1.36	nppiCopy_16u_C4CR	794
7.55.1.37	nppiCopy_16u_C4MR	795
7.55.1.38	nppiCopy_16u_C4P4R	795
7.55.1.39	nppiCopy_16u_C4R	795

7.55.1.40 nppiCopy_16u_P3C3R	796
7.55.1.41 nppiCopy_16u_P4C4R	796
7.55.1.42 nppiCopy_32f_AC4MR	796
7.55.1.43 nppiCopy_32f_AC4R	797
7.55.1.44 nppiCopy_32f_C1C3R	797
7.55.1.45 nppiCopy_32f_C1C4R	797
7.55.1.46 nppiCopy_32f_C1MR	798
7.55.1.47 nppiCopy_32f_C1R	798
7.55.1.48 nppiCopy_32f_C3C1R	798
7.55.1.49 nppiCopy_32f_C3CR	799
7.55.1.50 nppiCopy_32f_C3MR	799
7.55.1.51 nppiCopy_32f_C3P3R	799
7.55.1.52 nppiCopy_32f_C3R	800
7.55.1.53 nppiCopy_32f_C4C1R	800
7.55.1.54 nppiCopy_32f_C4CR	800
7.55.1.55 nppiCopy_32f_C4MR	801
7.55.1.56 nppiCopy_32f_C4P4R	801
7.55.1.57 nppiCopy_32f_C4R	801
7.55.1.58 nppiCopy_32f_P3C3R	802
7.55.1.59 nppiCopy_32f_P4C4R	802
7.55.1.60 nppiCopy_32fc_AC4R	802
7.55.1.61 nppiCopy_32fc_C1R	803
7.55.1.62 nppiCopy_32fc_C2R	803
7.55.1.63 nppiCopy_32fc_C3R	803
7.55.1.64 nppiCopy_32fc_C4R	804
7.55.1.65 nppiCopy_32s_AC4MR	804
7.55.1.66 nppiCopy_32s_AC4R	804
7.55.1.67 nppiCopy_32s_C1C3R	805
7.55.1.68 nppiCopy_32s_C1C4R	805
7.55.1.69 nppiCopy_32s_C1MR	805
7.55.1.70 nppiCopy_32s_C1R	806
7.55.1.71 nppiCopy_32s_C3C1R	806
7.55.1.72 nppiCopy_32s_C3CR	806
7.55.1.73 nppiCopy_32s_C3MR	807
7.55.1.74 nppiCopy_32s_C3P3R	807
7.55.1.75 nppiCopy_32s_C3R	807

7.55.1.76 nppiCopy_32s_C4C1R	808
7.55.1.77 nppiCopy_32s_C4CR	808
7.55.1.78 nppiCopy_32s_C4MR	808
7.55.1.79 nppiCopy_32s_C4P4R	809
7.55.1.80 nppiCopy_32s_C4R	809
7.55.1.81 nppiCopy_32s_P3C3R	809
7.55.1.82 nppiCopy_32s_P4C4R	810
7.55.1.83 nppiCopy_32sc_AC4R	810
7.55.1.84 nppiCopy_32sc_C1R	810
7.55.1.85 nppiCopy_32sc_C2R	811
7.55.1.86 nppiCopy_32sc_C3R	811
7.55.1.87 nppiCopy_32sc_C4R	811
7.55.1.88 nppiCopy_8s_AC4R	812
7.55.1.89 nppiCopy_8s_C1R	812
7.55.1.90 nppiCopy_8s_C2R	812
7.55.1.91 nppiCopy_8s_C3R	813
7.55.1.92 nppiCopy_8s_C4R	813
7.55.1.93 nppiCopy_8u_AC4MR	813
7.55.1.94 nppiCopy_8u_AC4R	814
7.55.1.95 nppiCopy_8u_C1C3R	814
7.55.1.96 nppiCopy_8u_C1C4R	814
7.55.1.97 nppiCopy_8u_C1MR	815
7.55.1.98 nppiCopy_8u_C1R	815
7.55.1.99 nppiCopy_8u_C3C1R	815
7.55.1.100 nppiCopy_8u_C3CR	816
7.55.1.101 nppiCopy_8u_C3MR	816
7.55.1.102 nppiCopy_8u_C3P3R	816
7.55.1.103 nppiCopy_8u_C3R	817
7.55.1.104 nppiCopy_8u_C4C1R	817
7.55.1.105 nppiCopy_8u_C4CR	817
7.55.1.106 nppiCopy_8u_C4MR	818
7.55.1.107 nppiCopy_8u_C4P4R	818
7.55.1.108 nppiCopy_8u_C4R	818
7.55.1.109 nppiCopy_8u_P3C3R	819
7.55.1.110 nppiCopy_8u_P4C4R	819
7.56 Convert	820

7.56.1	Function Documentation	828
7.56.1.1	nppiConvert_16s16u_C1Rs	828
7.56.1.2	nppiConvert_16s32f_AC4R	828
7.56.1.3	nppiConvert_16s32f_C1R	829
7.56.1.4	nppiConvert_16s32f_C3R	829
7.56.1.5	nppiConvert_16s32f_C4R	829
7.56.1.6	nppiConvert_16s32s_AC4R	830
7.56.1.7	nppiConvert_16s32s_C1R	830
7.56.1.8	nppiConvert_16s32s_C3R	830
7.56.1.9	nppiConvert_16s32s_C4R	831
7.56.1.10	nppiConvert_16s32u_C1Rs	831
7.56.1.11	nppiConvert_16s8s_C1RSfs	831
7.56.1.12	nppiConvert_16s8u_AC4R	832
7.56.1.13	nppiConvert_16s8u_C1R	832
7.56.1.14	nppiConvert_16s8u_C3R	833
7.56.1.15	nppiConvert_16s8u_C4R	833
7.56.1.16	nppiConvert_16u16s_C1RSfs	833
7.56.1.17	nppiConvert_16u32f_AC4R	834
7.56.1.18	nppiConvert_16u32f_C1R	834
7.56.1.19	nppiConvert_16u32f_C3R	834
7.56.1.20	nppiConvert_16u32f_C4R	835
7.56.1.21	nppiConvert_16u32s_AC4R	835
7.56.1.22	nppiConvert_16u32s_C1R	835
7.56.1.23	nppiConvert_16u32s_C3R	836
7.56.1.24	nppiConvert_16u32s_C4R	836
7.56.1.25	nppiConvert_16u32u_C1R	836
7.56.1.26	nppiConvert_16u8s_C1RSfs	837
7.56.1.27	nppiConvert_16u8u_AC4R	837
7.56.1.28	nppiConvert_16u8u_C1R	837
7.56.1.29	nppiConvert_16u8u_C3R	838
7.56.1.30	nppiConvert_16u8u_C4R	838
7.56.1.31	nppiConvert_32f16s_AC4R	838
7.56.1.32	nppiConvert_32f16s_C1R	839
7.56.1.33	nppiConvert_32f16s_C1RSfs	839
7.56.1.34	nppiConvert_32f16s_C3R	839
7.56.1.35	nppiConvert_32f16s_C4R	840

7.56.1.36 nppiConvert_32f16u_AC4R	840
7.56.1.37 nppiConvert_32f16u_C1R	841
7.56.1.38 nppiConvert_32f16u_C1RSfs	841
7.56.1.39 nppiConvert_32f16u_C3R	841
7.56.1.40 nppiConvert_32f16u_C4R	842
7.56.1.41 nppiConvert_32f32s_C1RSfs	842
7.56.1.42 nppiConvert_32f32u_C1RSfs	843
7.56.1.43 nppiConvert_32f8s_AC4R	843
7.56.1.44 nppiConvert_32f8s_C1R	843
7.56.1.45 nppiConvert_32f8s_C1RSfs	844
7.56.1.46 nppiConvert_32f8s_C3R	844
7.56.1.47 nppiConvert_32f8s_C4R	845
7.56.1.48 nppiConvert_32f8u_AC4R	845
7.56.1.49 nppiConvert_32f8u_C1R	845
7.56.1.50 nppiConvert_32f8u_C1RSfs	846
7.56.1.51 nppiConvert_32f8u_C3R	846
7.56.1.52 nppiConvert_32f8u_C4R	846
7.56.1.53 nppiConvert_32s16s_C1RSfs	847
7.56.1.54 nppiConvert_32s16u_C1RSfs	847
7.56.1.55 nppiConvert_32s32f_C1R	848
7.56.1.56 nppiConvert_32s32u_C1Rs	848
7.56.1.57 nppiConvert_32s8s_AC4R	848
7.56.1.58 nppiConvert_32s8s_C1R	849
7.56.1.59 nppiConvert_32s8s_C3R	849
7.56.1.60 nppiConvert_32s8s_C4R	849
7.56.1.61 nppiConvert_32s8u_AC4R	850
7.56.1.62 nppiConvert_32s8u_C1R	850
7.56.1.63 nppiConvert_32s8u_C3R	850
7.56.1.64 nppiConvert_32s8u_C4R	851
7.56.1.65 nppiConvert_32u16s_C1RSfs	851
7.56.1.66 nppiConvert_32u16u_C1RSfs	851
7.56.1.67 nppiConvert_32u32f_C1R	852
7.56.1.68 nppiConvert_32u32s_C1RSfs	852
7.56.1.69 nppiConvert_32u8s_C1RSfs	853
7.56.1.70 nppiConvert_32u8u_C1RSfs	853
7.56.1.71 nppiConvert_8s16s_C1R	853

7.56.1.72	nppiConvert_8s16u_C1Rs	854
7.56.1.73	nppiConvert_8s32f_AC4R	854
7.56.1.74	nppiConvert_8s32f_C1R	854
7.56.1.75	nppiConvert_8s32f_C3R	855
7.56.1.76	nppiConvert_8s32f_C4R	855
7.56.1.77	nppiConvert_8s32s_AC4R	856
7.56.1.78	nppiConvert_8s32s_C1R	856
7.56.1.79	nppiConvert_8s32s_C3R	856
7.56.1.80	nppiConvert_8s32s_C4R	857
7.56.1.81	nppiConvert_8s32u_C1Rs	857
7.56.1.82	nppiConvert_8s8u_C1Rs	857
7.56.1.83	nppiConvert_8u16s_AC4R	858
7.56.1.84	nppiConvert_8u16s_C1R	858
7.56.1.85	nppiConvert_8u16s_C3R	858
7.56.1.86	nppiConvert_8u16s_C4R	859
7.56.1.87	nppiConvert_8u16u_AC4R	859
7.56.1.88	nppiConvert_8u16u_C1R	859
7.56.1.89	nppiConvert_8u16u_C3R	860
7.56.1.90	nppiConvert_8u16u_C4R	860
7.56.1.91	nppiConvert_8u32f_AC4R	860
7.56.1.92	nppiConvert_8u32f_C1R	861
7.56.1.93	nppiConvert_8u32f_C3R	861
7.56.1.94	nppiConvert_8u32f_C4R	861
7.56.1.95	nppiConvert_8u32s_AC4R	862
7.56.1.96	nppiConvert_8u32s_C1R	862
7.56.1.97	nppiConvert_8u32s_C3R	862
7.56.1.98	nppiConvert_8u32s_C4R	863
7.56.1.99	nppiConvert_8u8s_C1RSfs	863
7.57	Scale	864
7.57.1	Function Documentation	867
7.57.1.1	nppiScale_16s8u_AC4R	867
7.57.1.2	nppiScale_16s8u_C1R	867
7.57.1.3	nppiScale_16s8u_C3R	867
7.57.1.4	nppiScale_16s8u_C4R	868
7.57.1.5	nppiScale_16u8u_AC4R	868
7.57.1.6	nppiScale_16u8u_C1R	869

7.57.1.7	nppiScale_16u8u_C3R	869
7.57.1.8	nppiScale_16u8u_C4R	869
7.57.1.9	nppiScale_32f8u_AC4R	870
7.57.1.10	nppiScale_32f8u_C1R	870
7.57.1.11	nppiScale_32f8u_C3R	870
7.57.1.12	nppiScale_32f8u_C4R	871
7.57.1.13	nppiScale_32s8u_AC4R	871
7.57.1.14	nppiScale_32s8u_C1R	872
7.57.1.15	nppiScale_32s8u_C3R	872
7.57.1.16	nppiScale_32s8u_C4R	872
7.57.1.17	nppiScale_8u16s_AC4R	873
7.57.1.18	nppiScale_8u16s_C1R	873
7.57.1.19	nppiScale_8u16s_C3R	873
7.57.1.20	nppiScale_8u16s_C4R	874
7.57.1.21	nppiScale_8u16u_AC4R	874
7.57.1.22	nppiScale_8u16u_C1R	874
7.57.1.23	nppiScale_8u16u_C3R	875
7.57.1.24	nppiScale_8u16u_C4R	875
7.57.1.25	nppiScale_8u32f_AC4R	875
7.57.1.26	nppiScale_8u32f_C1R	876
7.57.1.27	nppiScale_8u32f_C3R	876
7.57.1.28	nppiScale_8u32f_C4R	876
7.57.1.29	nppiScale_8u32s_AC4R	877
7.57.1.30	nppiScale_8u32s_C1R	877
7.57.1.31	nppiScale_8u32s_C3R	878
7.57.1.32	nppiScale_8u32s_C4R	878
7.58	Copy Constant Border	879
7.58.1	Function Documentation	881
7.58.1.1	nppiCopyConstBorder_16s_AC4R	881
7.58.1.2	nppiCopyConstBorder_16s_C1R	881
7.58.1.3	nppiCopyConstBorder_16s_C3R	882
7.58.1.4	nppiCopyConstBorder_16s_C4R	882
7.58.1.5	nppiCopyConstBorder_16u_AC4R	883
7.58.1.6	nppiCopyConstBorder_16u_C1R	883
7.58.1.7	nppiCopyConstBorder_16u_C3R	884
7.58.1.8	nppiCopyConstBorder_16u_C4R	884

7.58.1.9	nppiCopyConstBorder_32f_AC4R	885
7.58.1.10	nppiCopyConstBorder_32f_C1R	885
7.58.1.11	nppiCopyConstBorder_32f_C3R	886
7.58.1.12	nppiCopyConstBorder_32f_C4R	886
7.58.1.13	nppiCopyConstBorder_32s_AC4R	887
7.58.1.14	nppiCopyConstBorder_32s_C1R	887
7.58.1.15	nppiCopyConstBorder_32s_C3R	888
7.58.1.16	nppiCopyConstBorder_32s_C4R	888
7.58.1.17	nppiCopyConstBorder_8u_AC4R	889
7.58.1.18	nppiCopyConstBorder_8u_C1R	889
7.58.1.19	nppiCopyConstBorder_8u_C3R	890
7.58.1.20	nppiCopyConstBorder_8u_C4R	890
7.59	Copy Replicate Border	892
7.59.1	Function Documentation	894
7.59.1.1	nppiCopyReplicateBorder_16s_AC4R	894
7.59.1.2	nppiCopyReplicateBorder_16s_C1R	894
7.59.1.3	nppiCopyReplicateBorder_16s_C3R	895
7.59.1.4	nppiCopyReplicateBorder_16s_C4R	895
7.59.1.5	nppiCopyReplicateBorder_16u_AC4R	896
7.59.1.6	nppiCopyReplicateBorder_16u_C1R	896
7.59.1.7	nppiCopyReplicateBorder_16u_C3R	897
7.59.1.8	nppiCopyReplicateBorder_16u_C4R	897
7.59.1.9	nppiCopyReplicateBorder_32f_AC4R	898
7.59.1.10	nppiCopyReplicateBorder_32f_C1R	898
7.59.1.11	nppiCopyReplicateBorder_32f_C3R	899
7.59.1.12	nppiCopyReplicateBorder_32f_C4R	899
7.59.1.13	nppiCopyReplicateBorder_32s_AC4R	900
7.59.1.14	nppiCopyReplicateBorder_32s_C1R	900
7.59.1.15	nppiCopyReplicateBorder_32s_C3R	901
7.59.1.16	nppiCopyReplicateBorder_32s_C4R	901
7.59.1.17	nppiCopyReplicateBorder_8u_AC4R	902
7.59.1.18	nppiCopyReplicateBorder_8u_C1R	902
7.59.1.19	nppiCopyReplicateBorder_8u_C3R	903
7.59.1.20	nppiCopyReplicateBorder_8u_C4R	903
7.60	Copy Wrap Border	904
7.60.1	Function Documentation	906

7.60.1.1	nppiCopyWrapBorder_16s_AC4R	906
7.60.1.2	nppiCopyWrapBorder_16s_C1R	907
7.60.1.3	nppiCopyWrapBorder_16s_C3R	907
7.60.1.4	nppiCopyWrapBorder_16s_C4R	908
7.60.1.5	nppiCopyWrapBorder_16u_AC4R	908
7.60.1.6	nppiCopyWrapBorder_16u_C1R	909
7.60.1.7	nppiCopyWrapBorder_16u_C3R	909
7.60.1.8	nppiCopyWrapBorder_16u_C4R	910
7.60.1.9	nppiCopyWrapBorder_32f_AC4R	910
7.60.1.10	nppiCopyWrapBorder_32f_C1R	911
7.60.1.11	nppiCopyWrapBorder_32f_C3R	911
7.60.1.12	nppiCopyWrapBorder_32f_C4R	912
7.60.1.13	nppiCopyWrapBorder_32s_AC4R	912
7.60.1.14	nppiCopyWrapBorder_32s_C1R	913
7.60.1.15	nppiCopyWrapBorder_32s_C3R	913
7.60.1.16	nppiCopyWrapBorder_32s_C4R	914
7.60.1.17	nppiCopyWrapBorder_8u_AC4R	914
7.60.1.18	nppiCopyWrapBorder_8u_C1R	915
7.60.1.19	nppiCopyWrapBorder_8u_C3R	915
7.60.1.20	nppiCopyWrapBorder_8u_C4R	916
7.61	Copy Sub-Pixel	917
7.61.1	Function Documentation	918
7.61.1.1	nppiCopySubpix_16s_AC4R	918
7.61.1.2	nppiCopySubpix_16s_C1R	919
7.61.1.3	nppiCopySubpix_16s_C3R	919
7.61.1.4	nppiCopySubpix_16s_C4R	920
7.61.1.5	nppiCopySubpix_16u_AC4R	920
7.61.1.6	nppiCopySubpix_16u_C1R	921
7.61.1.7	nppiCopySubpix_16u_C3R	921
7.61.1.8	nppiCopySubpix_16u_C4R	921
7.61.1.9	nppiCopySubpix_32f_AC4R	922
7.61.1.10	nppiCopySubpix_32f_C1R	922
7.61.1.11	nppiCopySubpix_32f_C3R	923
7.61.1.12	nppiCopySubpix_32f_C4R	923
7.61.1.13	nppiCopySubpix_32s_AC4R	923
7.61.1.14	nppiCopySubpix_32s_C1R	924

7.61.1.15	nppiCopySubpix_32s_C3R	924
7.61.1.16	nppiCopySubpix_32s_C4R	925
7.61.1.17	nppiCopySubpix_8u_AC4R	925
7.61.1.18	nppiCopySubpix_8u_C1R	926
7.61.1.19	nppiCopySubpix_8u_C3R	926
7.61.1.20	nppiCopySubpix_8u_C4R	926
7.62	Duplicate Channel	928
7.62.1	Function Documentation	929
7.62.1.1	nppiDup_16s_C1AC4R	929
7.62.1.2	nppiDup_16s_C1C3R	929
7.62.1.3	nppiDup_16s_C1C4R	930
7.62.1.4	nppiDup_16u_C1AC4R	930
7.62.1.5	nppiDup_16u_C1C3R	931
7.62.1.6	nppiDup_16u_C1C4R	931
7.62.1.7	nppiDup_32f_C1AC4R	931
7.62.1.8	nppiDup_32f_C1C3R	932
7.62.1.9	nppiDup_32f_C1C4R	932
7.62.1.10	nppiDup_32s_C1AC4R	932
7.62.1.11	nppiDup_32s_C1C3R	933
7.62.1.12	nppiDup_32s_C1C4R	933
7.62.1.13	nppiDup_8u_C1AC4R	933
7.62.1.14	nppiDup_8u_C1C3R	934
7.62.1.15	nppiDup_8u_C1C4R	934
7.63	Transpose	935
7.63.1	Function Documentation	936
7.63.1.1	nppiTranspose_16s_C1R	936
7.63.1.2	nppiTranspose_16s_C3R	936
7.63.1.3	nppiTranspose_16s_C4R	937
7.63.1.4	nppiTranspose_16u_C1R	937
7.63.1.5	nppiTranspose_16u_C3R	937
7.63.1.6	nppiTranspose_16u_C4R	938
7.63.1.7	nppiTranspose_32f_C1R	938
7.63.1.8	nppiTranspose_32f_C3R	938
7.63.1.9	nppiTranspose_32f_C4R	939
7.63.1.10	nppiTranspose_32s_C1R	939
7.63.1.11	nppiTranspose_32s_C3R	940

7.63.1.12	<code>npptTranspose_32s_C4R</code>	940
7.63.1.13	<code>npptTranspose_8u_C1R</code>	940
7.63.1.14	<code>npptTranspose_8u_C3R</code>	941
7.63.1.15	<code>npptTranspose_8u_C4R</code>	941
7.64	Swap Channels	942
7.64.1	Function Documentation	945
7.64.1.1	<code>npptSwapChannels_16s_AC4R</code>	945
7.64.1.2	<code>npptSwapChannels_16s_C3C4R</code>	945
7.64.1.3	<code>npptSwapChannels_16s_C3IR</code>	946
7.64.1.4	<code>npptSwapChannels_16s_C3R</code>	946
7.64.1.5	<code>npptSwapChannels_16s_C4C3R</code>	946
7.64.1.6	<code>npptSwapChannels_16s_C4IR</code>	947
7.64.1.7	<code>npptSwapChannels_16s_C4R</code>	947
7.64.1.8	<code>npptSwapChannels_16u_AC4R</code>	948
7.64.1.9	<code>npptSwapChannels_16u_C3C4R</code>	948
7.64.1.10	<code>npptSwapChannels_16u_C3IR</code>	949
7.64.1.11	<code>npptSwapChannels_16u_C3R</code>	949
7.64.1.12	<code>npptSwapChannels_16u_C4C3R</code>	949
7.64.1.13	<code>npptSwapChannels_16u_C4IR</code>	950
7.64.1.14	<code>npptSwapChannels_16u_C4R</code>	950
7.64.1.15	<code>npptSwapChannels_32f_AC4R</code>	951
7.64.1.16	<code>npptSwapChannels_32f_C3C4R</code>	951
7.64.1.17	<code>npptSwapChannels_32f_C3IR</code>	952
7.64.1.18	<code>npptSwapChannels_32f_C3R</code>	952
7.64.1.19	<code>npptSwapChannels_32f_C4C3R</code>	952
7.64.1.20	<code>npptSwapChannels_32f_C4IR</code>	953
7.64.1.21	<code>npptSwapChannels_32f_C4R</code>	953
7.64.1.22	<code>npptSwapChannels_32s_AC4R</code>	954
7.64.1.23	<code>npptSwapChannels_32s_C3C4R</code>	954
7.64.1.24	<code>npptSwapChannels_32s_C3IR</code>	955
7.64.1.25	<code>npptSwapChannels_32s_C3R</code>	955
7.64.1.26	<code>npptSwapChannels_32s_C4C3R</code>	955
7.64.1.27	<code>npptSwapChannels_32s_C4IR</code>	956
7.64.1.28	<code>npptSwapChannels_32s_C4R</code>	956
7.64.1.29	<code>npptSwapChannels_8u_AC4R</code>	957
7.64.1.30	<code>npptSwapChannels_8u_C3C4R</code>	957

7.64.1.31	nppiSwapChannels_8u_C3IR	958
7.64.1.32	nppiSwapChannels_8u_C3R	958
7.64.1.33	nppiSwapChannels_8u_C4C3R	958
7.64.1.34	nppiSwapChannels_8u_C4IR	959
7.64.1.35	nppiSwapChannels_8u_C4R	959
7.65	Filtering Functions	960
7.65.1	Detailed Description	989
7.65.2	Function Documentation	989
7.65.2.1	nppiFilterGauss_16s_AC4R	989
7.65.2.2	nppiFilterGauss_16s_C1R	989
7.65.2.3	nppiFilterGauss_16s_C3R	990
7.65.2.4	nppiFilterGauss_16s_C4R	990
7.65.2.5	nppiFilterGauss_16u_AC4R	990
7.65.2.6	nppiFilterGauss_16u_C1R	991
7.65.2.7	nppiFilterGauss_16u_C3R	991
7.65.2.8	nppiFilterGauss_16u_C4R	991
7.65.2.9	nppiFilterGauss_32f_AC4R	992
7.65.2.10	nppiFilterGauss_32f_C1R	992
7.65.2.11	nppiFilterGauss_32f_C3R	992
7.65.2.12	nppiFilterGauss_32f_C4R	993
7.65.2.13	nppiFilterGauss_8u_AC4R	993
7.65.2.14	nppiFilterGauss_8u_C1R	993
7.65.2.15	nppiFilterGauss_8u_C3R	994
7.65.2.16	nppiFilterGauss_8u_C4R	994
7.65.2.17	nppiFilterGaussAdvanced_16s_AC4R	994
7.65.2.18	nppiFilterGaussAdvanced_16s_C1R	995
7.65.2.19	nppiFilterGaussAdvanced_16s_C3R	995
7.65.2.20	nppiFilterGaussAdvanced_16s_C4R	996
7.65.2.21	nppiFilterGaussAdvanced_16u_AC4R	996
7.65.2.22	nppiFilterGaussAdvanced_16u_C1R	996
7.65.2.23	nppiFilterGaussAdvanced_16u_C3R	997
7.65.2.24	nppiFilterGaussAdvanced_16u_C4R	997
7.65.2.25	nppiFilterGaussAdvanced_32f_AC4R	998
7.65.2.26	nppiFilterGaussAdvanced_32f_C1R	998
7.65.2.27	nppiFilterGaussAdvanced_32f_C3R	998
7.65.2.28	nppiFilterGaussAdvanced_32f_C4R	999

7.65.2.29	nppiFilterGaussAdvanced_8u_AC4R	999
7.65.2.30	nppiFilterGaussAdvanced_8u_C1R	1000
7.65.2.31	nppiFilterGaussAdvanced_8u_C3R	1000
7.65.2.32	nppiFilterGaussAdvanced_8u_C4R	1000
7.65.2.33	nppiFilterGaussAdvancedBorder_16s_AC4R	1001
7.65.2.34	nppiFilterGaussAdvancedBorder_16s_C1R	1001
7.65.2.35	nppiFilterGaussAdvancedBorder_16s_C3R	1002
7.65.2.36	nppiFilterGaussAdvancedBorder_16s_C4R	1002
7.65.2.37	nppiFilterGaussAdvancedBorder_16u_AC4R	1003
7.65.2.38	nppiFilterGaussAdvancedBorder_16u_C1R	1003
7.65.2.39	nppiFilterGaussAdvancedBorder_16u_C3R	1004
7.65.2.40	nppiFilterGaussAdvancedBorder_16u_C4R	1004
7.65.2.41	nppiFilterGaussAdvancedBorder_32f_AC4R	1005
7.65.2.42	nppiFilterGaussAdvancedBorder_32f_C1R	1005
7.65.2.43	nppiFilterGaussAdvancedBorder_32f_C3R	1006
7.65.2.44	nppiFilterGaussAdvancedBorder_32f_C4R	1006
7.65.2.45	nppiFilterGaussAdvancedBorder_8u_AC4R	1007
7.65.2.46	nppiFilterGaussAdvancedBorder_8u_C1R	1007
7.65.2.47	nppiFilterGaussAdvancedBorder_8u_C3R	1008
7.65.2.48	nppiFilterGaussAdvancedBorder_8u_C4R	1008
7.65.2.49	nppiFilterGaussBorder_16s_AC4R	1009
7.65.2.50	nppiFilterGaussBorder_16s_C1R	1009
7.65.2.51	nppiFilterGaussBorder_16s_C3R	1010
7.65.2.52	nppiFilterGaussBorder_16s_C4R	1010
7.65.2.53	nppiFilterGaussBorder_16u_AC4R	1011
7.65.2.54	nppiFilterGaussBorder_16u_C1R	1011
7.65.2.55	nppiFilterGaussBorder_16u_C3R	1012
7.65.2.56	nppiFilterGaussBorder_16u_C4R	1012
7.65.2.57	nppiFilterGaussBorder_32f_AC4R	1013
7.65.2.58	nppiFilterGaussBorder_32f_C1R	1013
7.65.2.59	nppiFilterGaussBorder_32f_C3R	1014
7.65.2.60	nppiFilterGaussBorder_32f_C4R	1014
7.65.2.61	nppiFilterGaussBorder_8u_AC4R	1015
7.65.2.62	nppiFilterGaussBorder_8u_C1R	1015
7.65.2.63	nppiFilterGaussBorder_8u_C3R	1016
7.65.2.64	nppiFilterGaussBorder_8u_C4R	1016

7.65.2.65 nppiFilterHighPass_16s_AC4R	1017
7.65.2.66 nppiFilterHighPass_16s_C1R	1017
7.65.2.67 nppiFilterHighPass_16s_C3R	1017
7.65.2.68 nppiFilterHighPass_16s_C4R	1018
7.65.2.69 nppiFilterHighPass_16u_AC4R	1018
7.65.2.70 nppiFilterHighPass_16u_C1R	1018
7.65.2.71 nppiFilterHighPass_16u_C3R	1019
7.65.2.72 nppiFilterHighPass_16u_C4R	1019
7.65.2.73 nppiFilterHighPass_32f_AC4R	1019
7.65.2.74 nppiFilterHighPass_32f_C1R	1020
7.65.2.75 nppiFilterHighPass_32f_C3R	1020
7.65.2.76 nppiFilterHighPass_32f_C4R	1020
7.65.2.77 nppiFilterHighPass_8u_AC4R	1021
7.65.2.78 nppiFilterHighPass_8u_C1R	1021
7.65.2.79 nppiFilterHighPass_8u_C3R	1021
7.65.2.80 nppiFilterHighPass_8u_C4R	1022
7.65.2.81 nppiFilterHighPassBorder_16s_AC4R	1022
7.65.2.82 nppiFilterHighPassBorder_16s_C1R	1022
7.65.2.83 nppiFilterHighPassBorder_16s_C3R	1023
7.65.2.84 nppiFilterHighPassBorder_16s_C4R	1023
7.65.2.85 nppiFilterHighPassBorder_16u_AC4R	1024
7.65.2.86 nppiFilterHighPassBorder_16u_C1R	1024
7.65.2.87 nppiFilterHighPassBorder_16u_C3R	1025
7.65.2.88 nppiFilterHighPassBorder_16u_C4R	1025
7.65.2.89 nppiFilterHighPassBorder_32f_AC4R	1026
7.65.2.90 nppiFilterHighPassBorder_32f_C1R	1026
7.65.2.91 nppiFilterHighPassBorder_32f_C3R	1027
7.65.2.92 nppiFilterHighPassBorder_32f_C4R	1027
7.65.2.93 nppiFilterHighPassBorder_8u_AC4R	1028
7.65.2.94 nppiFilterHighPassBorder_8u_C1R	1028
7.65.2.95 nppiFilterHighPassBorder_8u_C3R	1029
7.65.2.96 nppiFilterHighPassBorder_8u_C4R	1029
7.65.2.97 nppiFilterLaplace_16s_AC4R	1030
7.65.2.98 nppiFilterLaplace_16s_C1R	1030
7.65.2.99 nppiFilterLaplace_16s_C3R	1030
7.65.2.100 nppiFilterLaplace_16s_C4R	1031

7.65.2.101nppiFilterLaplace_32f_AC4R	1031
7.65.2.102nppiFilterLaplace_32f_C1R	1031
7.65.2.103nppiFilterLaplace_32f_C3R	1032
7.65.2.104nppiFilterLaplace_32f_C4R	1032
7.65.2.105nppiFilterLaplace_8s16s_C1R	1032
7.65.2.106nppiFilterLaplace_8u16s_C1R	1033
7.65.2.107nppiFilterLaplace_8u_AC4R	1033
7.65.2.108nppiFilterLaplace_8u_C1R	1033
7.65.2.109nppiFilterLaplace_8u_C3R	1034
7.65.2.110nppiFilterLaplace_8u_C4R	1034
7.65.2.111nppiFilterLaplaceBorder_16s_AC4R	1034
7.65.2.112nppiFilterLaplaceBorder_16s_C1R	1035
7.65.2.113nppiFilterLaplaceBorder_16s_C3R	1035
7.65.2.114nppiFilterLaplaceBorder_16s_C4R	1036
7.65.2.115nppiFilterLaplaceBorder_32f_AC4R	1036
7.65.2.116nppiFilterLaplaceBorder_32f_C1R	1037
7.65.2.117nppiFilterLaplaceBorder_32f_C3R	1037
7.65.2.118nppiFilterLaplaceBorder_32f_C4R	1038
7.65.2.119nppiFilterLaplaceBorder_8s16s_C1R	1038
7.65.2.120nppiFilterLaplaceBorder_8u16s_C1R	1039
7.65.2.121nppiFilterLaplaceBorder_8u_AC4R	1039
7.65.2.122nppiFilterLaplaceBorder_8u_C1R	1040
7.65.2.123nppiFilterLaplaceBorder_8u_C3R	1040
7.65.2.124nppiFilterLaplaceBorder_8u_C4R	1041
7.65.2.125nppiFilterLowPass_16s_AC4R	1041
7.65.2.126nppiFilterLowPass_16s_C1R	1041
7.65.2.127nppiFilterLowPass_16s_C3R	1042
7.65.2.128nppiFilterLowPass_16s_C4R	1042
7.65.2.129nppiFilterLowPass_16u_AC4R	1043
7.65.2.130nppiFilterLowPass_16u_C1R	1043
7.65.2.131nppiFilterLowPass_16u_C3R	1043
7.65.2.132nppiFilterLowPass_16u_C4R	1044
7.65.2.133nppiFilterLowPass_32f_AC4R	1044
7.65.2.134nppiFilterLowPass_32f_C1R	1044
7.65.2.135nppiFilterLowPass_32f_C3R	1045
7.65.2.136nppiFilterLowPass_32f_C4R	1045

7.65.2.137nppiFilterLowPass_8u_AC4R	1045
7.65.2.138nppiFilterLowPass_8u_C1R	1046
7.65.2.139nppiFilterLowPass_8u_C3R	1046
7.65.2.140nppiFilterLowPass_8u_C4R	1046
7.65.2.141nppiFilterLowPassBorder_16s_AC4R	1047
7.65.2.142nppiFilterLowPassBorder_16s_C1R	1047
7.65.2.143nppiFilterLowPassBorder_16s_C3R	1048
7.65.2.144nppiFilterLowPassBorder_16s_C4R	1048
7.65.2.145nppiFilterLowPassBorder_16u_AC4R	1049
7.65.2.146nppiFilterLowPassBorder_16u_C1R	1049
7.65.2.147nppiFilterLowPassBorder_16u_C3R	1050
7.65.2.148nppiFilterLowPassBorder_16u_C4R	1050
7.65.2.149nppiFilterLowPassBorder_32f_AC4R	1051
7.65.2.150nppiFilterLowPassBorder_32f_C1R	1051
7.65.2.151nppiFilterLowPassBorder_32f_C3R	1052
7.65.2.152nppiFilterLowPassBorder_32f_C4R	1052
7.65.2.153nppiFilterLowPassBorder_8u_AC4R	1053
7.65.2.154nppiFilterLowPassBorder_8u_C1R	1053
7.65.2.155nppiFilterLowPassBorder_8u_C3R	1054
7.65.2.156nppiFilterLowPassBorder_8u_C4R	1054
7.65.2.157nppiFilterRobertsDown_16s_AC4R	1055
7.65.2.158nppiFilterRobertsDown_16s_C1R	1055
7.65.2.159nppiFilterRobertsDown_16s_C3R	1055
7.65.2.160nppiFilterRobertsDown_16s_C4R	1056
7.65.2.161nppiFilterRobertsDown_32f_AC4R	1056
7.65.2.162nppiFilterRobertsDown_32f_C1R	1056
7.65.2.163nppiFilterRobertsDown_32f_C3R	1057
7.65.2.164nppiFilterRobertsDown_32f_C4R	1057
7.65.2.165nppiFilterRobertsDown_8u_AC4R	1057
7.65.2.166nppiFilterRobertsDown_8u_C1R	1058
7.65.2.167nppiFilterRobertsDown_8u_C3R	1058
7.65.2.168nppiFilterRobertsDown_8u_C4R	1058
7.65.2.169nppiFilterRobertsDownBorder_16s_AC4R	1059
7.65.2.170nppiFilterRobertsDownBorder_16s_C1R	1059
7.65.2.171nppiFilterRobertsDownBorder_16s_C3R	1059
7.65.2.172nppiFilterRobertsDownBorder_16s_C4R	1060

7.65.2.173nppiFilterRobertsDownBorder_32f_AC4R	1060
7.65.2.174nppiFilterRobertsDownBorder_32f_C1R	1061
7.65.2.175nppiFilterRobertsDownBorder_32f_C3R	1061
7.65.2.176nppiFilterRobertsDownBorder_32f_C4R	1062
7.65.2.177nppiFilterRobertsDownBorder_8u_AC4R	1062
7.65.2.178nppiFilterRobertsDownBorder_8u_C1R	1062
7.65.2.179nppiFilterRobertsDownBorder_8u_C3R	1063
7.65.2.180nppiFilterRobertsDownBorder_8u_C4R	1063
7.65.2.181nppiFilterRobertsUp_16s_AC4R	1064
7.65.2.182nppiFilterRobertsUp_16s_C1R	1064
7.65.2.183nppiFilterRobertsUp_16s_C3R	1065
7.65.2.184nppiFilterRobertsUp_16s_C4R	1065
7.65.2.185nppiFilterRobertsUp_32f_AC4R	1065
7.65.2.186nppiFilterRobertsUp_32f_C1R	1066
7.65.2.187nppiFilterRobertsUp_32f_C3R	1066
7.65.2.188nppiFilterRobertsUp_32f_C4R	1066
7.65.2.189nppiFilterRobertsUp_8u_AC4R	1067
7.65.2.190nppiFilterRobertsUp_8u_C1R	1067
7.65.2.191nppiFilterRobertsUp_8u_C3R	1067
7.65.2.192nppiFilterRobertsUp_8u_C4R	1068
7.65.2.193nppiFilterRobertsUpBorder_16s_AC4R	1068
7.65.2.194nppiFilterRobertsUpBorder_16s_C1R	1068
7.65.2.195nppiFilterRobertsUpBorder_16s_C3R	1069
7.65.2.196nppiFilterRobertsUpBorder_16s_C4R	1069
7.65.2.197nppiFilterRobertsUpBorder_32f_AC4R	1070
7.65.2.198nppiFilterRobertsUpBorder_32f_C1R	1070
7.65.2.199nppiFilterRobertsUpBorder_32f_C3R	1070
7.65.2.200nppiFilterRobertsUpBorder_32f_C4R	1071
7.65.2.201nppiFilterRobertsUpBorder_8u_AC4R	1071
7.65.2.202nppiFilterRobertsUpBorder_8u_C1R	1072
7.65.2.203nppiFilterRobertsUpBorder_8u_C3R	1072
7.65.2.204nppiFilterRobertsUpBorder_8u_C4R	1073
7.65.2.205nppiFilterSharpen_16s_AC4R	1073
7.65.2.206nppiFilterSharpen_16s_C1R	1073
7.65.2.207nppiFilterSharpen_16s_C3R	1074
7.65.2.208nppiFilterSharpen_16s_C4R	1074

7.65.2.209	nppiFilterSharpen_16u_AC4R	1074
7.65.2.210	nppiFilterSharpen_16u_C1R	1075
7.65.2.211	nppiFilterSharpen_16u_C3R	1075
7.65.2.212	nppiFilterSharpen_16u_C4R	1076
7.65.2.213	nppiFilterSharpen_32f_AC4R	1076
7.65.2.214	nppiFilterSharpen_32f_C1R	1076
7.65.2.215	nppiFilterSharpen_32f_C3R	1077
7.65.2.216	nppiFilterSharpen_32f_C4R	1077
7.65.2.217	nppiFilterSharpen_8u_AC4R	1077
7.65.2.218	nppiFilterSharpen_8u_C1R	1078
7.65.2.219	nppiFilterSharpen_8u_C3R	1078
7.65.2.220	nppiFilterSharpen_8u_C4R	1078
7.65.2.221	nppiFilterSharpenBorder_16s_AC4R	1079
7.65.2.222	nppiFilterSharpenBorder_16s_C1R	1079
7.65.2.223	nppiFilterSharpenBorder_16s_C3R	1079
7.65.2.224	nppiFilterSharpenBorder_16s_C4R	1080
7.65.2.225	nppiFilterSharpenBorder_16u_AC4R	1080
7.65.2.226	nppiFilterSharpenBorder_16u_C1R	1081
7.65.2.227	nppiFilterSharpenBorder_16u_C3R	1081
7.65.2.228	nppiFilterSharpenBorder_16u_C4R	1082
7.65.2.229	nppiFilterSharpenBorder_32f_AC4R	1082
7.65.2.230	nppiFilterSharpenBorder_32f_C1R	1082
7.65.2.231	nppiFilterSharpenBorder_32f_C3R	1083
7.65.2.232	nppiFilterSharpenBorder_32f_C4R	1083
7.65.2.233	nppiFilterSharpenBorder_8u_AC4R	1084
7.65.2.234	nppiFilterSharpenBorder_8u_C1R	1084
7.65.2.235	nppiFilterSharpenBorder_8u_C3R	1085
7.65.2.236	nppiFilterSharpenBorder_8u_C4R	1085
7.65.2.237	nppiFilterSobelCrossBorder_32f_C1R	1085
7.65.2.238	nppiFilterSobelCrossBorder_8s16s_C1R	1086
7.65.2.239	nppiFilterSobelCrossBorder_8u16s_C1R	1086
7.65.2.240	nppiFilterSobelVertSecondBorder_32f_C1R	1087
7.65.2.241	nppiFilterSobelVertSecondBorder_8s16s_C1R	1087
7.65.2.242	nppiFilterSobelVertSecondBorder_8u16s_C1R	1088
7.65.2.243	nppiFilterUnsharpBorder_16s_AC4R	1088
7.65.2.244	nppiFilterUnsharpBorder_16s_C1R	1089

7.65.2.245	<code>nppiFilterUnsharpBorder_16s_C3R</code>	1089
7.65.2.246	<code>nppiFilterUnsharpBorder_16s_C4R</code>	1090
7.65.2.247	<code>nppiFilterUnsharpBorder_16u_AC4R</code>	1091
7.65.2.248	<code>nppiFilterUnsharpBorder_16u_C1R</code>	1091
7.65.2.249	<code>nppiFilterUnsharpBorder_16u_C3R</code>	1092
7.65.2.250	<code>nppiFilterUnsharpBorder_16u_C4R</code>	1092
7.65.2.251	<code>nppiFilterUnsharpBorder_32f_AC4R</code>	1093
7.65.2.252	<code>nppiFilterUnsharpBorder_32f_C1R</code>	1093
7.65.2.253	<code>nppiFilterUnsharpBorder_32f_C3R</code>	1094
7.65.2.254	<code>nppiFilterUnsharpBorder_32f_C4R</code>	1095
7.65.2.255	<code>nppiFilterUnsharpBorder_8u_AC4R</code>	1095
7.65.2.256	<code>nppiFilterUnsharpBorder_8u_C1R</code>	1096
7.65.2.257	<code>nppiFilterUnsharpBorder_8u_C3R</code>	1096
7.65.2.258	<code>nppiFilterUnsharpBorder_8u_C4R</code>	1097
7.65.2.259	<code>nppiFilterUnsharpGetBufferSize_16s_AC4R</code>	1097
7.65.2.260	<code>nppiFilterUnsharpGetBufferSize_16s_C1R</code>	1098
7.65.2.261	<code>nppiFilterUnsharpGetBufferSize_16s_C3R</code>	1098
7.65.2.262	<code>nppiFilterUnsharpGetBufferSize_16s_C4R</code>	1098
7.65.2.263	<code>nppiFilterUnsharpGetBufferSize_16u_AC4R</code>	1099
7.65.2.264	<code>nppiFilterUnsharpGetBufferSize_16u_C1R</code>	1099
7.65.2.265	<code>nppiFilterUnsharpGetBufferSize_16u_C3R</code>	1099
7.65.2.266	<code>nppiFilterUnsharpGetBufferSize_16u_C4R</code>	1099
7.65.2.267	<code>nppiFilterUnsharpGetBufferSize_32f_AC4R</code>	1100
7.65.2.268	<code>nppiFilterUnsharpGetBufferSize_32f_C1R</code>	1100
7.65.2.269	<code>nppiFilterUnsharpGetBufferSize_32f_C3R</code>	1100
7.65.2.270	<code>nppiFilterUnsharpGetBufferSize_32f_C4R</code>	1101
7.65.2.271	<code>nppiFilterUnsharpGetBufferSize_8u_AC4R</code>	1101
7.65.2.272	<code>nppiFilterUnsharpGetBufferSize_8u_C1R</code>	1101
7.65.2.273	<code>nppiFilterUnsharpGetBufferSize_8u_C3R</code>	1101
7.65.2.274	<code>nppiFilterUnsharpGetBufferSize_8u_C4R</code>	1102
7.66	1D Linear Filter	1103
7.66.1	Function Documentation	1120
7.66.1.1	<code>nppiFilterColumn32f_16s_AC4R</code>	1120
7.66.1.2	<code>nppiFilterColumn32f_16s_C1R</code>	1121
7.66.1.3	<code>nppiFilterColumn32f_16s_C3R</code>	1121
7.66.1.4	<code>nppiFilterColumn32f_16s_C4R</code>	1122

7.66.1.5	nppiFilterColumn32f_16u_AC4R	1122
7.66.1.6	nppiFilterColumn32f_16u_C1R	1123
7.66.1.7	nppiFilterColumn32f_16u_C3R	1123
7.66.1.8	nppiFilterColumn32f_16u_C4R	1124
7.66.1.9	nppiFilterColumn32f_8u_AC4R	1124
7.66.1.10	nppiFilterColumn32f_8u_C1R	1125
7.66.1.11	nppiFilterColumn32f_8u_C3R	1125
7.66.1.12	nppiFilterColumn32f_8u_C4R	1126
7.66.1.13	nppiFilterColumn_16s_AC4R	1126
7.66.1.14	nppiFilterColumn_16s_C1R	1127
7.66.1.15	nppiFilterColumn_16s_C3R	1127
7.66.1.16	nppiFilterColumn_16s_C4R	1128
7.66.1.17	nppiFilterColumn_16u_AC4R	1128
7.66.1.18	nppiFilterColumn_16u_C1R	1129
7.66.1.19	nppiFilterColumn_16u_C3R	1129
7.66.1.20	nppiFilterColumn_16u_C4R	1130
7.66.1.21	nppiFilterColumn_32f_AC4R	1130
7.66.1.22	nppiFilterColumn_32f_C1R	1131
7.66.1.23	nppiFilterColumn_32f_C3R	1131
7.66.1.24	nppiFilterColumn_32f_C4R	1132
7.66.1.25	nppiFilterColumn_64f_C1R	1132
7.66.1.26	nppiFilterColumn_8u_AC4R	1133
7.66.1.27	nppiFilterColumn_8u_C1R	1133
7.66.1.28	nppiFilterColumn_8u_C3R	1134
7.66.1.29	nppiFilterColumn_8u_C4R	1134
7.66.1.30	nppiFilterColumnBorder32f_16s_AC4R	1135
7.66.1.31	nppiFilterColumnBorder32f_16s_C1R	1135
7.66.1.32	nppiFilterColumnBorder32f_16s_C3R	1136
7.66.1.33	nppiFilterColumnBorder32f_16s_C4R	1136
7.66.1.34	nppiFilterColumnBorder32f_16u_AC4R	1137
7.66.1.35	nppiFilterColumnBorder32f_16u_C1R	1137
7.66.1.36	nppiFilterColumnBorder32f_16u_C3R	1138
7.66.1.37	nppiFilterColumnBorder32f_16u_C4R	1138
7.66.1.38	nppiFilterColumnBorder32f_8u_AC4R	1139
7.66.1.39	nppiFilterColumnBorder32f_8u_C1R	1139
7.66.1.40	nppiFilterColumnBorder32f_8u_C3R	1140

7.66.1.41 nppiFilterColumnBorder32f_8u_C4R	1140
7.66.1.42 nppiFilterColumnBorder_16s_AC4R	1141
7.66.1.43 nppiFilterColumnBorder_16s_C1R	1141
7.66.1.44 nppiFilterColumnBorder_16s_C3R	1142
7.66.1.45 nppiFilterColumnBorder_16s_C4R	1142
7.66.1.46 nppiFilterColumnBorder_16u_AC4R	1143
7.66.1.47 nppiFilterColumnBorder_16u_C1R	1144
7.66.1.48 nppiFilterColumnBorder_16u_C3R	1144
7.66.1.49 nppiFilterColumnBorder_16u_C4R	1145
7.66.1.50 nppiFilterColumnBorder_32f_AC4R	1145
7.66.1.51 nppiFilterColumnBorder_32f_C1R	1146
7.66.1.52 nppiFilterColumnBorder_32f_C3R	1146
7.66.1.53 nppiFilterColumnBorder_32f_C4R	1147
7.66.1.54 nppiFilterColumnBorder_8u_AC4R	1148
7.66.1.55 nppiFilterColumnBorder_8u_C1R	1148
7.66.1.56 nppiFilterColumnBorder_8u_C3R	1149
7.66.1.57 nppiFilterColumnBorder_8u_C4R	1149
7.66.1.58 nppiFilterRow32f_16s_AC4R	1150
7.66.1.59 nppiFilterRow32f_16s_C1R	1150
7.66.1.60 nppiFilterRow32f_16s_C3R	1151
7.66.1.61 nppiFilterRow32f_16s_C4R	1151
7.66.1.62 nppiFilterRow32f_16u_AC4R	1152
7.66.1.63 nppiFilterRow32f_16u_C1R	1152
7.66.1.64 nppiFilterRow32f_16u_C3R	1153
7.66.1.65 nppiFilterRow32f_16u_C4R	1153
7.66.1.66 nppiFilterRow32f_8u_AC4R	1154
7.66.1.67 nppiFilterRow32f_8u_C1R	1154
7.66.1.68 nppiFilterRow32f_8u_C3R	1155
7.66.1.69 nppiFilterRow32f_8u_C4R	1155
7.66.1.70 nppiFilterRow_16s_AC4R	1156
7.66.1.71 nppiFilterRow_16s_C1R	1156
7.66.1.72 nppiFilterRow_16s_C3R	1157
7.66.1.73 nppiFilterRow_16s_C4R	1157
7.66.1.74 nppiFilterRow_16u_AC4R	1158
7.66.1.75 nppiFilterRow_16u_C1R	1158
7.66.1.76 nppiFilterRow_16u_C3R	1159

7.66.1.77 nppiFilterRow_16u_C4R	1159
7.66.1.78 nppiFilterRow_32f_AC4R	1160
7.66.1.79 nppiFilterRow_32f_C1R	1160
7.66.1.80 nppiFilterRow_32f_C3R	1161
7.66.1.81 nppiFilterRow_32f_C4R	1161
7.66.1.82 nppiFilterRow_64f_C1R	1162
7.66.1.83 nppiFilterRow_8u_AC4R	1162
7.66.1.84 nppiFilterRow_8u_C1R	1163
7.66.1.85 nppiFilterRow_8u_C3R	1163
7.66.1.86 nppiFilterRow_8u_C4R	1164
7.66.1.87 nppiFilterRowBorder32f_16s_AC4R	1164
7.66.1.88 nppiFilterRowBorder32f_16s_C1R	1165
7.66.1.89 nppiFilterRowBorder32f_16s_C3R	1165
7.66.1.90 nppiFilterRowBorder32f_16s_C4R	1166
7.66.1.91 nppiFilterRowBorder32f_16u_AC4R	1166
7.66.1.92 nppiFilterRowBorder32f_16u_C1R	1167
7.66.1.93 nppiFilterRowBorder32f_16u_C3R	1167
7.66.1.94 nppiFilterRowBorder32f_16u_C4R	1168
7.66.1.95 nppiFilterRowBorder32f_8u_AC4R	1168
7.66.1.96 nppiFilterRowBorder32f_8u_C1R	1169
7.66.1.97 nppiFilterRowBorder32f_8u_C3R	1169
7.66.1.98 nppiFilterRowBorder32f_8u_C4R	1170
7.66.1.99 nppiFilterRowBorder_16s_AC4R	1170
7.66.1.100 nppiFilterRowBorder_16s_C1R	1171
7.66.1.101 nppiFilterRowBorder_16s_C3R	1171
7.66.1.102 nppiFilterRowBorder_16s_C4R	1172
7.66.1.103 nppiFilterRowBorder_16u_AC4R	1173
7.66.1.104 nppiFilterRowBorder_16u_C1R	1173
7.66.1.105 nppiFilterRowBorder_16u_C3R	1174
7.66.1.106 nppiFilterRowBorder_16u_C4R	1174
7.66.1.107 nppiFilterRowBorder_32f_AC4R	1175
7.66.1.108 nppiFilterRowBorder_32f_C1R	1176
7.66.1.109 nppiFilterRowBorder_32f_C3R	1176
7.66.1.110 nppiFilterRowBorder_32f_C4R	1177
7.66.1.111 nppiFilterRowBorder_8u_AC4R	1177
7.66.1.112 nppiFilterRowBorder_8u_C1R	1178

7.66.1.113nppiFilterRowBorder_8u_C3R	1178
7.66.1.114nppiFilterRowBorder_8u_C4R	1179
7.66.1.115nppiFilterSobelCross_32f_C1R	1180
7.66.1.116nppiFilterSobelCross_8s16s_C1R	1180
7.66.1.117nppiFilterSobelCross_8u16s_C1R	1180
7.66.1.118nppiFilterSobelHorizBorder_16s_AC4R	1181
7.66.1.119nppiFilterSobelHorizBorder_16s_C1R	1181
7.66.1.120nppiFilterSobelHorizBorder_16s_C3R	1181
7.66.1.121nppiFilterSobelHorizBorder_16s_C4R	1182
7.66.1.122nppiFilterSobelHorizBorder_32f_AC4R	1182
7.66.1.123nppiFilterSobelHorizBorder_32f_C1R	1183
7.66.1.124nppiFilterSobelHorizBorder_32f_C3R	1183
7.66.1.125nppiFilterSobelHorizBorder_32f_C4R	1184
7.66.1.126nppiFilterSobelHorizBorder_8s16s_C1R	1184
7.66.1.127nppiFilterSobelHorizBorder_8u16s_C1R	1185
7.66.1.128nppiFilterSobelHorizBorder_8u_AC4R	1185
7.66.1.129nppiFilterSobelHorizBorder_8u_C1R	1186
7.66.1.130nppiFilterSobelHorizBorder_8u_C3R	1186
7.66.1.131nppiFilterSobelHorizBorder_8u_C4R	1186
7.66.1.132nppiFilterSobelHorizMaskBorder_32f_C1R	1187
7.66.1.133nppiFilterSobelHorizSecondBorder_32f_C1R	1187
7.66.1.134nppiFilterSobelHorizSecondBorder_8s16s_C1R	1188
7.66.1.135nppiFilterSobelHorizSecondBorder_8u16s_C1R	1188
7.66.1.136nppiFilterSobelVertBorder_16s_AC4R	1189
7.66.1.137nppiFilterSobelVertBorder_16s_C1R	1189
7.66.1.138nppiFilterSobelVertBorder_16s_C3R	1190
7.66.1.139nppiFilterSobelVertBorder_16s_C4R	1190
7.66.1.140nppiFilterSobelVertBorder_32f_AC4R	1190
7.66.1.141nppiFilterSobelVertBorder_32f_C1R	1191
7.66.1.142nppiFilterSobelVertBorder_32f_C3R	1191
7.66.1.143nppiFilterSobelVertBorder_32f_C4R	1192
7.66.1.144nppiFilterSobelVertBorder_8s16s_C1R	1192
7.66.1.145nppiFilterSobelVertBorder_8u16s_C1R	1193
7.66.1.146nppiFilterSobelVertBorder_8u_AC4R	1193
7.66.1.147nppiFilterSobelVertBorder_8u_C1R	1194
7.66.1.148nppiFilterSobelVertBorder_8u_C3R	1194

7.66.1.149	ippiFilterSobelVertBorder_8u_C4R	1194
7.66.1.150	ippiFilterSobelVertMaskBorder_32f_C1R	1195
7.66.1.151	ippiFilterSobelVertSecond_32f_C1R	1195
7.66.1.152	ippiFilterSobelVertSecond_8s16s_C1R	1196
7.66.1.153	ippiFilterSobelVertSecond_8u16s_C1R	1196
7.67	1D Window Sum	1197
7.67.1	Function Documentation	1198
7.67.1.1	ippiSumWindowColumn_16s32f_C1R	1198
7.67.1.2	ippiSumWindowColumn_16s32f_C3R	1199
7.67.1.3	ippiSumWindowColumn_16s32f_C4R	1199
7.67.1.4	ippiSumWindowColumn_16u32f_C1R	1200
7.67.1.5	ippiSumWindowColumn_16u32f_C3R	1200
7.67.1.6	ippiSumWindowColumn_16u32f_C4R	1201
7.67.1.7	ippiSumWindowColumn_8u32f_C1R	1201
7.67.1.8	ippiSumWindowColumn_8u32f_C3R	1202
7.67.1.9	ippiSumWindowColumn_8u32f_C4R	1202
7.67.1.10	ippiSumWindowRow_16s32f_C1R	1203
7.67.1.11	ippiSumWindowRow_16s32f_C3R	1203
7.67.1.12	ippiSumWindowRow_16s32f_C4R	1204
7.67.1.13	ippiSumWindowRow_16u32f_C1R	1204
7.67.1.14	ippiSumWindowRow_16u32f_C3R	1205
7.67.1.15	ippiSumWindowRow_16u32f_C4R	1205
7.67.1.16	ippiSumWindowRow_8u32f_C1R	1206
7.67.1.17	ippiSumWindowRow_8u32f_C3R	1206
7.67.1.18	ippiSumWindowRow_8u32f_C4R	1207
7.68	1D Window Sum with Border Control	1208
7.68.1	Function Documentation	1210
7.68.1.1	ippiSumWindowColumnBorder_16s32f_C1R	1210
7.68.1.2	ippiSumWindowColumnBorder_16s32f_C3R	1210
7.68.1.3	ippiSumWindowColumnBorder_16s32f_C4R	1211
7.68.1.4	ippiSumWindowColumnBorder_16u32f_C1R	1211
7.68.1.5	ippiSumWindowColumnBorder_16u32f_C3R	1212
7.68.1.6	ippiSumWindowColumnBorder_16u32f_C4R	1213
7.68.1.7	ippiSumWindowColumnBorder_8u32f_C1R	1213
7.68.1.8	ippiSumWindowColumnBorder_8u32f_C3R	1214
7.68.1.9	ippiSumWindowColumnBorder_8u32f_C4R	1214

7.68.1.10	nppiSumWindowRowBorder_16s32f_C1R	1215
7.68.1.11	nppiSumWindowRowBorder_16s32f_C3R	1215
7.68.1.12	nppiSumWindowRowBorder_16s32f_C4R	1216
7.68.1.13	nppiSumWindowRowBorder_16u32f_C1R	1217
7.68.1.14	nppiSumWindowRowBorder_16u32f_C3R	1217
7.68.1.15	nppiSumWindowRowBorder_16u32f_C4R	1218
7.68.1.16	nppiSumWindowRowBorder_8u32f_C1R	1218
7.68.1.17	nppiSumWindowRowBorder_8u32f_C3R	1219
7.68.1.18	nppiSumWindowRowBorder_8u32f_C4R	1219
7.69	Convolution	1221
7.69.1	Function Documentation	1230
7.69.1.1	nppiFilter32f_16s_AC4R	1230
7.69.1.2	nppiFilter32f_16s_C1R	1231
7.69.1.3	nppiFilter32f_16s_C3R	1231
7.69.1.4	nppiFilter32f_16s_C4R	1232
7.69.1.5	nppiFilter32f_16u_AC4R	1232
7.69.1.6	nppiFilter32f_16u_C1R	1233
7.69.1.7	nppiFilter32f_16u_C3R	1233
7.69.1.8	nppiFilter32f_16u_C4R	1234
7.69.1.9	nppiFilter32f_32s_AC4R	1234
7.69.1.10	nppiFilter32f_32s_C1R	1235
7.69.1.11	nppiFilter32f_32s_C3R	1235
7.69.1.12	nppiFilter32f_32s_C4R	1236
7.69.1.13	nppiFilter32f_8s16s_AC4R	1236
7.69.1.14	nppiFilter32f_8s16s_C1R	1237
7.69.1.15	nppiFilter32f_8s16s_C3R	1237
7.69.1.16	nppiFilter32f_8s16s_C4R	1238
7.69.1.17	nppiFilter32f_8s_AC4R	1238
7.69.1.18	nppiFilter32f_8s_C1R	1239
7.69.1.19	nppiFilter32f_8s_C2R	1239
7.69.1.20	nppiFilter32f_8s_C3R	1240
7.69.1.21	nppiFilter32f_8s_C4R	1240
7.69.1.22	nppiFilter32f_8u16s_AC4R	1241
7.69.1.23	nppiFilter32f_8u16s_C1R	1241
7.69.1.24	nppiFilter32f_8u16s_C3R	1242
7.69.1.25	nppiFilter32f_8u16s_C4R	1242

7.69.1.26 nppiFilter32f_8u_AC4R	1243
7.69.1.27 nppiFilter32f_8u_C1R	1243
7.69.1.28 nppiFilter32f_8u_C2R	1244
7.69.1.29 nppiFilter32f_8u_C3R	1244
7.69.1.30 nppiFilter32f_8u_C4R	1245
7.69.1.31 nppiFilter_16s_AC4R	1245
7.69.1.32 nppiFilter_16s_C1R	1246
7.69.1.33 nppiFilter_16s_C3R	1246
7.69.1.34 nppiFilter_16s_C4R	1247
7.69.1.35 nppiFilter_16u_AC4R	1247
7.69.1.36 nppiFilter_16u_C1R	1248
7.69.1.37 nppiFilter_16u_C3R	1248
7.69.1.38 nppiFilter_16u_C4R	1249
7.69.1.39 nppiFilter_32f_AC4R	1249
7.69.1.40 nppiFilter_32f_C1R	1250
7.69.1.41 nppiFilter_32f_C2R	1250
7.69.1.42 nppiFilter_32f_C3R	1251
7.69.1.43 nppiFilter_32f_C4R	1251
7.69.1.44 nppiFilter_64f_C1R	1252
7.69.1.45 nppiFilter_8u_AC4R	1252
7.69.1.46 nppiFilter_8u_C1R	1253
7.69.1.47 nppiFilter_8u_C3R	1253
7.69.1.48 nppiFilter_8u_C4R	1254
7.69.1.49 nppiFilterBorder32f_16s_AC4R	1254
7.69.1.50 nppiFilterBorder32f_16s_C1R	1255
7.69.1.51 nppiFilterBorder32f_16s_C3R	1255
7.69.1.52 nppiFilterBorder32f_16s_C4R	1256
7.69.1.53 nppiFilterBorder32f_16u_AC4R	1256
7.69.1.54 nppiFilterBorder32f_16u_C1R	1257
7.69.1.55 nppiFilterBorder32f_16u_C3R	1257
7.69.1.56 nppiFilterBorder32f_16u_C4R	1258
7.69.1.57 nppiFilterBorder32f_32s_AC4R	1258
7.69.1.58 nppiFilterBorder32f_32s_C1R	1259
7.69.1.59 nppiFilterBorder32f_32s_C3R	1259
7.69.1.60 nppiFilterBorder32f_32s_C4R	1260
7.69.1.61 nppiFilterBorder32f_8s16s_AC4R	1260

7.69.1.62	nppiFilterBorder32f_8s16s_C1R	1261
7.69.1.63	nppiFilterBorder32f_8s16s_C3R	1261
7.69.1.64	nppiFilterBorder32f_8s16s_C4R	1262
7.69.1.65	nppiFilterBorder32f_8s_AC4R	1262
7.69.1.66	nppiFilterBorder32f_8s_C1R	1263
7.69.1.67	nppiFilterBorder32f_8s_C2R	1263
7.69.1.68	nppiFilterBorder32f_8s_C3R	1264
7.69.1.69	nppiFilterBorder32f_8s_C4R	1264
7.69.1.70	nppiFilterBorder32f_8u16s_AC4R	1265
7.69.1.71	nppiFilterBorder32f_8u16s_C1R	1265
7.69.1.72	nppiFilterBorder32f_8u16s_C3R	1266
7.69.1.73	nppiFilterBorder32f_8u16s_C4R	1266
7.69.1.74	nppiFilterBorder32f_8u_AC4R	1267
7.69.1.75	nppiFilterBorder32f_8u_C1R	1267
7.69.1.76	nppiFilterBorder32f_8u_C2R	1268
7.69.1.77	nppiFilterBorder32f_8u_C3R	1268
7.69.1.78	nppiFilterBorder32f_8u_C4R	1269
7.69.1.79	nppiFilterBorder_16s_AC4R	1269
7.69.1.80	nppiFilterBorder_16s_C1R	1270
7.69.1.81	nppiFilterBorder_16s_C3R	1270
7.69.1.82	nppiFilterBorder_16s_C4R	1271
7.69.1.83	nppiFilterBorder_16u_AC4R	1272
7.69.1.84	nppiFilterBorder_16u_C1R	1272
7.69.1.85	nppiFilterBorder_16u_C3R	1273
7.69.1.86	nppiFilterBorder_16u_C4R	1273
7.69.1.87	nppiFilterBorder_32f_AC4R	1274
7.69.1.88	nppiFilterBorder_32f_C1R	1275
7.69.1.89	nppiFilterBorder_32f_C2R	1275
7.69.1.90	nppiFilterBorder_32f_C3R	1276
7.69.1.91	nppiFilterBorder_32f_C4R	1276
7.69.1.92	nppiFilterBorder_8u_AC4R	1277
7.69.1.93	nppiFilterBorder_8u_C1R	1277
7.69.1.94	nppiFilterBorder_8u_C3R	1278
7.69.1.95	nppiFilterBorder_8u_C4R	1278
7.70	2D Fixed Linear Filters	1280
7.70.1	Function Documentation	1283

7.70.1.1	nppiFilterBox_16s_AC4R	1283
7.70.1.2	nppiFilterBox_16s_C1R	1283
7.70.1.3	nppiFilterBox_16s_C3R	1284
7.70.1.4	nppiFilterBox_16s_C4R	1284
7.70.1.5	nppiFilterBox_16u_AC4R	1284
7.70.1.6	nppiFilterBox_16u_C1R	1285
7.70.1.7	nppiFilterBox_16u_C3R	1285
7.70.1.8	nppiFilterBox_16u_C4R	1286
7.70.1.9	nppiFilterBox_32f_AC4R	1286
7.70.1.10	nppiFilterBox_32f_C1R	1286
7.70.1.11	nppiFilterBox_32f_C3R	1287
7.70.1.12	nppiFilterBox_32f_C4R	1287
7.70.1.13	nppiFilterBox_64f_C1R	1288
7.70.1.14	nppiFilterBox_8u_AC4R	1288
7.70.1.15	nppiFilterBox_8u_C1R	1288
7.70.1.16	nppiFilterBox_8u_C3R	1289
7.70.1.17	nppiFilterBox_8u_C4R	1289
7.70.1.18	nppiFilterBoxBorder_16s_AC4R	1290
7.70.1.19	nppiFilterBoxBorder_16s_C1R	1290
7.70.1.20	nppiFilterBoxBorder_16s_C3R	1291
7.70.1.21	nppiFilterBoxBorder_16s_C4R	1291
7.70.1.22	nppiFilterBoxBorder_16u_AC4R	1292
7.70.1.23	nppiFilterBoxBorder_16u_C1R	1292
7.70.1.24	nppiFilterBoxBorder_16u_C3R	1293
7.70.1.25	nppiFilterBoxBorder_16u_C4R	1293
7.70.1.26	nppiFilterBoxBorder_32f_AC4R	1294
7.70.1.27	nppiFilterBoxBorder_32f_C1R	1294
7.70.1.28	nppiFilterBoxBorder_32f_C3R	1295
7.70.1.29	nppiFilterBoxBorder_32f_C4R	1295
7.70.1.30	nppiFilterBoxBorder_8u_AC4R	1296
7.70.1.31	nppiFilterBoxBorder_8u_C1R	1296
7.70.1.32	nppiFilterBoxBorder_8u_C3R	1297
7.70.1.33	nppiFilterBoxBorder_8u_C4R	1297
7.71	Rank Filters	1298
7.71.1	Function Documentation	1307
7.71.1.1	nppiFilterMax_16s_AC4R	1307

7.71.1.2	nppiFilterMax_16s_C1R	1307
7.71.1.3	nppiFilterMax_16s_C3R	1308
7.71.1.4	nppiFilterMax_16s_C4R	1308
7.71.1.5	nppiFilterMax_16u_AC4R	1308
7.71.1.6	nppiFilterMax_16u_C1R	1309
7.71.1.7	nppiFilterMax_16u_C3R	1309
7.71.1.8	nppiFilterMax_16u_C4R	1310
7.71.1.9	nppiFilterMax_32f_AC4R	1310
7.71.1.10	nppiFilterMax_32f_C1R	1310
7.71.1.11	nppiFilterMax_32f_C3R	1311
7.71.1.12	nppiFilterMax_32f_C4R	1311
7.71.1.13	nppiFilterMax_8u_AC4R	1312
7.71.1.14	nppiFilterMax_8u_C1R	1312
7.71.1.15	nppiFilterMax_8u_C3R	1312
7.71.1.16	nppiFilterMax_8u_C4R	1313
7.71.1.17	nppiFilterMaxBorder_16s_AC4R	1313
7.71.1.18	nppiFilterMaxBorder_16s_C1R	1314
7.71.1.19	nppiFilterMaxBorder_16s_C3R	1314
7.71.1.20	nppiFilterMaxBorder_16s_C4R	1315
7.71.1.21	nppiFilterMaxBorder_16u_AC4R	1315
7.71.1.22	nppiFilterMaxBorder_16u_C1R	1316
7.71.1.23	nppiFilterMaxBorder_16u_C3R	1316
7.71.1.24	nppiFilterMaxBorder_16u_C4R	1317
7.71.1.25	nppiFilterMaxBorder_32f_AC4R	1317
7.71.1.26	nppiFilterMaxBorder_32f_C1R	1318
7.71.1.27	nppiFilterMaxBorder_32f_C3R	1318
7.71.1.28	nppiFilterMaxBorder_32f_C4R	1319
7.71.1.29	nppiFilterMaxBorder_8u_AC4R	1319
7.71.1.30	nppiFilterMaxBorder_8u_C1R	1320
7.71.1.31	nppiFilterMaxBorder_8u_C3R	1320
7.71.1.32	nppiFilterMaxBorder_8u_C4R	1321
7.71.1.33	nppiFilterMedian_16s_AC4R	1321
7.71.1.34	nppiFilterMedian_16s_C1R	1322
7.71.1.35	nppiFilterMedian_16s_C3R	1322
7.71.1.36	nppiFilterMedian_16s_C4R	1322
7.71.1.37	nppiFilterMedian_16u_AC4R	1323

7.71.1.38	nppiFilterMedian_16u_C1R	1323
7.71.1.39	nppiFilterMedian_16u_C3R	1324
7.71.1.40	nppiFilterMedian_16u_C4R	1324
7.71.1.41	nppiFilterMedian_32f_AC4R	1325
7.71.1.42	nppiFilterMedian_32f_C1R	1325
7.71.1.43	nppiFilterMedian_32f_C3R	1325
7.71.1.44	nppiFilterMedian_32f_C4R	1326
7.71.1.45	nppiFilterMedian_8u_AC4R	1326
7.71.1.46	nppiFilterMedian_8u_C1R	1327
7.71.1.47	nppiFilterMedian_8u_C3R	1327
7.71.1.48	nppiFilterMedian_8u_C4R	1328
7.71.1.49	nppiFilterMedianGetBufferSize_16s_AC4R	1328
7.71.1.50	nppiFilterMedianGetBufferSize_16s_C1R	1328
7.71.1.51	nppiFilterMedianGetBufferSize_16s_C3R	1329
7.71.1.52	nppiFilterMedianGetBufferSize_16s_C4R	1329
7.71.1.53	nppiFilterMedianGetBufferSize_16u_AC4R	1329
7.71.1.54	nppiFilterMedianGetBufferSize_16u_C1R	1329
7.71.1.55	nppiFilterMedianGetBufferSize_16u_C3R	1330
7.71.1.56	nppiFilterMedianGetBufferSize_16u_C4R	1330
7.71.1.57	nppiFilterMedianGetBufferSize_32f_AC4R	1330
7.71.1.58	nppiFilterMedianGetBufferSize_32f_C1R	1331
7.71.1.59	nppiFilterMedianGetBufferSize_32f_C3R	1331
7.71.1.60	nppiFilterMedianGetBufferSize_32f_C4R	1331
7.71.1.61	nppiFilterMedianGetBufferSize_8u_AC4R	1331
7.71.1.62	nppiFilterMedianGetBufferSize_8u_C1R	1332
7.71.1.63	nppiFilterMedianGetBufferSize_8u_C3R	1332
7.71.1.64	nppiFilterMedianGetBufferSize_8u_C4R	1332
7.71.1.65	nppiFilterMin_16s_AC4R	1333
7.71.1.66	nppiFilterMin_16s_C1R	1333
7.71.1.67	nppiFilterMin_16s_C3R	1333
7.71.1.68	nppiFilterMin_16s_C4R	1334
7.71.1.69	nppiFilterMin_16u_AC4R	1334
7.71.1.70	nppiFilterMin_16u_C1R	1335
7.71.1.71	nppiFilterMin_16u_C3R	1335
7.71.1.72	nppiFilterMin_16u_C4R	1335
7.71.1.73	nppiFilterMin_32f_AC4R	1336

7.71.1.74	<code>nppiFilterMin_32f_C1R</code>	1336
7.71.1.75	<code>nppiFilterMin_32f_C3R</code>	1337
7.71.1.76	<code>nppiFilterMin_32f_C4R</code>	1337
7.71.1.77	<code>nppiFilterMin_8u_AC4R</code>	1337
7.71.1.78	<code>nppiFilterMin_8u_C1R</code>	1338
7.71.1.79	<code>nppiFilterMin_8u_C3R</code>	1338
7.71.1.80	<code>nppiFilterMin_8u_C4R</code>	1339
7.71.1.81	<code>nppiFilterMinBorder_16s_AC4R</code>	1339
7.71.1.82	<code>nppiFilterMinBorder_16s_C1R</code>	1339
7.71.1.83	<code>nppiFilterMinBorder_16s_C3R</code>	1340
7.71.1.84	<code>nppiFilterMinBorder_16s_C4R</code>	1340
7.71.1.85	<code>nppiFilterMinBorder_16u_AC4R</code>	1341
7.71.1.86	<code>nppiFilterMinBorder_16u_C1R</code>	1341
7.71.1.87	<code>nppiFilterMinBorder_16u_C3R</code>	1342
7.71.1.88	<code>nppiFilterMinBorder_16u_C4R</code>	1342
7.71.1.89	<code>nppiFilterMinBorder_32f_AC4R</code>	1343
7.71.1.90	<code>nppiFilterMinBorder_32f_C1R</code>	1343
7.71.1.91	<code>nppiFilterMinBorder_32f_C3R</code>	1344
7.71.1.92	<code>nppiFilterMinBorder_32f_C4R</code>	1344
7.71.1.93	<code>nppiFilterMinBorder_8u_AC4R</code>	1345
7.71.1.94	<code>nppiFilterMinBorder_8u_C1R</code>	1345
7.71.1.95	<code>nppiFilterMinBorder_8u_C3R</code>	1346
7.71.1.96	<code>nppiFilterMinBorder_8u_C4R</code>	1346
7.72	Fixed Filters	1348
7.72.1	Detailed Description	1358
7.72.2	Function Documentation	1358
7.72.2.1	<code>nppiFilterPrewittHoriz_16s_AC4R</code>	1358
7.72.2.2	<code>nppiFilterPrewittHoriz_16s_C1R</code>	1358
7.72.2.3	<code>nppiFilterPrewittHoriz_16s_C3R</code>	1359
7.72.2.4	<code>nppiFilterPrewittHoriz_16s_C4R</code>	1359
7.72.2.5	<code>nppiFilterPrewittHoriz_32f_AC4R</code>	1360
7.72.2.6	<code>nppiFilterPrewittHoriz_32f_C1R</code>	1360
7.72.2.7	<code>nppiFilterPrewittHoriz_32f_C3R</code>	1360
7.72.2.8	<code>nppiFilterPrewittHoriz_32f_C4R</code>	1361
7.72.2.9	<code>nppiFilterPrewittHoriz_8u_AC4R</code>	1361
7.72.2.10	<code>nppiFilterPrewittHoriz_8u_C1R</code>	1361

7.72.2.11 nppiFilterPrewittHoriz_8u_C3R	1362
7.72.2.12 nppiFilterPrewittHoriz_8u_C4R	1362
7.72.2.13 nppiFilterPrewittHorizBorder_16s_AC4R	1362
7.72.2.14 nppiFilterPrewittHorizBorder_16s_C1R	1363
7.72.2.15 nppiFilterPrewittHorizBorder_16s_C3R	1363
7.72.2.16 nppiFilterPrewittHorizBorder_16s_C4R	1364
7.72.2.17 nppiFilterPrewittHorizBorder_32f_AC4R	1364
7.72.2.18 nppiFilterPrewittHorizBorder_32f_C1R	1364
7.72.2.19 nppiFilterPrewittHorizBorder_32f_C3R	1365
7.72.2.20 nppiFilterPrewittHorizBorder_32f_C4R	1365
7.72.2.21 nppiFilterPrewittHorizBorder_8u_AC4R	1366
7.72.2.22 nppiFilterPrewittHorizBorder_8u_C1R	1366
7.72.2.23 nppiFilterPrewittHorizBorder_8u_C3R	1367
7.72.2.24 nppiFilterPrewittHorizBorder_8u_C4R	1367
7.72.2.25 nppiFilterPrewittVert_16s_AC4R	1367
7.72.2.26 nppiFilterPrewittVert_16s_C1R	1368
7.72.2.27 nppiFilterPrewittVert_16s_C3R	1368
7.72.2.28 nppiFilterPrewittVert_16s_C4R	1369
7.72.2.29 nppiFilterPrewittVert_32f_AC4R	1369
7.72.2.30 nppiFilterPrewittVert_32f_C1R	1369
7.72.2.31 nppiFilterPrewittVert_32f_C3R	1370
7.72.2.32 nppiFilterPrewittVert_32f_C4R	1370
7.72.2.33 nppiFilterPrewittVert_8u_AC4R	1370
7.72.2.34 nppiFilterPrewittVert_8u_C1R	1371
7.72.2.35 nppiFilterPrewittVert_8u_C3R	1371
7.72.2.36 nppiFilterPrewittVert_8u_C4R	1371
7.72.2.37 nppiFilterPrewittVertBorder_16s_AC4R	1372
7.72.2.38 nppiFilterPrewittVertBorder_16s_C1R	1372
7.72.2.39 nppiFilterPrewittVertBorder_16s_C3R	1372
7.72.2.40 nppiFilterPrewittVertBorder_16s_C4R	1373
7.72.2.41 nppiFilterPrewittVertBorder_32f_AC4R	1373
7.72.2.42 nppiFilterPrewittVertBorder_32f_C1R	1374
7.72.2.43 nppiFilterPrewittVertBorder_32f_C3R	1374
7.72.2.44 nppiFilterPrewittVertBorder_32f_C4R	1375
7.72.2.45 nppiFilterPrewittVertBorder_8u_AC4R	1375
7.72.2.46 nppiFilterPrewittVertBorder_8u_C1R	1375

7.72.2.47	nppiFilterPrewittVertBorder_8u_C3R	1376
7.72.2.48	nppiFilterPrewittVertBorder_8u_C4R	1376
7.72.2.49	nppiFilterScharrHoriz_32f_C1R	1377
7.72.2.50	nppiFilterScharrHoriz_8s16s_C1R	1377
7.72.2.51	nppiFilterScharrHoriz_8u16s_C1R	1378
7.72.2.52	nppiFilterScharrHorizBorder_32f_C1R	1378
7.72.2.53	nppiFilterScharrHorizBorder_8s16s_C1R	1378
7.72.2.54	nppiFilterScharrHorizBorder_8u16s_C1R	1379
7.72.2.55	nppiFilterScharrVert_32f_C1R	1379
7.72.2.56	nppiFilterScharrVert_8s16s_C1R	1380
7.72.2.57	nppiFilterScharrVert_8u16s_C1R	1380
7.72.2.58	nppiFilterScharrVertBorder_32f_C1R	1380
7.72.2.59	nppiFilterScharrVertBorder_8s16s_C1R	1381
7.72.2.60	nppiFilterScharrVertBorder_8u16s_C1R	1381
7.72.2.61	nppiFilterSobelHoriz_16s_AC4R	1382
7.72.2.62	nppiFilterSobelHoriz_16s_C1R	1382
7.72.2.63	nppiFilterSobelHoriz_16s_C3R	1382
7.72.2.64	nppiFilterSobelHoriz_16s_C4R	1383
7.72.2.65	nppiFilterSobelHoriz_32f_AC4R	1383
7.72.2.66	nppiFilterSobelHoriz_32f_C1R	1383
7.72.2.67	nppiFilterSobelHoriz_32f_C3R	1384
7.72.2.68	nppiFilterSobelHoriz_32f_C4R	1384
7.72.2.69	nppiFilterSobelHoriz_8s16s_C1R	1384
7.72.2.70	nppiFilterSobelHoriz_8u16s_C1R	1385
7.72.2.71	nppiFilterSobelHoriz_8u_AC4R	1385
7.72.2.72	nppiFilterSobelHoriz_8u_C1R	1385
7.72.2.73	nppiFilterSobelHoriz_8u_C3R	1386
7.72.2.74	nppiFilterSobelHoriz_8u_C4R	1386
7.72.2.75	nppiFilterSobelHorizMask_32f_C1R	1386
7.72.2.76	nppiFilterSobelHorizSecond_32f_C1R	1387
7.72.2.77	nppiFilterSobelHorizSecond_8s16s_C1R	1387
7.72.2.78	nppiFilterSobelHorizSecond_8u16s_C1R	1387
7.72.2.79	nppiFilterSobelVert_16s_AC4R	1388
7.72.2.80	nppiFilterSobelVert_16s_C1R	1388
7.72.2.81	nppiFilterSobelVert_16s_C3R	1388
7.72.2.82	nppiFilterSobelVert_16s_C4R	1389

7.72.2.83	nppiFilterSobelVert_32f_AC4R	1389
7.72.2.84	nppiFilterSobelVert_32f_C1R	1390
7.72.2.85	nppiFilterSobelVert_32f_C3R	1390
7.72.2.86	nppiFilterSobelVert_32f_C4R	1390
7.72.2.87	nppiFilterSobelVert_8s16s_C1R	1391
7.72.2.88	nppiFilterSobelVert_8u16s_C1R	1391
7.72.2.89	nppiFilterSobelVert_8u_AC4R	1391
7.72.2.90	nppiFilterSobelVert_8u_C1R	1392
7.72.2.91	nppiFilterSobelVert_8u_C3R	1392
7.72.2.92	nppiFilterSobelVert_8u_C4R	1392
7.72.2.93	nppiFilterSobelVertMask_32f_C1R	1393
7.73	Geometry Transforms	1394
7.73.1	Detailed Description	1394
7.73.2	Geometric Transform API Specifics	1394
7.73.2.1	Geometric Transforms and ROIs	1394
7.73.2.2	Pixel Interpolation	1394
7.74	ResizeSqrPixel	1396
7.74.1	Detailed Description	1399
7.74.2	Error Codes	1400
7.74.3	Function Documentation	1400
7.74.3.1	nppiGetResizeRect	1400
7.74.3.2	nppiResizeAdvancedGetBufferHostSize_8u_C1R	1400
7.74.3.3	nppiResizeSqrPixel_16s_AC4R	1401
7.74.3.4	nppiResizeSqrPixel_16s_C1R	1401
7.74.3.5	nppiResizeSqrPixel_16s_C3R	1402
7.74.3.6	nppiResizeSqrPixel_16s_C4R	1402
7.74.3.7	nppiResizeSqrPixel_16s_P3R	1403
7.74.3.8	nppiResizeSqrPixel_16s_P4R	1403
7.74.3.9	nppiResizeSqrPixel_16u_AC4R	1404
7.74.3.10	nppiResizeSqrPixel_16u_C1R	1404
7.74.3.11	nppiResizeSqrPixel_16u_C3R	1405
7.74.3.12	nppiResizeSqrPixel_16u_C4R	1405
7.74.3.13	nppiResizeSqrPixel_16u_P3R	1406
7.74.3.14	nppiResizeSqrPixel_16u_P4R	1407
7.74.3.15	nppiResizeSqrPixel_32f_AC4R	1407
7.74.3.16	nppiResizeSqrPixel_32f_C1R	1408

7.74.3.17	nppiResizeSqrPixel_32f_C3R	1408
7.74.3.18	nppiResizeSqrPixel_32f_C4R	1409
7.74.3.19	nppiResizeSqrPixel_32f_P3R	1409
7.74.3.20	nppiResizeSqrPixel_32f_P4R	1410
7.74.3.21	nppiResizeSqrPixel_64f_AC4R	1411
7.74.3.22	nppiResizeSqrPixel_64f_C1R	1411
7.74.3.23	nppiResizeSqrPixel_64f_C3R	1412
7.74.3.24	nppiResizeSqrPixel_64f_C4R	1412
7.74.3.25	nppiResizeSqrPixel_64f_P3R	1413
7.74.3.26	nppiResizeSqrPixel_64f_P4R	1413
7.74.3.27	nppiResizeSqrPixel_8u_AC4R	1414
7.74.3.28	nppiResizeSqrPixel_8u_C1R	1414
7.74.3.29	nppiResizeSqrPixel_8u_C1R_Advanced	1415
7.74.3.30	nppiResizeSqrPixel_8u_C3R	1415
7.74.3.31	nppiResizeSqrPixel_8u_C4R	1416
7.74.3.32	nppiResizeSqrPixel_8u_P3R	1417
7.74.3.33	nppiResizeSqrPixel_8u_P4R	1417
7.75	Resize	1419
7.75.1	Detailed Description	1420
7.75.2	Error Codes	1421
7.75.3	Function Documentation	1421
7.75.3.1	nppiResize_16u_AC4R	1421
7.75.3.2	nppiResize_16u_C1R	1422
7.75.3.3	nppiResize_16u_C3R	1422
7.75.3.4	nppiResize_16u_C4R	1423
7.75.3.5	nppiResize_16u_P3R	1423
7.75.3.6	nppiResize_16u_P4R	1424
7.75.3.7	nppiResize_32f_AC4R	1424
7.75.3.8	nppiResize_32f_C1R	1425
7.75.3.9	nppiResize_32f_C3R	1425
7.75.3.10	nppiResize_32f_C4R	1426
7.75.3.11	nppiResize_32f_P3R	1426
7.75.3.12	nppiResize_32f_P4R	1427
7.75.3.13	nppiResize_8u_AC4R	1427
7.75.3.14	nppiResize_8u_C1R	1428
7.75.3.15	nppiResize_8u_C3R	1428

7.75.3.16	nppiResize_8u_C4R	1429
7.75.3.17	nppiResize_8u_P3R	1429
7.75.3.18	nppiResize_8u_P4R	1430
7.76	Remap	1431
7.76.1	Detailed Description	1434
7.76.2	Error Codes	1434
7.76.3	Function Documentation	1434
7.76.3.1	nppiRemap_16s_AC4R	1434
7.76.3.2	nppiRemap_16s_C1R	1435
7.76.3.3	nppiRemap_16s_C3R	1436
7.76.3.4	nppiRemap_16s_C4R	1436
7.76.3.5	nppiRemap_16s_P3R	1437
7.76.3.6	nppiRemap_16s_P4R	1437
7.76.3.7	nppiRemap_16u_AC4R	1438
7.76.3.8	nppiRemap_16u_C1R	1439
7.76.3.9	nppiRemap_16u_C3R	1439
7.76.3.10	nppiRemap_16u_C4R	1440
7.76.3.11	nppiRemap_16u_P3R	1440
7.76.3.12	nppiRemap_16u_P4R	1441
7.76.3.13	nppiRemap_32f_AC4R	1442
7.76.3.14	nppiRemap_32f_C1R	1442
7.76.3.15	nppiRemap_32f_C3R	1443
7.76.3.16	nppiRemap_32f_C4R	1443
7.76.3.17	nppiRemap_32f_P3R	1444
7.76.3.18	nppiRemap_32f_P4R	1445
7.76.3.19	nppiRemap_64f_AC4R	1445
7.76.3.20	nppiRemap_64f_C1R	1446
7.76.3.21	nppiRemap_64f_C3R	1446
7.76.3.22	nppiRemap_64f_C4R	1447
7.76.3.23	nppiRemap_64f_P3R	1448
7.76.3.24	nppiRemap_64f_P4R	1448
7.76.3.25	nppiRemap_8u_AC4R	1449
7.76.3.26	nppiRemap_8u_C1R	1449
7.76.3.27	nppiRemap_8u_C3R	1450
7.76.3.28	nppiRemap_8u_C4R	1451
7.76.3.29	nppiRemap_8u_P3R	1451

7.76.3.30	nppiRemap_8u_P4R	1452
7.77	Rotate	1453
7.77.1	Detailed Description	1454
7.77.2	Rotate Error Codes	1454
7.77.3	Function Documentation	1454
7.77.3.1	nppiGetRotateBound	1454
7.77.3.2	nppiGetRotateQuad	1455
7.77.3.3	nppiRotate_16u_AC4R	1455
7.77.3.4	nppiRotate_16u_C1R	1456
7.77.3.5	nppiRotate_16u_C3R	1456
7.77.3.6	nppiRotate_16u_C4R	1457
7.77.3.7	nppiRotate_32f_AC4R	1457
7.77.3.8	nppiRotate_32f_C1R	1458
7.77.3.9	nppiRotate_32f_C3R	1458
7.77.3.10	nppiRotate_32f_C4R	1459
7.77.3.11	nppiRotate_8u_AC4R	1459
7.77.3.12	nppiRotate_8u_C1R	1460
7.77.3.13	nppiRotate_8u_C3R	1460
7.77.3.14	nppiRotate_8u_C4R	1461
7.78	Mirror	1462
7.78.1	Detailed Description	1465
7.78.2	Mirror Error Codes	1465
7.78.3	Function Documentation	1465
7.78.3.1	nppiMirror_16s_AC4IR	1465
7.78.3.2	nppiMirror_16s_AC4R	1465
7.78.3.3	nppiMirror_16s_C1IR	1466
7.78.3.4	nppiMirror_16s_C1R	1466
7.78.3.5	nppiMirror_16s_C3IR	1466
7.78.3.6	nppiMirror_16s_C3R	1467
7.78.3.7	nppiMirror_16s_C4IR	1467
7.78.3.8	nppiMirror_16s_C4R	1467
7.78.3.9	nppiMirror_16u_AC4IR	1468
7.78.3.10	nppiMirror_16u_AC4R	1468
7.78.3.11	nppiMirror_16u_C1IR	1469
7.78.3.12	nppiMirror_16u_C1R	1469
7.78.3.13	nppiMirror_16u_C3IR	1469

7.78.3.14	nppiMirror_16u_C3R	1470
7.78.3.15	nppiMirror_16u_C4IR	1470
7.78.3.16	nppiMirror_16u_C4R	1470
7.78.3.17	nppiMirror_32f_AC4IR	1471
7.78.3.18	nppiMirror_32f_AC4R	1471
7.78.3.19	nppiMirror_32f_C1IR	1471
7.78.3.20	nppiMirror_32f_C1R	1472
7.78.3.21	nppiMirror_32f_C3IR	1472
7.78.3.22	nppiMirror_32f_C3R	1472
7.78.3.23	nppiMirror_32f_C4IR	1473
7.78.3.24	nppiMirror_32f_C4R	1473
7.78.3.25	nppiMirror_32s_AC4IR	1473
7.78.3.26	nppiMirror_32s_AC4R	1474
7.78.3.27	nppiMirror_32s_C1IR	1474
7.78.3.28	nppiMirror_32s_C1R	1474
7.78.3.29	nppiMirror_32s_C3IR	1475
7.78.3.30	nppiMirror_32s_C3R	1475
7.78.3.31	nppiMirror_32s_C4IR	1475
7.78.3.32	nppiMirror_32s_C4R	1476
7.78.3.33	nppiMirror_8u_AC4IR	1476
7.78.3.34	nppiMirror_8u_AC4R	1476
7.78.3.35	nppiMirror_8u_C1IR	1477
7.78.3.36	nppiMirror_8u_C1R	1477
7.78.3.37	nppiMirror_8u_C3IR	1477
7.78.3.38	nppiMirror_8u_C3R	1478
7.78.3.39	nppiMirror_8u_C4IR	1478
7.78.3.40	nppiMirror_8u_C4R	1478
7.79	Affine Transforms	1479
7.79.1	Detailed Description	1488
7.79.2	Affine Transform Error Codes	1488
7.79.3	Function Documentation	1488
7.79.3.1	nppiGetAffineBound	1488
7.79.3.2	nppiGetAffineQuad	1488
7.79.3.3	nppiGetAffineTransform	1489
7.79.3.4	nppiWarpAffine_16u_AC4R	1490
7.79.3.5	nppiWarpAffine_16u_C1R	1490

7.79.3.6	nppiWarpAffine_16u_C3R	1491
7.79.3.7	nppiWarpAffine_16u_C4R	1491
7.79.3.8	nppiWarpAffine_16u_P3R	1492
7.79.3.9	nppiWarpAffine_16u_P4R	1492
7.79.3.10	nppiWarpAffine_32f_AC4R	1493
7.79.3.11	nppiWarpAffine_32f_C1R	1493
7.79.3.12	nppiWarpAffine_32f_C3R	1494
7.79.3.13	nppiWarpAffine_32f_C4R	1494
7.79.3.14	nppiWarpAffine_32f_P3R	1495
7.79.3.15	nppiWarpAffine_32f_P4R	1495
7.79.3.16	nppiWarpAffine_32s_AC4R	1496
7.79.3.17	nppiWarpAffine_32s_C1R	1496
7.79.3.18	nppiWarpAffine_32s_C3R	1497
7.79.3.19	nppiWarpAffine_32s_C4R	1497
7.79.3.20	nppiWarpAffine_32s_P3R	1498
7.79.3.21	nppiWarpAffine_32s_P4R	1498
7.79.3.22	nppiWarpAffine_64f_AC4R	1499
7.79.3.23	nppiWarpAffine_64f_C1R	1499
7.79.3.24	nppiWarpAffine_64f_C3R	1500
7.79.3.25	nppiWarpAffine_64f_C4R	1500
7.79.3.26	nppiWarpAffine_64f_P3R	1501
7.79.3.27	nppiWarpAffine_64f_P4R	1501
7.79.3.28	nppiWarpAffine_8u_AC4R	1502
7.79.3.29	nppiWarpAffine_8u_C1R	1502
7.79.3.30	nppiWarpAffine_8u_C3R	1503
7.79.3.31	nppiWarpAffine_8u_C4R	1503
7.79.3.32	nppiWarpAffine_8u_P3R	1504
7.79.3.33	nppiWarpAffine_8u_P4R	1504
7.79.3.34	nppiWarpAffineBack_16u_AC4R	1505
7.79.3.35	nppiWarpAffineBack_16u_C1R	1505
7.79.3.36	nppiWarpAffineBack_16u_C3R	1506
7.79.3.37	nppiWarpAffineBack_16u_C4R	1506
7.79.3.38	nppiWarpAffineBack_16u_P3R	1507
7.79.3.39	nppiWarpAffineBack_16u_P4R	1507
7.79.3.40	nppiWarpAffineBack_32f_AC4R	1508
7.79.3.41	nppiWarpAffineBack_32f_C1R	1508

7.79.3.42 nppiWarpAffineBack_32f_C3R	1509
7.79.3.43 nppiWarpAffineBack_32f_C4R	1509
7.79.3.44 nppiWarpAffineBack_32f_P3R	1510
7.79.3.45 nppiWarpAffineBack_32f_P4R	1510
7.79.3.46 nppiWarpAffineBack_32s_AC4R	1511
7.79.3.47 nppiWarpAffineBack_32s_C1R	1511
7.79.3.48 nppiWarpAffineBack_32s_C3R	1512
7.79.3.49 nppiWarpAffineBack_32s_C4R	1512
7.79.3.50 nppiWarpAffineBack_32s_P3R	1513
7.79.3.51 nppiWarpAffineBack_32s_P4R	1513
7.79.3.52 nppiWarpAffineBack_8u_AC4R	1514
7.79.3.53 nppiWarpAffineBack_8u_C1R	1514
7.79.3.54 nppiWarpAffineBack_8u_C3R	1515
7.79.3.55 nppiWarpAffineBack_8u_C4R	1515
7.79.3.56 nppiWarpAffineBack_8u_P3R	1516
7.79.3.57 nppiWarpAffineBack_8u_P4R	1516
7.79.3.58 nppiWarpAffineQuad_16u_AC4R	1517
7.79.3.59 nppiWarpAffineQuad_16u_C1R	1517
7.79.3.60 nppiWarpAffineQuad_16u_C3R	1518
7.79.3.61 nppiWarpAffineQuad_16u_C4R	1518
7.79.3.62 nppiWarpAffineQuad_16u_P3R	1519
7.79.3.63 nppiWarpAffineQuad_16u_P4R	1519
7.79.3.64 nppiWarpAffineQuad_32f_AC4R	1520
7.79.3.65 nppiWarpAffineQuad_32f_C1R	1520
7.79.3.66 nppiWarpAffineQuad_32f_C3R	1521
7.79.3.67 nppiWarpAffineQuad_32f_C4R	1521
7.79.3.68 nppiWarpAffineQuad_32f_P3R	1522
7.79.3.69 nppiWarpAffineQuad_32f_P4R	1522
7.79.3.70 nppiWarpAffineQuad_32s_AC4R	1523
7.79.3.71 nppiWarpAffineQuad_32s_C1R	1523
7.79.3.72 nppiWarpAffineQuad_32s_C3R	1524
7.79.3.73 nppiWarpAffineQuad_32s_C4R	1524
7.79.3.74 nppiWarpAffineQuad_32s_P3R	1525
7.79.3.75 nppiWarpAffineQuad_32s_P4R	1525
7.79.3.76 nppiWarpAffineQuad_8u_AC4R	1526
7.79.3.77 nppiWarpAffineQuad_8u_C1R	1526

7.79.3.78	nppiWarpAffineQuad_8u_C3R	1527
7.79.3.79	nppiWarpAffineQuad_8u_C4R	1527
7.79.3.80	nppiWarpAffineQuad_8u_P3R	1528
7.79.3.81	nppiWarpAffineQuad_8u_P4R	1528
7.80	Perspective Transform	1529
7.80.1	Detailed Description	1537
7.80.2	Perspective Transform Error Codes	1537
7.80.3	Function Documentation	1537
7.80.3.1	nppiGetPerspectiveBound	1537
7.80.3.2	nppiGetPerspectiveQuad	1538
7.80.3.3	nppiGetPerspectiveTransform	1538
7.80.3.4	nppiWarpPerspective_16u_AC4R	1538
7.80.3.5	nppiWarpPerspective_16u_C1R	1539
7.80.3.6	nppiWarpPerspective_16u_C3R	1539
7.80.3.7	nppiWarpPerspective_16u_C4R	1540
7.80.3.8	nppiWarpPerspective_16u_P3R	1540
7.80.3.9	nppiWarpPerspective_16u_P4R	1541
7.80.3.10	nppiWarpPerspective_32f_AC4R	1541
7.80.3.11	nppiWarpPerspective_32f_C1R	1542
7.80.3.12	nppiWarpPerspective_32f_C3R	1542
7.80.3.13	nppiWarpPerspective_32f_C4R	1543
7.80.3.14	nppiWarpPerspective_32f_P3R	1543
7.80.3.15	nppiWarpPerspective_32f_P4R	1544
7.80.3.16	nppiWarpPerspective_32s_AC4R	1544
7.80.3.17	nppiWarpPerspective_32s_C1R	1545
7.80.3.18	nppiWarpPerspective_32s_C3R	1545
7.80.3.19	nppiWarpPerspective_32s_C4R	1546
7.80.3.20	nppiWarpPerspective_32s_P3R	1546
7.80.3.21	nppiWarpPerspective_32s_P4R	1547
7.80.3.22	nppiWarpPerspective_8u_AC4R	1547
7.80.3.23	nppiWarpPerspective_8u_C1R	1548
7.80.3.24	nppiWarpPerspective_8u_C3R	1548
7.80.3.25	nppiWarpPerspective_8u_C4R	1549
7.80.3.26	nppiWarpPerspective_8u_P3R	1549
7.80.3.27	nppiWarpPerspective_8u_P4R	1550
7.80.3.28	nppiWarpPerspectiveBack_16u_AC4R	1550

7.80.3.29	nppiWarpPerspectiveBack_16u_C1R	1551
7.80.3.30	nppiWarpPerspectiveBack_16u_C3R	1551
7.80.3.31	nppiWarpPerspectiveBack_16u_C4R	1552
7.80.3.32	nppiWarpPerspectiveBack_16u_P3R	1552
7.80.3.33	nppiWarpPerspectiveBack_16u_P4R	1553
7.80.3.34	nppiWarpPerspectiveBack_32f_AC4R	1553
7.80.3.35	nppiWarpPerspectiveBack_32f_C1R	1554
7.80.3.36	nppiWarpPerspectiveBack_32f_C3R	1554
7.80.3.37	nppiWarpPerspectiveBack_32f_C4R	1555
7.80.3.38	nppiWarpPerspectiveBack_32f_P3R	1555
7.80.3.39	nppiWarpPerspectiveBack_32f_P4R	1556
7.80.3.40	nppiWarpPerspectiveBack_32s_AC4R	1556
7.80.3.41	nppiWarpPerspectiveBack_32s_C1R	1557
7.80.3.42	nppiWarpPerspectiveBack_32s_C3R	1557
7.80.3.43	nppiWarpPerspectiveBack_32s_C4R	1558
7.80.3.44	nppiWarpPerspectiveBack_32s_P3R	1558
7.80.3.45	nppiWarpPerspectiveBack_32s_P4R	1559
7.80.3.46	nppiWarpPerspectiveBack_8u_AC4R	1559
7.80.3.47	nppiWarpPerspectiveBack_8u_C1R	1560
7.80.3.48	nppiWarpPerspectiveBack_8u_C3R	1560
7.80.3.49	nppiWarpPerspectiveBack_8u_C4R	1561
7.80.3.50	nppiWarpPerspectiveBack_8u_P3R	1561
7.80.3.51	nppiWarpPerspectiveBack_8u_P4R	1562
7.80.3.52	nppiWarpPerspectiveQuad_16u_AC4R	1562
7.80.3.53	nppiWarpPerspectiveQuad_16u_C1R	1563
7.80.3.54	nppiWarpPerspectiveQuad_16u_C3R	1563
7.80.3.55	nppiWarpPerspectiveQuad_16u_C4R	1564
7.80.3.56	nppiWarpPerspectiveQuad_16u_P3R	1564
7.80.3.57	nppiWarpPerspectiveQuad_16u_P4R	1565
7.80.3.58	nppiWarpPerspectiveQuad_32f_AC4R	1565
7.80.3.59	nppiWarpPerspectiveQuad_32f_C1R	1566
7.80.3.60	nppiWarpPerspectiveQuad_32f_C3R	1566
7.80.3.61	nppiWarpPerspectiveQuad_32f_C4R	1567
7.80.3.62	nppiWarpPerspectiveQuad_32f_P3R	1567
7.80.3.63	nppiWarpPerspectiveQuad_32f_P4R	1568
7.80.3.64	nppiWarpPerspectiveQuad_32s_AC4R	1568

7.80.3.65	nppiWarpPerspectiveQuad_32s_C1R	1569
7.80.3.66	nppiWarpPerspectiveQuad_32s_C3R	1569
7.80.3.67	nppiWarpPerspectiveQuad_32s_C4R	1570
7.80.3.68	nppiWarpPerspectiveQuad_32s_P3R	1570
7.80.3.69	nppiWarpPerspectiveQuad_32s_P4R	1571
7.80.3.70	nppiWarpPerspectiveQuad_8u_AC4R	1571
7.80.3.71	nppiWarpPerspectiveQuad_8u_C1R	1572
7.80.3.72	nppiWarpPerspectiveQuad_8u_C3R	1572
7.80.3.73	nppiWarpPerspectiveQuad_8u_C4R	1573
7.80.3.74	nppiWarpPerspectiveQuad_8u_P3R	1573
7.80.3.75	nppiWarpPerspectiveQuad_8u_P4R	1574
7.81	Linear Transforms	1575
7.81.1	Detailed Description	1575
7.82	Fourier Transforms	1576
7.82.1	Function Documentation	1576
7.82.1.1	nppiMagnitude_32fc32f_C1R	1576
7.82.1.2	nppiMagnitudeSqr_32fc32f_C1R	1576
7.83	Morphological Operations	1578
7.83.1	Detailed Description	1578
7.84	Dilation	1579
7.84.1	Detailed Description	1580
7.84.2	Function Documentation	1580
7.84.2.1	nppiDilate_16u_AC4R	1580
7.84.2.2	nppiDilate_16u_C1R	1580
7.84.2.3	nppiDilate_16u_C3R	1581
7.84.2.4	nppiDilate_16u_C4R	1581
7.84.2.5	nppiDilate_32f_AC4R	1582
7.84.2.6	nppiDilate_32f_C1R	1582
7.84.2.7	nppiDilate_32f_C3R	1582
7.84.2.8	nppiDilate_32f_C4R	1583
7.84.2.9	nppiDilate_8u_AC4R	1583
7.84.2.10	nppiDilate_8u_C1R	1584
7.84.2.11	nppiDilate_8u_C3R	1584
7.84.2.12	nppiDilate_8u_C4R	1585
7.85	Dilation with border control	1586
7.85.1	Detailed Description	1587

7.85.2	Function Documentation	1587
7.85.2.1	nppiDilateBorder_16u_AC4R	1587
7.85.2.2	nppiDilateBorder_16u_C1R	1588
7.85.2.3	nppiDilateBorder_16u_C3R	1588
7.85.2.4	nppiDilateBorder_16u_C4R	1589
7.85.2.5	nppiDilateBorder_32f_AC4R	1589
7.85.2.6	nppiDilateBorder_32f_C1R	1590
7.85.2.7	nppiDilateBorder_32f_C3R	1590
7.85.2.8	nppiDilateBorder_32f_C4R	1591
7.85.2.9	nppiDilateBorder_8u_AC4R	1591
7.85.2.10	nppiDilateBorder_8u_C1R	1592
7.85.2.11	nppiDilateBorder_8u_C3R	1592
7.85.2.12	nppiDilateBorder_8u_C4R	1593
7.86	Dilate3x3	1594
7.86.1	Detailed Description	1595
7.86.2	Function Documentation	1595
7.86.2.1	nppiDilate3x3_16u_AC4R	1595
7.86.2.2	nppiDilate3x3_16u_C1R	1595
7.86.2.3	nppiDilate3x3_16u_C3R	1596
7.86.2.4	nppiDilate3x3_16u_C4R	1596
7.86.2.5	nppiDilate3x3_32f_AC4R	1596
7.86.2.6	nppiDilate3x3_32f_C1R	1597
7.86.2.7	nppiDilate3x3_32f_C3R	1597
7.86.2.8	nppiDilate3x3_32f_C4R	1597
7.86.2.9	nppiDilate3x3_64f_C1R	1598
7.86.2.10	nppiDilate3x3_8u_AC4R	1598
7.86.2.11	nppiDilate3x3_8u_C1R	1598
7.86.2.12	nppiDilate3x3_8u_C3R	1599
7.86.2.13	nppiDilate3x3_8u_C4R	1599
7.87	Dilate3x3Border	1600
7.87.1	Detailed Description	1601
7.87.2	Function Documentation	1601
7.87.2.1	nppiDilate3x3Border_16u_AC4R	1601
7.87.2.2	nppiDilate3x3Border_16u_C1R	1602
7.87.2.3	nppiDilate3x3Border_16u_C3R	1602
7.87.2.4	nppiDilate3x3Border_16u_C4R	1602

7.87.2.5	nppiDilate3x3Border_32f_AC4R	1603
7.87.2.6	nppiDilate3x3Border_32f_C1R	1603
7.87.2.7	nppiDilate3x3Border_32f_C3R	1604
7.87.2.8	nppiDilate3x3Border_32f_C4R	1604
7.87.2.9	nppiDilate3x3Border_8u_AC4R	1605
7.87.2.10	nppiDilate3x3Border_8u_C1R	1605
7.87.2.11	nppiDilate3x3Border_8u_C3R	1605
7.87.2.12	nppiDilate3x3Border_8u_C4R	1606
7.88	Erode	1607
7.88.1	Detailed Description	1608
7.88.2	Function Documentation	1608
7.88.2.1	nppiErode_16u_AC4R	1608
7.88.2.2	nppiErode_16u_C1R	1608
7.88.2.3	nppiErode_16u_C3R	1609
7.88.2.4	nppiErode_16u_C4R	1609
7.88.2.5	nppiErode_32f_AC4R	1610
7.88.2.6	nppiErode_32f_C1R	1610
7.88.2.7	nppiErode_32f_C3R	1610
7.88.2.8	nppiErode_32f_C4R	1611
7.88.2.9	nppiErode_8u_AC4R	1611
7.88.2.10	nppiErode_8u_C1R	1612
7.88.2.11	nppiErode_8u_C3R	1612
7.88.2.12	nppiErode_8u_C4R	1613
7.89	Erosion with border control	1614
7.89.1	Detailed Description	1615
7.89.2	Function Documentation	1615
7.89.2.1	nppiErodeBorder_16u_AC4R	1615
7.89.2.2	nppiErodeBorder_16u_C1R	1616
7.89.2.3	nppiErodeBorder_16u_C3R	1616
7.89.2.4	nppiErodeBorder_16u_C4R	1617
7.89.2.5	nppiErodeBorder_32f_AC4R	1617
7.89.2.6	nppiErodeBorder_32f_C1R	1618
7.89.2.7	nppiErodeBorder_32f_C3R	1618
7.89.2.8	nppiErodeBorder_32f_C4R	1619
7.89.2.9	nppiErodeBorder_8u_AC4R	1619
7.89.2.10	nppiErodeBorder_8u_C1R	1620

7.89.2.11	nppiErodeBorder_8u_C3R	1620
7.89.2.12	nppiErodeBorder_8u_C4R	1621
7.90	Erode3x3	1622
7.90.1	Detailed Description	1623
7.90.2	Function Documentation	1623
7.90.2.1	nppiErode3x3_16u_AC4R	1623
7.90.2.2	nppiErode3x3_16u_C1R	1623
7.90.2.3	nppiErode3x3_16u_C3R	1624
7.90.2.4	nppiErode3x3_16u_C4R	1624
7.90.2.5	nppiErode3x3_32f_AC4R	1624
7.90.2.6	nppiErode3x3_32f_C1R	1625
7.90.2.7	nppiErode3x3_32f_C3R	1625
7.90.2.8	nppiErode3x3_32f_C4R	1625
7.90.2.9	nppiErode3x3_64f_C1R	1626
7.90.2.10	nppiErode3x3_8u_AC4R	1626
7.90.2.11	nppiErode3x3_8u_C1R	1626
7.90.2.12	nppiErode3x3_8u_C3R	1627
7.90.2.13	nppiErode3x3_8u_C4R	1627
7.91	Erode3x3Border	1628
7.91.1	Detailed Description	1629
7.91.2	Function Documentation	1629
7.91.2.1	nppiErode3x3Border_16u_AC4R	1629
7.91.2.2	nppiErode3x3Border_16u_C1R	1630
7.91.2.3	nppiErode3x3Border_16u_C3R	1630
7.91.2.4	nppiErode3x3Border_16u_C4R	1630
7.91.2.5	nppiErode3x3Border_32f_AC4R	1631
7.91.2.6	nppiErode3x3Border_32f_C1R	1631
7.91.2.7	nppiErode3x3Border_32f_C3R	1632
7.91.2.8	nppiErode3x3Border_32f_C4R	1632
7.91.2.9	nppiErode3x3Border_8u_AC4R	1633
7.91.2.10	nppiErode3x3Border_8u_C1R	1633
7.91.2.11	nppiErode3x3Border_8u_C3R	1633
7.91.2.12	nppiErode3x3Border_8u_C4R	1634
7.92	Statistical Operations	1635
7.92.1	Detailed Description	1651
7.92.2	Function Documentation	1651

7.92.2.1	nppiAverageErrorGetBufferHostSize_16s_C1R	1651
7.92.2.2	nppiAverageErrorGetBufferHostSize_16s_C2R	1651
7.92.2.3	nppiAverageErrorGetBufferHostSize_16s_C3R	1651
7.92.2.4	nppiAverageErrorGetBufferHostSize_16s_C4R	1652
7.92.2.5	nppiAverageErrorGetBufferHostSize_16sc_C1R	1652
7.92.2.6	nppiAverageErrorGetBufferHostSize_16sc_C2R	1652
7.92.2.7	nppiAverageErrorGetBufferHostSize_16sc_C3R	1652
7.92.2.8	nppiAverageErrorGetBufferHostSize_16sc_C4R	1653
7.92.2.9	nppiAverageErrorGetBufferHostSize_16u_C1R	1653
7.92.2.10	nppiAverageErrorGetBufferHostSize_16u_C2R	1653
7.92.2.11	nppiAverageErrorGetBufferHostSize_16u_C3R	1654
7.92.2.12	nppiAverageErrorGetBufferHostSize_16u_C4R	1654
7.92.2.13	nppiAverageErrorGetBufferHostSize_32f_C1R	1654
7.92.2.14	nppiAverageErrorGetBufferHostSize_32f_C2R	1654
7.92.2.15	nppiAverageErrorGetBufferHostSize_32f_C3R	1655
7.92.2.16	nppiAverageErrorGetBufferHostSize_32f_C4R	1655
7.92.2.17	nppiAverageErrorGetBufferHostSize_32fc_C1R	1655
7.92.2.18	nppiAverageErrorGetBufferHostSize_32fc_C2R	1656
7.92.2.19	nppiAverageErrorGetBufferHostSize_32fc_C3R	1656
7.92.2.20	nppiAverageErrorGetBufferHostSize_32fc_C4R	1656
7.92.2.21	nppiAverageErrorGetBufferHostSize_32s_C1R	1656
7.92.2.22	nppiAverageErrorGetBufferHostSize_32s_C2R	1657
7.92.2.23	nppiAverageErrorGetBufferHostSize_32s_C3R	1657
7.92.2.24	nppiAverageErrorGetBufferHostSize_32s_C4R	1657
7.92.2.25	nppiAverageErrorGetBufferHostSize_32sc_C1R	1658
7.92.2.26	nppiAverageErrorGetBufferHostSize_32sc_C2R	1658
7.92.2.27	nppiAverageErrorGetBufferHostSize_32sc_C3R	1658
7.92.2.28	nppiAverageErrorGetBufferHostSize_32sc_C4R	1658
7.92.2.29	nppiAverageErrorGetBufferHostSize_32u_C1R	1659
7.92.2.30	nppiAverageErrorGetBufferHostSize_32u_C2R	1659
7.92.2.31	nppiAverageErrorGetBufferHostSize_32u_C3R	1659
7.92.2.32	nppiAverageErrorGetBufferHostSize_32u_C4R	1660
7.92.2.33	nppiAverageErrorGetBufferHostSize_64f_C1R	1660
7.92.2.34	nppiAverageErrorGetBufferHostSize_64f_C2R	1660
7.92.2.35	nppiAverageErrorGetBufferHostSize_64f_C3R	1660
7.92.2.36	nppiAverageErrorGetBufferHostSize_64f_C4R	1661

7.92.2.37	nppiAverageErrorGetBufferHostSize_8s_C1R	1661
7.92.2.38	nppiAverageErrorGetBufferHostSize_8s_C2R	1661
7.92.2.39	nppiAverageErrorGetBufferHostSize_8s_C3R	1662
7.92.2.40	nppiAverageErrorGetBufferHostSize_8s_C4R	1662
7.92.2.41	nppiAverageErrorGetBufferHostSize_8u_C1R	1662
7.92.2.42	nppiAverageErrorGetBufferHostSize_8u_C2R	1662
7.92.2.43	nppiAverageErrorGetBufferHostSize_8u_C3R	1663
7.92.2.44	nppiAverageErrorGetBufferHostSize_8u_C4R	1663
7.92.2.45	nppiAverageRelativeErrorGetBufferHostSize_16s_C1R	1663
7.92.2.46	nppiAverageRelativeErrorGetBufferHostSize_16s_C2R	1664
7.92.2.47	nppiAverageRelativeErrorGetBufferHostSize_16s_C3R	1664
7.92.2.48	nppiAverageRelativeErrorGetBufferHostSize_16s_C4R	1664
7.92.2.49	nppiAverageRelativeErrorGetBufferHostSize_16sc_C1R	1664
7.92.2.50	nppiAverageRelativeErrorGetBufferHostSize_16sc_C2R	1665
7.92.2.51	nppiAverageRelativeErrorGetBufferHostSize_16sc_C3R	1665
7.92.2.52	nppiAverageRelativeErrorGetBufferHostSize_16sc_C4R	1665
7.92.2.53	nppiAverageRelativeErrorGetBufferHostSize_16u_C1R	1666
7.92.2.54	nppiAverageRelativeErrorGetBufferHostSize_16u_C2R	1666
7.92.2.55	nppiAverageRelativeErrorGetBufferHostSize_16u_C3R	1666
7.92.2.56	nppiAverageRelativeErrorGetBufferHostSize_16u_C4R	1666
7.92.2.57	nppiAverageRelativeErrorGetBufferHostSize_32f_C1R	1667
7.92.2.58	nppiAverageRelativeErrorGetBufferHostSize_32f_C2R	1667
7.92.2.59	nppiAverageRelativeErrorGetBufferHostSize_32f_C3R	1667
7.92.2.60	nppiAverageRelativeErrorGetBufferHostSize_32f_C4R	1668
7.92.2.61	nppiAverageRelativeErrorGetBufferHostSize_32fc_C1R	1668
7.92.2.62	nppiAverageRelativeErrorGetBufferHostSize_32fc_C2R	1668
7.92.2.63	nppiAverageRelativeErrorGetBufferHostSize_32fc_C3R	1668
7.92.2.64	nppiAverageRelativeErrorGetBufferHostSize_32fc_C4R	1669
7.92.2.65	nppiAverageRelativeErrorGetBufferHostSize_32s_C1R	1669
7.92.2.66	nppiAverageRelativeErrorGetBufferHostSize_32s_C2R	1669
7.92.2.67	nppiAverageRelativeErrorGetBufferHostSize_32s_C3R	1670
7.92.2.68	nppiAverageRelativeErrorGetBufferHostSize_32s_C4R	1670
7.92.2.69	nppiAverageRelativeErrorGetBufferHostSize_32sc_C1R	1670
7.92.2.70	nppiAverageRelativeErrorGetBufferHostSize_32sc_C2R	1670
7.92.2.71	nppiAverageRelativeErrorGetBufferHostSize_32sc_C3R	1671
7.92.2.72	nppiAverageRelativeErrorGetBufferHostSize_32sc_C4R	1671

7.92.2.73	nppiAverageRelativeErrorGetBufferHostSize_32u_C1R	1671
7.92.2.74	nppiAverageRelativeErrorGetBufferHostSize_32u_C2R	1672
7.92.2.75	nppiAverageRelativeErrorGetBufferHostSize_32u_C3R	1672
7.92.2.76	nppiAverageRelativeErrorGetBufferHostSize_32u_C4R	1672
7.92.2.77	nppiAverageRelativeErrorGetBufferHostSize_64f_C1R	1672
7.92.2.78	nppiAverageRelativeErrorGetBufferHostSize_64f_C2R	1673
7.92.2.79	nppiAverageRelativeErrorGetBufferHostSize_64f_C3R	1673
7.92.2.80	nppiAverageRelativeErrorGetBufferHostSize_64f_C4R	1673
7.92.2.81	nppiAverageRelativeErrorGetBufferHostSize_8s_C1R	1674
7.92.2.82	nppiAverageRelativeErrorGetBufferHostSize_8s_C2R	1674
7.92.2.83	nppiAverageRelativeErrorGetBufferHostSize_8s_C3R	1674
7.92.2.84	nppiAverageRelativeErrorGetBufferHostSize_8s_C4R	1674
7.92.2.85	nppiAverageRelativeErrorGetBufferHostSize_8u_C1R	1675
7.92.2.86	nppiAverageRelativeErrorGetBufferHostSize_8u_C2R	1675
7.92.2.87	nppiAverageRelativeErrorGetBufferHostSize_8u_C3R	1675
7.92.2.88	nppiAverageRelativeErrorGetBufferHostSize_8u_C4R	1676
7.92.2.89	nppiMaximumErrorGetBufferHostSize_16s_C1R	1676
7.92.2.90	nppiMaximumErrorGetBufferHostSize_16s_C2R	1676
7.92.2.91	nppiMaximumErrorGetBufferHostSize_16s_C3R	1676
7.92.2.92	nppiMaximumErrorGetBufferHostSize_16s_C4R	1677
7.92.2.93	nppiMaximumErrorGetBufferHostSize_16sc_C1R	1677
7.92.2.94	nppiMaximumErrorGetBufferHostSize_16sc_C2R	1677
7.92.2.95	nppiMaximumErrorGetBufferHostSize_16sc_C3R	1678
7.92.2.96	nppiMaximumErrorGetBufferHostSize_16sc_C4R	1678
7.92.2.97	nppiMaximumErrorGetBufferHostSize_16u_C1R	1678
7.92.2.98	nppiMaximumErrorGetBufferHostSize_16u_C2R	1678
7.92.2.99	nppiMaximumErrorGetBufferHostSize_16u_C3R	1679
7.92.2.100	nppiMaximumErrorGetBufferHostSize_16u_C4R	1679
7.92.2.101	nppiMaximumErrorGetBufferHostSize_32f_C1R	1679
7.92.2.102	nppiMaximumErrorGetBufferHostSize_32f_C2R	1680
7.92.2.103	nppiMaximumErrorGetBufferHostSize_32f_C3R	1680
7.92.2.104	nppiMaximumErrorGetBufferHostSize_32f_C4R	1680
7.92.2.105	nppiMaximumErrorGetBufferHostSize_32fc_C1R	1680
7.92.2.106	nppiMaximumErrorGetBufferHostSize_32fc_C2R	1681
7.92.2.107	nppiMaximumErrorGetBufferHostSize_32fc_C3R	1681
7.92.2.108	nppiMaximumErrorGetBufferHostSize_32fc_C4R	1681

7.92.2.109	nppiMaximumErrorGetBufferHostSize_32s_C1R	1682
7.92.2.110	nppiMaximumErrorGetBufferHostSize_32s_C2R	1682
7.92.2.111	nppiMaximumErrorGetBufferHostSize_32s_C3R	1682
7.92.2.112	nppiMaximumErrorGetBufferHostSize_32s_C4R	1682
7.92.2.113	nppiMaximumErrorGetBufferHostSize_32sc_C1R	1683
7.92.2.114	nppiMaximumErrorGetBufferHostSize_32sc_C2R	1683
7.92.2.115	nppiMaximumErrorGetBufferHostSize_32sc_C3R	1683
7.92.2.116	nppiMaximumErrorGetBufferHostSize_32sc_C4R	1684
7.92.2.117	nppiMaximumErrorGetBufferHostSize_32u_C1R	1684
7.92.2.118	nppiMaximumErrorGetBufferHostSize_32u_C2R	1684
7.92.2.119	nppiMaximumErrorGetBufferHostSize_32u_C3R	1684
7.92.2.120	nppiMaximumErrorGetBufferHostSize_32u_C4R	1685
7.92.2.121	nppiMaximumErrorGetBufferHostSize_64f_C1R	1685
7.92.2.122	nppiMaximumErrorGetBufferHostSize_64f_C2R	1685
7.92.2.123	nppiMaximumErrorGetBufferHostSize_64f_C3R	1686
7.92.2.124	nppiMaximumErrorGetBufferHostSize_64f_C4R	1686
7.92.2.125	nppiMaximumErrorGetBufferHostSize_8s_C1R	1686
7.92.2.126	nppiMaximumErrorGetBufferHostSize_8s_C2R	1686
7.92.2.127	nppiMaximumErrorGetBufferHostSize_8s_C3R	1687
7.92.2.128	nppiMaximumErrorGetBufferHostSize_8s_C4R	1687
7.92.2.129	nppiMaximumErrorGetBufferHostSize_8u_C1R	1687
7.92.2.130	nppiMaximumErrorGetBufferHostSize_8u_C2R	1688
7.92.2.131	nppiMaximumErrorGetBufferHostSize_8u_C3R	1688
7.92.2.132	nppiMaximumErrorGetBufferHostSize_8u_C4R	1688
7.92.2.133	nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R	1688
7.92.2.134	nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R	1689
7.92.2.135	nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R	1689
7.92.2.136	nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R	1689
7.92.2.137	nppiMaximumRelativeErrorGetBufferHostSize_16sc_C1R	1690
7.92.2.138	nppiMaximumRelativeErrorGetBufferHostSize_16sc_C2R	1690
7.92.2.139	nppiMaximumRelativeErrorGetBufferHostSize_16sc_C3R	1690
7.92.2.140	nppiMaximumRelativeErrorGetBufferHostSize_16sc_C4R	1690
7.92.2.141	nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R	1691
7.92.2.142	nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R	1691
7.92.2.143	nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R	1691
7.92.2.144	nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R	1692

7.92.2.145	<code>nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R</code>	1692
7.92.2.146	<code>nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R</code>	1692
7.92.2.147	<code>nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R</code>	1692
7.92.2.148	<code>nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R</code>	1693
7.92.2.149	<code>nppiMaximumRelativeErrorGetBufferHostSize_32fc_C1R</code>	1693
7.92.2.150	<code>nppiMaximumRelativeErrorGetBufferHostSize_32fc_C2R</code>	1693
7.92.2.151	<code>nppiMaximumRelativeErrorGetBufferHostSize_32fc_C3R</code>	1694
7.92.2.152	<code>nppiMaximumRelativeErrorGetBufferHostSize_32fc_C4R</code>	1694
7.92.2.153	<code>nppiMaximumRelativeErrorGetBufferHostSize_32s_C1R</code>	1694
7.92.2.154	<code>nppiMaximumRelativeErrorGetBufferHostSize_32s_C2R</code>	1694
7.92.2.155	<code>nppiMaximumRelativeErrorGetBufferHostSize_32s_C3R</code>	1695
7.92.2.156	<code>nppiMaximumRelativeErrorGetBufferHostSize_32s_C4R</code>	1695
7.92.2.157	<code>nppiMaximumRelativeErrorGetBufferHostSize_32sc_C1R</code>	1695
7.92.2.158	<code>nppiMaximumRelativeErrorGetBufferHostSize_32sc_C2R</code>	1696
7.92.2.159	<code>nppiMaximumRelativeErrorGetBufferHostSize_32sc_C3R</code>	1696
7.92.2.160	<code>nppiMaximumRelativeErrorGetBufferHostSize_32sc_C4R</code>	1696
7.92.2.161	<code>nppiMaximumRelativeErrorGetBufferHostSize_32u_C1R</code>	1696
7.92.2.162	<code>nppiMaximumRelativeErrorGetBufferHostSize_32u_C2R</code>	1697
7.92.2.163	<code>nppiMaximumRelativeErrorGetBufferHostSize_32u_C3R</code>	1697
7.92.2.164	<code>nppiMaximumRelativeErrorGetBufferHostSize_32u_C4R</code>	1697
7.92.2.165	<code>nppiMaximumRelativeErrorGetBufferHostSize_64f_C1R</code>	1698
7.92.2.166	<code>nppiMaximumRelativeErrorGetBufferHostSize_64f_C2R</code>	1698
7.92.2.167	<code>nppiMaximumRelativeErrorGetBufferHostSize_64f_C3R</code>	1698
7.92.2.168	<code>nppiMaximumRelativeErrorGetBufferHostSize_64f_C4R</code>	1698
7.92.2.169	<code>nppiMaximumRelativeErrorGetBufferHostSize_8s_C1R</code>	1699
7.92.2.170	<code>nppiMaximumRelativeErrorGetBufferHostSize_8s_C2R</code>	1699
7.92.2.171	<code>nppiMaximumRelativeErrorGetBufferHostSize_8s_C3R</code>	1699
7.92.2.172	<code>nppiMaximumRelativeErrorGetBufferHostSize_8s_C4R</code>	1700
7.92.2.173	<code>nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R</code>	1700
7.92.2.174	<code>nppiMaximumRelativeErrorGetBufferHostSize_8u_C2R</code>	1700
7.92.2.175	<code>nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R</code>	1700
7.92.2.176	<code>nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R</code>	1701
7.93	Sum	1702
7.93.1	Detailed Description	1704
7.93.2	Function Documentation	1705
7.93.2.1	<code>nppiSum_16s_AC4R</code>	1705

7.93.2.2	nppiSum_16s_C1R	1705
7.93.2.3	nppiSum_16s_C3R	1705
7.93.2.4	nppiSum_16s_C4R	1706
7.93.2.5	nppiSum_16u_AC4R	1706
7.93.2.6	nppiSum_16u_C1R	1706
7.93.2.7	nppiSum_16u_C3R	1707
7.93.2.8	nppiSum_16u_C4R	1707
7.93.2.9	nppiSum_32f_AC4R	1708
7.93.2.10	nppiSum_32f_C1R	1708
7.93.2.11	nppiSum_32f_C3R	1708
7.93.2.12	nppiSum_32f_C4R	1709
7.93.2.13	nppiSum_8u64s_C1R	1709
7.93.2.14	nppiSum_8u64s_C4R	1709
7.93.2.15	nppiSum_8u_AC4R	1710
7.93.2.16	nppiSum_8u_C1R	1710
7.93.2.17	nppiSum_8u_C3R	1711
7.93.2.18	nppiSum_8u_C4R	1711
7.93.2.19	nppiSumGetBufferHostSize_16s_AC4R	1711
7.93.2.20	nppiSumGetBufferHostSize_16s_C1R	1712
7.93.2.21	nppiSumGetBufferHostSize_16s_C3R	1712
7.93.2.22	nppiSumGetBufferHostSize_16s_C4R	1712
7.93.2.23	nppiSumGetBufferHostSize_16u_AC4R	1712
7.93.2.24	nppiSumGetBufferHostSize_16u_C1R	1713
7.93.2.25	nppiSumGetBufferHostSize_16u_C3R	1713
7.93.2.26	nppiSumGetBufferHostSize_16u_C4R	1713
7.93.2.27	nppiSumGetBufferHostSize_32f_AC4R	1714
7.93.2.28	nppiSumGetBufferHostSize_32f_C1R	1714
7.93.2.29	nppiSumGetBufferHostSize_32f_C3R	1714
7.93.2.30	nppiSumGetBufferHostSize_32f_C4R	1714
7.93.2.31	nppiSumGetBufferHostSize_8u64s_C1R	1715
7.93.2.32	nppiSumGetBufferHostSize_8u64s_C4R	1715
7.93.2.33	nppiSumGetBufferHostSize_8u_AC4R	1715
7.93.2.34	nppiSumGetBufferHostSize_8u_C1R	1716
7.93.2.35	nppiSumGetBufferHostSize_8u_C3R	1716
7.93.2.36	nppiSumGetBufferHostSize_8u_C4R	1716
7.94	Min	1717

7.94.1	Detailed Description	1719
7.94.2	Function Documentation	1719
7.94.2.1	nppiMin_16s_AC4R	1719
7.94.2.2	nppiMin_16s_C1R	1720
7.94.2.3	nppiMin_16s_C3R	1720
7.94.2.4	nppiMin_16s_C4R	1720
7.94.2.5	nppiMin_16u_AC4R	1721
7.94.2.6	nppiMin_16u_C1R	1721
7.94.2.7	nppiMin_16u_C3R	1721
7.94.2.8	nppiMin_16u_C4R	1722
7.94.2.9	nppiMin_32f_AC4R	1722
7.94.2.10	nppiMin_32f_C1R	1722
7.94.2.11	nppiMin_32f_C3R	1723
7.94.2.12	nppiMin_32f_C4R	1723
7.94.2.13	nppiMin_8u_AC4R	1724
7.94.2.14	nppiMin_8u_C1R	1724
7.94.2.15	nppiMin_8u_C3R	1724
7.94.2.16	nppiMin_8u_C4R	1725
7.94.2.17	nppiMinGetBufferHostSize_16s_AC4R	1725
7.94.2.18	nppiMinGetBufferHostSize_16s_C1R	1725
7.94.2.19	nppiMinGetBufferHostSize_16s_C3R	1726
7.94.2.20	nppiMinGetBufferHostSize_16s_C4R	1726
7.94.2.21	nppiMinGetBufferHostSize_16u_AC4R	1726
7.94.2.22	nppiMinGetBufferHostSize_16u_C1R	1726
7.94.2.23	nppiMinGetBufferHostSize_16u_C3R	1727
7.94.2.24	nppiMinGetBufferHostSize_16u_C4R	1727
7.94.2.25	nppiMinGetBufferHostSize_32f_AC4R	1727
7.94.2.26	nppiMinGetBufferHostSize_32f_C1R	1728
7.94.2.27	nppiMinGetBufferHostSize_32f_C3R	1728
7.94.2.28	nppiMinGetBufferHostSize_32f_C4R	1728
7.94.2.29	nppiMinGetBufferHostSize_8u_AC4R	1728
7.94.2.30	nppiMinGetBufferHostSize_8u_C1R	1729
7.94.2.31	nppiMinGetBufferHostSize_8u_C3R	1729
7.94.2.32	nppiMinGetBufferHostSize_8u_C4R	1729
7.95	MinIdx	1730
7.95.1	Detailed Description	1732

7.95.2	Function Documentation	1732
7.95.2.1	nppiMinIndx_16s_AC4R	1732
7.95.2.2	nppiMinIndx_16s_C1R	1733
7.95.2.3	nppiMinIndx_16s_C3R	1733
7.95.2.4	nppiMinIndx_16s_C4R	1734
7.95.2.5	nppiMinIndx_16u_AC4R	1734
7.95.2.6	nppiMinIndx_16u_C1R	1734
7.95.2.7	nppiMinIndx_16u_C3R	1735
7.95.2.8	nppiMinIndx_16u_C4R	1735
7.95.2.9	nppiMinIndx_32f_AC4R	1736
7.95.2.10	nppiMinIndx_32f_C1R	1736
7.95.2.11	nppiMinIndx_32f_C3R	1736
7.95.2.12	nppiMinIndx_32f_C4R	1737
7.95.2.13	nppiMinIndx_8u_AC4R	1737
7.95.2.14	nppiMinIndx_8u_C1R	1738
7.95.2.15	nppiMinIndx_8u_C3R	1738
7.95.2.16	nppiMinIndx_8u_C4R	1738
7.95.2.17	nppiMinIndxGetBufferHostSize_16s_AC4R	1739
7.95.2.18	nppiMinIndxGetBufferHostSize_16s_C1R	1739
7.95.2.19	nppiMinIndxGetBufferHostSize_16s_C3R	1739
7.95.2.20	nppiMinIndxGetBufferHostSize_16s_C4R	1740
7.95.2.21	nppiMinIndxGetBufferHostSize_16u_AC4R	1740
7.95.2.22	nppiMinIndxGetBufferHostSize_16u_C1R	1740
7.95.2.23	nppiMinIndxGetBufferHostSize_16u_C3R	1741
7.95.2.24	nppiMinIndxGetBufferHostSize_16u_C4R	1741
7.95.2.25	nppiMinIndxGetBufferHostSize_32f_AC4R	1741
7.95.2.26	nppiMinIndxGetBufferHostSize_32f_C1R	1741
7.95.2.27	nppiMinIndxGetBufferHostSize_32f_C3R	1742
7.95.2.28	nppiMinIndxGetBufferHostSize_32f_C4R	1742
7.95.2.29	nppiMinIndxGetBufferHostSize_8u_AC4R	1742
7.95.2.30	nppiMinIndxGetBufferHostSize_8u_C1R	1743
7.95.2.31	nppiMinIndxGetBufferHostSize_8u_C3R	1743
7.95.2.32	nppiMinIndxGetBufferHostSize_8u_C4R	1743
7.96	Max	1744
7.96.1	Detailed Description	1746
7.96.2	Function Documentation	1746

7.96.2.1	nppiMax_16s_AC4R	1746
7.96.2.2	nppiMax_16s_C1R	1747
7.96.2.3	nppiMax_16s_C3R	1747
7.96.2.4	nppiMax_16s_C4R	1747
7.96.2.5	nppiMax_16u_AC4R	1748
7.96.2.6	nppiMax_16u_C1R	1748
7.96.2.7	nppiMax_16u_C3R	1748
7.96.2.8	nppiMax_16u_C4R	1749
7.96.2.9	nppiMax_32f_AC4R	1749
7.96.2.10	nppiMax_32f_C1R	1749
7.96.2.11	nppiMax_32f_C3R	1750
7.96.2.12	nppiMax_32f_C4R	1750
7.96.2.13	nppiMax_8u_AC4R	1751
7.96.2.14	nppiMax_8u_C1R	1751
7.96.2.15	nppiMax_8u_C3R	1751
7.96.2.16	nppiMax_8u_C4R	1752
7.96.2.17	nppiMaxGetBufferHostSize_16s_AC4R	1752
7.96.2.18	nppiMaxGetBufferHostSize_16s_C1R	1752
7.96.2.19	nppiMaxGetBufferHostSize_16s_C3R	1753
7.96.2.20	nppiMaxGetBufferHostSize_16s_C4R	1753
7.96.2.21	nppiMaxGetBufferHostSize_16u_AC4R	1753
7.96.2.22	nppiMaxGetBufferHostSize_16u_C1R	1753
7.96.2.23	nppiMaxGetBufferHostSize_16u_C3R	1754
7.96.2.24	nppiMaxGetBufferHostSize_16u_C4R	1754
7.96.2.25	nppiMaxGetBufferHostSize_32f_AC4R	1754
7.96.2.26	nppiMaxGetBufferHostSize_32f_C1R	1755
7.96.2.27	nppiMaxGetBufferHostSize_32f_C3R	1755
7.96.2.28	nppiMaxGetBufferHostSize_32f_C4R	1755
7.96.2.29	nppiMaxGetBufferHostSize_8u_AC4R	1755
7.96.2.30	nppiMaxGetBufferHostSize_8u_C1R	1756
7.96.2.31	nppiMaxGetBufferHostSize_8u_C3R	1756
7.96.2.32	nppiMaxGetBufferHostSize_8u_C4R	1756
7.97	MaxIndx	1757
7.97.1	Detailed Description	1759
7.97.2	Function Documentation	1759
7.97.2.1	nppiMaxIndx_16s_AC4R	1759

7.97.2.2	nppiMaxIndx_16s_C1R	1760
7.97.2.3	nppiMaxIndx_16s_C3R	1760
7.97.2.4	nppiMaxIndx_16s_C4R	1761
7.97.2.5	nppiMaxIndx_16u_AC4R	1761
7.97.2.6	nppiMaxIndx_16u_C1R	1761
7.97.2.7	nppiMaxIndx_16u_C3R	1762
7.97.2.8	nppiMaxIndx_16u_C4R	1762
7.97.2.9	nppiMaxIndx_32f_AC4R	1763
7.97.2.10	nppiMaxIndx_32f_C1R	1763
7.97.2.11	nppiMaxIndx_32f_C3R	1763
7.97.2.12	nppiMaxIndx_32f_C4R	1764
7.97.2.13	nppiMaxIndx_8u_AC4R	1764
7.97.2.14	nppiMaxIndx_8u_C1R	1765
7.97.2.15	nppiMaxIndx_8u_C3R	1765
7.97.2.16	nppiMaxIndx_8u_C4R	1765
7.97.2.17	nppiMaxIndxGetBufferHostSize_16s_AC4R	1766
7.97.2.18	nppiMaxIndxGetBufferHostSize_16s_C1R	1766
7.97.2.19	nppiMaxIndxGetBufferHostSize_16s_C3R	1766
7.97.2.20	nppiMaxIndxGetBufferHostSize_16s_C4R	1767
7.97.2.21	nppiMaxIndxGetBufferHostSize_16u_AC4R	1767
7.97.2.22	nppiMaxIndxGetBufferHostSize_16u_C1R	1767
7.97.2.23	nppiMaxIndxGetBufferHostSize_16u_C3R	1768
7.97.2.24	nppiMaxIndxGetBufferHostSize_16u_C4R	1768
7.97.2.25	nppiMaxIndxGetBufferHostSize_32f_AC4R	1768
7.97.2.26	nppiMaxIndxGetBufferHostSize_32f_C1R	1768
7.97.2.27	nppiMaxIndxGetBufferHostSize_32f_C3R	1769
7.97.2.28	nppiMaxIndxGetBufferHostSize_32f_C4R	1769
7.97.2.29	nppiMaxIndxGetBufferHostSize_8u_AC4R	1769
7.97.2.30	nppiMaxIndxGetBufferHostSize_8u_C1R	1770
7.97.2.31	nppiMaxIndxGetBufferHostSize_8u_C3R	1770
7.97.2.32	nppiMaxIndxGetBufferHostSize_8u_C4R	1770
7.98	MinMax	1771
7.98.1	Detailed Description	1773
7.98.2	Function Documentation	1773
7.98.2.1	nppiMinMax_16s_AC4R	1773
7.98.2.2	nppiMinMax_16s_C1R	1774

7.98.2.3	nppiMinMax_16s_C3R	1774
7.98.2.4	nppiMinMax_16s_C4R	1774
7.98.2.5	nppiMinMax_16u_AC4R	1775
7.98.2.6	nppiMinMax_16u_C1R	1775
7.98.2.7	nppiMinMax_16u_C3R	1776
7.98.2.8	nppiMinMax_16u_C4R	1776
7.98.2.9	nppiMinMax_32f_AC4R	1776
7.98.2.10	nppiMinMax_32f_C1R	1777
7.98.2.11	nppiMinMax_32f_C3R	1777
7.98.2.12	nppiMinMax_32f_C4R	1778
7.98.2.13	nppiMinMax_8u_AC4R	1778
7.98.2.14	nppiMinMax_8u_C1R	1778
7.98.2.15	nppiMinMax_8u_C3R	1779
7.98.2.16	nppiMinMax_8u_C4R	1779
7.98.2.17	nppiMinMaxGetBufferHostSize_16s_AC4R	1780
7.98.2.18	nppiMinMaxGetBufferHostSize_16s_C1R	1780
7.98.2.19	nppiMinMaxGetBufferHostSize_16s_C3R	1780
7.98.2.20	nppiMinMaxGetBufferHostSize_16s_C4R	1780
7.98.2.21	nppiMinMaxGetBufferHostSize_16u_AC4R	1781
7.98.2.22	nppiMinMaxGetBufferHostSize_16u_C1R	1781
7.98.2.23	nppiMinMaxGetBufferHostSize_16u_C3R	1781
7.98.2.24	nppiMinMaxGetBufferHostSize_16u_C4R	1782
7.98.2.25	nppiMinMaxGetBufferHostSize_32f_AC4R	1782
7.98.2.26	nppiMinMaxGetBufferHostSize_32f_C1R	1782
7.98.2.27	nppiMinMaxGetBufferHostSize_32f_C3R	1782
7.98.2.28	nppiMinMaxGetBufferHostSize_32f_C4R	1783
7.98.2.29	nppiMinMaxGetBufferHostSize_8u_AC4R	1783
7.98.2.30	nppiMinMaxGetBufferHostSize_8u_C1R	1783
7.98.2.31	nppiMinMaxGetBufferHostSize_8u_C3R	1784
7.98.2.32	nppiMinMaxGetBufferHostSize_8u_C4R	1784
7.99	MinMaxIndx	1785
7.99.1	Detailed Description	1788
7.99.2	Function Documentation	1788
7.99.2.1	nppiMinMaxIndx_16u_C1MR	1788
7.99.2.2	nppiMinMaxIndx_16u_C1R	1789
7.99.2.3	nppiMinMaxIndx_16u_C3CMR	1789

7.99.2.4	nppiMinMaxIndx_16u_C3CR	1790
7.99.2.5	nppiMinMaxIndx_32f_C1MR	1791
7.99.2.6	nppiMinMaxIndx_32f_C1R	1791
7.99.2.7	nppiMinMaxIndx_32f_C3CMR	1792
7.99.2.8	nppiMinMaxIndx_32f_C3CR	1792
7.99.2.9	nppiMinMaxIndx_8s_C1MR	1793
7.99.2.10	nppiMinMaxIndx_8s_C1R	1793
7.99.2.11	nppiMinMaxIndx_8s_C3CMR	1794
7.99.2.12	nppiMinMaxIndx_8s_C3CR	1794
7.99.2.13	nppiMinMaxIndx_8u_C1MR	1795
7.99.2.14	nppiMinMaxIndx_8u_C1R	1796
7.99.2.15	nppiMinMaxIndx_8u_C3CMR	1796
7.99.2.16	nppiMinMaxIndx_8u_C3CR	1797
7.99.2.17	nppiMinMaxIndxGetBufferHostSize_16u_C1MR	1797
7.99.2.18	nppiMinMaxIndxGetBufferHostSize_16u_C1R	1797
7.99.2.19	nppiMinMaxIndxGetBufferHostSize_16u_C3CMR	1798
7.99.2.20	nppiMinMaxIndxGetBufferHostSize_16u_C3CR	1798
7.99.2.21	nppiMinMaxIndxGetBufferHostSize_32f_C1MR	1798
7.99.2.22	nppiMinMaxIndxGetBufferHostSize_32f_C1R	1798
7.99.2.23	nppiMinMaxIndxGetBufferHostSize_32f_C3CMR	1799
7.99.2.24	nppiMinMaxIndxGetBufferHostSize_32f_C3CR	1799
7.99.2.25	nppiMinMaxIndxGetBufferHostSize_8s_C1MR	1799
7.99.2.26	nppiMinMaxIndxGetBufferHostSize_8s_C1R	1800
7.99.2.27	nppiMinMaxIndxGetBufferHostSize_8s_C3CMR	1800
7.99.2.28	nppiMinMaxIndxGetBufferHostSize_8s_C3CR	1800
7.99.2.29	nppiMinMaxIndxGetBufferHostSize_8u_C1MR	1800
7.99.2.30	nppiMinMaxIndxGetBufferHostSize_8u_C1R	1801
7.99.2.31	nppiMinMaxIndxGetBufferHostSize_8u_C3CMR	1801
7.99.2.32	nppiMinMaxIndxGetBufferHostSize_8u_C3CR	1801
7.100	Mean	1802
7.100.1	Detailed Description	1805
7.100.2	Function Documentation	1806
7.100.2.1	nppiMean_16s_AC4R	1806
7.100.2.2	nppiMean_16s_C1R	1806
7.100.2.3	nppiMean_16s_C3R	1806
7.100.2.4	nppiMean_16s_C4R	1807

7.100.2.5 nppiMean_16u_AC4R	1807
7.100.2.6 nppiMean_16u_C1MR	1807
7.100.2.7 nppiMean_16u_C1R	1808
7.100.2.8 nppiMean_16u_C3CMR	1808
7.100.2.9 nppiMean_16u_C3R	1809
7.100.2.10 nppiMean_16u_C4R	1809
7.100.2.11 nppiMean_32f_AC4R	1809
7.100.2.12 nppiMean_32f_C1MR	1810
7.100.2.13 nppiMean_32f_C1R	1810
7.100.2.14 nppiMean_32f_C3CMR	1811
7.100.2.15 nppiMean_32f_C3R	1811
7.100.2.16 nppiMean_32f_C4R	1811
7.100.2.17 nppiMean_8s_C1MR	1812
7.100.2.18 nppiMean_8s_C3CMR	1812
7.100.2.19 nppiMean_8u_AC4R	1813
7.100.2.20 nppiMean_8u_C1MR	1813
7.100.2.21 nppiMean_8u_C1R	1814
7.100.2.22 nppiMean_8u_C3CMR	1814
7.100.2.23 nppiMean_8u_C3R	1814
7.100.2.24 nppiMean_8u_C4R	1815
7.100.2.25 nppiMeanGetBufferHostSize_16s_AC4R	1815
7.100.2.26 nppiMeanGetBufferHostSize_16s_C1R	1815
7.100.2.27 nppiMeanGetBufferHostSize_16s_C3R	1816
7.100.2.28 nppiMeanGetBufferHostSize_16s_C4R	1816
7.100.2.29 nppiMeanGetBufferHostSize_16u_AC4R	1816
7.100.2.30 nppiMeanGetBufferHostSize_16u_C1MR	1817
7.100.2.31 nppiMeanGetBufferHostSize_16u_C1R	1817
7.100.2.32 nppiMeanGetBufferHostSize_16u_C3CMR	1817
7.100.2.33 nppiMeanGetBufferHostSize_16u_C3R	1817
7.100.2.34 nppiMeanGetBufferHostSize_16u_C4R	1818
7.100.2.35 nppiMeanGetBufferHostSize_32f_AC4R	1818
7.100.2.36 nppiMeanGetBufferHostSize_32f_C1MR	1818
7.100.2.37 nppiMeanGetBufferHostSize_32f_C1R	1819
7.100.2.38 nppiMeanGetBufferHostSize_32f_C3CMR	1819
7.100.2.39 nppiMeanGetBufferHostSize_32f_C3R	1819
7.100.2.40 nppiMeanGetBufferHostSize_32f_C4R	1819

7.100.2.41	ppiMeanGetBufferHostSize_8s_C1MR	1820
7.100.2.42	ppiMeanGetBufferHostSize_8s_C3CMR	1820
7.100.2.43	ppiMeanGetBufferHostSize_8u_AC4R	1820
7.100.2.44	ppiMeanGetBufferHostSize_8u_C1MR	1821
7.100.2.45	ppiMeanGetBufferHostSize_8u_C1R	1821
7.100.2.46	ppiMeanGetBufferHostSize_8u_C3CMR	1821
7.100.2.47	ppiMeanGetBufferHostSize_8u_C3R	1821
7.100.2.48	ppiMeanGetBufferHostSize_8u_C4R	1822
7.101	Mean_StdDev	1823
7.101.1	Detailed Description	1826
7.101.2	Function Documentation	1826
7.101.2.1	ppiMean_StdDev_16u_C1MR	1826
7.101.2.2	ppiMean_StdDev_16u_C1R	1827
7.101.2.3	ppiMean_StdDev_16u_C3CMR	1827
7.101.2.4	ppiMean_StdDev_16u_C3CR	1828
7.101.2.5	ppiMean_StdDev_32f_C1MR	1828
7.101.2.6	ppiMean_StdDev_32f_C1R	1829
7.101.2.7	ppiMean_StdDev_32f_C3CMR	1829
7.101.2.8	ppiMean_StdDev_32f_C3CR	1830
7.101.2.9	ppiMean_StdDev_8s_C1MR	1830
7.101.2.10	ppiMean_StdDev_8s_C1R	1831
7.101.2.11	ppiMean_StdDev_8s_C3CMR	1831
7.101.2.12	ppiMean_StdDev_8s_C3CR	1832
7.101.2.13	ppiMean_StdDev_8u_C1MR	1832
7.101.2.14	ppiMean_StdDev_8u_C1R	1833
7.101.2.15	ppiMean_StdDev_8u_C3CMR	1833
7.101.2.16	ppiMean_StdDev_8u_C3CR	1834
7.101.2.17	ppiMeanStdDevGetBufferHostSize_16u_C1MR	1834
7.101.2.18	ppiMeanStdDevGetBufferHostSize_16u_C1R	1834
7.101.2.19	ppiMeanStdDevGetBufferHostSize_16u_C3CMR	1835
7.101.2.20	ppiMeanStdDevGetBufferHostSize_16u_C3CR	1835
7.101.2.21	ppiMeanStdDevGetBufferHostSize_32f_C1MR	1835
7.101.2.22	ppiMeanStdDevGetBufferHostSize_32f_C1R	1835
7.101.2.23	ppiMeanStdDevGetBufferHostSize_32f_C3CMR	1836
7.101.2.24	ppiMeanStdDevGetBufferHostSize_32f_C3CR	1836
7.101.2.25	ppiMeanStdDevGetBufferHostSize_8s_C1MR	1836

7.101.2.26	ippiMeanStdDevGetBufferHostSize_8s_C1R	1837
7.101.2.27	ippiMeanStdDevGetBufferHostSize_8s_C3CMR	1837
7.101.2.28	ippiMeanStdDevGetBufferHostSize_8s_C3CR	1837
7.101.2.29	ippiMeanStdDevGetBufferHostSize_8u_C1MR	1837
7.101.2.30	ippiMeanStdDevGetBufferHostSize_8u_C1R	1838
7.101.2.31	ippiMeanStdDevGetBufferHostSize_8u_C3CMR	1838
7.101.2.32	ippiMeanStdDevGetBufferHostSize_8u_C3CR	1838
7.102	Image Norms	1839
7.102.1	Detailed Description	1839
7.103	Norm_Inf	1841
7.103.1	Detailed Description	1845
7.103.2	Function Documentation	1845
7.103.2.1	ippiNorm_Inf_16s_AC4R	1845
7.103.2.2	ippiNorm_Inf_16s_C1R	1845
7.103.2.3	ippiNorm_Inf_16s_C3R	1845
7.103.2.4	ippiNorm_Inf_16s_C4R	1846
7.103.2.5	ippiNorm_Inf_16u_AC4R	1846
7.103.2.6	ippiNorm_Inf_16u_C1MR	1847
7.103.2.7	ippiNorm_Inf_16u_C1R	1847
7.103.2.8	ippiNorm_Inf_16u_C3CMR	1847
7.103.2.9	ippiNorm_Inf_16u_C3R	1848
7.103.2.10	ippiNorm_Inf_16u_C4R	1848
7.103.2.11	ippiNorm_Inf_32f_AC4R	1849
7.103.2.12	ippiNorm_Inf_32f_C1MR	1849
7.103.2.13	ippiNorm_Inf_32f_C1R	1849
7.103.2.14	ippiNorm_Inf_32f_C3CMR	1850
7.103.2.15	ippiNorm_Inf_32f_C3R	1850
7.103.2.16	ippiNorm_Inf_32f_C4R	1851
7.103.2.17	ippiNorm_Inf_32s_C1R	1851
7.103.2.18	ippiNorm_Inf_8s_C1MR	1851
7.103.2.19	ippiNorm_Inf_8s_C3CMR	1852
7.103.2.20	ippiNorm_Inf_8u_AC4R	1852
7.103.2.21	ippiNorm_Inf_8u_C1MR	1853
7.103.2.22	ippiNorm_Inf_8u_C1R	1853
7.103.2.23	ippiNorm_Inf_8u_C3CMR	1853
7.103.2.24	ippiNorm_Inf_8u_C3R	1854

7.103.2.25	nppiNorm_Inf_8u_C4R	1854
7.103.2.26	nppiNormInfGetBufferHostSize_16s_AC4R	1855
7.103.2.27	nppiNormInfGetBufferHostSize_16s_C1R	1855
7.103.2.28	nppiNormInfGetBufferHostSize_16s_C3R	1855
7.103.2.29	nppiNormInfGetBufferHostSize_16s_C4R	1855
7.103.2.30	nppiNormInfGetBufferHostSize_16u_AC4R	1856
7.103.2.31	nppiNormInfGetBufferHostSize_16u_C1MR	1856
7.103.2.32	nppiNormInfGetBufferHostSize_16u_C1R	1856
7.103.2.33	nppiNormInfGetBufferHostSize_16u_C3CMR	1857
7.103.2.34	nppiNormInfGetBufferHostSize_16u_C3R	1857
7.103.2.35	nppiNormInfGetBufferHostSize_16u_C4R	1857
7.103.2.36	nppiNormInfGetBufferHostSize_32f_AC4R	1857
7.103.2.37	nppiNormInfGetBufferHostSize_32f_C1MR	1858
7.103.2.38	nppiNormInfGetBufferHostSize_32f_C1R	1858
7.103.2.39	nppiNormInfGetBufferHostSize_32f_C3CMR	1858
7.103.2.40	nppiNormInfGetBufferHostSize_32f_C3R	1859
7.103.2.41	nppiNormInfGetBufferHostSize_32f_C4R	1859
7.103.2.42	nppiNormInfGetBufferHostSize_32s_C1R	1859
7.103.2.43	nppiNormInfGetBufferHostSize_8s_C1MR	1859
7.103.2.44	nppiNormInfGetBufferHostSize_8s_C3CMR	1860
7.103.2.45	nppiNormInfGetBufferHostSize_8u_AC4R	1860
7.103.2.46	nppiNormInfGetBufferHostSize_8u_C1MR	1860
7.103.2.47	nppiNormInfGetBufferHostSize_8u_C1R	1861
7.103.2.48	nppiNormInfGetBufferHostSize_8u_C3CMR	1861
7.103.2.49	nppiNormInfGetBufferHostSize_8u_C3R	1861
7.103.2.50	nppiNormInfGetBufferHostSize_8u_C4R	1861
7.104	Norm_L1	1863
7.104.1	Detailed Description	1866
7.104.2	Function Documentation	1867
7.104.2.1	nppiNorm_L1_16s_AC4R	1867
7.104.2.2	nppiNorm_L1_16s_C1R	1867
7.104.2.3	nppiNorm_L1_16s_C3R	1867
7.104.2.4	nppiNorm_L1_16s_C4R	1868
7.104.2.5	nppiNorm_L1_16u_AC4R	1868
7.104.2.6	nppiNorm_L1_16u_C1MR	1868
7.104.2.7	nppiNorm_L1_16u_C1R	1869

7.104.2.8 nppiNorm_L1_16u_C3CMR	1869
7.104.2.9 nppiNorm_L1_16u_C3R	1870
7.104.2.10 nppiNorm_L1_16u_C4R	1870
7.104.2.11 nppiNorm_L1_32f_AC4R	1870
7.104.2.12 nppiNorm_L1_32f_C1MR	1871
7.104.2.13 nppiNorm_L1_32f_C1R	1871
7.104.2.14 nppiNorm_L1_32f_C3CMR	1872
7.104.2.15 nppiNorm_L1_32f_C3R	1872
7.104.2.16 nppiNorm_L1_32f_C4R	1872
7.104.2.17 nppiNorm_L1_8s_C1MR	1873
7.104.2.18 nppiNorm_L1_8s_C3CMR	1873
7.104.2.19 nppiNorm_L1_8u_AC4R	1874
7.104.2.20 nppiNorm_L1_8u_C1MR	1874
7.104.2.21 nppiNorm_L1_8u_C1R	1874
7.104.2.22 nppiNorm_L1_8u_C3CMR	1875
7.104.2.23 nppiNorm_L1_8u_C3R	1875
7.104.2.24 nppiNorm_L1_8u_C4R	1876
7.104.2.25 nppiNormL1GetBufferHostSize_16s_AC4R	1876
7.104.2.26 nppiNormL1GetBufferHostSize_16s_C1R	1876
7.104.2.27 nppiNormL1GetBufferHostSize_16s_C3R	1877
7.104.2.28 nppiNormL1GetBufferHostSize_16s_C4R	1877
7.104.2.29 nppiNormL1GetBufferHostSize_16u_AC4R	1877
7.104.2.30 nppiNormL1GetBufferHostSize_16u_C1MR	1877
7.104.2.31 nppiNormL1GetBufferHostSize_16u_C1R	1878
7.104.2.32 nppiNormL1GetBufferHostSize_16u_C3CMR	1878
7.104.2.33 nppiNormL1GetBufferHostSize_16u_C3R	1878
7.104.2.34 nppiNormL1GetBufferHostSize_16u_C4R	1879
7.104.2.35 nppiNormL1GetBufferHostSize_32f_AC4R	1879
7.104.2.36 nppiNormL1GetBufferHostSize_32f_C1MR	1879
7.104.2.37 nppiNormL1GetBufferHostSize_32f_C1R	1879
7.104.2.38 nppiNormL1GetBufferHostSize_32f_C3CMR	1880
7.104.2.39 nppiNormL1GetBufferHostSize_32f_C3R	1880
7.104.2.40 nppiNormL1GetBufferHostSize_32f_C4R	1880
7.104.2.41 nppiNormL1GetBufferHostSize_8s_C1MR	1881
7.104.2.42 nppiNormL1GetBufferHostSize_8s_C3CMR	1881
7.104.2.43 nppiNormL1GetBufferHostSize_8u_AC4R	1881

7.104.2.44	nppiNormL1GetBufferHostSize_8u_C1MR	1881
7.104.2.45	nppiNormL1GetBufferHostSize_8u_C1R	1882
7.104.2.46	nppiNormL1GetBufferHostSize_8u_C3CMR	1882
7.104.2.47	nppiNormL1GetBufferHostSize_8u_C3R	1882
7.104.2.48	nppiNormL1GetBufferHostSize_8u_C4R	1883
7.105	Norm_L2	1884
7.105.1	Detailed Description	1887
7.105.2	Function Documentation	1888
7.105.2.1	nppiNorm_L2_16s_AC4R	1888
7.105.2.2	nppiNorm_L2_16s_C1R	1888
7.105.2.3	nppiNorm_L2_16s_C3R	1888
7.105.2.4	nppiNorm_L2_16s_C4R	1889
7.105.2.5	nppiNorm_L2_16u_AC4R	1889
7.105.2.6	nppiNorm_L2_16u_C1MR	1889
7.105.2.7	nppiNorm_L2_16u_C1R	1890
7.105.2.8	nppiNorm_L2_16u_C3CMR	1890
7.105.2.9	nppiNorm_L2_16u_C3R	1891
7.105.2.10	nppiNorm_L2_16u_C4R	1891
7.105.2.11	nppiNorm_L2_32f_AC4R	1891
7.105.2.12	nppiNorm_L2_32f_C1MR	1892
7.105.2.13	nppiNorm_L2_32f_C1R	1892
7.105.2.14	nppiNorm_L2_32f_C3CMR	1893
7.105.2.15	nppiNorm_L2_32f_C3R	1893
7.105.2.16	nppiNorm_L2_32f_C4R	1893
7.105.2.17	nppiNorm_L2_8s_C1MR	1894
7.105.2.18	nppiNorm_L2_8s_C3CMR	1894
7.105.2.19	nppiNorm_L2_8u_AC4R	1895
7.105.2.20	nppiNorm_L2_8u_C1MR	1895
7.105.2.21	nppiNorm_L2_8u_C1R	1895
7.105.2.22	nppiNorm_L2_8u_C3CMR	1896
7.105.2.23	nppiNorm_L2_8u_C3R	1896
7.105.2.24	nppiNorm_L2_8u_C4R	1897
7.105.2.25	nppiNormL2GetBufferHostSize_16s_AC4R	1897
7.105.2.26	nppiNormL2GetBufferHostSize_16s_C1R	1897
7.105.2.27	nppiNormL2GetBufferHostSize_16s_C3R	1898
7.105.2.28	nppiNormL2GetBufferHostSize_16s_C4R	1898

7.105.2.29	nppiNormL2GetBufferHostSize_16u_AC4R	1898
7.105.2.30	nppiNormL2GetBufferHostSize_16u_C1MR	1898
7.105.2.31	nppiNormL2GetBufferHostSize_16u_C1R	1899
7.105.2.32	nppiNormL2GetBufferHostSize_16u_C3CMR	1899
7.105.2.33	nppiNormL2GetBufferHostSize_16u_C3R	1899
7.105.2.34	nppiNormL2GetBufferHostSize_16u_C4R	1900
7.105.2.35	nppiNormL2GetBufferHostSize_32f_AC4R	1900
7.105.2.36	nppiNormL2GetBufferHostSize_32f_C1MR	1900
7.105.2.37	nppiNormL2GetBufferHostSize_32f_C1R	1900
7.105.2.38	nppiNormL2GetBufferHostSize_32f_C3CMR	1901
7.105.2.39	nppiNormL2GetBufferHostSize_32f_C3R	1901
7.105.2.40	nppiNormL2GetBufferHostSize_32f_C4R	1901
7.105.2.41	nppiNormL2GetBufferHostSize_8s_C1MR	1902
7.105.2.42	nppiNormL2GetBufferHostSize_8s_C3CMR	1902
7.105.2.43	nppiNormL2GetBufferHostSize_8u_AC4R	1902
7.105.2.44	nppiNormL2GetBufferHostSize_8u_C1MR	1902
7.105.2.45	nppiNormL2GetBufferHostSize_8u_C1R	1903
7.105.2.46	nppiNormL2GetBufferHostSize_8u_C3CMR	1903
7.105.2.47	nppiNormL2GetBufferHostSize_8u_C3R	1903
7.105.2.48	nppiNormL2GetBufferHostSize_8u_C4R	1904
7.106	NormDiff_Inf	1905
7.106.1	Detailed Description	1909
7.106.2	Function Documentation	1909
7.106.2.1	nppiNormDiff_Inf_16s_AC4R	1909
7.106.2.2	nppiNormDiff_Inf_16s_C1R	1910
7.106.2.3	nppiNormDiff_Inf_16s_C3R	1910
7.106.2.4	nppiNormDiff_Inf_16s_C4R	1910
7.106.2.5	nppiNormDiff_Inf_16u_AC4R	1911
7.106.2.6	nppiNormDiff_Inf_16u_C1MR	1911
7.106.2.7	nppiNormDiff_Inf_16u_C1R	1912
7.106.2.8	nppiNormDiff_Inf_16u_C3CMR	1912
7.106.2.9	nppiNormDiff_Inf_16u_C3R	1913
7.106.2.10	nppiNormDiff_Inf_16u_C4R	1913
7.106.2.11	nppiNormDiff_Inf_32f_AC4R	1914
7.106.2.12	nppiNormDiff_Inf_32f_C1MR	1914
7.106.2.13	nppiNormDiff_Inf_32f_C1R	1915

7.106.2.14nppiNormDiff_Inf_32f_C3CMR	1915
7.106.2.15nppiNormDiff_Inf_32f_C3R	1916
7.106.2.16nppiNormDiff_Inf_32f_C4R	1916
7.106.2.17nppiNormDiff_Inf_8s_C1MR	1916
7.106.2.18nppiNormDiff_Inf_8s_C3CMR	1917
7.106.2.19nppiNormDiff_Inf_8u_AC4R	1917
7.106.2.20nppiNormDiff_Inf_8u_C1MR	1918
7.106.2.21nppiNormDiff_Inf_8u_C1R	1918
7.106.2.22nppiNormDiff_Inf_8u_C3CMR	1919
7.106.2.23nppiNormDiff_Inf_8u_C3R	1919
7.106.2.24nppiNormDiff_Inf_8u_C4R	1920
7.106.2.25nppiNormDiffInfGetBufferHostSize_16s_AC4R	1920
7.106.2.26nppiNormDiffInfGetBufferHostSize_16s_C1R	1921
7.106.2.27nppiNormDiffInfGetBufferHostSize_16s_C3R	1921
7.106.2.28nppiNormDiffInfGetBufferHostSize_16s_C4R	1921
7.106.2.29nppiNormDiffInfGetBufferHostSize_16u_AC4R	1921
7.106.2.30nppiNormDiffInfGetBufferHostSize_16u_C1MR	1922
7.106.2.31nppiNormDiffInfGetBufferHostSize_16u_C1R	1922
7.106.2.32nppiNormDiffInfGetBufferHostSize_16u_C3CMR	1922
7.106.2.33nppiNormDiffInfGetBufferHostSize_16u_C3R	1923
7.106.2.34nppiNormDiffInfGetBufferHostSize_16u_C4R	1923
7.106.2.35nppiNormDiffInfGetBufferHostSize_32f_AC4R	1923
7.106.2.36nppiNormDiffInfGetBufferHostSize_32f_C1MR	1923
7.106.2.37nppiNormDiffInfGetBufferHostSize_32f_C1R	1924
7.106.2.38nppiNormDiffInfGetBufferHostSize_32f_C3CMR	1924
7.106.2.39nppiNormDiffInfGetBufferHostSize_32f_C3R	1924
7.106.2.40nppiNormDiffInfGetBufferHostSize_32f_C4R	1925
7.106.2.41nppiNormDiffInfGetBufferHostSize_8s_C1MR	1925
7.106.2.42nppiNormDiffInfGetBufferHostSize_8s_C3CMR	1925
7.106.2.43nppiNormDiffInfGetBufferHostSize_8u_AC4R	1925
7.106.2.44nppiNormDiffInfGetBufferHostSize_8u_C1MR	1926
7.106.2.45nppiNormDiffInfGetBufferHostSize_8u_C1R	1926
7.106.2.46nppiNormDiffInfGetBufferHostSize_8u_C3CMR	1926
7.106.2.47nppiNormDiffInfGetBufferHostSize_8u_C3R	1927
7.106.2.48nppiNormDiffInfGetBufferHostSize_8u_C4R	1927
7.107NormDiff_L1	1928

7.107.1 Detailed Description	1932
7.107.2 Function Documentation	1932
7.107.2.1 nppiNormDiff_L1_16s_AC4R	1932
7.107.2.2 nppiNormDiff_L1_16s_C1R	1932
7.107.2.3 nppiNormDiff_L1_16s_C3R	1933
7.107.2.4 nppiNormDiff_L1_16s_C4R	1933
7.107.2.5 nppiNormDiff_L1_16u_AC4R	1934
7.107.2.6 nppiNormDiff_L1_16u_C1MR	1934
7.107.2.7 nppiNormDiff_L1_16u_C1R	1935
7.107.2.8 nppiNormDiff_L1_16u_C3CMR	1935
7.107.2.9 nppiNormDiff_L1_16u_C3R	1936
7.107.2.10 nppiNormDiff_L1_16u_C4R	1936
7.107.2.11 nppiNormDiff_L1_32f_AC4R	1936
7.107.2.12 nppiNormDiff_L1_32f_C1MR	1937
7.107.2.13 nppiNormDiff_L1_32f_C1R	1937
7.107.2.14 nppiNormDiff_L1_32f_C3CMR	1938
7.107.2.15 nppiNormDiff_L1_32f_C3R	1938
7.107.2.16 nppiNormDiff_L1_32f_C4R	1939
7.107.2.17 nppiNormDiff_L1_8s_C1MR	1939
7.107.2.18 nppiNormDiff_L1_8s_C3CMR	1940
7.107.2.19 nppiNormDiff_L1_8u_AC4R	1940
7.107.2.20 nppiNormDiff_L1_8u_C1MR	1941
7.107.2.21 nppiNormDiff_L1_8u_C1R	1941
7.107.2.22 nppiNormDiff_L1_8u_C3CMR	1942
7.107.2.23 nppiNormDiff_L1_8u_C3R	1942
7.107.2.24 nppiNormDiff_L1_8u_C4R	1943
7.107.2.25 nppiNormDiffL1GetBufferHostSize_16s_AC4R	1943
7.107.2.26 nppiNormDiffL1GetBufferHostSize_16s_C1R	1943
7.107.2.27 nppiNormDiffL1GetBufferHostSize_16s_C3R	1944
7.107.2.28 nppiNormDiffL1GetBufferHostSize_16s_C4R	1944
7.107.2.29 nppiNormDiffL1GetBufferHostSize_16u_AC4R	1944
7.107.2.30 nppiNormDiffL1GetBufferHostSize_16u_C1MR	1944
7.107.2.31 nppiNormDiffL1GetBufferHostSize_16u_C1R	1945
7.107.2.32 nppiNormDiffL1GetBufferHostSize_16u_C3CMR	1945
7.107.2.33 nppiNormDiffL1GetBufferHostSize_16u_C3R	1945
7.107.2.34 nppiNormDiffL1GetBufferHostSize_16u_C4R	1946

7.107.2.35	nppiNormDiffL1GetBufferHostSize_32f_AC4R	1946
7.107.2.36	nppiNormDiffL1GetBufferHostSize_32f_C1MR	1946
7.107.2.37	nppiNormDiffL1GetBufferHostSize_32f_C1R	1946
7.107.2.38	nppiNormDiffL1GetBufferHostSize_32f_C3CMR	1947
7.107.2.39	nppiNormDiffL1GetBufferHostSize_32f_C3R	1947
7.107.2.40	nppiNormDiffL1GetBufferHostSize_32f_C4R	1947
7.107.2.41	nppiNormDiffL1GetBufferHostSize_8s_C1MR	1948
7.107.2.42	nppiNormDiffL1GetBufferHostSize_8s_C3CMR	1948
7.107.2.43	nppiNormDiffL1GetBufferHostSize_8u_AC4R	1948
7.107.2.44	nppiNormDiffL1GetBufferHostSize_8u_C1MR	1948
7.107.2.45	nppiNormDiffL1GetBufferHostSize_8u_C1R	1949
7.107.2.46	nppiNormDiffL1GetBufferHostSize_8u_C3CMR	1949
7.107.2.47	nppiNormDiffL1GetBufferHostSize_8u_C3R	1949
7.107.2.48	nppiNormDiffL1GetBufferHostSize_8u_C4R	1950
7.108	NormDiff_L2	1951
7.108.1	Detailed Description	1955
7.108.2	Function Documentation	1955
7.108.2.1	nppiNormDiff_L2_16s_AC4R	1955
7.108.2.2	nppiNormDiff_L2_16s_C1R	1955
7.108.2.3	nppiNormDiff_L2_16s_C3R	1956
7.108.2.4	nppiNormDiff_L2_16s_C4R	1956
7.108.2.5	nppiNormDiff_L2_16u_AC4R	1957
7.108.2.6	nppiNormDiff_L2_16u_C1MR	1957
7.108.2.7	nppiNormDiff_L2_16u_C1R	1958
7.108.2.8	nppiNormDiff_L2_16u_C3CMR	1958
7.108.2.9	nppiNormDiff_L2_16u_C3R	1959
7.108.2.10	nppiNormDiff_L2_16u_C4R	1959
7.108.2.11	nppiNormDiff_L2_32f_AC4R	1959
7.108.2.12	nppiNormDiff_L2_32f_C1MR	1960
7.108.2.13	nppiNormDiff_L2_32f_C1R	1960
7.108.2.14	nppiNormDiff_L2_32f_C3CMR	1961
7.108.2.15	nppiNormDiff_L2_32f_C3R	1961
7.108.2.16	nppiNormDiff_L2_32f_C4R	1962
7.108.2.17	nppiNormDiff_L2_8s_C1MR	1962
7.108.2.18	nppiNormDiff_L2_8s_C3CMR	1963
7.108.2.19	nppiNormDiff_L2_8u_AC4R	1963

7.108.2.20	<code>nppiNormDiff_L2_8u_C1MR</code>	1964
7.108.2.21	<code>nppiNormDiff_L2_8u_C1R</code>	1964
7.108.2.22	<code>nppiNormDiff_L2_8u_C3CMR</code>	1965
7.108.2.23	<code>nppiNormDiff_L2_8u_C3R</code>	1965
7.108.2.24	<code>nppiNormDiff_L2_8u_C4R</code>	1966
7.108.2.25	<code>nppiNormDiffL2GetBufferHostSize_16s_AC4R</code>	1966
7.108.2.26	<code>nppiNormDiffL2GetBufferHostSize_16s_C1R</code>	1966
7.108.2.27	<code>nppiNormDiffL2GetBufferHostSize_16s_C3R</code>	1967
7.108.2.28	<code>nppiNormDiffL2GetBufferHostSize_16s_C4R</code>	1967
7.108.2.29	<code>nppiNormDiffL2GetBufferHostSize_16u_AC4R</code>	1967
7.108.2.30	<code>nppiNormDiffL2GetBufferHostSize_16u_C1MR</code>	1967
7.108.2.31	<code>nppiNormDiffL2GetBufferHostSize_16u_C1R</code>	1968
7.108.2.32	<code>nppiNormDiffL2GetBufferHostSize_16u_C3CMR</code>	1968
7.108.2.33	<code>nppiNormDiffL2GetBufferHostSize_16u_C3R</code>	1968
7.108.2.34	<code>nppiNormDiffL2GetBufferHostSize_16u_C4R</code>	1969
7.108.2.35	<code>nppiNormDiffL2GetBufferHostSize_32f_AC4R</code>	1969
7.108.2.36	<code>nppiNormDiffL2GetBufferHostSize_32f_C1MR</code>	1969
7.108.2.37	<code>nppiNormDiffL2GetBufferHostSize_32f_C1R</code>	1969
7.108.2.38	<code>nppiNormDiffL2GetBufferHostSize_32f_C3CMR</code>	1970
7.108.2.39	<code>nppiNormDiffL2GetBufferHostSize_32f_C3R</code>	1970
7.108.2.40	<code>nppiNormDiffL2GetBufferHostSize_32f_C4R</code>	1970
7.108.2.41	<code>nppiNormDiffL2GetBufferHostSize_8s_C1MR</code>	1971
7.108.2.42	<code>nppiNormDiffL2GetBufferHostSize_8s_C3CMR</code>	1971
7.108.2.43	<code>nppiNormDiffL2GetBufferHostSize_8u_AC4R</code>	1971
7.108.2.44	<code>nppiNormDiffL2GetBufferHostSize_8u_C1MR</code>	1971
7.108.2.45	<code>nppiNormDiffL2GetBufferHostSize_8u_C1R</code>	1972
7.108.2.46	<code>nppiNormDiffL2GetBufferHostSize_8u_C3CMR</code>	1972
7.108.2.47	<code>nppiNormDiffL2GetBufferHostSize_8u_C3R</code>	1972
7.108.2.48	<code>nppiNormDiffL2GetBufferHostSize_8u_C4R</code>	1973
7.109	<code>NormRel_Inf</code>	1974
7.109.1	Detailed Description	1978
7.109.2	Function Documentation	1978
7.109.2.1	<code>nppiNormRel_Inf_16s_AC4R</code>	1978
7.109.2.2	<code>nppiNormRel_Inf_16s_C1R</code>	1978
7.109.2.3	<code>nppiNormRel_Inf_16s_C3R</code>	1979
7.109.2.4	<code>nppiNormRel_Inf_16s_C4R</code>	1979

7.109.2.5 nppiNormRel_Inf_16u_AC4R	1980
7.109.2.6 nppiNormRel_Inf_16u_C1MR	1980
7.109.2.7 nppiNormRel_Inf_16u_C1R	1981
7.109.2.8 nppiNormRel_Inf_16u_C3CMR	1981
7.109.2.9 nppiNormRel_Inf_16u_C3R	1982
7.109.2.10 nppiNormRel_Inf_16u_C4R	1982
7.109.2.11 nppiNormRel_Inf_32f_AC4R	1983
7.109.2.12 nppiNormRel_Inf_32f_C1MR	1983
7.109.2.13 nppiNormRel_Inf_32f_C1R	1984
7.109.2.14 nppiNormRel_Inf_32f_C3CMR	1984
7.109.2.15 nppiNormRel_Inf_32f_C3R	1985
7.109.2.16 nppiNormRel_Inf_32f_C4R	1985
7.109.2.17 nppiNormRel_Inf_8s_C1MR	1986
7.109.2.18 nppiNormRel_Inf_8s_C3CMR	1986
7.109.2.19 nppiNormRel_Inf_8u_AC4R	1987
7.109.2.20 nppiNormRel_Inf_8u_C1MR	1987
7.109.2.21 nppiNormRel_Inf_8u_C1R	1988
7.109.2.22 nppiNormRel_Inf_8u_C3CMR	1988
7.109.2.23 nppiNormRel_Inf_8u_C3R	1989
7.109.2.24 nppiNormRel_Inf_8u_C4R	1989
7.109.2.25 nppiNormRelInfGetBufferHostSize_16s_AC4R	1989
7.109.2.26 nppiNormRelInfGetBufferHostSize_16s_C1R	1990
7.109.2.27 nppiNormRelInfGetBufferHostSize_16s_C3R	1990
7.109.2.28 nppiNormRelInfGetBufferHostSize_16s_C4R	1990
7.109.2.29 nppiNormRelInfGetBufferHostSize_16u_AC4R	1991
7.109.2.30 nppiNormRelInfGetBufferHostSize_16u_C1MR	1991
7.109.2.31 nppiNormRelInfGetBufferHostSize_16u_C1R	1991
7.109.2.32 nppiNormRelInfGetBufferHostSize_16u_C3CMR	1991
7.109.2.33 nppiNormRelInfGetBufferHostSize_16u_C3R	1992
7.109.2.34 nppiNormRelInfGetBufferHostSize_16u_C4R	1992
7.109.2.35 nppiNormRelInfGetBufferHostSize_32f_AC4R	1992
7.109.2.36 nppiNormRelInfGetBufferHostSize_32f_C1MR	1993
7.109.2.37 nppiNormRelInfGetBufferHostSize_32f_C1R	1993
7.109.2.38 nppiNormRelInfGetBufferHostSize_32f_C3CMR	1993
7.109.2.39 nppiNormRelInfGetBufferHostSize_32f_C3R	1993
7.109.2.40 nppiNormRelInfGetBufferHostSize_32f_C4R	1994

7.109.2.4	nppiNormRelInfGetBufferHostSize_32s_C1R	1994
7.109.2.42	nppiNormRelInfGetBufferHostSize_8s_C1MR	1994
7.109.2.43	nppiNormRelInfGetBufferHostSize_8s_C3CMR	1995
7.109.2.44	nppiNormRelInfGetBufferHostSize_8u_AC4R	1995
7.109.2.45	nppiNormRelInfGetBufferHostSize_8u_C1MR	1995
7.109.2.46	nppiNormRelInfGetBufferHostSize_8u_C1R	1995
7.109.2.47	nppiNormRelInfGetBufferHostSize_8u_C3CMR	1996
7.109.2.48	nppiNormRelInfGetBufferHostSize_8u_C3R	1996
7.109.2.49	nppiNormRelInfGetBufferHostSize_8u_C4R	1996
7.110	NormRel_L1	1997
7.110.1	Detailed Description	2001
7.110.2	Function Documentation	2001
7.110.2.1	nppiNormRel_L1_16s_AC4R	2001
7.110.2.2	nppiNormRel_L1_16s_C1R	2001
7.110.2.3	nppiNormRel_L1_16s_C3R	2002
7.110.2.4	nppiNormRel_L1_16s_C4R	2002
7.110.2.5	nppiNormRel_L1_16u_AC4R	2003
7.110.2.6	nppiNormRel_L1_16u_C1MR	2003
7.110.2.7	nppiNormRel_L1_16u_C1R	2004
7.110.2.8	nppiNormRel_L1_16u_C3CMR	2004
7.110.2.9	nppiNormRel_L1_16u_C3R	2005
7.110.2.10	nppiNormRel_L1_16u_C4R	2005
7.110.2.11	nppiNormRel_L1_32f_AC4R	2005
7.110.2.12	nppiNormRel_L1_32f_C1MR	2006
7.110.2.13	nppiNormRel_L1_32f_C1R	2006
7.110.2.14	nppiNormRel_L1_32f_C3CMR	2007
7.110.2.15	nppiNormRel_L1_32f_C3R	2007
7.110.2.16	nppiNormRel_L1_32f_C4R	2008
7.110.2.17	nppiNormRel_L1_8s_C1MR	2008
7.110.2.18	nppiNormRel_L1_8s_C3CMR	2009
7.110.2.19	nppiNormRel_L1_8u_AC4R	2009
7.110.2.20	nppiNormRel_L1_8u_C1MR	2010
7.110.2.21	nppiNormRel_L1_8u_C1R	2010
7.110.2.22	nppiNormRel_L1_8u_C3CMR	2011
7.110.2.23	nppiNormRel_L1_8u_C3R	2011
7.110.2.24	nppiNormRel_L1_8u_C4R	2012

7.110.2.25	nppiNormRelL1GetBufferHostSize_16s_AC4R	2012
7.110.2.26	nppiNormRelL1GetBufferHostSize_16s_C1R	2013
7.110.2.27	nppiNormRelL1GetBufferHostSize_16s_C3R	2013
7.110.2.28	nppiNormRelL1GetBufferHostSize_16s_C4R	2013
7.110.2.29	nppiNormRelL1GetBufferHostSize_16u_AC4R	2013
7.110.2.30	nppiNormRelL1GetBufferHostSize_16u_C1MR	2014
7.110.2.31	nppiNormRelL1GetBufferHostSize_16u_C1R	2014
7.110.2.32	nppiNormRelL1GetBufferHostSize_16u_C3CMR	2014
7.110.2.33	nppiNormRelL1GetBufferHostSize_16u_C3R	2015
7.110.2.34	nppiNormRelL1GetBufferHostSize_16u_C4R	2015
7.110.2.35	nppiNormRelL1GetBufferHostSize_32f_AC4R	2015
7.110.2.36	nppiNormRelL1GetBufferHostSize_32f_C1MR	2015
7.110.2.37	nppiNormRelL1GetBufferHostSize_32f_C1R	2016
7.110.2.38	nppiNormRelL1GetBufferHostSize_32f_C3CMR	2016
7.110.2.39	nppiNormRelL1GetBufferHostSize_32f_C3R	2016
7.110.2.40	nppiNormRelL1GetBufferHostSize_32f_C4R	2017
7.110.2.41	nppiNormRelL1GetBufferHostSize_8s_C1MR	2017
7.110.2.42	nppiNormRelL1GetBufferHostSize_8s_C3CMR	2017
7.110.2.43	nppiNormRelL1GetBufferHostSize_8u_AC4R	2017
7.110.2.44	nppiNormRelL1GetBufferHostSize_8u_C1MR	2018
7.110.2.45	nppiNormRelL1GetBufferHostSize_8u_C1R	2018
7.110.2.46	nppiNormRelL1GetBufferHostSize_8u_C3CMR	2018
7.110.2.47	nppiNormRelL1GetBufferHostSize_8u_C3R	2019
7.110.2.48	nppiNormRelL1GetBufferHostSize_8u_C4R	2019
7.111	NormRel_L2	2020
7.111.1	Detailed Description	2024
7.111.2	Function Documentation	2024
7.111.2.1	nppiNormRel_L2_16s_AC4R	2024
7.111.2.2	nppiNormRel_L2_16s_C1R	2024
7.111.2.3	nppiNormRel_L2_16s_C3R	2025
7.111.2.4	nppiNormRel_L2_16s_C4R	2025
7.111.2.5	nppiNormRel_L2_16u_AC4R	2026
7.111.2.6	nppiNormRel_L2_16u_C1MR	2026
7.111.2.7	nppiNormRel_L2_16u_C1R	2027
7.111.2.8	nppiNormRel_L2_16u_C3CMR	2027
7.111.2.9	nppiNormRel_L2_16u_C3R	2028

7.111.2.10nppiNormRel_L2_16u_C4R	2028
7.111.2.11nppiNormRel_L2_32f_AC4R	2028
7.111.2.12nppiNormRel_L2_32f_C1MR	2029
7.111.2.13nppiNormRel_L2_32f_C1R	2029
7.111.2.14nppiNormRel_L2_32f_C3CMR	2030
7.111.2.15nppiNormRel_L2_32f_C3R	2030
7.111.2.16nppiNormRel_L2_32f_C4R	2031
7.111.2.17nppiNormRel_L2_8s_C1MR	2031
7.111.2.18nppiNormRel_L2_8s_C3CMR	2032
7.111.2.19nppiNormRel_L2_8u_AC4R	2032
7.111.2.20nppiNormRel_L2_8u_C1MR	2033
7.111.2.21nppiNormRel_L2_8u_C1R	2033
7.111.2.22nppiNormRel_L2_8u_C3CMR	2034
7.111.2.23nppiNormRel_L2_8u_C3R	2034
7.111.2.24nppiNormRel_L2_8u_C4R	2035
7.111.2.25nppiNormRelL2GetBufferHostSize_16s_AC4R	2035
7.111.2.26nppiNormRelL2GetBufferHostSize_16s_C1R	2036
7.111.2.27nppiNormRelL2GetBufferHostSize_16s_C3R	2036
7.111.2.28nppiNormRelL2GetBufferHostSize_16s_C4R	2036
7.111.2.29nppiNormRelL2GetBufferHostSize_16u_AC4R	2036
7.111.2.30nppiNormRelL2GetBufferHostSize_16u_C1MR	2037
7.111.2.31nppiNormRelL2GetBufferHostSize_16u_C1R	2037
7.111.2.32nppiNormRelL2GetBufferHostSize_16u_C3CMR	2037
7.111.2.33nppiNormRelL2GetBufferHostSize_16u_C3R	2038
7.111.2.34nppiNormRelL2GetBufferHostSize_16u_C4R	2038
7.111.2.35nppiNormRelL2GetBufferHostSize_32f_AC4R	2038
7.111.2.36nppiNormRelL2GetBufferHostSize_32f_C1MR	2038
7.111.2.37nppiNormRelL2GetBufferHostSize_32f_C1R	2039
7.111.2.38nppiNormRelL2GetBufferHostSize_32f_C3CMR	2039
7.111.2.39nppiNormRelL2GetBufferHostSize_32f_C3R	2039
7.111.2.40nppiNormRelL2GetBufferHostSize_32f_C4R	2040
7.111.2.41nppiNormRelL2GetBufferHostSize_8s_C1MR	2040
7.111.2.42nppiNormRelL2GetBufferHostSize_8s_C3CMR	2040
7.111.2.43nppiNormRelL2GetBufferHostSize_8u_AC4R	2040
7.111.2.44nppiNormRelL2GetBufferHostSize_8u_C1MR	2041
7.111.2.45nppiNormRelL2GetBufferHostSize_8u_C1R	2041

7.111.2.46	npapiNormRelL2GetBufferHostSize_8u_C3CMR	2041
7.111.2.47	npapiNormRelL2GetBufferHostSize_8u_C3R	2042
7.111.2.48	npapiNormRelL2GetBufferHostSize_8u_C4R	2042
7.112	DotProd	2043
7.112.1	Detailed Description	2047
7.112.2	Function Documentation	2047
7.112.2.1	npapiDotProd_16s64f_AC4R	2047
7.112.2.2	npapiDotProd_16s64f_C1R	2048
7.112.2.3	npapiDotProd_16s64f_C3R	2048
7.112.2.4	npapiDotProd_16s64f_C4R	2048
7.112.2.5	npapiDotProd_16u64f_AC4R	2049
7.112.2.6	npapiDotProd_16u64f_C1R	2049
7.112.2.7	npapiDotProd_16u64f_C3R	2050
7.112.2.8	npapiDotProd_16u64f_C4R	2050
7.112.2.9	npapiDotProd_32f64f_AC4R	2051
7.112.2.10	npapiDotProd_32f64f_C1R	2051
7.112.2.11	npapiDotProd_32f64f_C3R	2051
7.112.2.12	npapiDotProd_32f64f_C4R	2052
7.112.2.13	npapiDotProd_32s64f_AC4R	2052
7.112.2.14	npapiDotProd_32s64f_C1R	2053
7.112.2.15	npapiDotProd_32s64f_C3R	2053
7.112.2.16	npapiDotProd_32s64f_C4R	2054
7.112.2.17	npapiDotProd_32u64f_AC4R	2054
7.112.2.18	npapiDotProd_32u64f_C1R	2054
7.112.2.19	npapiDotProd_32u64f_C3R	2055
7.112.2.20	npapiDotProd_32u64f_C4R	2055
7.112.2.21	npapiDotProd_8s64f_AC4R	2056
7.112.2.22	npapiDotProd_8s64f_C1R	2056
7.112.2.23	npapiDotProd_8s64f_C3R	2057
7.112.2.24	npapiDotProd_8s64f_C4R	2057
7.112.2.25	npapiDotProd_8u64f_AC4R	2057
7.112.2.26	npapiDotProd_8u64f_C1R	2058
7.112.2.27	npapiDotProd_8u64f_C3R	2058
7.112.2.28	npapiDotProd_8u64f_C4R	2059
7.112.2.29	npapiDotProdGetBufferHostSize_16s64f_AC4R	2059
7.112.2.30	npapiDotProdGetBufferHostSize_16s64f_C1R	2059

7.112.2.3	nppiDotProdGetBufferHostSize_16s64f_C3R	2060
7.112.2.32	nppiDotProdGetBufferHostSize_16s64f_C4R	2060
7.112.2.33	nppiDotProdGetBufferHostSize_16u64f_AC4R	2060
7.112.2.34	nppiDotProdGetBufferHostSize_16u64f_C1R	2061
7.112.2.35	nppiDotProdGetBufferHostSize_16u64f_C3R	2061
7.112.2.36	nppiDotProdGetBufferHostSize_16u64f_C4R	2061
7.112.2.37	nppiDotProdGetBufferHostSize_32f64f_AC4R	2061
7.112.2.38	nppiDotProdGetBufferHostSize_32f64f_C1R	2062
7.112.2.39	nppiDotProdGetBufferHostSize_32f64f_C3R	2062
7.112.2.40	nppiDotProdGetBufferHostSize_32f64f_C4R	2062
7.112.2.41	nppiDotProdGetBufferHostSize_32s64f_AC4R	2063
7.112.2.42	nppiDotProdGetBufferHostSize_32s64f_C1R	2063
7.112.2.43	nppiDotProdGetBufferHostSize_32s64f_C3R	2063
7.112.2.44	nppiDotProdGetBufferHostSize_32s64f_C4R	2063
7.112.2.45	nppiDotProdGetBufferHostSize_32u64f_AC4R	2064
7.112.2.46	nppiDotProdGetBufferHostSize_32u64f_C1R	2064
7.112.2.47	nppiDotProdGetBufferHostSize_32u64f_C3R	2064
7.112.2.48	nppiDotProdGetBufferHostSize_32u64f_C4R	2065
7.112.2.49	nppiDotProdGetBufferHostSize_8s64f_AC4R	2065
7.112.2.50	nppiDotProdGetBufferHostSize_8s64f_C1R	2065
7.112.2.51	nppiDotProdGetBufferHostSize_8s64f_C3R	2065
7.112.2.52	nppiDotProdGetBufferHostSize_8s64f_C4R	2066
7.112.2.53	nppiDotProdGetBufferHostSize_8u64f_AC4R	2066
7.112.2.54	nppiDotProdGetBufferHostSize_8u64f_C1R	2066
7.112.2.55	nppiDotProdGetBufferHostSize_8u64f_C3R	2067
7.112.2.56	nppiDotProdGetBufferHostSize_8u64f_C4R	2067
7.113	CountInRange	2068
7.113.1	Detailed Description	2069
7.113.2	Function Documentation	2069
7.113.2.1	nppiCountInRange_32f_AC4R	2069
7.113.2.2	nppiCountInRange_32f_C1R	2069
7.113.2.3	nppiCountInRange_32f_C3R	2070
7.113.2.4	nppiCountInRange_8u_AC4R	2070
7.113.2.5	nppiCountInRange_8u_C1R	2071
7.113.2.6	nppiCountInRange_8u_C3R	2071
7.113.2.7	nppiCountInRangeGetBufferHostSize_32f_AC4R	2072

7.113.2.8 nppiCountInRangeGetBufferHostSize_32f_C1R	2072
7.113.2.9 nppiCountInRangeGetBufferHostSize_32f_C3R	2072
7.113.2.10 nppiCountInRangeGetBufferHostSize_8u_AC4R	2073
7.113.2.11 nppiCountInRangeGetBufferHostSize_8u_C1R	2073
7.113.2.12 nppiCountInRangeGetBufferHostSize_8u_C3R	2073
7.114 MaxEvery	2074
7.114.1 Detailed Description	2075
7.114.2 Function Documentation	2075
7.114.2.1 nppiMaxEvery_16s_AC4IR	2075
7.114.2.2 nppiMaxEvery_16s_C1IR	2076
7.114.2.3 nppiMaxEvery_16s_C3IR	2076
7.114.2.4 nppiMaxEvery_16s_C4IR	2076
7.114.2.5 nppiMaxEvery_16u_AC4IR	2077
7.114.2.6 nppiMaxEvery_16u_C1IR	2077
7.114.2.7 nppiMaxEvery_16u_C3IR	2077
7.114.2.8 nppiMaxEvery_16u_C4IR	2078
7.114.2.9 nppiMaxEvery_32f_AC4IR	2078
7.114.2.10 nppiMaxEvery_32f_C1IR	2078
7.114.2.11 nppiMaxEvery_32f_C3IR	2079
7.114.2.12 nppiMaxEvery_32f_C4IR	2079
7.114.2.13 nppiMaxEvery_8u_AC4IR	2079
7.114.2.14 nppiMaxEvery_8u_C1IR	2080
7.114.2.15 nppiMaxEvery_8u_C3IR	2080
7.114.2.16 nppiMaxEvery_8u_C4IR	2080
7.115 MinEvery	2081
7.115.1 Detailed Description	2082
7.115.2 Function Documentation	2082
7.115.2.1 nppiMinEvery_16s_AC4IR	2082
7.115.2.2 nppiMinEvery_16s_C1IR	2083
7.115.2.3 nppiMinEvery_16s_C3IR	2083
7.115.2.4 nppiMinEvery_16s_C4IR	2083
7.115.2.5 nppiMinEvery_16u_AC4IR	2084
7.115.2.6 nppiMinEvery_16u_C1IR	2084
7.115.2.7 nppiMinEvery_16u_C3IR	2084
7.115.2.8 nppiMinEvery_16u_C4IR	2085
7.115.2.9 nppiMinEvery_32f_AC4IR	2085

7.115.2.10	nppiMinEvery_32f_C1IR	2085
7.115.2.11	nppiMinEvery_32f_C3IR	2086
7.115.2.12	nppiMinEvery_32f_C4IR	2086
7.115.2.13	nppiMinEvery_8u_AC4IR	2086
7.115.2.14	nppiMinEvery_8u_C1IR	2087
7.115.2.15	nppiMinEvery_8u_C3IR	2087
7.115.2.16	nppiMinEvery_8u_C4IR	2087
7.116	Integral	2088
7.116.1	Detailed Description	2088
7.116.2	Function Documentation	2088
7.116.2.1	nppiIntegral_8u32f_C1R	2088
7.116.2.2	nppiIntegral_8u32s_C1R	2089
7.117	SqrIntegral	2090
7.117.1	Detailed Description	2090
7.117.2	Function Documentation	2090
7.117.2.1	nppiSqrIntegral_8u32f64f_C1R	2090
7.117.2.2	nppiSqrIntegral_8u32s64f_C1R	2091
7.117.2.3	nppiSqrIntegral_8u32s_C1R	2091
7.118	RectStdDev	2093
7.118.1	Detailed Description	2093
7.118.2	Function Documentation	2093
7.118.2.1	nppiRectStdDev_32f_C1R	2093
7.118.2.2	nppiRectStdDev_32s32f_C1R	2094
7.118.2.3	nppiRectStdDev_32s_C1RSfs	2094
7.119	HistogramEven	2096
7.119.1	Detailed Description	2098
7.119.2	Function Documentation	2098
7.119.2.1	nppiEvenLevelsHost_32s	2098
7.119.2.2	nppiHistogramEven_16s_AC4R	2099
7.119.2.3	nppiHistogramEven_16s_C1R	2099
7.119.2.4	nppiHistogramEven_16s_C3R	2100
7.119.2.5	nppiHistogramEven_16s_C4R	2100
7.119.2.6	nppiHistogramEven_16u_AC4R	2101
7.119.2.7	nppiHistogramEven_16u_C1R	2101
7.119.2.8	nppiHistogramEven_16u_C3R	2102
7.119.2.9	nppiHistogramEven_16u_C4R	2102

7.119.2.10	nppiHistogramEven_8u_AC4R	2103
7.119.2.11	nppiHistogramEven_8u_C1R	2103
7.119.2.12	nppiHistogramEven_8u_C3R	2103
7.119.2.13	nppiHistogramEven_8u_C4R	2104
7.119.2.14	nppiHistogramEvenGetBufferSize_16s_AC4R	2104
7.119.2.15	nppiHistogramEvenGetBufferSize_16s_C1R	2105
7.119.2.16	nppiHistogramEvenGetBufferSize_16s_C3R	2105
7.119.2.17	nppiHistogramEvenGetBufferSize_16s_C4R	2105
7.119.2.18	nppiHistogramEvenGetBufferSize_16u_AC4R	2106
7.119.2.19	nppiHistogramEvenGetBufferSize_16u_C1R	2106
7.119.2.20	nppiHistogramEvenGetBufferSize_16u_C3R	2106
7.119.2.21	nppiHistogramEvenGetBufferSize_16u_C4R	2107
7.119.2.22	nppiHistogramEvenGetBufferSize_8u_AC4R	2107
7.119.2.23	nppiHistogramEvenGetBufferSize_8u_C1R	2107
7.119.2.24	nppiHistogramEvenGetBufferSize_8u_C3R	2108
7.119.2.25	nppiHistogramEvenGetBufferSize_8u_C4R	2108
7.120	HistogramRange	2109
7.120.1	Detailed Description	2111
7.120.2	Function Documentation	2112
7.120.2.1	nppiHistogramRange_16s_AC4R	2112
7.120.2.2	nppiHistogramRange_16s_C1R	2112
7.120.2.3	nppiHistogramRange_16s_C3R	2113
7.120.2.4	nppiHistogramRange_16s_C4R	2113
7.120.2.5	nppiHistogramRange_16u_AC4R	2113
7.120.2.6	nppiHistogramRange_16u_C1R	2114
7.120.2.7	nppiHistogramRange_16u_C3R	2114
7.120.2.8	nppiHistogramRange_16u_C4R	2115
7.120.2.9	nppiHistogramRange_32f_AC4R	2115
7.120.2.10	nppiHistogramRange_32f_C1R	2116
7.120.2.11	nppiHistogramRange_32f_C3R	2116
7.120.2.12	nppiHistogramRange_32f_C4R	2117
7.120.2.13	nppiHistogramRange_8u_AC4R	2117
7.120.2.14	nppiHistogramRange_8u_C1R	2118
7.120.2.15	nppiHistogramRange_8u_C3R	2118
7.120.2.16	nppiHistogramRange_8u_C4R	2118
7.120.2.17	nppiHistogramRangeGetBufferSize_16s_AC4R	2119

7.120.2.18	<code>ippiHistogramRangeGetBufferSize_16s_C1R</code>	2119
7.120.2.19	<code>ippiHistogramRangeGetBufferSize_16s_C3R</code>	2120
7.120.2.20	<code>ippiHistogramRangeGetBufferSize_16s_C4R</code>	2120
7.120.2.21	<code>ippiHistogramRangeGetBufferSize_16u_AC4R</code>	2120
7.120.2.22	<code>ippiHistogramRangeGetBufferSize_16u_C1R</code>	2121
7.120.2.23	<code>ippiHistogramRangeGetBufferSize_16u_C3R</code>	2121
7.120.2.24	<code>ippiHistogramRangeGetBufferSize_16u_C4R</code>	2121
7.120.2.25	<code>ippiHistogramRangeGetBufferSize_32f_AC4R</code>	2122
7.120.2.26	<code>ippiHistogramRangeGetBufferSize_32f_C1R</code>	2122
7.120.2.27	<code>ippiHistogramRangeGetBufferSize_32f_C3R</code>	2122
7.120.2.28	<code>ippiHistogramRangeGetBufferSize_32f_C4R</code>	2123
7.120.2.29	<code>ippiHistogramRangeGetBufferSize_8u_AC4R</code>	2123
7.120.2.30	<code>ippiHistogramRangeGetBufferSize_8u_C1R</code>	2123
7.120.2.31	<code>ippiHistogramRangeGetBufferSize_8u_C3R</code>	2124
7.120.2.32	<code>ippiHistogramRangeGetBufferSize_8u_C4R</code>	2124
7.121	Image Proximity	2125
7.121.1	Detailed Description	2125
7.121.2	General Introduction	2125
7.121.3	Categorizations	2127
7.122	<code>SqrDistanceFull_Norm</code>	2128
7.122.1	Detailed Description	2129
7.122.2	Function Documentation	2130
7.122.2.1	<code>ippiSqrDistanceFull_Norm_16u32f_AC4R</code>	2130
7.122.2.2	<code>ippiSqrDistanceFull_Norm_16u32f_C1R</code>	2130
7.122.2.3	<code>ippiSqrDistanceFull_Norm_16u32f_C3R</code>	2131
7.122.2.4	<code>ippiSqrDistanceFull_Norm_16u32f_C4R</code>	2131
7.122.2.5	<code>ippiSqrDistanceFull_Norm_32f_AC4R</code>	2131
7.122.2.6	<code>ippiSqrDistanceFull_Norm_32f_C1R</code>	2132
7.122.2.7	<code>ippiSqrDistanceFull_Norm_32f_C3R</code>	2132
7.122.2.8	<code>ippiSqrDistanceFull_Norm_32f_C4R</code>	2133
7.122.2.9	<code>ippiSqrDistanceFull_Norm_8s32f_AC4R</code>	2133
7.122.2.10	<code>ippiSqrDistanceFull_Norm_8s32f_C1R</code>	2134
7.122.2.11	<code>ippiSqrDistanceFull_Norm_8s32f_C3R</code>	2134
7.122.2.12	<code>ippiSqrDistanceFull_Norm_8s32f_C4R</code>	2134
7.122.2.13	<code>ippiSqrDistanceFull_Norm_8u32f_AC4R</code>	2135
7.122.2.14	<code>ippiSqrDistanceFull_Norm_8u32f_C1R</code>	2135

7.122.2.15	nppiSqrDistanceFull_Norm_8u32f_C3R	2136
7.122.2.16	nppiSqrDistanceFull_Norm_8u32f_C4R	2136
7.122.2.17	nppiSqrDistanceFull_Norm_8u_AC4RSfs	2137
7.122.2.18	nppiSqrDistanceFull_Norm_8u_C1RSfs	2137
7.122.2.19	nppiSqrDistanceFull_Norm_8u_C3RSfs	2138
7.122.2.20	nppiSqrDistanceFull_Norm_8u_C4RSfs	2138
7.123	SqrDistanceSame_Norm	2139
7.123.1	Detailed Description	2141
7.123.2	Function Documentation	2141
7.123.2.1	nppiSqrDistanceSame_Norm_16u32f_AC4R	2141
7.123.2.2	nppiSqrDistanceSame_Norm_16u32f_C1R	2141
7.123.2.3	nppiSqrDistanceSame_Norm_16u32f_C3R	2142
7.123.2.4	nppiSqrDistanceSame_Norm_16u32f_C4R	2142
7.123.2.5	nppiSqrDistanceSame_Norm_32f_AC4R	2143
7.123.2.6	nppiSqrDistanceSame_Norm_32f_C1R	2143
7.123.2.7	nppiSqrDistanceSame_Norm_32f_C3R	2143
7.123.2.8	nppiSqrDistanceSame_Norm_32f_C4R	2144
7.123.2.9	nppiSqrDistanceSame_Norm_8s32f_AC4R	2144
7.123.2.10	nppiSqrDistanceSame_Norm_8s32f_C1R	2145
7.123.2.11	nppiSqrDistanceSame_Norm_8s32f_C3R	2145
7.123.2.12	nppiSqrDistanceSame_Norm_8s32f_C4R	2146
7.123.2.13	nppiSqrDistanceSame_Norm_8u32f_AC4R	2146
7.123.2.14	nppiSqrDistanceSame_Norm_8u32f_C1R	2146
7.123.2.15	nppiSqrDistanceSame_Norm_8u32f_C3R	2147
7.123.2.16	nppiSqrDistanceSame_Norm_8u32f_C4R	2147
7.123.2.17	nppiSqrDistanceSame_Norm_8u_AC4RSfs	2148
7.123.2.18	nppiSqrDistanceSame_Norm_8u_C1RSfs	2148
7.123.2.19	nppiSqrDistanceSame_Norm_8u_C3RSfs	2149
7.123.2.20	nppiSqrDistanceSame_Norm_8u_C4RSfs	2149
7.124	SqrDistanceValid_Norm	2150
7.124.1	Detailed Description	2152
7.124.2	Function Documentation	2152
7.124.2.1	nppiSqrDistanceValid_Norm_16u32f_AC4R	2152
7.124.2.2	nppiSqrDistanceValid_Norm_16u32f_C1R	2152
7.124.2.3	nppiSqrDistanceValid_Norm_16u32f_C3R	2153
7.124.2.4	nppiSqrDistanceValid_Norm_16u32f_C4R	2153

7.124.2.5 nppiSqrDistanceValid_Norm_32f_AC4R	2154
7.124.2.6 nppiSqrDistanceValid_Norm_32f_C1R	2154
7.124.2.7 nppiSqrDistanceValid_Norm_32f_C3R	2154
7.124.2.8 nppiSqrDistanceValid_Norm_32f_C4R	2155
7.124.2.9 nppiSqrDistanceValid_Norm_8s32f_AC4R	2155
7.124.2.10 nppiSqrDistanceValid_Norm_8s32f_C1R	2156
7.124.2.11 nppiSqrDistanceValid_Norm_8s32f_C3R	2156
7.124.2.12 nppiSqrDistanceValid_Norm_8s32f_C4R	2157
7.124.2.13 nppiSqrDistanceValid_Norm_8u32f_AC4R	2157
7.124.2.14 nppiSqrDistanceValid_Norm_8u32f_C1R	2157
7.124.2.15 nppiSqrDistanceValid_Norm_8u32f_C3R	2158
7.124.2.16 nppiSqrDistanceValid_Norm_8u32f_C4R	2158
7.124.2.17 nppiSqrDistanceValid_Norm_8u_AC4RSfs	2159
7.124.2.18 nppiSqrDistanceValid_Norm_8u_C1RSfs	2159
7.124.2.19 nppiSqrDistanceValid_Norm_8u_C3RSfs	2160
7.124.2.20 nppiSqrDistanceValid_Norm_8u_C4RSfs	2160
7.125 CrossCorrFull_Norm	2161
7.125.1 Detailed Description	2162
7.125.2 Function Documentation	2163
7.125.2.1 nppiCrossCorrFull_Norm_16u32f_AC4R	2163
7.125.2.2 nppiCrossCorrFull_Norm_16u32f_C1R	2163
7.125.2.3 nppiCrossCorrFull_Norm_16u32f_C3R	2164
7.125.2.4 nppiCrossCorrFull_Norm_16u32f_C4R	2164
7.125.2.5 nppiCrossCorrFull_Norm_32f_AC4R	2164
7.125.2.6 nppiCrossCorrFull_Norm_32f_C1R	2165
7.125.2.7 nppiCrossCorrFull_Norm_32f_C3R	2165
7.125.2.8 nppiCrossCorrFull_Norm_32f_C4R	2166
7.125.2.9 nppiCrossCorrFull_Norm_8s32f_AC4R	2166
7.125.2.10 nppiCrossCorrFull_Norm_8s32f_C1R	2167
7.125.2.11 nppiCrossCorrFull_Norm_8s32f_C3R	2167
7.125.2.12 nppiCrossCorrFull_Norm_8s32f_C4R	2167
7.125.2.13 nppiCrossCorrFull_Norm_8u32f_AC4R	2168
7.125.2.14 nppiCrossCorrFull_Norm_8u32f_C1R	2168
7.125.2.15 nppiCrossCorrFull_Norm_8u32f_C3R	2169
7.125.2.16 nppiCrossCorrFull_Norm_8u32f_C4R	2169
7.125.2.17 nppiCrossCorrFull_Norm_8u_AC4RSfs	2170

7.125.2.18	nppiCrossCorrFull_Norm_8u_C1RSfs	2170
7.125.2.19	nppiCrossCorrFull_Norm_8u_C3RSfs	2171
7.125.2.20	nppiCrossCorrFull_Norm_8u_C4RSfs	2171
7.126	CrossCorrSame_Norm	2172
7.126.1	Detailed Description	2173
7.126.2	Function Documentation	2174
7.126.2.1	nppiCrossCorrSame_Norm_16u32f_AC4R	2174
7.126.2.2	nppiCrossCorrSame_Norm_16u32f_C1R	2174
7.126.2.3	nppiCrossCorrSame_Norm_16u32f_C3R	2175
7.126.2.4	nppiCrossCorrSame_Norm_16u32f_C4R	2175
7.126.2.5	nppiCrossCorrSame_Norm_32f_AC4R	2175
7.126.2.6	nppiCrossCorrSame_Norm_32f_C1R	2176
7.126.2.7	nppiCrossCorrSame_Norm_32f_C3R	2176
7.126.2.8	nppiCrossCorrSame_Norm_32f_C4R	2177
7.126.2.9	nppiCrossCorrSame_Norm_8s32f_AC4R	2177
7.126.2.10	nppiCrossCorrSame_Norm_8s32f_C1R	2178
7.126.2.11	nppiCrossCorrSame_Norm_8s32f_C3R	2178
7.126.2.12	nppiCrossCorrSame_Norm_8s32f_C4R	2178
7.126.2.13	nppiCrossCorrSame_Norm_8u32f_AC4R	2179
7.126.2.14	nppiCrossCorrSame_Norm_8u32f_C1R	2179
7.126.2.15	nppiCrossCorrSame_Norm_8u32f_C3R	2180
7.126.2.16	nppiCrossCorrSame_Norm_8u32f_C4R	2180
7.126.2.17	nppiCrossCorrSame_Norm_8u_AC4RSfs	2181
7.126.2.18	nppiCrossCorrSame_Norm_8u_C1RSfs	2181
7.126.2.19	nppiCrossCorrSame_Norm_8u_C3RSfs	2182
7.126.2.20	nppiCrossCorrSame_Norm_8u_C4RSfs	2182
7.127	CrossCorrValid_Norm	2183
7.127.1	Detailed Description	2184
7.127.2	Function Documentation	2185
7.127.2.1	nppiCrossCorrValid_Norm_16u32f_AC4R	2185
7.127.2.2	nppiCrossCorrValid_Norm_16u32f_C1R	2185
7.127.2.3	nppiCrossCorrValid_Norm_16u32f_C3R	2186
7.127.2.4	nppiCrossCorrValid_Norm_16u32f_C4R	2186
7.127.2.5	nppiCrossCorrValid_Norm_32f_AC4R	2186
7.127.2.6	nppiCrossCorrValid_Norm_32f_C1R	2187
7.127.2.7	nppiCrossCorrValid_Norm_32f_C3R	2187

7.127.2.8	nppiCrossCorrValid_Norm_32f_C4R	2188
7.127.2.9	nppiCrossCorrValid_Norm_8s32f_AC4R	2188
7.127.2.10	nppiCrossCorrValid_Norm_8s32f_C1R	2189
7.127.2.11	nppiCrossCorrValid_Norm_8s32f_C3R	2189
7.127.2.12	nppiCrossCorrValid_Norm_8s32f_C4R	2189
7.127.2.13	nppiCrossCorrValid_Norm_8u32f_AC4R	2190
7.127.2.14	nppiCrossCorrValid_Norm_8u32f_C1R	2190
7.127.2.15	nppiCrossCorrValid_Norm_8u32f_C3R	2191
7.127.2.16	nppiCrossCorrValid_Norm_8u32f_C4R	2191
7.127.2.17	nppiCrossCorrValid_Norm_8u_AC4RSfs	2192
7.127.2.18	nppiCrossCorrValid_Norm_8u_C1RSfs	2192
7.127.2.19	nppiCrossCorrValid_Norm_8u_C3RSfs	2193
7.127.2.20	nppiCrossCorrValid_Norm_8u_C4RSfs	2193
7.128	CrossCorrValid	2194
7.128.1	Detailed Description	2194
7.128.2	Function Documentation	2194
7.128.2.1	nppiCrossCorrValid_16u32f_C1R	2194
7.128.2.2	nppiCrossCorrValid_32f_C1R	2195
7.128.2.3	nppiCrossCorrValid_8s32f_C1R	2195
7.128.2.4	nppiCrossCorrValid_8u32f_C1R	2196
7.129	CrossCorrFull_NormLevel	2197
7.129.1	Detailed Description	2200
7.129.2	Function Documentation	2201
7.129.2.1	nppiCrossCorrFull_NormLevel_16u32f_AC4R	2201
7.129.2.2	nppiCrossCorrFull_NormLevel_16u32f_C1R	2201
7.129.2.3	nppiCrossCorrFull_NormLevel_16u32f_C3R	2202
7.129.2.4	nppiCrossCorrFull_NormLevel_16u32f_C4R	2202
7.129.2.5	nppiCrossCorrFull_NormLevel_32f_AC4R	2203
7.129.2.6	nppiCrossCorrFull_NormLevel_32f_C1R	2203
7.129.2.7	nppiCrossCorrFull_NormLevel_32f_C3R	2204
7.129.2.8	nppiCrossCorrFull_NormLevel_32f_C4R	2204
7.129.2.9	nppiCrossCorrFull_NormLevel_8s32f_AC4R	2205
7.129.2.10	nppiCrossCorrFull_NormLevel_8s32f_C1R	2205
7.129.2.11	nppiCrossCorrFull_NormLevel_8s32f_C3R	2206
7.129.2.12	nppiCrossCorrFull_NormLevel_8s32f_C4R	2206
7.129.2.13	nppiCrossCorrFull_NormLevel_8u32f_AC4R	2207

7.129.2.14	nppiCrossCorrFull_NormLevel_8u32f_C1R	2207
7.129.2.15	nppiCrossCorrFull_NormLevel_8u32f_C3R	2208
7.129.2.16	nppiCrossCorrFull_NormLevel_8u32f_C4R	2208
7.129.2.17	nppiCrossCorrFull_NormLevel_8u_AC4RSfs	2209
7.129.2.18	nppiCrossCorrFull_NormLevel_8u_C1RSfs	2209
7.129.2.19	nppiCrossCorrFull_NormLevel_8u_C3RSfs	2210
7.129.2.20	nppiCrossCorrFull_NormLevel_8u_C4RSfs	2210
7.129.2.21	nppiFullNormLevelGetBufferHostSize_16u32f_AC4R	2211
7.129.2.22	nppiFullNormLevelGetBufferHostSize_16u32f_C1R	2211
7.129.2.23	nppiFullNormLevelGetBufferHostSize_16u32f_C3R	2211
7.129.2.24	nppiFullNormLevelGetBufferHostSize_16u32f_C4R	2211
7.129.2.25	nppiFullNormLevelGetBufferHostSize_32f_AC4R	2212
7.129.2.26	nppiFullNormLevelGetBufferHostSize_32f_C1R	2212
7.129.2.27	nppiFullNormLevelGetBufferHostSize_32f_C3R	2212
7.129.2.28	nppiFullNormLevelGetBufferHostSize_32f_C4R	2213
7.129.2.29	nppiFullNormLevelGetBufferHostSize_8s32f_AC4R	2213
7.129.2.30	nppiFullNormLevelGetBufferHostSize_8s32f_C1R	2213
7.129.2.31	nppiFullNormLevelGetBufferHostSize_8s32f_C3R	2213
7.129.2.32	nppiFullNormLevelGetBufferHostSize_8s32f_C4R	2214
7.129.2.33	nppiFullNormLevelGetBufferHostSize_8u32f_AC4R	2214
7.129.2.34	nppiFullNormLevelGetBufferHostSize_8u32f_C1R	2214
7.129.2.35	nppiFullNormLevelGetBufferHostSize_8u32f_C3R	2215
7.129.2.36	nppiFullNormLevelGetBufferHostSize_8u32f_C4R	2215
7.129.2.37	nppiFullNormLevelGetBufferHostSize_8u_AC4RSfs	2215
7.129.2.38	nppiFullNormLevelGetBufferHostSize_8u_C1RSfs	2215
7.129.2.39	nppiFullNormLevelGetBufferHostSize_8u_C3RSfs	2216
7.129.2.40	nppiFullNormLevelGetBufferHostSize_8u_C4RSfs	2216
7.130	CrossCorrSame_NormLevel	2217
7.130.1	Detailed Description	2220
7.130.2	Function Documentation	2221
7.130.2.1	nppiCrossCorrSame_NormLevel_16u32f_AC4R	2221
7.130.2.2	nppiCrossCorrSame_NormLevel_16u32f_C1R	2221
7.130.2.3	nppiCrossCorrSame_NormLevel_16u32f_C3R	2222
7.130.2.4	nppiCrossCorrSame_NormLevel_16u32f_C4R	2222
7.130.2.5	nppiCrossCorrSame_NormLevel_32f_AC4R	2223
7.130.2.6	nppiCrossCorrSame_NormLevel_32f_C1R	2223

7.130.2.7 nppiCrossCorrSame_NormLevel_32f_C3R	2224
7.130.2.8 nppiCrossCorrSame_NormLevel_32f_C4R	2224
7.130.2.9 nppiCrossCorrSame_NormLevel_8s32f_AC4R	2225
7.130.2.10 nppiCrossCorrSame_NormLevel_8s32f_C1R	2225
7.130.2.11 nppiCrossCorrSame_NormLevel_8s32f_C3R	2226
7.130.2.12 nppiCrossCorrSame_NormLevel_8s32f_C4R	2226
7.130.2.13 nppiCrossCorrSame_NormLevel_8u32f_AC4R	2227
7.130.2.14 nppiCrossCorrSame_NormLevel_8u32f_C1R	2227
7.130.2.15 nppiCrossCorrSame_NormLevel_8u32f_C3R	2228
7.130.2.16 nppiCrossCorrSame_NormLevel_8u32f_C4R	2228
7.130.2.17 nppiCrossCorrSame_NormLevel_8u_AC4RSfs	2229
7.130.2.18 nppiCrossCorrSame_NormLevel_8u_C1RSfs	2229
7.130.2.19 nppiCrossCorrSame_NormLevel_8u_C3RSfs	2230
7.130.2.20 nppiCrossCorrSame_NormLevel_8u_C4RSfs	2230
7.130.2.21 nppiSameNormLevelGetBufferHostSize_16u32f_AC4R	2231
7.130.2.22 nppiSameNormLevelGetBufferHostSize_16u32f_C1R	2231
7.130.2.23 nppiSameNormLevelGetBufferHostSize_16u32f_C3R	2231
7.130.2.24 nppiSameNormLevelGetBufferHostSize_16u32f_C4R	2231
7.130.2.25 nppiSameNormLevelGetBufferHostSize_32f_AC4R	2232
7.130.2.26 nppiSameNormLevelGetBufferHostSize_32f_C1R	2232
7.130.2.27 nppiSameNormLevelGetBufferHostSize_32f_C3R	2232
7.130.2.28 nppiSameNormLevelGetBufferHostSize_32f_C4R	2233
7.130.2.29 nppiSameNormLevelGetBufferHostSize_8s32f_AC4R	2233
7.130.2.30 nppiSameNormLevelGetBufferHostSize_8s32f_C1R	2233
7.130.2.31 nppiSameNormLevelGetBufferHostSize_8s32f_C3R	2233
7.130.2.32 nppiSameNormLevelGetBufferHostSize_8s32f_C4R	2234
7.130.2.33 nppiSameNormLevelGetBufferHostSize_8u32f_AC4R	2234
7.130.2.34 nppiSameNormLevelGetBufferHostSize_8u32f_C1R	2234
7.130.2.35 nppiSameNormLevelGetBufferHostSize_8u32f_C3R	2235
7.130.2.36 nppiSameNormLevelGetBufferHostSize_8u32f_C4R	2235
7.130.2.37 nppiSameNormLevelGetBufferHostSize_8u_AC4RSfs	2235
7.130.2.38 nppiSameNormLevelGetBufferHostSize_8u_C1RSfs	2235
7.130.2.39 nppiSameNormLevelGetBufferHostSize_8u_C3RSfs	2236
7.130.2.40 nppiSameNormLevelGetBufferHostSize_8u_C4RSfs	2236
7.131 CrossCorrValid_NormLevel	2237
7.131.1 Detailed Description	2240

7.131.2 Function Documentation	2241
7.131.2.1 nppiCrossCorrValid_NormLevel_16u32f_AC4R	2241
7.131.2.2 nppiCrossCorrValid_NormLevel_16u32f_C1R	2241
7.131.2.3 nppiCrossCorrValid_NormLevel_16u32f_C3R	2242
7.131.2.4 nppiCrossCorrValid_NormLevel_16u32f_C4R	2242
7.131.2.5 nppiCrossCorrValid_NormLevel_32f_AC4R	2243
7.131.2.6 nppiCrossCorrValid_NormLevel_32f_C1R	2243
7.131.2.7 nppiCrossCorrValid_NormLevel_32f_C3R	2244
7.131.2.8 nppiCrossCorrValid_NormLevel_32f_C4R	2244
7.131.2.9 nppiCrossCorrValid_NormLevel_8s32f_AC4R	2245
7.131.2.10 nppiCrossCorrValid_NormLevel_8s32f_C1R	2245
7.131.2.11 nppiCrossCorrValid_NormLevel_8s32f_C3R	2246
7.131.2.12 nppiCrossCorrValid_NormLevel_8s32f_C4R	2246
7.131.2.13 nppiCrossCorrValid_NormLevel_8u32f_AC4R	2247
7.131.2.14 nppiCrossCorrValid_NormLevel_8u32f_C1R	2247
7.131.2.15 nppiCrossCorrValid_NormLevel_8u32f_C3R	2248
7.131.2.16 nppiCrossCorrValid_NormLevel_8u32f_C4R	2248
7.131.2.17 nppiCrossCorrValid_NormLevel_8u_AC4RSfs	2249
7.131.2.18 nppiCrossCorrValid_NormLevel_8u_C1RSfs	2249
7.131.2.19 nppiCrossCorrValid_NormLevel_8u_C3RSfs	2250
7.131.2.20 nppiCrossCorrValid_NormLevel_8u_C4RSfs	2250
7.131.2.21 nppiValidNormLevelGetBufferHostSize_16u32f_AC4R	2251
7.131.2.22 nppiValidNormLevelGetBufferHostSize_16u32f_C1R	2251
7.131.2.23 nppiValidNormLevelGetBufferHostSize_16u32f_C3R	2251
7.131.2.24 nppiValidNormLevelGetBufferHostSize_16u32f_C4R	2251
7.131.2.25 nppiValidNormLevelGetBufferHostSize_32f_AC4R	2252
7.131.2.26 nppiValidNormLevelGetBufferHostSize_32f_C1R	2252
7.131.2.27 nppiValidNormLevelGetBufferHostSize_32f_C3R	2252
7.131.2.28 nppiValidNormLevelGetBufferHostSize_32f_C4R	2253
7.131.2.29 nppiValidNormLevelGetBufferHostSize_8s32f_AC4R	2253
7.131.2.30 nppiValidNormLevelGetBufferHostSize_8s32f_C1R	2253
7.131.2.31 nppiValidNormLevelGetBufferHostSize_8s32f_C3R	2253
7.131.2.32 nppiValidNormLevelGetBufferHostSize_8s32f_C4R	2254
7.131.2.33 nppiValidNormLevelGetBufferHostSize_8u32f_AC4R	2254
7.131.2.34 nppiValidNormLevelGetBufferHostSize_8u32f_C1R	2254
7.131.2.35 nppiValidNormLevelGetBufferHostSize_8u32f_C3R	2255

7.131.2.36	nppiValidNormLevelGetBufferHostSize_8u32f_C4R	2255
7.131.2.37	nppiValidNormLevelGetBufferHostSize_8u_AC4RSfs	2255
7.131.2.38	nppiValidNormLevelGetBufferHostSize_8u_C1RSfs	2255
7.131.2.39	nppiValidNormLevelGetBufferHostSize_8u_C3RSfs	2256
7.131.2.40	nppiValidNormLevelGetBufferHostSize_8u_C4RSfs	2256
7.132	Image Quality Index	2257
7.132.1	Detailed Description	2259
7.132.2	Function Documentation	2259
7.132.2.1	nppiQualityIndex_16u32f_AC4R	2259
7.132.2.2	nppiQualityIndex_16u32f_C1R	2259
7.132.2.3	nppiQualityIndex_16u32f_C3R	2260
7.132.2.4	nppiQualityIndex_32f_AC4R	2260
7.132.2.5	nppiQualityIndex_32f_C1R	2261
7.132.2.6	nppiQualityIndex_32f_C3R	2261
7.132.2.7	nppiQualityIndex_8u32f_AC4R	2262
7.132.2.8	nppiQualityIndex_8u32f_C1R	2262
7.132.2.9	nppiQualityIndex_8u32f_C3R	2262
7.132.2.10	nppiQualityIndexGetBufferHostSize_16u32f_AC4R	2263
7.132.2.11	nppiQualityIndexGetBufferHostSize_16u32f_C1R	2263
7.132.2.12	nppiQualityIndexGetBufferHostSize_16u32f_C3R	2264
7.132.2.13	nppiQualityIndexGetBufferHostSize_32f_AC4R	2264
7.132.2.14	nppiQualityIndexGetBufferHostSize_32f_C1R	2264
7.132.2.15	nppiQualityIndexGetBufferHostSize_32f_C3R	2264
7.132.2.16	nppiQualityIndexGetBufferHostSize_8u32f_AC4R	2265
7.132.2.17	nppiQualityIndexGetBufferHostSize_8u32f_C1R	2265
7.132.2.18	nppiQualityIndexGetBufferHostSize_8u32f_C3R	2265
7.133	MaximumError	2266
7.133.1	Detailed Description	2269
7.133.2	Function Documentation	2269
7.133.2.1	nppiMaximumError_16s_C1R	2269
7.133.2.2	nppiMaximumError_16s_C2R	2270
7.133.2.3	nppiMaximumError_16s_C3R	2270
7.133.2.4	nppiMaximumError_16s_C4R	2271
7.133.2.5	nppiMaximumError_16sc_C1R	2271
7.133.2.6	nppiMaximumError_16sc_C2R	2271
7.133.2.7	nppiMaximumError_16sc_C3R	2272

7.133.2.8 nppiMaximumError_16sc_C4R	2272
7.133.2.9 nppiMaximumError_16u_C1R	2273
7.133.2.10 nppiMaximumError_16u_C2R	2273
7.133.2.11 nppiMaximumError_16u_C3R	2274
7.133.2.12 nppiMaximumError_16u_C4R	2274
7.133.2.13 nppiMaximumError_32f_C1R	2274
7.133.2.14 nppiMaximumError_32f_C2R	2275
7.133.2.15 nppiMaximumError_32f_C3R	2275
7.133.2.16 nppiMaximumError_32f_C4R	2276
7.133.2.17 nppiMaximumError_32fc_C1R	2276
7.133.2.18 nppiMaximumError_32fc_C2R	2277
7.133.2.19 nppiMaximumError_32fc_C3R	2277
7.133.2.20 nppiMaximumError_32fc_C4R	2278
7.133.2.21 nppiMaximumError_32s_C1R	2278
7.133.2.22 nppiMaximumError_32s_C2R	2278
7.133.2.23 nppiMaximumError_32s_C3R	2279
7.133.2.24 nppiMaximumError_32s_C4R	2279
7.133.2.25 nppiMaximumError_32sc_C1R	2280
7.133.2.26 nppiMaximumError_32sc_C2R	2280
7.133.2.27 nppiMaximumError_32sc_C3R	2281
7.133.2.28 nppiMaximumError_32sc_C4R	2281
7.133.2.29 nppiMaximumError_32u_C1R	2281
7.133.2.30 nppiMaximumError_32u_C2R	2282
7.133.2.31 nppiMaximumError_32u_C3R	2282
7.133.2.32 nppiMaximumError_32u_C4R	2283
7.133.2.33 nppiMaximumError_64f_C1R	2283
7.133.2.34 nppiMaximumError_64f_C2R	2284
7.133.2.35 nppiMaximumError_64f_C3R	2284
7.133.2.36 nppiMaximumError_64f_C4R	2284
7.133.2.37 nppiMaximumError_8s_C1R	2285
7.133.2.38 nppiMaximumError_8s_C2R	2285
7.133.2.39 nppiMaximumError_8s_C3R	2286
7.133.2.40 nppiMaximumError_8s_C4R	2286
7.133.2.41 nppiMaximumError_8u_C1R	2287
7.133.2.42 nppiMaximumError_8u_C2R	2287
7.133.2.43 nppiMaximumError_8u_C3R	2287

7.133.2.44	<code>ippiMaximumError_8u_C4R</code>	2288
7.134	<code>AverageError</code>	2289
7.134.1	Detailed Description	2292
7.134.2	Function Documentation	2292
7.134.2.1	<code>ippiAverageError_16s_C1R</code>	2292
7.134.2.2	<code>ippiAverageError_16s_C2R</code>	2293
7.134.2.3	<code>ippiAverageError_16s_C3R</code>	2293
7.134.2.4	<code>ippiAverageError_16s_C4R</code>	2294
7.134.2.5	<code>ippiAverageError_16sc_C1R</code>	2294
7.134.2.6	<code>ippiAverageError_16sc_C2R</code>	2295
7.134.2.7	<code>ippiAverageError_16sc_C3R</code>	2295
7.134.2.8	<code>ippiAverageError_16sc_C4R</code>	2295
7.134.2.9	<code>ippiAverageError_16u_C1R</code>	2296
7.134.2.10	<code>ippiAverageError_16u_C2R</code>	2296
7.134.2.11	<code>ippiAverageError_16u_C3R</code>	2297
7.134.2.12	<code>ippiAverageError_16u_C4R</code>	2297
7.134.2.13	<code>ippiAverageError_32f_C1R</code>	2298
7.134.2.14	<code>ippiAverageError_32f_C2R</code>	2298
7.134.2.15	<code>ippiAverageError_32f_C3R</code>	2298
7.134.2.16	<code>ippiAverageError_32f_C4R</code>	2299
7.134.2.17	<code>ippiAverageError_32fc_C1R</code>	2299
7.134.2.18	<code>ippiAverageError_32fc_C2R</code>	2300
7.134.2.19	<code>ippiAverageError_32fc_C3R</code>	2300
7.134.2.20	<code>ippiAverageError_32fc_C4R</code>	2301
7.134.2.21	<code>ippiAverageError_32s_C1R</code>	2301
7.134.2.22	<code>ippiAverageError_32s_C2R</code>	2302
7.134.2.23	<code>ippiAverageError_32s_C3R</code>	2302
7.134.2.24	<code>ippiAverageError_32s_C4R</code>	2302
7.134.2.25	<code>ippiAverageError_32sc_C1R</code>	2303
7.134.2.26	<code>ippiAverageError_32sc_C2R</code>	2303
7.134.2.27	<code>ippiAverageError_32sc_C3R</code>	2304
7.134.2.28	<code>ippiAverageError_32sc_C4R</code>	2304
7.134.2.29	<code>ippiAverageError_32u_C1R</code>	2305
7.134.2.30	<code>ippiAverageError_32u_C2R</code>	2305
7.134.2.31	<code>ippiAverageError_32u_C3R</code>	2305
7.134.2.32	<code>ippiAverageError_32u_C4R</code>	2306

7.134.2.33	nppiAverageError_64f_C1R	2306
7.134.2.34	nppiAverageError_64f_C2R	2307
7.134.2.35	nppiAverageError_64f_C3R	2307
7.134.2.36	nppiAverageError_64f_C4R	2308
7.134.2.37	nppiAverageError_8s_C1R	2308
7.134.2.38	nppiAverageError_8s_C2R	2309
7.134.2.39	nppiAverageError_8s_C3R	2309
7.134.2.40	nppiAverageError_8s_C4R	2309
7.134.2.41	nppiAverageError_8u_C1R	2310
7.134.2.42	nppiAverageError_8u_C2R	2310
7.134.2.43	nppiAverageError_8u_C3R	2311
7.134.2.44	nppiAverageError_8u_C4R	2311
7.135	MaximumRelativeError	2312
7.135.1	Detailed Description	2315
7.135.2	Function Documentation	2315
7.135.2.1	nppiMaximumRelativeError_16s_C1R	2315
7.135.2.2	nppiMaximumRelativeError_16s_C2R	2316
7.135.2.3	nppiMaximumRelativeError_16s_C3R	2316
7.135.2.4	nppiMaximumRelativeError_16s_C4R	2317
7.135.2.5	nppiMaximumRelativeError_16sc_C1R	2317
7.135.2.6	nppiMaximumRelativeError_16sc_C2R	2318
7.135.2.7	nppiMaximumRelativeError_16sc_C3R	2318
7.135.2.8	nppiMaximumRelativeError_16sc_C4R	2319
7.135.2.9	nppiMaximumRelativeError_16u_C1R	2319
7.135.2.10	nppiMaximumRelativeError_16u_C2R	2319
7.135.2.11	nppiMaximumRelativeError_16u_C3R	2320
7.135.2.12	nppiMaximumRelativeError_16u_C4R	2320
7.135.2.13	nppiMaximumRelativeError_32f_C1R	2321
7.135.2.14	nppiMaximumRelativeError_32f_C2R	2321
7.135.2.15	nppiMaximumRelativeError_32f_C3R	2322
7.135.2.16	nppiMaximumRelativeError_32f_C4R	2322
7.135.2.17	nppiMaximumRelativeError_32fc_C1R	2323
7.135.2.18	nppiMaximumRelativeError_32fc_C2R	2323
7.135.2.19	nppiMaximumRelativeError_32fc_C3R	2324
7.135.2.20	nppiMaximumRelativeError_32fc_C4R	2324
7.135.2.21	nppiMaximumRelativeError_32s_C1R	2325

7.135.2.22	<code>nppiMaximumRelativeError_32s_C2R</code>	2325
7.135.2.23	<code>nppiMaximumRelativeError_32s_C3R</code>	2325
7.135.2.24	<code>nppiMaximumRelativeError_32s_C4R</code>	2326
7.135.2.25	<code>nppiMaximumRelativeError_32sc_C1R</code>	2326
7.135.2.26	<code>nppiMaximumRelativeError_32sc_C2R</code>	2327
7.135.2.27	<code>nppiMaximumRelativeError_32sc_C3R</code>	2327
7.135.2.28	<code>nppiMaximumRelativeError_32sc_C4R</code>	2328
7.135.2.29	<code>nppiMaximumRelativeError_32u_C1R</code>	2328
7.135.2.30	<code>nppiMaximumRelativeError_32u_C2R</code>	2329
7.135.2.31	<code>nppiMaximumRelativeError_32u_C3R</code>	2329
7.135.2.32	<code>nppiMaximumRelativeError_32u_C4R</code>	2329
7.135.2.33	<code>nppiMaximumRelativeError_64f_C1R</code>	2330
7.135.2.34	<code>nppiMaximumRelativeError_64f_C2R</code>	2330
7.135.2.35	<code>nppiMaximumRelativeError_64f_C3R</code>	2331
7.135.2.36	<code>nppiMaximumRelativeError_64f_C4R</code>	2331
7.135.2.37	<code>nppiMaximumRelativeError_8s_C1R</code>	2332
7.135.2.38	<code>nppiMaximumRelativeError_8s_C2R</code>	2332
7.135.2.39	<code>nppiMaximumRelativeError_8s_C3R</code>	2333
7.135.2.40	<code>nppiMaximumRelativeError_8s_C4R</code>	2333
7.135.2.41	<code>nppiMaximumRelativeError_8u_C1R</code>	2334
7.135.2.42	<code>nppiMaximumRelativeError_8u_C2R</code>	2334
7.135.2.43	<code>nppiMaximumRelativeError_8u_C3R</code>	2334
7.135.2.44	<code>nppiMaximumRelativeError_8u_C4R</code>	2335
7.136	<code>AverageRelativeError</code>	2336
7.136.1	Detailed Description	2339
7.136.2	Function Documentation	2339
7.136.2.1	<code>nppiAverageRelativeError_16s_C1R</code>	2339
7.136.2.2	<code>nppiAverageRelativeError_16s_C2R</code>	2340
7.136.2.3	<code>nppiAverageRelativeError_16s_C3R</code>	2340
7.136.2.4	<code>nppiAverageRelativeError_16s_C4R</code>	2341
7.136.2.5	<code>nppiAverageRelativeError_16sc_C1R</code>	2341
7.136.2.6	<code>nppiAverageRelativeError_16sc_C2R</code>	2342
7.136.2.7	<code>nppiAverageRelativeError_16sc_C3R</code>	2342
7.136.2.8	<code>nppiAverageRelativeError_16sc_C4R</code>	2343
7.136.2.9	<code>nppiAverageRelativeError_16u_C1R</code>	2343
7.136.2.10	<code>nppiAverageRelativeError_16u_C2R</code>	2343

7.136.2.1	lnppiAverageRelativeError_16u_C3R	2344
7.136.2.12	npipiAverageRelativeError_16u_C4R	2344
7.136.2.13	npipiAverageRelativeError_32f_C1R	2345
7.136.2.14	npipiAverageRelativeError_32f_C2R	2345
7.136.2.15	npipiAverageRelativeError_32f_C3R	2346
7.136.2.16	npipiAverageRelativeError_32f_C4R	2346
7.136.2.17	npipiAverageRelativeError_32fc_C1R	2347
7.136.2.18	npipiAverageRelativeError_32fc_C2R	2347
7.136.2.19	npipiAverageRelativeError_32fc_C3R	2348
7.136.2.20	npipiAverageRelativeError_32fc_C4R	2348
7.136.2.21	lnppiAverageRelativeError_32s_C1R	2349
7.136.2.22	npipiAverageRelativeError_32s_C2R	2349
7.136.2.23	npipiAverageRelativeError_32s_C3R	2349
7.136.2.24	npipiAverageRelativeError_32s_C4R	2350
7.136.2.25	npipiAverageRelativeError_32sc_C1R	2350
7.136.2.26	npipiAverageRelativeError_32sc_C2R	2351
7.136.2.27	npipiAverageRelativeError_32sc_C3R	2351
7.136.2.28	npipiAverageRelativeError_32sc_C4R	2352
7.136.2.29	npipiAverageRelativeError_32u_C1R	2352
7.136.2.30	npipiAverageRelativeError_32u_C2R	2353
7.136.2.31	lnppiAverageRelativeError_32u_C3R	2353
7.136.2.32	npipiAverageRelativeError_32u_C4R	2353
7.136.2.33	npipiAverageRelativeError_64f_C1R	2354
7.136.2.34	npipiAverageRelativeError_64f_C2R	2354
7.136.2.35	npipiAverageRelativeError_64f_C3R	2355
7.136.2.36	npipiAverageRelativeError_64f_C4R	2355
7.136.2.37	npipiAverageRelativeError_8s_C1R	2356
7.136.2.38	npipiAverageRelativeError_8s_C2R	2356
7.136.2.39	npipiAverageRelativeError_8s_C3R	2357
7.136.2.40	npipiAverageRelativeError_8s_C4R	2357
7.136.2.41	lnppiAverageRelativeError_8u_C1R	2358
7.136.2.42	npipiAverageRelativeError_8u_C2R	2358
7.136.2.43	npipiAverageRelativeError_8u_C3R	2358
7.136.2.44	npipiAverageRelativeError_8u_C4R	2359
7.137	Memory Management	2360
7.137.1	Detailed Description	2362

7.137.2 Function Documentation	2362
7.137.2.1 nppiFree	2362
7.137.2.2 nppiMalloc_16s_C1	2362
7.137.2.3 nppiMalloc_16s_C2	2363
7.137.2.4 nppiMalloc_16s_C4	2363
7.137.2.5 nppiMalloc_16sc_C1	2363
7.137.2.6 nppiMalloc_16sc_C2	2363
7.137.2.7 nppiMalloc_16sc_C3	2364
7.137.2.8 nppiMalloc_16sc_C4	2364
7.137.2.9 nppiMalloc_16u_C1	2364
7.137.2.10 nppiMalloc_16u_C2	2365
7.137.2.11 nppiMalloc_16u_C3	2365
7.137.2.12 nppiMalloc_16u_C4	2365
7.137.2.13 nppiMalloc_32f_C1	2365
7.137.2.14 nppiMalloc_32f_C2	2366
7.137.2.15 nppiMalloc_32f_C3	2366
7.137.2.16 nppiMalloc_32f_C4	2366
7.137.2.17 nppiMalloc_32fc_C1	2367
7.137.2.18 nppiMalloc_32fc_C2	2367
7.137.2.19 nppiMalloc_32fc_C3	2367
7.137.2.20 nppiMalloc_32fc_C4	2367
7.137.2.21 nppiMalloc_32s_C1	2368
7.137.2.22 nppiMalloc_32s_C3	2368
7.137.2.23 nppiMalloc_32s_C4	2368
7.137.2.24 nppiMalloc_32sc_C1	2369
7.137.2.25 nppiMalloc_32sc_C2	2369
7.137.2.26 nppiMalloc_32sc_C3	2369
7.137.2.27 nppiMalloc_32sc_C4	2369
7.137.2.28 nppiMalloc_8u_C1	2370
7.137.2.29 nppiMalloc_8u_C2	2370
7.137.2.30 nppiMalloc_8u_C3	2370
7.137.2.31 nppiMalloc_8u_C4	2371
7.138 Threshold and Compare Operations	2372
7.138.1 Detailed Description	2372
7.139 Threshold Operations	2373
7.139.1 Detailed Description	2387

7.139.2 Function Documentation	2387
7.139.2.1 nppiThreshold_16s_AC4IR	2387
7.139.2.2 nppiThreshold_16s_AC4R	2387
7.139.2.3 nppiThreshold_16s_C1IR	2388
7.139.2.4 nppiThreshold_16s_C1R	2388
7.139.2.5 nppiThreshold_16s_C3IR	2389
7.139.2.6 nppiThreshold_16s_C3R	2389
7.139.2.7 nppiThreshold_16u_AC4IR	2390
7.139.2.8 nppiThreshold_16u_AC4R	2390
7.139.2.9 nppiThreshold_16u_C1IR	2391
7.139.2.10 nppiThreshold_16u_C1R	2391
7.139.2.11 nppiThreshold_16u_C3IR	2391
7.139.2.12 nppiThreshold_16u_C3R	2392
7.139.2.13 nppiThreshold_32f_AC4IR	2392
7.139.2.14 nppiThreshold_32f_AC4R	2393
7.139.2.15 nppiThreshold_32f_C1IR	2393
7.139.2.16 nppiThreshold_32f_C1R	2394
7.139.2.17 nppiThreshold_32f_C3IR	2394
7.139.2.18 nppiThreshold_32f_C3R	2395
7.139.2.19 nppiThreshold_8u_AC4IR	2395
7.139.2.20 nppiThreshold_8u_AC4R	2396
7.139.2.21 nppiThreshold_8u_C1IR	2396
7.139.2.22 nppiThreshold_8u_C1R	2397
7.139.2.23 nppiThreshold_8u_C3IR	2397
7.139.2.24 nppiThreshold_8u_C3R	2398
7.139.2.25 nppiThreshold_GT_16s_AC4IR	2398
7.139.2.26 nppiThreshold_GT_16s_AC4R	2398
7.139.2.27 nppiThreshold_GT_16s_C1IR	2399
7.139.2.28 nppiThreshold_GT_16s_C1R	2399
7.139.2.29 nppiThreshold_GT_16s_C3IR	2400
7.139.2.30 nppiThreshold_GT_16s_C3R	2400
7.139.2.31 nppiThreshold_GT_16u_AC4IR	2400
7.139.2.32 nppiThreshold_GT_16u_AC4R	2401
7.139.2.33 nppiThreshold_GT_16u_C1IR	2401
7.139.2.34 nppiThreshold_GT_16u_C1R	2402
7.139.2.35 nppiThreshold_GT_16u_C3IR	2402

7.139.2.36	nppiThreshold_GT_16u_C3R	2402
7.139.2.37	nppiThreshold_GT_32f_AC4IR	2403
7.139.2.38	nppiThreshold_GT_32f_AC4R	2403
7.139.2.39	nppiThreshold_GT_32f_C1IR	2404
7.139.2.40	nppiThreshold_GT_32f_C1R	2404
7.139.2.41	nppiThreshold_GT_32f_C3IR	2404
7.139.2.42	nppiThreshold_GT_32f_C3R	2405
7.139.2.43	nppiThreshold_GT_8u_AC4IR	2405
7.139.2.44	nppiThreshold_GT_8u_AC4R	2406
7.139.2.45	nppiThreshold_GT_8u_C1IR	2406
7.139.2.46	nppiThreshold_GT_8u_C1R	2406
7.139.2.47	nppiThreshold_GT_8u_C3IR	2407
7.139.2.48	nppiThreshold_GT_8u_C3R	2407
7.139.2.49	nppiThreshold_GTVal_16s_AC4IR	2408
7.139.2.50	nppiThreshold_GTVal_16s_AC4R	2408
7.139.2.51	nppiThreshold_GTVal_16s_C1IR	2408
7.139.2.52	nppiThreshold_GTVal_16s_C1R	2409
7.139.2.53	nppiThreshold_GTVal_16s_C3IR	2409
7.139.2.54	nppiThreshold_GTVal_16s_C3R	2410
7.139.2.55	nppiThreshold_GTVal_16u_AC4IR	2410
7.139.2.56	nppiThreshold_GTVal_16u_AC4R	2410
7.139.2.57	nppiThreshold_GTVal_16u_C1IR	2411
7.139.2.58	nppiThreshold_GTVal_16u_C1R	2411
7.139.2.59	nppiThreshold_GTVal_16u_C3IR	2412
7.139.2.60	nppiThreshold_GTVal_16u_C3R	2412
7.139.2.61	nppiThreshold_GTVal_32f_AC4IR	2413
7.139.2.62	nppiThreshold_GTVal_32f_AC4R	2413
7.139.2.63	nppiThreshold_GTVal_32f_C1IR	2413
7.139.2.64	nppiThreshold_GTVal_32f_C1R	2414
7.139.2.65	nppiThreshold_GTVal_32f_C3IR	2414
7.139.2.66	nppiThreshold_GTVal_32f_C3R	2415
7.139.2.67	nppiThreshold_GTVal_8u_AC4IR	2415
7.139.2.68	nppiThreshold_GTVal_8u_AC4R	2415
7.139.2.69	nppiThreshold_GTVal_8u_C1IR	2416
7.139.2.70	nppiThreshold_GTVal_8u_C1R	2416
7.139.2.71	nppiThreshold_GTVal_8u_C3IR	2417

7.139.2.72	nppiThreshold_GTVal_8u_C3R	2417
7.139.2.73	nppiThreshold_LT_16s_AC4IR	2418
7.139.2.74	nppiThreshold_LT_16s_AC4R	2418
7.139.2.75	nppiThreshold_LT_16s_C1IR	2418
7.139.2.76	nppiThreshold_LT_16s_C1R	2419
7.139.2.77	nppiThreshold_LT_16s_C3IR	2419
7.139.2.78	nppiThreshold_LT_16s_C3R	2420
7.139.2.79	nppiThreshold_LT_16u_AC4IR	2420
7.139.2.80	nppiThreshold_LT_16u_AC4R	2420
7.139.2.81	nppiThreshold_LT_16u_C1IR	2421
7.139.2.82	nppiThreshold_LT_16u_C1R	2421
7.139.2.83	nppiThreshold_LT_16u_C3IR	2422
7.139.2.84	nppiThreshold_LT_16u_C3R	2422
7.139.2.85	nppiThreshold_LT_32f_AC4IR	2422
7.139.2.86	nppiThreshold_LT_32f_AC4R	2423
7.139.2.87	nppiThreshold_LT_32f_C1IR	2423
7.139.2.88	nppiThreshold_LT_32f_C1R	2424
7.139.2.89	nppiThreshold_LT_32f_C3IR	2424
7.139.2.90	nppiThreshold_LT_32f_C3R	2424
7.139.2.91	nppiThreshold_LT_8u_AC4IR	2425
7.139.2.92	nppiThreshold_LT_8u_AC4R	2425
7.139.2.93	nppiThreshold_LT_8u_C1IR	2426
7.139.2.94	nppiThreshold_LT_8u_C1R	2426
7.139.2.95	nppiThreshold_LT_8u_C3IR	2426
7.139.2.96	nppiThreshold_LT_8u_C3R	2427
7.139.2.97	nppiThreshold_LTVal_16s_AC4IR	2427
7.139.2.98	nppiThreshold_LTVal_16s_AC4R	2428
7.139.2.99	nppiThreshold_LTVal_16s_C1IR	2428
7.139.2.100	nppiThreshold_LTVal_16s_C1R	2428
7.139.2.101	nppiThreshold_LTVal_16s_C3IR	2429
7.139.2.102	nppiThreshold_LTVal_16s_C3R	2429
7.139.2.103	nppiThreshold_LTVal_16u_AC4IR	2430
7.139.2.104	nppiThreshold_LTVal_16u_AC4R	2430
7.139.2.105	nppiThreshold_LTVal_16u_C1IR	2431
7.139.2.106	nppiThreshold_LTVal_16u_C1R	2431
7.139.2.107	nppiThreshold_LTVal_16u_C3IR	2431

7.139.2.108ppiThreshold_LTVal_16u_C3R	2432
7.139.2.109ppiThreshold_LTVal_32f_AC4IR	2432
7.139.2.110ppiThreshold_LTVal_32f_AC4R	2433
7.139.2.111ppiThreshold_LTVal_32f_C1IR	2433
7.139.2.112ppiThreshold_LTVal_32f_C1R	2433
7.139.2.113ppiThreshold_LTVal_32f_C3IR	2434
7.139.2.114ppiThreshold_LTVal_32f_C3R	2434
7.139.2.115ppiThreshold_LTVal_8u_AC4IR	2435
7.139.2.116ppiThreshold_LTVal_8u_AC4R	2435
7.139.2.117ppiThreshold_LTVal_8u_C1IR	2436
7.139.2.118ppiThreshold_LTVal_8u_C1R	2436
7.139.2.119ppiThreshold_LTVal_8u_C3IR	2436
7.139.2.120ppiThreshold_LTVal_8u_C3R	2437
7.139.2.121ppiThreshold_LTValGTVal_16s_AC4IR	2437
7.139.2.122ppiThreshold_LTValGTVal_16s_AC4R	2438
7.139.2.123ppiThreshold_LTValGTVal_16s_C1IR	2438
7.139.2.124ppiThreshold_LTValGTVal_16s_C1R	2439
7.139.2.125ppiThreshold_LTValGTVal_16s_C3IR	2439
7.139.2.126ppiThreshold_LTValGTVal_16s_C3R	2440
7.139.2.127ppiThreshold_LTValGTVal_16u_AC4IR	2440
7.139.2.128ppiThreshold_LTValGTVal_16u_AC4R	2441
7.139.2.129ppiThreshold_LTValGTVal_16u_C1IR	2441
7.139.2.130ppiThreshold_LTValGTVal_16u_C1R	2442
7.139.2.131ppiThreshold_LTValGTVal_16u_C3IR	2442
7.139.2.132ppiThreshold_LTValGTVal_16u_C3R	2443
7.139.2.133ppiThreshold_LTValGTVal_32f_AC4IR	2443
7.139.2.134ppiThreshold_LTValGTVal_32f_AC4R	2444
7.139.2.135ppiThreshold_LTValGTVal_32f_C1IR	2444
7.139.2.136ppiThreshold_LTValGTVal_32f_C1R	2445
7.139.2.137ppiThreshold_LTValGTVal_32f_C3IR	2445
7.139.2.138ppiThreshold_LTValGTVal_32f_C3R	2446
7.139.2.139ppiThreshold_LTValGTVal_8u_AC4IR	2446
7.139.2.140ppiThreshold_LTValGTVal_8u_AC4R	2447
7.139.2.141ppiThreshold_LTValGTVal_8u_C1IR	2447
7.139.2.142ppiThreshold_LTValGTVal_8u_C1R	2448
7.139.2.143ppiThreshold_LTValGTVal_8u_C3IR	2448

7.139.2.144	nppiThreshold_LTVaGTVa_8u_C3R	2449
7.139.2.145	nppiThreshold_Val_16s_AC4IR	2449
7.139.2.146	nppiThreshold_Val_16s_AC4R	2450
7.139.2.147	nppiThreshold_Val_16s_C1IR	2450
7.139.2.148	nppiThreshold_Val_16s_C1R	2451
7.139.2.149	nppiThreshold_Val_16s_C3IR	2451
7.139.2.150	nppiThreshold_Val_16s_C3R	2452
7.139.2.151	nppiThreshold_Val_16u_AC4IR	2452
7.139.2.152	nppiThreshold_Val_16u_AC4R	2453
7.139.2.153	nppiThreshold_Val_16u_C1IR	2453
7.139.2.154	nppiThreshold_Val_16u_C1R	2454
7.139.2.155	nppiThreshold_Val_16u_C3IR	2454
7.139.2.156	nppiThreshold_Val_16u_C3R	2455
7.139.2.157	nppiThreshold_Val_32f_AC4IR	2455
7.139.2.158	nppiThreshold_Val_32f_AC4R	2456
7.139.2.159	nppiThreshold_Val_32f_C1IR	2456
7.139.2.160	nppiThreshold_Val_32f_C1R	2457
7.139.2.161	nppiThreshold_Val_32f_C3IR	2457
7.139.2.162	nppiThreshold_Val_32f_C3R	2458
7.139.2.163	nppiThreshold_Val_8u_AC4IR	2458
7.139.2.164	nppiThreshold_Val_8u_AC4R	2459
7.139.2.165	nppiThreshold_Val_8u_C1IR	2459
7.139.2.166	nppiThreshold_Val_8u_C1R	2460
7.139.2.167	nppiThreshold_Val_8u_C3IR	2460
7.139.2.168	nppiThreshold_Val_8u_C3R	2461
7.140	Compare Operations	2462
7.140.1	Detailed Description	2465
7.140.2	Function Documentation	2465
7.140.2.1	nppiCompare_16s_AC4R	2465
7.140.2.2	nppiCompare_16s_C1R	2466
7.140.2.3	nppiCompare_16s_C3R	2466
7.140.2.4	nppiCompare_16s_C4R	2467
7.140.2.5	nppiCompare_16u_AC4R	2467
7.140.2.6	nppiCompare_16u_C1R	2468
7.140.2.7	nppiCompare_16u_C3R	2468
7.140.2.8	nppiCompare_16u_C4R	2469

7.140.2.9 nppiCompare_32f_AC4R	2469
7.140.2.10 nppiCompare_32f_C1R	2470
7.140.2.11 nppiCompare_32f_C3R	2470
7.140.2.12 nppiCompare_32f_C4R	2471
7.140.2.13 nppiCompare_8u_AC4R	2471
7.140.2.14 nppiCompare_8u_C1R	2472
7.140.2.15 nppiCompare_8u_C3R	2472
7.140.2.16 nppiCompare_8u_C4R	2473
7.140.2.17 nppiCompareC_16s_AC4R	2473
7.140.2.18 nppiCompareC_16s_C1R	2473
7.140.2.19 nppiCompareC_16s_C3R	2474
7.140.2.20 nppiCompareC_16s_C4R	2474
7.140.2.21 nppiCompareC_16u_AC4R	2475
7.140.2.22 nppiCompareC_16u_C1R	2475
7.140.2.23 nppiCompareC_16u_C3R	2476
7.140.2.24 nppiCompareC_16u_C4R	2476
7.140.2.25 nppiCompareC_32f_AC4R	2476
7.140.2.26 nppiCompareC_32f_C1R	2477
7.140.2.27 nppiCompareC_32f_C3R	2477
7.140.2.28 nppiCompareC_32f_C4R	2478
7.140.2.29 nppiCompareC_8u_AC4R	2478
7.140.2.30 nppiCompareC_8u_C1R	2479
7.140.2.31 nppiCompareC_8u_C3R	2479
7.140.2.32 nppiCompareC_8u_C4R	2479
7.140.2.33 nppiCompareEqualEps_32f_AC4R	2480
7.140.2.34 nppiCompareEqualEps_32f_C1R	2480
7.140.2.35 nppiCompareEqualEps_32f_C3R	2481
7.140.2.36 nppiCompareEqualEps_32f_C4R	2481
7.140.2.37 nppiCompareEqualEpsC_32f_AC4R	2482
7.140.2.38 nppiCompareEqualEpsC_32f_C1R	2482
7.140.2.39 nppiCompareEqualEpsC_32f_C3R	2483
7.140.2.40 nppiCompareEqualEpsC_32f_C4R	2483
7.141 NPP Signal Processing	2485
7.142 Arithmetic and Logical Operations	2486
7.143 Arithmetic Operations	2487
7.144 AddC	2489

7.144.1 Detailed Description	2490
7.144.2 Function Documentation	2490
7.144.2.1 nppsAddC_16s_ISfs	2490
7.144.2.2 nppsAddC_16s_Sfs	2491
7.144.2.3 nppsAddC_16sc_ISfs	2491
7.144.2.4 nppsAddC_16sc_Sfs	2491
7.144.2.5 nppsAddC_16u_ISfs	2492
7.144.2.6 nppsAddC_16u_Sfs	2492
7.144.2.7 nppsAddC_32f	2492
7.144.2.8 nppsAddC_32f_I	2493
7.144.2.9 nppsAddC_32fc	2493
7.144.2.10 nppsAddC_32fc_I	2493
7.144.2.11 nppsAddC_32s_ISfs	2493
7.144.2.12 nppsAddC_32s_Sfs	2494
7.144.2.13 nppsAddC_32sc_ISfs	2494
7.144.2.14 nppsAddC_32sc_Sfs	2495
7.144.2.15 nppsAddC_64f	2495
7.144.2.16 nppsAddC_64f_I	2495
7.144.2.17 nppsAddC_64fc	2496
7.144.2.18 nppsAddC_64fc_I	2496
7.144.2.19 nppsAddC_8u_ISfs	2496
7.144.2.20 nppsAddC_8u_Sfs	2497
7.145 AddProductC	2498
7.145.1 Detailed Description	2498
7.145.2 Function Documentation	2498
7.145.2.1 nppsAddProductC_32f	2498
7.146 MulC	2499
7.146.1 Detailed Description	2500
7.146.2 Function Documentation	2500
7.146.2.1 nppsMulC_16s_ISfs	2500
7.146.2.2 nppsMulC_16s_Sfs	2501
7.146.2.3 nppsMulC_16sc_ISfs	2501
7.146.2.4 nppsMulC_16sc_Sfs	2502
7.146.2.5 nppsMulC_16u_ISfs	2502
7.146.2.6 nppsMulC_16u_Sfs	2502
7.146.2.7 nppsMulC_32f	2503

7.146.2.8	nppsMulC_32f16s_Sfs	2503
7.146.2.9	nppsMulC_32f_I	2503
7.146.2.10	nppsMulC_32fc	2504
7.146.2.11	nppsMulC_32fc_I	2504
7.146.2.12	nppsMulC_32s_ISfs	2504
7.146.2.13	nppsMulC_32s_Sfs	2505
7.146.2.14	nppsMulC_32sc_ISfs	2505
7.146.2.15	nppsMulC_32sc_Sfs	2505
7.146.2.16	nppsMulC_64f	2506
7.146.2.17	nppsMulC_64f64s_ISfs	2506
7.146.2.18	nppsMulC_64f_I	2506
7.146.2.19	nppsMulC_64fc	2507
7.146.2.20	nppsMulC_64fc_I	2507
7.146.2.21	nppsMulC_8u_ISfs	2507
7.146.2.22	nppsMulC_8u_Sfs	2508
7.146.2.23	nppsMulC_Low_32f16s	2508
7.147	SubC	2509
7.147.1	Detailed Description	2510
7.147.2	Function Documentation	2510
7.147.2.1	nppsSubC_16s_ISfs	2510
7.147.2.2	nppsSubC_16s_Sfs	2511
7.147.2.3	nppsSubC_16sc_ISfs	2511
7.147.2.4	nppsSubC_16sc_Sfs	2511
7.147.2.5	nppsSubC_16u_ISfs	2512
7.147.2.6	nppsSubC_16u_Sfs	2512
7.147.2.7	nppsSubC_32f	2512
7.147.2.8	nppsSubC_32f_I	2513
7.147.2.9	nppsSubC_32fc	2513
7.147.2.10	nppsSubC_32fc_I	2513
7.147.2.11	nppsSubC_32s_ISfs	2513
7.147.2.12	nppsSubC_32s_Sfs	2514
7.147.2.13	nppsSubC_32sc_ISfs	2514
7.147.2.14	nppsSubC_32sc_Sfs	2515
7.147.2.15	nppsSubC_64f	2515
7.147.2.16	nppsSubC_64f_I	2515
7.147.2.17	nppsSubC_64fc	2516

7.147.2.18	nppsSubC_64fc_I	2516
7.147.2.19	nppsSubC_8u_ISfs	2516
7.147.2.20	nppsSubC_8u_Sfs	2517
7.148	SubCRev	2518
7.148.1	Detailed Description	2519
7.148.2	Function Documentation	2519
7.148.2.1	nppsSubCRev_16s_ISfs	2519
7.148.2.2	nppsSubCRev_16s_Sfs	2520
7.148.2.3	nppsSubCRev_16sc_ISfs	2520
7.148.2.4	nppsSubCRev_16sc_Sfs	2520
7.148.2.5	nppsSubCRev_16u_ISfs	2521
7.148.2.6	nppsSubCRev_16u_Sfs	2521
7.148.2.7	nppsSubCRev_32f	2521
7.148.2.8	nppsSubCRev_32f_I	2522
7.148.2.9	nppsSubCRev_32fc	2522
7.148.2.10	nppsSubCRev_32fc_I	2522
7.148.2.11	nppsSubCRev_32s_ISfs	2523
7.148.2.12	nppsSubCRev_32s_Sfs	2523
7.148.2.13	nppsSubCRev_32sc_ISfs	2523
7.148.2.14	nppsSubCRev_32sc_Sfs	2524
7.148.2.15	nppsSubCRev_64f	2524
7.148.2.16	nppsSubCRev_64f_I	2524
7.148.2.17	nppsSubCRev_64fc	2525
7.148.2.18	nppsSubCRev_64fc_I	2525
7.148.2.19	nppsSubCRev_8u_ISfs	2525
7.148.2.20	nppsSubCRev_8u_Sfs	2526
7.149	DivC	2527
7.149.1	Detailed Description	2528
7.149.2	Function Documentation	2528
7.149.2.1	nppsDivC_16s_ISfs	2528
7.149.2.2	nppsDivC_16s_Sfs	2528
7.149.2.3	nppsDivC_16sc_ISfs	2529
7.149.2.4	nppsDivC_16sc_Sfs	2529
7.149.2.5	nppsDivC_16u_ISfs	2529
7.149.2.6	nppsDivC_16u_Sfs	2530
7.149.2.7	nppsDivC_32f	2530

7.149.2.8	nppsDivC_32f_I	2530
7.149.2.9	nppsDivC_32fc	2531
7.149.2.10	nppsDivC_32fc_I	2531
7.149.2.11	nppsDivC_64f	2531
7.149.2.12	nppsDivC_64f_I	2532
7.149.2.13	nppsDivC_64fc	2532
7.149.2.14	nppsDivC_64fc_I	2532
7.149.2.15	nppsDivC_8u_ISfs	2532
7.149.2.16	nppsDivC_8u_Sfs	2533
7.150	DivCRev	2534
7.150.1	Detailed Description	2534
7.150.2	Function Documentation	2534
7.150.2.1	nppsDivCRev_16u	2534
7.150.2.2	nppsDivCRev_16u_I	2534
7.150.2.3	nppsDivCRev_32f	2535
7.150.2.4	nppsDivCRev_32f_I	2535
7.151	Add	2536
7.151.1	Detailed Description	2538
7.151.2	Function Documentation	2538
7.151.2.1	nppsAdd_16s	2538
7.151.2.2	nppsAdd_16s32f	2538
7.151.2.3	nppsAdd_16s32s_I	2539
7.151.2.4	nppsAdd_16s_I	2539
7.151.2.5	nppsAdd_16s_ISfs	2539
7.151.2.6	nppsAdd_16s_Sfs	2540
7.151.2.7	nppsAdd_16sc_ISfs	2540
7.151.2.8	nppsAdd_16sc_Sfs	2540
7.151.2.9	nppsAdd_16u	2541
7.151.2.10	nppsAdd_16u_ISfs	2541
7.151.2.11	nppsAdd_16u_Sfs	2541
7.151.2.12	nppsAdd_32f	2542
7.151.2.13	nppsAdd_32f_I	2542
7.151.2.14	nppsAdd_32fc	2542
7.151.2.15	nppsAdd_32fc_I	2543
7.151.2.16	nppsAdd_32s_ISfs	2543
7.151.2.17	nppsAdd_32s_Sfs	2543

7.151.2.18	nppsAdd_32sc_ISfs	2544
7.151.2.19	nppsAdd_32sc_Sfs	2544
7.151.2.20	nppsAdd_32u	2544
7.151.2.21	nppsAdd_64f	2545
7.151.2.22	nppsAdd_64f_I	2545
7.151.2.23	nppsAdd_64fc	2545
7.151.2.24	nppsAdd_64fc_I	2546
7.151.2.25	nppsAdd_64s_Sfs	2546
7.151.2.26	nppsAdd_8u16u	2546
7.151.2.27	nppsAdd_8u_ISfs	2547
7.151.2.28	nppsAdd_8u_Sfs	2547
7.152	AddProduct	2548
7.152.1	Detailed Description	2548
7.152.2	Function Documentation	2549
7.152.2.1	nppsAddProduct_16s32s_Sfs	2549
7.152.2.2	nppsAddProduct_16s_Sfs	2549
7.152.2.3	nppsAddProduct_32f	2549
7.152.2.4	nppsAddProduct_32fc	2550
7.152.2.5	nppsAddProduct_32s_Sfs	2550
7.152.2.6	nppsAddProduct_64f	2551
7.152.2.7	nppsAddProduct_64fc	2551
7.153	Mul	2552
7.153.1	Detailed Description	2554
7.153.2	Function Documentation	2554
7.153.2.1	nppsMul_16s	2554
7.153.2.2	nppsMul_16s32f	2555
7.153.2.3	nppsMul_16s32s_Sfs	2555
7.153.2.4	nppsMul_16s_I	2555
7.153.2.5	nppsMul_16s_ISfs	2556
7.153.2.6	nppsMul_16s_Sfs	2556
7.153.2.7	nppsMul_16sc_ISfs	2556
7.153.2.8	nppsMul_16sc_Sfs	2557
7.153.2.9	nppsMul_16u16s_Sfs	2557
7.153.2.10	nppsMul_16u_ISfs	2557
7.153.2.11	nppsMul_16u_Sfs	2558
7.153.2.12	nppsMul_32f	2558

7.153.2.13	nppsMul_32f32fc	2558
7.153.2.14	nppsMul_32f32fc_I	2559
7.153.2.15	nppsMul_32f_I	2559
7.153.2.16	nppsMul_32fc	2559
7.153.2.17	nppsMul_32fc_I	2560
7.153.2.18	nppsMul_32s32sc_ISfs	2560
7.153.2.19	nppsMul_32s32sc_Sfs	2560
7.153.2.20	nppsMul_32s_ISfs	2561
7.153.2.21	nppsMul_32s_Sfs	2561
7.153.2.22	nppsMul_32sc_ISfs	2561
7.153.2.23	nppsMul_32sc_Sfs	2562
7.153.2.24	nppsMul_64f	2562
7.153.2.25	nppsMul_64f_I	2562
7.153.2.26	nppsMul_64fc	2563
7.153.2.27	nppsMul_64fc_I	2563
7.153.2.28	nppsMul_8u16u	2563
7.153.2.29	nppsMul_8u_ISfs	2564
7.153.2.30	nppsMul_8u_Sfs	2564
7.153.2.31	nppsMul_Low_32s_Sfs	2564
7.154	Sub	2565
7.154.1	Detailed Description	2566
7.154.2	Function Documentation	2566
7.154.2.1	nppsSub_16s	2566
7.154.2.2	nppsSub_16s32f	2567
7.154.2.3	nppsSub_16s_I	2567
7.154.2.4	nppsSub_16s_ISfs	2567
7.154.2.5	nppsSub_16s_Sfs	2568
7.154.2.6	nppsSub_16sc_ISfs	2568
7.154.2.7	nppsSub_16sc_Sfs	2568
7.154.2.8	nppsSub_16u_ISfs	2569
7.154.2.9	nppsSub_16u_Sfs	2569
7.154.2.10	nppsSub_32f	2569
7.154.2.11	nppsSub_32f_I	2570
7.154.2.12	nppsSub_32fc	2570
7.154.2.13	nppsSub_32fc_I	2570
7.154.2.14	nppsSub_32s_ISfs	2571

7.154.2.15	nppsSub_32s_Sfs	2571
7.154.2.16	nppsSub_32sc_ISfs	2571
7.154.2.17	nppsSub_32sc_Sfs	2572
7.154.2.18	nppsSub_64f	2572
7.154.2.19	nppsSub_64f_I	2572
7.154.2.20	nppsSub_64fc	2573
7.154.2.21	nppsSub_64fc_I	2573
7.154.2.22	nppsSub_8u_ISfs	2573
7.154.2.23	nppsSub_8u_Sfs	2574
7.155	Div	2575
7.155.1	Detailed Description	2576
7.155.2	Function Documentation	2576
7.155.2.1	nppsDiv_16s_ISfs	2576
7.155.2.2	nppsDiv_16s_Sfs	2577
7.155.2.3	nppsDiv_16sc_ISfs	2577
7.155.2.4	nppsDiv_16sc_Sfs	2577
7.155.2.5	nppsDiv_16u_ISfs	2578
7.155.2.6	nppsDiv_16u_Sfs	2578
7.155.2.7	nppsDiv_32f	2578
7.155.2.8	nppsDiv_32f_I	2579
7.155.2.9	nppsDiv_32fc	2579
7.155.2.10	nppsDiv_32fc_I	2579
7.155.2.11	nppsDiv_32s16s_Sfs	2579
7.155.2.12	nppsDiv_32s_ISfs	2580
7.155.2.13	nppsDiv_32s_Sfs	2580
7.155.2.14	nppsDiv_64f	2581
7.155.2.15	nppsDiv_64f_I	2581
7.155.2.16	nppsDiv_64fc	2581
7.155.2.17	nppsDiv_64fc_I	2582
7.155.2.18	nppsDiv_8u_ISfs	2582
7.155.2.19	nppsDiv_8u_Sfs	2582
7.156	Div_Round	2583
7.156.1	Detailed Description	2583
7.156.2	Function Documentation	2583
7.156.2.1	nppsDiv_Round_16s_ISfs	2583
7.156.2.2	nppsDiv_Round_16s_Sfs	2584

7.156.2.3 nppsDiv_Round_16u_ISfs	2584
7.156.2.4 nppsDiv_Round_16u_Sfs	2584
7.156.2.5 nppsDiv_Round_8u_ISfs	2585
7.156.2.6 nppsDiv_Round_8u_Sfs	2585
7.157Abs	2586
7.157.1 Detailed Description	2586
7.157.2 Function Documentation	2586
7.157.2.1 nppsAbs_16s	2586
7.157.2.2 nppsAbs_16s_I	2587
7.157.2.3 nppsAbs_32f	2587
7.157.2.4 nppsAbs_32f_I	2587
7.157.2.5 nppsAbs_32s	2587
7.157.2.6 nppsAbs_32s_I	2588
7.157.2.7 nppsAbs_64f	2588
7.157.2.8 nppsAbs_64f_I	2588
7.158Sqr	2589
7.158.1 Detailed Description	2590
7.158.2 Function Documentation	2590
7.158.2.1 nppsSqr_16s_ISfs	2590
7.158.2.2 nppsSqr_16s_Sfs	2590
7.158.2.3 nppsSqr_16sc_ISfs	2590
7.158.2.4 nppsSqr_16sc_Sfs	2591
7.158.2.5 nppsSqr_16u_ISfs	2591
7.158.2.6 nppsSqr_16u_Sfs	2591
7.158.2.7 nppsSqr_32f	2592
7.158.2.8 nppsSqr_32f_I	2592
7.158.2.9 nppsSqr_32fc	2592
7.158.2.10nppsSqr_32fc_I	2592
7.158.2.11nppsSqr_64f	2593
7.158.2.12nppsSqr_64f_I	2593
7.158.2.13nppsSqr_64fc	2593
7.158.2.14nppsSqr_64fc_I	2593
7.158.2.15nppsSqr_8u_ISfs	2594
7.158.2.16nppsSqr_8u_Sfs	2594
7.159Sqrt	2595
7.159.1 Detailed Description	2596

7.159.2 Function Documentation	2596
7.159.2.1 nppsSqrt_16s_ISfs	2596
7.159.2.2 nppsSqrt_16s_Sfs	2596
7.159.2.3 nppsSqrt_16sc_ISfs	2597
7.159.2.4 nppsSqrt_16sc_Sfs	2597
7.159.2.5 nppsSqrt_16u_ISfs	2597
7.159.2.6 nppsSqrt_16u_Sfs	2598
7.159.2.7 nppsSqrt_32f	2598
7.159.2.8 nppsSqrt_32f_I	2598
7.159.2.9 nppsSqrt_32fc	2598
7.159.2.10 nppsSqrt_32fc_I	2599
7.159.2.11 nppsSqrt_32s16s_Sfs	2599
7.159.2.12 nppsSqrt_64f	2599
7.159.2.13 nppsSqrt_64f_I	2600
7.159.2.14 nppsSqrt_64fc	2600
7.159.2.15 nppsSqrt_64fc_I	2600
7.159.2.16 nppsSqrt_64s16s_Sfs	2600
7.159.2.17 nppsSqrt_64s_ISfs	2601
7.159.2.18 nppsSqrt_64s_Sfs	2601
7.159.2.19 nppsSqrt_8u_ISfs	2601
7.159.2.20 nppsSqrt_8u_Sfs	2601
7.160 Cubrt	2603
7.160.1 Detailed Description	2603
7.160.2 Function Documentation	2603
7.160.2.1 nppsCubrt_32f	2603
7.160.2.2 nppsCubrt_32s16s_Sfs	2603
7.161 Exp	2604
7.161.1 Detailed Description	2604
7.161.2 Function Documentation	2604
7.161.2.1 nppsExp_16s_ISfs	2604
7.161.2.2 nppsExp_16s_Sfs	2605
7.161.2.3 nppsExp_32f	2605
7.161.2.4 nppsExp_32f64f	2605
7.161.2.5 nppsExp_32f_I	2606
7.161.2.6 nppsExp_32s_ISfs	2606
7.161.2.7 nppsExp_32s_Sfs	2606

7.161.2.8 nppsExp_64f	2606
7.161.2.9 nppsExp_64f_I	2607
7.161.2.10 nppsExp_64s_ISfs	2607
7.161.2.11 nppsExp_64s_Sfs	2607
7.162Ln	2608
7.162.1 Detailed Description	2608
7.162.2 Function Documentation	2608
7.162.2.1 nppsLn_16s_ISfs	2608
7.162.2.2 nppsLn_16s_Sfs	2609
7.162.2.3 nppsLn_32f	2609
7.162.2.4 nppsLn_32f_I	2609
7.162.2.5 nppsLn_32s16s_Sfs	2610
7.162.2.6 nppsLn_32s_ISfs	2610
7.162.2.7 nppsLn_32s_Sfs	2610
7.162.2.8 nppsLn_64f	2611
7.162.2.9 nppsLn_64f32f	2611
7.162.2.10 nppsLn_64f_I	2611
7.16310Log10	2612
7.163.1 Detailed Description	2612
7.163.2 Function Documentation	2612
7.163.2.1 npps10Log10_32s_ISfs	2612
7.163.2.2 npps10Log10_32s_Sfs	2612
7.164SumLn	2613
7.164.1 Detailed Description	2613
7.164.2 Function Documentation	2613
7.164.2.1 nppsSumLn_16s32f	2613
7.164.2.2 nppsSumLn_32f	2614
7.164.2.3 nppsSumLn_32f64f	2614
7.164.2.4 nppsSumLn_64f	2614
7.164.2.5 nppsSumLnGetBufferSize_16s32f	2615
7.164.2.6 nppsSumLnGetBufferSize_32f	2615
7.164.2.7 nppsSumLnGetBufferSize_32f64f	2615
7.164.2.8 nppsSumLnGetBufferSize_64f	2616
7.165Arctan	2617
7.165.1 Detailed Description	2617
7.165.2 Function Documentation	2617

7.165.2.1 nppsArctan_32f	2617
7.165.2.2 nppsArctan_32f_I	2617
7.165.2.3 nppsArctan_64f	2618
7.165.2.4 nppsArctan_64f_I	2618
7.166 Normalize	2619
7.166.1 Detailed Description	2619
7.166.2 Function Documentation	2619
7.166.2.1 nppsNormalize_16s_Sfs	2619
7.166.2.2 nppsNormalize_16sc_Sfs	2620
7.166.2.3 nppsNormalize_32f	2620
7.166.2.4 nppsNormalize_32fc	2620
7.166.2.5 nppsNormalize_64f	2621
7.166.2.6 nppsNormalize_64fc	2621
7.167 Cauchy, CauchyD, and CauchyDD2	2622
7.167.1 Detailed Description	2622
7.167.2 Function Documentation	2622
7.167.2.1 nppsCauchy_32f_I	2622
7.167.2.2 nppsCauchyD_32f_I	2622
7.167.2.3 nppsCauchyDD2_32f_I	2623
7.168 Logical And Shift Operations	2624
7.169 AndC	2625
7.169.1 Detailed Description	2625
7.169.2 Function Documentation	2625
7.169.2.1 nppsAndC_16u	2625
7.169.2.2 nppsAndC_16u_I	2626
7.169.2.3 nppsAndC_32u	2626
7.169.2.4 nppsAndC_32u_I	2626
7.169.2.5 nppsAndC_8u	2626
7.169.2.6 nppsAndC_8u_I	2627
7.170 And	2628
7.170.1 Detailed Description	2628
7.170.2 Function Documentation	2628
7.170.2.1 nppsAnd_16u	2628
7.170.2.2 nppsAnd_16u_I	2629
7.170.2.3 nppsAnd_32u	2629
7.170.2.4 nppsAnd_32u_I	2629

7.170.2.5 nppsAnd_8u	2629
7.170.2.6 nppsAnd_8u_I	2630
7.171OrC	2631
7.171.1 Detailed Description	2631
7.171.2 Function Documentation	2631
7.171.2.1 nppsOrC_16u	2631
7.171.2.2 nppsOrC_16u_I	2632
7.171.2.3 nppsOrC_32u	2632
7.171.2.4 nppsOrC_32u_I	2632
7.171.2.5 nppsOrC_8u	2632
7.171.2.6 nppsOrC_8u_I	2633
7.172Or	2634
7.172.1 Detailed Description	2634
7.172.2 Function Documentation	2634
7.172.2.1 nppsOr_16u	2634
7.172.2.2 nppsOr_16u_I	2635
7.172.2.3 nppsOr_32u	2635
7.172.2.4 nppsOr_32u_I	2635
7.172.2.5 nppsOr_8u	2635
7.172.2.6 nppsOr_8u_I	2636
7.173XorC	2637
7.173.1 Detailed Description	2637
7.173.2 Function Documentation	2637
7.173.2.1 nppsXorC_16u	2637
7.173.2.2 nppsXorC_16u_I	2638
7.173.2.3 nppsXorC_32u	2638
7.173.2.4 nppsXorC_32u_I	2638
7.173.2.5 nppsXorC_8u	2638
7.173.2.6 nppsXorC_8u_I	2639
7.174Xor	2640
7.174.1 Detailed Description	2640
7.174.2 Function Documentation	2640
7.174.2.1 nppsXor_16u	2640
7.174.2.2 nppsXor_16u_I	2641
7.174.2.3 nppsXor_32u	2641
7.174.2.4 nppsXor_32u_I	2641

7.174.2.5 nppsXor_8u	2641
7.174.2.6 nppsXor_8u_I	2642
7.175Not	2643
7.175.1 Detailed Description	2643
7.175.2 Function Documentation	2643
7.175.2.1 nppsNot_16u	2643
7.175.2.2 nppsNot_16u_I	2644
7.175.2.3 nppsNot_32u	2644
7.175.2.4 nppsNot_32u_I	2644
7.175.2.5 nppsNot_8u	2644
7.175.2.6 nppsNot_8u_I	2645
7.176LShiftC	2646
7.176.1 Detailed Description	2646
7.176.2 Function Documentation	2646
7.176.2.1 nppsLShiftC_16s	2646
7.176.2.2 nppsLShiftC_16s_I	2647
7.176.2.3 nppsLShiftC_16u	2647
7.176.2.4 nppsLShiftC_16u_I	2647
7.176.2.5 nppsLShiftC_32s	2648
7.176.2.6 nppsLShiftC_32s_I	2648
7.176.2.7 nppsLShiftC_32u	2648
7.176.2.8 nppsLShiftC_32u_I	2649
7.176.2.9 nppsLShiftC_8u	2649
7.176.2.10 nppsLShiftC_8u_I	2649
7.177RShiftC	2650
7.177.1 Detailed Description	2650
7.177.2 Function Documentation	2650
7.177.2.1 nppsRShiftC_16s	2650
7.177.2.2 nppsRShiftC_16s_I	2651
7.177.2.3 nppsRShiftC_16u	2651
7.177.2.4 nppsRShiftC_16u_I	2651
7.177.2.5 nppsRShiftC_32s	2652
7.177.2.6 nppsRShiftC_32s_I	2652
7.177.2.7 nppsRShiftC_32u	2652
7.177.2.8 nppsRShiftC_32u_I	2653
7.177.2.9 nppsRShiftC_8u	2653

7.177.2.10	nppsRShiftC_8u_I	2653
7.178	Conversion Functions	2654
7.179	Convert	2655
7.179.1	Function Documentation	2657
7.179.1.1	nppsConvert_16s32f	2657
7.179.1.2	nppsConvert_16s32f_Sfs	2657
7.179.1.3	nppsConvert_16s32s	2657
7.179.1.4	nppsConvert_16s64f_Sfs	2657
7.179.1.5	nppsConvert_16s8s_Sfs	2657
7.179.1.6	nppsConvert_16u32f	2657
7.179.1.7	nppsConvert_32f16s_Sfs	2657
7.179.1.8	nppsConvert_32f16u_Sfs	2657
7.179.1.9	nppsConvert_32f32s_Sfs	2657
7.179.1.10	nppsConvert_32f64f	2657
7.179.1.11	nppsConvert_32f8s_Sfs	2657
7.179.1.12	nppsConvert_32f8u_Sfs	2657
7.179.1.13	nppsConvert_32s16s	2657
7.179.1.14	nppsConvert_32s16s_Sfs	2657
7.179.1.15	nppsConvert_32s32f	2657
7.179.1.16	nppsConvert_32s32f_Sfs	2657
7.179.1.17	nppsConvert_32s64f	2657
7.179.1.18	nppsConvert_32s64f_Sfs	2657
7.179.1.19	nppsConvert_64f16s_Sfs	2657
7.179.1.20	nppsConvert_64f32f	2657
7.179.1.21	nppsConvert_64f32s_Sfs	2657
7.179.1.22	nppsConvert_64f64s_Sfs	2657
7.179.1.23	nppsConvert_64s32s_Sfs	2657
7.179.1.24	nppsConvert_64s64f	2657
7.179.1.25	nppsConvert_8s16s	2657
7.179.1.26	nppsConvert_8s32f	2657
7.179.1.27	nppsConvert_8u32f	2657
7.180	Threshold	2658
7.180.1	Function Documentation	2662
7.180.1.1	nppsThreshold_16s	2662
7.180.1.2	nppsThreshold_16s_I	2663
7.180.1.3	nppsThreshold_16sc	2663

7.180.1.4 nppsThreshold_16sc_I	2663
7.180.1.5 nppsThreshold_32f	2664
7.180.1.6 nppsThreshold_32f_I	2664
7.180.1.7 nppsThreshold_32fc	2664
7.180.1.8 nppsThreshold_32fc_I	2665
7.180.1.9 nppsThreshold_64f	2665
7.180.1.10 nppsThreshold_64f_I	2666
7.180.1.11 nppsThreshold_64fc	2666
7.180.1.12 nppsThreshold_64fc_I	2666
7.180.1.13 nppsThreshold_GT_16s	2667
7.180.1.14 nppsThreshold_GT_16s_I	2667
7.180.1.15 nppsThreshold_GT_16sc	2667
7.180.1.16 nppsThreshold_GT_16sc_I	2668
7.180.1.17 nppsThreshold_GT_32f	2668
7.180.1.18 nppsThreshold_GT_32f_I	2668
7.180.1.19 nppsThreshold_GT_32fc	2669
7.180.1.20 nppsThreshold_GT_32fc_I	2669
7.180.1.21 nppsThreshold_GT_64f	2669
7.180.1.22 nppsThreshold_GT_64f_I	2670
7.180.1.23 nppsThreshold_GT_64fc	2670
7.180.1.24 nppsThreshold_GT_64fc_I	2670
7.180.1.25 nppsThreshold_GTVal_16s	2671
7.180.1.26 nppsThreshold_GTVal_16s_I	2671
7.180.1.27 nppsThreshold_GTVal_16sc	2671
7.180.1.28 nppsThreshold_GTVal_16sc_I	2672
7.180.1.29 nppsThreshold_GTVal_32f	2672
7.180.1.30 nppsThreshold_GTVal_32f_I	2672
7.180.1.31 nppsThreshold_GTVal_32fc	2673
7.180.1.32 nppsThreshold_GTVal_32fc_I	2673
7.180.1.33 nppsThreshold_GTVal_64f	2673
7.180.1.34 nppsThreshold_GTVal_64f_I	2674
7.180.1.35 nppsThreshold_GTVal_64fc	2674
7.180.1.36 nppsThreshold_GTVal_64fc_I	2674
7.180.1.37 nppsThreshold_LT_16s	2675
7.180.1.38 nppsThreshold_LT_16s_I	2675
7.180.1.39 nppsThreshold_LT_16sc	2675

7.180.1.40	nppsThreshold_LT_16sc_I	2676
7.180.1.41	nppsThreshold_LT_32f	2676
7.180.1.42	nppsThreshold_LT_32f_I	2676
7.180.1.43	nppsThreshold_LT_32fc	2677
7.180.1.44	nppsThreshold_LT_32fc_I	2677
7.180.1.45	nppsThreshold_LT_64f	2677
7.180.1.46	nppsThreshold_LT_64f_I	2678
7.180.1.47	nppsThreshold_LT_64fc	2678
7.180.1.48	nppsThreshold_LT_64fc_I	2678
7.180.1.49	nppsThreshold_LTVVal_16s	2679
7.180.1.50	nppsThreshold_LTVVal_16s_I	2679
7.180.1.51	nppsThreshold_LTVVal_16sc	2679
7.180.1.52	nppsThreshold_LTVVal_16sc_I	2680
7.180.1.53	nppsThreshold_LTVVal_32f	2680
7.180.1.54	nppsThreshold_LTVVal_32f_I	2680
7.180.1.55	nppsThreshold_LTVVal_32fc	2681
7.180.1.56	nppsThreshold_LTVVal_32fc_I	2681
7.180.1.57	nppsThreshold_LTVVal_64f	2681
7.180.1.58	nppsThreshold_LTVVal_64f_I	2682
7.180.1.59	nppsThreshold_LTVVal_64fc	2682
7.180.1.60	nppsThreshold_LTVVal_64fc_I	2682
7.181	Filtering Functions	2683
7.181.1	Detailed Description	2683
7.182	Integral	2684
7.182.1	Detailed Description	2684
7.182.2	Function Documentation	2684
7.182.2.1	nppsIntegral_32s	2684
7.182.2.2	nppsIntegralGetBufferSize_32s	2684
7.183	Initialization	2685
7.184	Set	2686
7.184.1	Function Documentation	2687
7.184.1.1	nppsSet_16s	2687
7.184.1.2	nppsSet_16sc	2687
7.184.1.3	nppsSet_16u	2687
7.184.1.4	nppsSet_32f	2687
7.184.1.5	nppsSet_32fc	2688

7.184.1.6 nppsSet_32s	2688
7.184.1.7 nppsSet_32sc	2688
7.184.1.8 nppsSet_32u	2689
7.184.1.9 nppsSet_64f	2689
7.184.1.10nppsSet_64fc	2689
7.184.1.11nppsSet_64s	2689
7.184.1.12nppsSet_64sc	2690
7.184.1.13nppsSet_8s	2690
7.184.1.14nppsSet_8u	2690
7.185Zero	2691
7.185.1 Function Documentation	2691
7.185.1.1 nppsZero_16s	2691
7.185.1.2 nppsZero_16sc	2692
7.185.1.3 nppsZero_32f	2692
7.185.1.4 nppsZero_32fc	2692
7.185.1.5 nppsZero_32s	2692
7.185.1.6 nppsZero_32sc	2693
7.185.1.7 nppsZero_64f	2693
7.185.1.8 nppsZero_64fc	2693
7.185.1.9 nppsZero_64s	2693
7.185.1.10nppsZero_64sc	2694
7.185.1.11nppsZero_8u	2694
7.186Copy	2695
7.186.1 Function Documentation	2695
7.186.1.1 nppsCopy_16s	2695
7.186.1.2 nppsCopy_16sc	2696
7.186.1.3 nppsCopy_32f	2696
7.186.1.4 nppsCopy_32fc	2696
7.186.1.5 nppsCopy_32s	2697
7.186.1.6 nppsCopy_32sc	2697
7.186.1.7 nppsCopy_64fc	2697
7.186.1.8 nppsCopy_64s	2697
7.186.1.9 nppsCopy_64sc	2698
7.186.1.10nppsCopy_8u	2698
7.187Statistical Functions	2699
7.187.1 Detailed Description	2699

7.188MinEvery And MaxEvery Functions	2700
7.188.1 Detailed Description	2700
7.188.2 Function Documentation	2700
7.188.2.1 nppsMaxEvery_16s_I	2700
7.188.2.2 nppsMaxEvery_16u_I	2701
7.188.2.3 nppsMaxEvery_32f_I	2701
7.188.2.4 nppsMaxEvery_32s_I	2701
7.188.2.5 nppsMaxEvery_8u_I	2702
7.188.2.6 nppsMinEvery_16s_I	2702
7.188.2.7 nppsMinEvery_16u_I	2702
7.188.2.8 nppsMinEvery_32f_I	2702
7.188.2.9 nppsMinEvery_32s_I	2703
7.188.2.10nppsMinEvery_64f_I	2703
7.188.2.11nppsMinEvery_8u_I	2703
7.189Sum	2704
7.189.1 Detailed Description	2705
7.189.2 Function Documentation	2705
7.189.2.1 nppsSum_16s32s_Sfs	2705
7.189.2.2 nppsSum_16s_Sfs	2705
7.189.2.3 nppsSum_16sc32sc_Sfs	2706
7.189.2.4 nppsSum_16sc_Sfs	2706
7.189.2.5 nppsSum_32f	2707
7.189.2.6 nppsSum_32fc	2707
7.189.2.7 nppsSum_32s_Sfs	2707
7.189.2.8 nppsSum_64f	2708
7.189.2.9 nppsSum_64fc	2708
7.189.2.10nppsSumGetBufferSize_16s32s_Sfs	2708
7.189.2.11nppsSumGetBufferSize_16s_Sfs	2709
7.189.2.12nppsSumGetBufferSize_16sc32sc_Sfs	2709
7.189.2.13nppsSumGetBufferSize_16sc_Sfs	2709
7.189.2.14nppsSumGetBufferSize_32f	2709
7.189.2.15nppsSumGetBufferSize_32fc	2710
7.189.2.16nppsSumGetBufferSize_32s_Sfs	2710
7.189.2.17nppsSumGetBufferSize_64f	2710
7.189.2.18nppsSumGetBufferSize_64fc	2710
7.190Maximum	2711

7.190.1 Function Documentation	2712
7.190.1.1 nppsMax_16s	2712
7.190.1.2 nppsMax_32f	2713
7.190.1.3 nppsMax_32s	2713
7.190.1.4 nppsMax_64f	2713
7.190.1.5 nppsMaxAbs_16s	2714
7.190.1.6 nppsMaxAbs_32s	2714
7.190.1.7 nppsMaxAbsGetBufferSize_16s	2714
7.190.1.8 nppsMaxAbsGetBufferSize_32s	2715
7.190.1.9 nppsMaxAbsIndx_16s	2715
7.190.1.10 nppsMaxAbsIndx_32s	2715
7.190.1.11 nppsMaxAbsIndxGetBufferSize_16s	2716
7.190.1.12 nppsMaxAbsIndxGetBufferSize_32s	2716
7.190.1.13 nppsMaxGetBufferSize_16s	2716
7.190.1.14 nppsMaxGetBufferSize_32f	2717
7.190.1.15 nppsMaxGetBufferSize_32s	2717
7.190.1.16 nppsMaxGetBufferSize_64f	2717
7.190.1.17 nppsMaxIndx_16s	2717
7.190.1.18 nppsMaxIndx_32f	2718
7.190.1.19 nppsMaxIndx_32s	2718
7.190.1.20 nppsMaxIndx_64f	2719
7.190.1.21 nppsMaxIndxGetBufferSize_16s	2719
7.190.1.22 nppsMaxIndxGetBufferSize_32f	2719
7.190.1.23 nppsMaxIndxGetBufferSize_32s	2720
7.190.1.24 nppsMaxIndxGetBufferSize_64f	2720
7.191 Minimum	2721
7.191.1 Function Documentation	2722
7.191.1.1 nppsMin_16s	2722
7.191.1.2 nppsMin_32f	2723
7.191.1.3 nppsMin_32s	2723
7.191.1.4 nppsMin_64f	2723
7.191.1.5 nppsMinAbs_16s	2724
7.191.1.6 nppsMinAbs_32s	2724
7.191.1.7 nppsMinAbsGetBufferSize_16s	2724
7.191.1.8 nppsMinAbsGetBufferSize_32s	2725
7.191.1.9 nppsMinAbsIndx_16s	2725

7.191.1.10	nppsMinAbsIndx_32s	2725
7.191.1.11	nppsMinAbsIndxGetBufferSize_16s	2726
7.191.1.12	nppsMinAbsIndxGetBufferSize_32s	2726
7.191.1.13	nppsMinGetBufferSize_16s	2726
7.191.1.14	nppsMinGetBufferSize_32f	2727
7.191.1.15	nppsMinGetBufferSize_32s	2727
7.191.1.16	nppsMinGetBufferSize_64f	2727
7.191.1.17	nppsMinIndx_16s	2727
7.191.1.18	nppsMinIndx_32f	2728
7.191.1.19	nppsMinIndx_32s	2728
7.191.1.20	nppsMinIndx_64f	2729
7.191.1.21	nppsMinIndxGetBufferSize_16s	2729
7.191.1.22	nppsMinIndxGetBufferSize_32f	2729
7.191.1.23	nppsMinIndxGetBufferSize_32s	2730
7.191.1.24	nppsMinIndxGetBufferSize_64f	2730
7.192	Mean	2731
7.192.1	Function Documentation	2732
7.192.1.1	nppsMean_16s_Sfs	2732
7.192.1.2	nppsMean_16sc_Sfs	2732
7.192.1.3	nppsMean_32f	2732
7.192.1.4	nppsMean_32fc	2733
7.192.1.5	nppsMean_32s_Sfs	2733
7.192.1.6	nppsMean_64f	2734
7.192.1.7	nppsMean_64fc	2734
7.192.1.8	nppsMeanGetBufferSize_16s_Sfs	2734
7.192.1.9	nppsMeanGetBufferSize_16sc_Sfs	2735
7.192.1.10	nppsMeanGetBufferSize_32f	2735
7.192.1.11	nppsMeanGetBufferSize_32fc	2735
7.192.1.12	nppsMeanGetBufferSize_32s_Sfs	2735
7.192.1.13	nppsMeanGetBufferSize_64f	2736
7.192.1.14	nppsMeanGetBufferSize_64fc	2736
7.193	Standard Deviation	2737
7.193.1	Function Documentation	2737
7.193.1.1	nppsStdDev_16s32s_Sfs	2737
7.193.1.2	nppsStdDev_16s_Sfs	2738
7.193.1.3	nppsStdDev_32f	2738

7.193.1.4 nppsStdDev_64f	2738
7.193.1.5 nppsStdDevGetBufferSize_16s32s_Sfs	2739
7.193.1.6 nppsStdDevGetBufferSize_16s_Sfs	2739
7.193.1.7 nppsStdDevGetBufferSize_32f	2739
7.193.1.8 nppsStdDevGetBufferSize_64f	2739
7.194 Mean And Standard Deviation	2740
7.194.1 Function Documentation	2740
7.194.1.1 nppsMeanStdDev_16s32s_Sfs	2740
7.194.1.2 nppsMeanStdDev_16s_Sfs	2741
7.194.1.3 nppsMeanStdDev_32f	2741
7.194.1.4 nppsMeanStdDev_64f	2741
7.194.1.5 nppsMeanStdDevGetBufferSize_16s32s_Sfs	2742
7.194.1.6 nppsMeanStdDevGetBufferSize_16s_Sfs	2742
7.194.1.7 nppsMeanStdDevGetBufferSize_32f	2742
7.194.1.8 nppsMeanStdDevGetBufferSize_64f	2743
7.195 Minimum_Maximum	2744
7.195.1 Function Documentation	2746
7.195.1.1 nppsMinMax_16s	2746
7.195.1.2 nppsMinMax_16u	2746
7.195.1.3 nppsMinMax_32f	2746
7.195.1.4 nppsMinMax_32s	2747
7.195.1.5 nppsMinMax_32u	2747
7.195.1.6 nppsMinMax_64f	2747
7.195.1.7 nppsMinMax_8u	2748
7.195.1.8 nppsMinMaxGetBufferSize_16s	2748
7.195.1.9 nppsMinMaxGetBufferSize_16u	2748
7.195.1.10 nppsMinMaxGetBufferSize_32f	2749
7.195.1.11 nppsMinMaxGetBufferSize_32s	2749
7.195.1.12 nppsMinMaxGetBufferSize_32u	2749
7.195.1.13 nppsMinMaxGetBufferSize_64f	2750
7.195.1.14 nppsMinMaxGetBufferSize_8u	2750
7.195.1.15 nppsMinMaxIndx_16s	2750
7.195.1.16 nppsMinMaxIndx_16u	2751
7.195.1.17 nppsMinMaxIndx_32f	2751
7.195.1.18 nppsMinMaxIndx_32s	2751
7.195.1.19 nppsMinMaxIndx_32u	2752

7.195.1.20	nppsMinMaxIndx_64f	2752
7.195.1.21	nppsMinMaxIndx_8u	2753
7.195.1.22	nppsMinMaxIndxGetBufferSize_16s	2753
7.195.1.23	nppsMinMaxIndxGetBufferSize_16u	2753
7.195.1.24	nppsMinMaxIndxGetBufferSize_32f	2754
7.195.1.25	nppsMinMaxIndxGetBufferSize_32s	2754
7.195.1.26	nppsMinMaxIndxGetBufferSize_32u	2754
7.195.1.27	nppsMinMaxIndxGetBufferSize_64f	2754
7.195.1.28	nppsMinMaxIndxGetBufferSize_8u	2755
7.196	Infinity Norm	2756
7.196.1	Function Documentation	2757
7.196.1.1	nppsNorm_Inf_16s32f	2757
7.196.1.2	nppsNorm_Inf_16s32s_Sfs	2757
7.196.1.3	nppsNorm_Inf_32f	2757
7.196.1.4	nppsNorm_Inf_32fc32f	2758
7.196.1.5	nppsNorm_Inf_64f	2758
7.196.1.6	nppsNorm_Inf_64fc64f	2758
7.196.1.7	nppsNormInfGetBufferSize_16s32f	2759
7.196.1.8	nppsNormInfGetBufferSize_16s32s_Sfs	2759
7.196.1.9	nppsNormInfGetBufferSize_32f	2759
7.196.1.10	nppsNormInfGetBufferSize_32fc32f	2759
7.196.1.11	nppsNormInfGetBufferSize_64f	2760
7.196.1.12	nppsNormInfGetBufferSize_64fc64f	2760
7.197	L1 Norm	2761
7.197.1	Function Documentation	2762
7.197.1.1	nppsNorm_L1_16s32f	2762
7.197.1.2	nppsNorm_L1_16s32s_Sfs	2762
7.197.1.3	nppsNorm_L1_16s64s_Sfs	2762
7.197.1.4	nppsNorm_L1_32f	2763
7.197.1.5	nppsNorm_L1_32fc64f	2763
7.197.1.6	nppsNorm_L1_64f	2763
7.197.1.7	nppsNorm_L1_64fc64f	2764
7.197.1.8	nppsNormL1GetBufferSize_16s32f	2764
7.197.1.9	nppsNormL1GetBufferSize_16s32s_Sfs	2764
7.197.1.10	nppsNormL1GetBufferSize_16s64s_Sfs	2765
7.197.1.11	nppsNormL1GetBufferSize_32f	2765

7.197.1.12	nppsNormL1GetBufferSize_32fc64f	2765
7.197.1.13	nppsNormL1GetBufferSize_64f	2765
7.197.1.14	nppsNormL1GetBufferSize_64fc64f	2766
7.198	L2 Norm	2767
7.198.1	Function Documentation	2768
7.198.1.1	nppsNorm_L2_16s32f	2768
7.198.1.2	nppsNorm_L2_16s32s_Sfs	2768
7.198.1.3	nppsNorm_L2_32f	2768
7.198.1.4	nppsNorm_L2_32fc64f	2769
7.198.1.5	nppsNorm_L2_64f	2769
7.198.1.6	nppsNorm_L2_64fc64f	2769
7.198.1.7	nppsNorm_L2Sqr_16s64s_Sfs	2770
7.198.1.8	nppsNormL2GetBufferSize_16s32f	2770
7.198.1.9	nppsNormL2GetBufferSize_16s32s_Sfs	2770
7.198.1.10	nppsNormL2GetBufferSize_32f	2771
7.198.1.11	nppsNormL2GetBufferSize_32fc64f	2771
7.198.1.12	nppsNormL2GetBufferSize_64f	2771
7.198.1.13	nppsNormL2GetBufferSize_64fc64f	2771
7.198.1.14	nppsNormL2SqrGetBufferSize_16s64s_Sfs	2772
7.199	Infinity Norm Diff	2773
7.199.1	Function Documentation	2774
7.199.1.1	nppsNormDiff_Inf_16s32f	2774
7.199.1.2	nppsNormDiff_Inf_16s32s_Sfs	2774
7.199.1.3	nppsNormDiff_Inf_32f	2774
7.199.1.4	nppsNormDiff_Inf_32fc32f	2775
7.199.1.5	nppsNormDiff_Inf_64f	2775
7.199.1.6	nppsNormDiff_Inf_64fc64f	2776
7.199.1.7	nppsNormDiffInfGetBufferSize_16s32f	2776
7.199.1.8	nppsNormDiffInfGetBufferSize_16s32s_Sfs	2776
7.199.1.9	nppsNormDiffInfGetBufferSize_32f	2776
7.199.1.10	nppsNormDiffInfGetBufferSize_32fc32f	2777
7.199.1.11	nppsNormDiffInfGetBufferSize_64f	2777
7.199.1.12	nppsNormDiffInfGetBufferSize_64fc64f	2777
7.200	L1 Norm Diff	2778
7.200.1	Function Documentation	2779
7.200.1.1	nppsNormDiff_L1_16s32f	2779

7.200.1.2	nppsNormDiff_L1_16s32s_Sfs	2779
7.200.1.3	nppsNormDiff_L1_16s64s_Sfs	2779
7.200.1.4	nppsNormDiff_L1_32f	2780
7.200.1.5	nppsNormDiff_L1_32fc64f	2780
7.200.1.6	nppsNormDiff_L1_64f	2781
7.200.1.7	nppsNormDiff_L1_64fc64f	2781
7.200.1.8	nppsNormDiffL1GetBufferSize_16s32f	2781
7.200.1.9	nppsNormDiffL1GetBufferSize_16s32s_Sfs	2782
7.200.1.10	nppsNormDiffL1GetBufferSize_16s64s_Sfs	2782
7.200.1.11	nppsNormDiffL1GetBufferSize_32f	2782
7.200.1.12	nppsNormDiffL1GetBufferSize_32fc64f	2782
7.200.1.13	nppsNormDiffL1GetBufferSize_64f	2783
7.200.1.14	nppsNormDiffL1GetBufferSize_64fc64f	2783
7.201	L2 Norm Diff	2784
7.201.1	Function Documentation	2785
7.201.1.1	nppsNormDiff_L2_16s32f	2785
7.201.1.2	nppsNormDiff_L2_16s32s_Sfs	2785
7.201.1.3	nppsNormDiff_L2_32f	2785
7.201.1.4	nppsNormDiff_L2_32fc64f	2786
7.201.1.5	nppsNormDiff_L2_64f	2786
7.201.1.6	nppsNormDiff_L2_64fc64f	2787
7.201.1.7	nppsNormDiff_L2Sqr_16s64s_Sfs	2787
7.201.1.8	nppsNormDiffL2GetBufferSize_16s32f	2787
7.201.1.9	nppsNormDiffL2GetBufferSize_16s32s_Sfs	2788
7.201.1.10	nppsNormDiffL2GetBufferSize_32f	2788
7.201.1.11	nppsNormDiffL2GetBufferSize_32fc64f	2788
7.201.1.12	nppsNormDiffL2GetBufferSize_64f	2788
7.201.1.13	nppsNormDiffL2GetBufferSize_64fc64f	2789
7.201.1.14	nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs	2789
7.202	Dot Product	2790
7.202.1	Function Documentation	2793
7.202.1.1	nppsDotProd_16s16sc32fc	2793
7.202.1.2	nppsDotProd_16s16sc32sc_Sfs	2794
7.202.1.3	nppsDotProd_16s16sc64sc	2794
7.202.1.4	nppsDotProd_16s16sc_Sfs	2795
7.202.1.5	nppsDotProd_16s32f	2795

7.202.1.6 nppsDotProd_16s32s32s_Sfs	2795
7.202.1.7 nppsDotProd_16s32s_Sfs	2796
7.202.1.8 nppsDotProd_16s64s	2796
7.202.1.9 nppsDotProd_16s_Sfs	2797
7.202.1.10 nppsDotProd_16sc32fc	2797
7.202.1.11 nppsDotProd_16sc32sc_Sfs	2797
7.202.1.12 nppsDotProd_16sc64sc	2798
7.202.1.13 nppsDotProd_16sc_Sfs	2798
7.202.1.14 nppsDotProd_32f	2799
7.202.1.15 nppsDotProd_32f32fc	2799
7.202.1.16 nppsDotProd_32f32fc64fc	2799
7.202.1.17 nppsDotProd_32f64f	2800
7.202.1.18 nppsDotProd_32fc	2800
7.202.1.19 nppsDotProd_32fc64fc	2800
7.202.1.20 nppsDotProd_32s32sc_Sfs	2801
7.202.1.21 nppsDotProd_32s_Sfs	2801
7.202.1.22 nppsDotProd_32sc_Sfs	2801
7.202.1.23 nppsDotProd_64f	2802
7.202.1.24 nppsDotProd_64f64fc	2802
7.202.1.25 nppsDotProd_64fc	2803
7.202.1.26 nppsDotProdGetBufferSize_16s16sc32fc	2803
7.202.1.27 nppsDotProdGetBufferSize_16s16sc32sc_Sfs	2803
7.202.1.28 nppsDotProdGetBufferSize_16s16sc64sc	2803
7.202.1.29 nppsDotProdGetBufferSize_16s16sc_Sfs	2804
7.202.1.30 nppsDotProdGetBufferSize_16s32f	2804
7.202.1.31 nppsDotProdGetBufferSize_16s32s32s_Sfs	2804
7.202.1.32 nppsDotProdGetBufferSize_16s32s_Sfs	2805
7.202.1.33 nppsDotProdGetBufferSize_16s64s	2805
7.202.1.34 nppsDotProdGetBufferSize_16s_Sfs	2805
7.202.1.35 nppsDotProdGetBufferSize_16sc32fc	2805
7.202.1.36 nppsDotProdGetBufferSize_16sc32sc_Sfs	2806
7.202.1.37 nppsDotProdGetBufferSize_16sc64sc	2806
7.202.1.38 nppsDotProdGetBufferSize_16sc_Sfs	2806
7.202.1.39 nppsDotProdGetBufferSize_32f	2806
7.202.1.40 nppsDotProdGetBufferSize_32f32fc	2807
7.202.1.41 nppsDotProdGetBufferSize_32f32fc64fc	2807

7.202.1.42	nppsDotProdGetBufferSize_32f64f	2807
7.202.1.43	nppsDotProdGetBufferSize_32fc	2807
7.202.1.44	nppsDotProdGetBufferSize_32fc64fc	2808
7.202.1.45	nppsDotProdGetBufferSize_32s32sc_Sfs	2808
7.202.1.46	nppsDotProdGetBufferSize_32s_Sfs	2808
7.202.1.47	nppsDotProdGetBufferSize_32sc_Sfs	2808
7.202.1.48	nppsDotProdGetBufferSize_64f	2809
7.202.1.49	nppsDotProdGetBufferSize_64f64fc	2809
7.202.1.50	nppsDotProdGetBufferSize_64fc	2809
7.203	Count In Range	2810
7.203.1	Function Documentation	2810
7.203.1.1	nppsCountInRange_32s	2810
7.203.1.2	nppsCountInRangeGetBufferSize_32s	2810
7.204	Count Zero Crossings	2811
7.204.1	Function Documentation	2811
7.204.1.1	nppsZeroCrossing_16s32f	2811
7.204.1.2	nppsZeroCrossing_32f	2811
7.204.1.3	nppsZeroCrossingGetBufferSize_16s32f	2812
7.204.1.4	nppsZeroCrossingGetBufferSize_32f	2812
7.205	MaximumError	2813
7.205.1	Detailed Description	2815
7.205.2	Function Documentation	2815
7.205.2.1	nppsMaximumError_16s	2815
7.205.2.2	nppsMaximumError_16sc	2815
7.205.2.3	nppsMaximumError_16u	2816
7.205.2.4	nppsMaximumError_32f	2816
7.205.2.5	nppsMaximumError_32fc	2816
7.205.2.6	nppsMaximumError_32s	2817
7.205.2.7	nppsMaximumError_32sc	2817
7.205.2.8	nppsMaximumError_32u	2817
7.205.2.9	nppsMaximumError_64f	2818
7.205.2.10	nppsMaximumError_64fc	2818
7.205.2.11	nppsMaximumError_64s	2818
7.205.2.12	nppsMaximumError_64sc	2819
7.205.2.13	nppsMaximumError_8s	2819
7.205.2.14	nppsMaximumError_8u	2819

7.205.2.15	nppsMaximumErrorGetBufferSize_16s	2820
7.205.2.16	nppsMaximumErrorGetBufferSize_16sc	2820
7.205.2.17	nppsMaximumErrorGetBufferSize_16u	2820
7.205.2.18	nppsMaximumErrorGetBufferSize_32f	2820
7.205.2.19	nppsMaximumErrorGetBufferSize_32fc	2821
7.205.2.20	nppsMaximumErrorGetBufferSize_32s	2821
7.205.2.21	nppsMaximumErrorGetBufferSize_32sc	2821
7.205.2.22	nppsMaximumErrorGetBufferSize_32u	2821
7.205.2.23	nppsMaximumErrorGetBufferSize_64f	2822
7.205.2.24	nppsMaximumErrorGetBufferSize_64fc	2822
7.205.2.25	nppsMaximumErrorGetBufferSize_64s	2822
7.205.2.26	nppsMaximumErrorGetBufferSize_64sc	2822
7.205.2.27	nppsMaximumErrorGetBufferSize_8s	2823
7.205.2.28	nppsMaximumErrorGetBufferSize_8u	2823
7.206	AverageError	2824
7.206.1	Detailed Description	2826
7.206.2	Function Documentation	2826
7.206.2.1	nppsAverageError_16s	2826
7.206.2.2	nppsAverageError_16sc	2826
7.206.2.3	nppsAverageError_16u	2827
7.206.2.4	nppsAverageError_32f	2827
7.206.2.5	nppsAverageError_32fc	2827
7.206.2.6	nppsAverageError_32s	2828
7.206.2.7	nppsAverageError_32sc	2828
7.206.2.8	nppsAverageError_32u	2828
7.206.2.9	nppsAverageError_64f	2829
7.206.2.10	nppsAverageError_64fc	2829
7.206.2.11	nppsAverageError_64s	2829
7.206.2.12	nppsAverageError_64sc	2830
7.206.2.13	nppsAverageError_8s	2830
7.206.2.14	nppsAverageError_8u	2830
7.206.2.15	nppsAverageErrorGetBufferSize_16s	2831
7.206.2.16	nppsAverageErrorGetBufferSize_16sc	2831
7.206.2.17	nppsAverageErrorGetBufferSize_16u	2831
7.206.2.18	nppsAverageErrorGetBufferSize_32f	2831
7.206.2.19	nppsAverageErrorGetBufferSize_32fc	2832

7.206.2.20	nppsAverageErrorGetBufferSize_32s	2832
7.206.2.21	nppsAverageErrorGetBufferSize_32sc	2832
7.206.2.22	nppsAverageErrorGetBufferSize_32u	2832
7.206.2.23	nppsAverageErrorGetBufferSize_64f	2833
7.206.2.24	nppsAverageErrorGetBufferSize_64fc	2833
7.206.2.25	nppsAverageErrorGetBufferSize_64s	2833
7.206.2.26	nppsAverageErrorGetBufferSize_64sc	2833
7.206.2.27	nppsAverageErrorGetBufferSize_8s	2834
7.206.2.28	nppsAverageErrorGetBufferSize_8u	2834
7.207	MaximumRelativeError	2835
7.207.1	Detailed Description	2837
7.207.2	Function Documentation	2837
7.207.2.1	nppsMaximumRelativeError_16s	2837
7.207.2.2	nppsMaximumRelativeError_16sc	2837
7.207.2.3	nppsMaximumRelativeError_16u	2838
7.207.2.4	nppsMaximumRelativeError_32f	2838
7.207.2.5	nppsMaximumRelativeError_32fc	2839
7.207.2.6	nppsMaximumRelativeError_32s	2839
7.207.2.7	nppsMaximumRelativeError_32sc	2839
7.207.2.8	nppsMaximumRelativeError_32u	2840
7.207.2.9	nppsMaximumRelativeError_64f	2840
7.207.2.10	nppsMaximumRelativeError_64fc	2841
7.207.2.11	nppsMaximumRelativeError_64s	2841
7.207.2.12	nppsMaximumRelativeError_64sc	2841
7.207.2.13	nppsMaximumRelativeError_8s	2842
7.207.2.14	nppsMaximumRelativeError_8u	2842
7.207.2.15	nppsMaximumRelativeErrorGetBufferSize_16s	2843
7.207.2.16	nppsMaximumRelativeErrorGetBufferSize_16sc	2843
7.207.2.17	nppsMaximumRelativeErrorGetBufferSize_16u	2843
7.207.2.18	nppsMaximumRelativeErrorGetBufferSize_32f	2843
7.207.2.19	nppsMaximumRelativeErrorGetBufferSize_32fc	2844
7.207.2.20	nppsMaximumRelativeErrorGetBufferSize_32s	2844
7.207.2.21	nppsMaximumRelativeErrorGetBufferSize_32sc	2844
7.207.2.22	nppsMaximumRelativeErrorGetBufferSize_32u	2844
7.207.2.23	nppsMaximumRelativeErrorGetBufferSize_64f	2845
7.207.2.24	nppsMaximumRelativeErrorGetBufferSize_64fc	2845

7.207.2.25	nppsMaximumRelativeErrorGetBufferSize_64s	2845
7.207.2.26	nppsMaximumRelativeErrorGetBufferSize_64sc	2845
7.207.2.27	nppsMaximumRelativeErrorGetBufferSize_8s	2846
7.207.2.28	nppsMaximumRelativeErrorGetBufferSize_8u	2846
7.208	AverageRelativeError	2847
7.208.1	Detailed Description	2849
7.208.2	Function Documentation	2849
7.208.2.1	nppsAverageRelativeError_16s	2849
7.208.2.2	nppsAverageRelativeError_16sc	2849
7.208.2.3	nppsAverageRelativeError_16u	2850
7.208.2.4	nppsAverageRelativeError_32f	2850
7.208.2.5	nppsAverageRelativeError_32fc	2851
7.208.2.6	nppsAverageRelativeError_32s	2851
7.208.2.7	nppsAverageRelativeError_32sc	2851
7.208.2.8	nppsAverageRelativeError_32u	2852
7.208.2.9	nppsAverageRelativeError_64f	2852
7.208.2.10	nppsAverageRelativeError_64fc	2853
7.208.2.11	nppsAverageRelativeError_64s	2853
7.208.2.12	nppsAverageRelativeError_64sc	2853
7.208.2.13	nppsAverageRelativeError_8s	2854
7.208.2.14	nppsAverageRelativeError_8u	2854
7.208.2.15	nppsAverageRelativeErrorGetBufferSize_16s	2855
7.208.2.16	nppsAverageRelativeErrorGetBufferSize_16sc	2855
7.208.2.17	nppsAverageRelativeErrorGetBufferSize_16u	2855
7.208.2.18	nppsAverageRelativeErrorGetBufferSize_32f	2855
7.208.2.19	nppsAverageRelativeErrorGetBufferSize_32fc	2856
7.208.2.20	nppsAverageRelativeErrorGetBufferSize_32s	2856
7.208.2.21	nppsAverageRelativeErrorGetBufferSize_32sc	2856
7.208.2.22	nppsAverageRelativeErrorGetBufferSize_32u	2856
7.208.2.23	nppsAverageRelativeErrorGetBufferSize_64f	2857
7.208.2.24	nppsAverageRelativeErrorGetBufferSize_64fc	2857
7.208.2.25	nppsAverageRelativeErrorGetBufferSize_64s	2857
7.208.2.26	nppsAverageRelativeErrorGetBufferSize_64sc	2857
7.208.2.27	nppsAverageRelativeErrorGetBufferSize_8s	2858
7.208.2.28	nppsAverageRelativeErrorGetBufferSize_8u	2858
7.209	Memory Management	2859

7.210Malloc	2860
7.210.1 Detailed Description	2861
7.210.2 Function Documentation	2861
7.210.2.1 nppsMalloc_16s	2861
7.210.2.2 nppsMalloc_16sc	2861
7.210.2.3 nppsMalloc_16u	2861
7.210.2.4 nppsMalloc_32f	2862
7.210.2.5 nppsMalloc_32fc	2862
7.210.2.6 nppsMalloc_32s	2862
7.210.2.7 nppsMalloc_32sc	2862
7.210.2.8 nppsMalloc_32u	2863
7.210.2.9 nppsMalloc_64f	2863
7.210.2.10 nppsMalloc_64fc	2863
7.210.2.11 nppsMalloc_64s	2863
7.210.2.12 nppsMalloc_64sc	2864
7.210.2.13 nppsMalloc_8s	2864
7.210.2.14 nppsMalloc_8u	2864
7.211Free	2865
7.211.1 Detailed Description	2865
7.211.2 Function Documentation	2865
7.211.2.1 nppsFree	2865
8 Data Structure Documentation	2867
8.1 NPP_ALIGN_16 Struct Reference	2867
8.1.1 Detailed Description	2867
8.1.2 Field Documentation	2867
8.1.2.1 im	2867
8.1.2.2 im	2868
8.1.2.3 re	2868
8.1.2.4 re	2868
8.2 NPP_ALIGN_8 Struct Reference	2869
8.2.1 Detailed Description	2869
8.2.2 Field Documentation	2869
8.2.2.1 im	2869
8.2.2.2 im	2869
8.2.2.3 im	2869
8.2.2.4 re	2870

8.2.2.5	re	2870
8.2.2.6	re	2870
8.3	NppiHaarBuffer Struct Reference	2871
8.3.1	Field Documentation	2871
8.3.1.1	haarBuffer	2871
8.3.1.2	haarBufferSize	2871
8.4	NppiHaarClassifier_32f Struct Reference	2872
8.4.1	Field Documentation	2872
8.4.1.1	classifiers	2872
8.4.1.2	classifierSize	2872
8.4.1.3	classifierStep	2872
8.4.1.4	counterDevice	2872
8.4.1.5	numClassifiers	2872
8.5	NppiPoint Struct Reference	2873
8.5.1	Detailed Description	2873
8.5.2	Field Documentation	2873
8.5.2.1	x	2873
8.5.2.2	y	2873
8.6	NppiRect Struct Reference	2874
8.6.1	Detailed Description	2874
8.6.2	Field Documentation	2874
8.6.2.1	height	2874
8.6.2.2	width	2874
8.6.2.3	x	2874
8.6.2.4	y	2874
8.7	NppiSize Struct Reference	2875
8.7.1	Detailed Description	2875
8.7.2	Field Documentation	2875
8.7.2.1	height	2875
8.7.2.2	width	2875
8.8	NppLibraryVersion Struct Reference	2876
8.8.1	Field Documentation	2876
8.8.1.1	build	2876
8.8.1.2	major	2876
8.8.1.3	minor	2876

Chapter 1

NVIDIA Performance Primitives

IMPORTANT SPECIAL NOTICE IMPORTANT SPECIAL NOTICE IMPORTANT SPECIAL NOTICE

Note: Starting with release 6.5, NPP is also provided as a static library (libnppc_static.a, libnppi_static.a, and libnpps_static.a) on Linux and Mac OSes in addition to being provided as a shared library. The static NPP libraries depend on a common thread abstraction layer library called cuLIBOS (libculibos.a) that is now distributed as part of the toolkit. Consequently, cuLIBOS must be provided to the linker when the static library is being linked against.

For example, on Linux, to compile a small application foo using NPP against the dynamic library, the following command can be used:

```
nvcc foo.c -lnppi -o foo
```

Whereas to compile against the static NPP library, the following command has to be used:

```
nvcc foo.c -lnppi_static -lculibos -o foo
```

It is also possible to use the native host C++ compiler. Depending on the host operating system, some additional libraries like pthread or dl might be needed on the linking line. The following command on Linux is suggested:

```
g++ foo.c -lnppi_static -lculibos -lcudart_static -lpthread -ldl  
-I <cuda-toolkit-path>/include -L <cuda-toolkit-path>/lib64 -o foo
```

NPP is a stateless API, as of NPP 6.5 the ONLY state that NPP remembers between function calls is the stream ID associated with each CPU thread that creates NPP streams. If an application intends to use NPP with multiple host threads then it is the responsibility of the application to call nppSetStream from each CPU thread to create an association between that thread and that stream within NPP. Earlier versions of NPP required a CPU thread mutex around the nppSetStream call and the one or more NPP function calls that followed because NPP only remembered the current stream ID (the one most recently set by an nppSetStream call) between NPP function calls. All NPP functions should be thread safe except for the following functions:

nppiGraphcut_32s8u - this function will be deprecated in a future release

nppiGraphcut_32f8u - this function will be deprecated in a future release

nppiGraphcut8_32s8u - this function will be deprecated in a future release

```
nppiGraphcut8_32f8u - this function will be deprecated in a future release  
  
nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R  
  
nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R
```

As of NPP version 5.0 and beyond a few parameters for a few pre-5.0 existing image LUT functions have changed from host memory pointers to device memory pointers. Your application will fail (crash or report an error) if you use these functions with host memory pointers. The functions are the `nppiLUT_Linear_8u_xxx` functions.

Also, pre-5.0 function `nppiMeanStdDev8uC1RGetBufferHostSize` has been renamed `nppiMeanStdDevGetBufferHostSize_8u_C1R`.

1.1 What is NPP?

NVIDIA NPP is a library of functions for performing CUDA accelerated processing. The initial set of functionality in the library focuses on imaging and video processing and is widely applicable for developers in these areas. NPP will evolve over time to encompass more of the compute heavy tasks in a variety of problem domains. The NPP library is written to maximize flexibility, while maintaining high performance.

NPP can be used in one of two ways:

- A stand-alone library for adding GPU acceleration to an application with minimal effort. Using this route allows developers to add GPU acceleration to their applications in a matter of hours.
- A cooperative library for interoperating with a developer's GPU code efficiently.

Either route allows developers to harness the massive compute resources of NVIDIA GPUs, while simultaneously reducing development times.

1.2 Documentation

- [General API Conventions](#)
- [Signal-Processing Specific API Conventions](#)
- [Imaging-Processing Specific API Conventions](#)

1.3 Technical Specifications

Supported Platforms:

- Microsoft Windows 7 and 8 (64-bit and 32-bit)
- Microsoft Windows Vista (64-bit and 32-bit)
- Linux (Centos & Ubuntu) (64-bit and 32-bit)
- Mac OS X (64-bit)
- Android on Arm V7

1.4 Files

NPP is comprised of the following files:

1.4.1 Header Files

- [nppdefs.h](#)
- [nppcore.h](#)
- [nppi.h](#)
- [npps.h](#)
- [nppversion.h](#)
- [npp.h](#)

All those header files are located in the CUDA Toolkit's

`/include/`

directory.

1.4.2 Library Files

Starting with Version 5.5 NPP's functionality is now split up into 3 distinct libraries:

- A core library (NPPC) containing basic functionality from the [npp.h](#) header files as well as functionality shared by the other two libraries.
- The image processing library NPPI. Any functions from the [nppi.h](#) header file (or the various header files named "nppi_xxx.h" are bundled into the NPPI library.
- The signal processing library NPPS. Any function from the [npps.h](#) header file (or the various header files named "npps_xxx.h" are bundled into the NPPS library.

On the Windows platform the NPP stub libraries are found in the CUDA Toolkit's library directory:

`/lib/nppc.lib`

`/lib/nppi.lib`

`/lib/npps.lib`

The matching DLLs are located in the CUDA Toolkit's binary directory. Example

`/bin/nppi64_55_<build_no>.dll` `// Dynamic image-processing library for 64-bit Windows.`

On Linux and Mac platforms the dynamic libraries are located in the lib directory

`/lib/libnppc32.so.5.5.<build_no>` `// NPP 32-bit dynamic core library for Linux`

`/lib/libnpps32.5.5.dylib` `// NPP 32-bit dynamic signal processing library for Mac`

1.5 Supported NVIDIA Hardware

NPP runs on all CUDA capable NVIDIA hardware. For details please see http://www.nvidia.com/object/cuda_learn_products.html

Chapter 2

General API Conventions

2.1 Memory Management

The design of all the NPP functions follows the same guidelines as other NVIDIA CUDA libraries like cuFFT and cuBLAS. That is that all pointer arguments in those APIs are device pointers.

This convention enables the individual developer to make smart choices about memory management that minimize the number of memory transfers. It also allows the user the maximum flexibility regarding which of the various memory transfer mechanisms offered by the CUDA runtime is used, e.g. synchronous or asynchronous memory transfers, zero-copy and pinned memory, etc.

The most basic steps involved in using NPP for processing data is as follows:

1. Transfer input data from the host to device using

```
cudaMemcpy(...)
```

2. Process data using one or several NPP functions or custom CUDA kernels

3. Transfer the result data from the device to the host using

```
cudaMemcpy(...)
```

2.1.1 Scratch Buffer and Host Pointer

Some primitives of NPP require additional device memory buffers (scratch buffers) for calculations, e.g. signal and image reductions (Sum, Max, Min, MinMax, etc.). In order to give the NPP user maximum control regarding memory allocations and performance, it is the user's responsibility to allocate and delete those temporary buffers. For one this has the benefit that the library will not allocate memory unbeknownst to the user. It also allows developers who invoke the same primitive repeatedly to allocate the scratch only once, improving performance and potential device-memory fragmentation.

Scratch-buffer memory is unstructured and may be passed to the primitive in uninitialized form. This allows for reuse of the same scratch buffers with any primitive require scratch memory, as long as it is sufficiently sized.

The minimum scratch-buffer size for a given primitive (e.g. `nppsSum_32f()`) can be obtained by a companion function (e.g. `nppsSumGetBufferSize_32f()`). The buffer size is returned via a host pointer as allocation of the scratch-buffer is performed via CUDA runtime host code.

An example to invoke signal sum primitive and allocate and free the necessary scratch memory:

```
// pSrc, pSum, pDeviceBuffer are all device pointers.
Npp32f * pSrc;
Npp32f * pSum;
Npp8u * pDeviceBuffer;
int nLength = 1024;

// Allocate the device memroy.
cudaMalloc((void **)&pSrc, sizeof(Npp32f) * nLength);
nppsSet_32f(1.0f, pSrc, nLength);
cudaMalloc((void **)&pSum, sizeof(Npp32f) * 1);

// Compute the appropriate size of the scratch-memory buffer
int nBufferSize;
nppsSumGetBufferSize_32f(nLength, &nBufferSize);
// Allocate the scratch buffer
cudaMalloc((void **)&pDeviceBuffer, nBufferSize);

// Call the primitive with the scratch buffer
```

```

nppsSum_32f(pSrc, nLength, pSum, pDeviceBuffer);
Npp32f nSumHost;
cudaMemcpy(&nSumHost, pSum, sizeof(Npp32f) * 1, cudaMemcpyDeviceToHost);
printf("sum = %f\n", nSumHost); // nSumHost = 1024.0f;

// Free the device memory
cudaFree(pSrc);
cudaFree(pDeviceBuffer);
cudaFree(pSum);

```

2.2 Function Naming

Since NPP is a C API and therefore does not allow for function overloading for different data-types the NPP naming convention addresses the need to differentiate between different flavors of the same algorithm or primitive function but for various data types. This disambiguation of different flavors of a primitive is done via a suffix containing data type and other disambiguating information.

In addition to the flavor suffix, all NPP functions are prefixed with by the letters "npp". Primitives belonging to NPP's image-processing module add the letter "i" to the npp prefix, i.e. are prefixed by "nppi". Similarly signal-processing primitives are prefixed with "npps".

The general naming scheme is:

`npp<module info><PrimitiveName>_<data-type info>[_<additional flavor info>](<parameter list>)`

The data-type information uses the same names as the [Basic NPP Data Types](#). For example the data-type information "8u" would imply that the primitive operates on [Npp8u](#) data.

If a primitive consumes different type data from what it produces, both types will be listed in the order of consumed to produced data type.

Details about the "additional flavor information" is provided for each of the NPP modules, since each problem domain uses different flavor information suffixes.

2.3 Integer Result Scaling

NPP signal processing and imaging primitives often operate on integer data. This integer data is usually a fixed point fractional representation of some physical magnitue (e.g. luminance). Because of this fixed-point nature of the representation many numerical operations (e.g. addition or multiplication) tend to produce results exceeding the original fixed-point range if treated as regular integers.

In cases where the results exceed the original range, these functions clamp the result values back to the valid range. E.g. the maximum positive value for a 16-bit unsigned integer is 32767. A multiplication operation of $4 * 10000 = 40000$ would exceed this range. The result would be clamped to be 32767.

To avoid the level of lost information due to clamping most integer primitives allow for result scaling. Primitives with result scaling have the "Sfs" suffix in their name and provide a parameter "nScaleFactor" that controls the amount of scaling. Before the results of an operation are clamped to the valid output-data range by multiplying them with $2^{-nScaleFactor}$.

Example: The primitive [nppsSqr_8u_Sfs\(\)](#) computes the square of 8-bit unsigned sample values in a signal (1D array of values). The maximum value of a 8-bit value is 255. The square of $255^2 = 65025$ which would be clamped to 255 if no result scaling is performed. In order to map the maximum value of 255 to 255 in the result, one would specify an integer result scaling factor of 8, i.e. multiply each result with $2^{-8} = \frac{1}{2^8} = \frac{1}{256}$. The final result for a signal value of 255 being squared and scaled would be:

$$255^2 \cdot 2^{-8} = 254.00390625$$

which would be rounded to a final result of 254.

A medium gray value of 128 would result in

$$128^2 * 2^{-8} = 64$$

2.4 Rounding Modes

Many NPP functions require converting floating-point values to integers. The [NppRoundMode](#) enum lists NPP's supported rounding modes. Not all primitives in NPP that perform rounding as part of their functionality allow the user to specify the round-mode used. Instead they use NPP's default rounding mode, which is [NPP_RND_FINANCIAL](#).

2.4.1 Rounding Mode Parameter

A subset of NPP functions performing rounding as part of their functionality do allow the user to specify which rounding mode is used through a parameter of the [NppRoundMode](#) type.

Chapter 3

Signal-Processing Specific API Conventions

3.1 Signal Data

Signal data is passed to and from NPPS primitives via a pointer to the signal's data type.

The general idea behind this fairly low-level way of passing signal data is ease-of-adoption into existing software projects:

- Passing the data pointer rather than a higher-level signal struct allows for easy adoption by not requiring a specific signal representation (that could include total signal size offset, or other additional information). This avoids awkward packing and unpacking of signal data from the host application to an NPP specific signal representation.

3.1.1 Parameter Names for Signal Data

There are three general cases of image-data passing throughout NPP detailed in the following sections.

Those are signals consumed by the algorithm.

3.1.1.1 Source Signal Pointer

The source signal data is generally passed via a pointer named

`pSrc`

The source signal pointer is generally defined constant, enforcing that the primitive does not change any image data pointed to by that pointer. E.g.

```
nppsPrimitive_32s(const Npp32s * pSrc, ...)
```

In case the primitive consumes multiple signals as inputs the source pointers are numbered like this:

`pSrc1, pSrc2, ...`

3.1.1.2 Destination Signal Pointer

The destination signal data is generally passed via a pointer named

`pDst`

In case the primitive consumes multiple signals as inputs the source pointers are numbered like this:

`pDst1, pDst2, ...`

3.1.1.3 In-Place Signal Pointer

In the case of in-place processing, source and destination are served by the same pointer and thus pointers to in-place signal data are called:

`pSrcDst`

3.1.2 Signal Data Alignment Requirements

NPP requires signal sample data to be naturally aligned, i.e. any pointer

```
NppType * p;
```

to a sample in a signal needs to fulfill:

```
assert(p % sizeof(p) == 0);
```

3.1.3 Signal Data Related Error Codes

All NPPI primitives operating on signal data validate the signal-data pointer for proper alignment and test that the point is not null.

Failed validation results in one of the following error codes being returned and the primitive not being executed:

- [NPP_NULL_POINTER_ERROR](#) is returned if the image-data pointer is 0 (NULL).
- [NPP_ALIGNMENT_ERROR](#) if the signal-data pointer address is not a multiple of the signal's data-type size.

3.2 Signal Length

The vast majority of NPPS functions take a

```
nLength
```

parameter that tells the primitive how many of the signal's samples starting from the given data pointer are to be processed.

3.2.1 Length Related Error Codes

All NPPS primitives taking a length parameter validate this input.

Failed validation results in the following error code being returned and the primitive not being executed:

- [NPP_SIZE_ERROR](#) is returned if the length is negative.

Chapter 4

Imaging-Processing Specific API Conventions

4.1 Function Naming

Image processing related functions use a number of suffixes to indicate various different flavors of a primitive beyond just different data types. The flavor suffix uses the following abbreviations:

- "A" if the image is a 4 channel image this indicates the result alpha channel is not affected by the primitive.
- "Cn" the image consists of n channel packed pixels, where n can be 1, 2, 3 or 4.
- "Pn" the image consists of n separate image planes, where n can be 1, 2, 3 or 4.
- "C" (following the channel information) indicates that the primitive only operates on one of the color channels, the "channel-of-interest". All other output channels are not affected by the primitive.
- "I" indicates that the primitive works "in-place". In this case the image-data pointer is usually named "pSrcDst" to indicate that the image data serves as source and destination at the same time.
- "M" indicates "masked operation". These types of primitives have an additional "mask image" as input. Each pixel in the destination image corresponds to a pixel in the mask image. Only pixels with a corresponding non-zero mask pixel are being processed.
- "R" indicates the primitive operates only on a rectangular "region-of-interest" or "ROI". All ROI primitives take an additional input parameter of type [NppiSize](#), which specifies the width and height of the rectangular region that the primitive should process. For details on how primitives operate on ROIs see: [Region-of-Interest \(ROI\)](#).
- "Sfs" indicates the result values are processed by fixed scaling and saturation before they're written out.

The suffixes above always appear in alphabetical order. E.g. a 4 channel primitive not affecting the alpha channel with masked operation, in place and with scaling/saturation and ROI would have the postfix: "AC4IMRSfs".

4.2 Image Data

Image data is passed to and from NPPI primitives via a pair of parameters:

1. A pointer to the image's underlying data type.
2. A line step in bytes (also sometimes called line stride).

The general idea behind this fairly low-level way of passing image data is ease-of-adoption into existing software projects:

- Passing a raw pointer to the underlying pixel data type, rather than structured (by color) channel pixel data allows usage of the function in a wide variety of situations avoiding risky type cast or expensive image data copies.
- Passing the data pointer and line step individually rather than a higher-level image struct again allows for easy adoption by not requiring a specific image representation and thus avoiding awkward packing and unpacking of image data from the host application to an NPP specific image representation.

4.2.1 Line Step

The line step (also called "line stride" or "row step") allows lines of oddly sized images to start on well-aligned addresses by adding a number of unused bytes at the ends of the lines. This type of line padding has been common practice in digital image processing for a long time and is not particular to GPU image processing.

The line step is the number of bytes in a line **including the padding**. An other way to interpret this number is to say that it is the number of bytes between the first pixel of successive rows in the image, or generally the number of bytes between two neighboring pixels in any column of pixels.

The general reason for the existence of the line step it is that uniformly aligned rows of pixel enable optimizations of memory-access patterns.

Even though all functions in NPP will work with arbitrarily aligned images, best performance can only be achieved with well aligned image data. Any image data allocated with the NPP image allocators or the 2D memory allocators in the CUDA runtime, is well aligned.

Particularly on older CUDA capable GPUs it is likely that the performance decrease for misaligned data is substantial (orders of magnitude).

All image data passed to NPPI primitives requires a line step to be provided. It is important to keep in mind that this line step is always specified in terms of bytes, not pixels.

4.2.2 Parameter Names for Image Data

There are three general cases of image-data passing throughout NPP detailed in the following sections.

4.2.2.1 Passing Source-Image Data

Those are images consumed by the algorithm.

4.2.2.1.1 Source-Image Pointer

The source image data is generally passed via a pointer named

```
pSrc
```

The source image pointer is generally defined constant, enforcing that the primitive does not change any image data pointed to by that pointer. E.g.

```
nppiPrimitive_32s_C1R(const Npp32s * pSrc, ...)
```

In case the primitive consumes multiple images as inputs the source pointers are numbered like this:

```
pSrc1, pSrc2, ...
```

4.2.2.1.2 Source-Planar-Image Pointer Array

The planar source image data is generally passed via an array of pointers named

```
pSrc[]
```

The planar source image pointer array is generally defined a constant array of constant pointers, enforcing that the primitive does not change any image data pointed to by those pointers. E.g.

```
nppiPrimitive_8u_P3R(const Npp8u * const pSrc[3], ...)
```

Each pointer in the array points to a different image plane.

4.2.2.1.3 Source-Planar-Image Pointer

The multiple plane source image data is passed via a set of pointers named

```
pSrc1, pSrc2, ...
```

The planar source image pointer is generally defined as one of a set of constant pointers with each pointer pointing to a different input image plane.

4.2.2.1.4 Source-Image Line Step

The source image line step is the number of bytes between successive rows in the image. The source image line step parameter is

```
nSrcStep
```

or in the case of multiple source images

```
nSrcStep1, nSrcStep2, ...
```

4.2.2.1.5 Source-Planar-Image Line Step Array

The source planar image line step array is an array where each element of the array contains the number of bytes between successive rows for a particular plane in the input image. The source planar image line step array parameter is

```
rSrcStep[]
```

4.2.2.1.6 Source-Planar-Image Line Step

The source planar image line step is the number of bytes between successive rows in a particular plane of the multiplane input image. The source planar image line step parameter is

```
nSrcStep1, nSrcStep2, ...
```

4.2.2.2 Passing Destination-Image Data

Those are images produced by the algorithm.

4.2.2.2.1 Destination-Image Pointer

The destination image data is generally passed via a pointer named

```
pDst
```

In case the primitive generates multiple images as outputs the destination pointers are numbered like this:

```
pDst1, pDst2, ...
```

4.2.2.2.2 Destination-Planar-Image Pointer Array

The planar destination image data pointers are generally passed via an array of pointers named

```
pDst[]
```

Each pointer in the array points to a different image plane.

4.2.2.2.3 Destination-Planar-Image Pointer

The destination planar image data is generally passed via a pointer to each plane of a multiplane output image named

```
pDst1, pDst2, ...
```

4.2.2.2.4 Destination-Image Line Step

The destination image line step parameter is

```
nDstStep
```

or in the case of multiple destination images

```
nDstStep1, nDstStep2, ...
```

4.2.2.2.5 Destination-Planar-Image Line Step Array

The destination planar image line step array is an array where each element of the array contains the number of bytes between successive rows for a particular plane in the output image. The destination planar image line step array parameter is

```
rDstStep[]
```

4.2.2.2.6 Destination-Planar-Image Line Step

The destination planar image line step is the number of bytes between successive rows for a particular plane in a multiplane output image. The destination planar image line step parameter is

```
nDstStep1, nDstStep2, ...
```

4.2.2.3 Passing In-Place Image Data

4.2.2.3.1 In-Place Image Pointer

In the case of in-place processing, source and destination are served by the same pointer and thus pointers to in-place image data are called:

`pSrcDst`

4.2.2.3.2 In-Place-Image Line Step

The in-place line step parameter is

`nSrcDstStep`

4.2.2.4 Passing Mask-Image Data

Some image processing primitives have variants supporting [Masked Operation](#).

4.2.2.4.1 Mask-Image Pointer

The mask-image data is generally passed via a pointer named

`pMask`

4.2.2.4.2 Mask-Image Line Step

The mask-image line step parameter is

`nMaskStep`

4.2.2.5 Passing Channel-of-Interest Data

Some image processing primitives support [Channel-of-Interest API](#).

4.2.2.5.1 Channel_of_Interest Number

The channel-of-interest data is generally an integer (either 1, 2, or 3):

`nCOI`

4.2.3 Image Data Alignment Requirements

NPP requires pixel data to adhere to certain alignment constraints: For 2 and 4 channel images the following alignment requirement holds: `data_pointer % (#channels * sizeof(channel type)) == 0`. E.g. a 4 channel image with underlying type [Npp8u](#) (8-bit unsigned) would require all pixels to fall on addresses that are multiples of 4 (4 channels * 1 byte size).

As a logical consequence of all pixels being aligned to their natural size the image line steps of 2 and 4 channel images also need to be multiples of the pixel size.

1 and 3 channel images only require that pixel pointers are aligned to the underlying data type, i.e. `pData % sizeof(data type) == 0`. And consequentially line steps are also held to this requirement.

4.2.4 Image Data Related Error Codes

All NPPI primitives operating on image data validate the image-data pointer for proper alignment and test that the point is not null. They also validate the line stride for proper alignment and guard against the step being less or equal to 0. Failed validation results in one of the following error codes being returned and the primitive not being executed:

- `NPP_STEP_ERROR` is returned if the data step is 0 or negative.
- `NPP_NOT_EVEN_STEP_ERROR` is returned if the line step is not a multiple of the pixel size for 2 and 4 channel images.
- `NPP_NULL_POINTER_ERROR` is returned if the image-data pointer is 0 (NULL).
- `NPP_ALIGNMENT_ERROR` if the image-data pointer address is not a multiple of the pixel size for 2 and 4 channel images.

4.3 Region-of-Interest (ROI)

In practice processing a rectangular sub-region of an image is often more common than processing complete images. The vast majority of NPP's image-processing primitives allow for processing of such sub regions also referred to as regions-of-interest or ROIs.

All primitives supporting ROI processing are marked by a "R" in their name suffix. In most cases the ROI is passed as a single `NppiSize` struct, which provides the width and height of the ROI. This raises the question how the primitive knows where in the image this rectangle of (width, height) is located. The "start pixel" of the ROI is implicitly given by the image-data pointer. I.e. instead of explicitly passing a pixel coordinate for the upper-right corner, the user simply offsets the image-data pointers to point to the first pixel of the ROI.

In practice this means that for an image (`pSrc`, `nSrcStep`) and the start-pixel of the ROI being at location (`xROI`, `yROI`), one would pass

`pSrcOffset = pSrc + yROI * nSrcStep + xROI * PixelSize;`

as the image-data source to the primitive. `PixelSize` is typically computed as

`PixelSize = NumberOfColorChannels * sizeof(PixelDataType).`

E.g. for a primitive like `nppiSet_16s_C4R()` we would have

- `NumberOfColorChannels == 4;`
- `sizeof(Npp16s) == 2;`
- and thus `PixelSize = 4 * 2 = 8;`

4.3.1 ROI Related Error Codes

All NPPI primitives operating on ROIs of image data validate the ROI size and image's step size. Failed validation results in one of the following error codes being returned and the primitive not being executed:

- `NPP_SIZE_ERROR` is returned if either the ROI width or ROI height are negative.
- `NPP_STEP_ERROR` is returned if the ROI width exceeds the image's line step. In mathematical terms $(\text{widthROI} * \text{PixelSize}) > \text{nLinStep}$ indicates an error.

4.4 Masked Operation

Some primitive support masked operation. An "M" in the suffix of those variants indicates masked operation. Primitives supporting masked operation consume an additional input image provided via a [Mask-Image Pointer](#) and [Mask-Image Line Step](#). The mask image is interpreted by these primitives as a boolean image. The values of type `Npp8u` are interpreted as boolean values where a values of 0 indicates false, any non-zero values true.

Unless otherwise indicated the operation is only performed on pixels where its spatially corresponding mask pixel is true (non-zero). E.g. a masked copy operation would only copy those pixels in the ROI that have corresponding non-zero mask pixels.

4.5 Channel-of-Interest API

Some primitives allow restricting operations to a single channel of interest within a multi-channel image. These primitives are suffixed with the letter "C" (after the channel information, e.g. `nppiCopy_8u_C3CR(...)`). The channel-of-interest is generally selected by offsetting the image-data pointer to point directly to the channel- of-interest rather than the base of the first pixel in the ROI. Some primitives also explicitly specify the selected channel number and pass it via an integer, e.g. `nppiMean_StdDev_8u_C3CR(...)`.

4.5.1 Select-Channel Source-Image Pointer

This is a pointer to the channel-of-interest within the first pixel of the source image. E.g. if `pSrc` is the pointer to the first pixel inside the ROI of a three channel image. Using the appropriate select-channel copy primitive one could copy the second channel of this source image into the first channel of a destination image given by `pDst` by offsetting the pointer by one:

```
nppiCopy_8u_C3CR(pSrc + 1, nSrcStep, pDst, nDstStep, oSizeROI);
```

4.5.2 Select-Channel Source-Image

Some primitives allow the user to select the channel-of-interest by specifying the channel number (`nCOI`). This approach is typically used in the image statistical functions. For example,

```
nppiMean_StdDev_8u_C3CR(pSrc, nSrcStep, oSizeROI, nCOI, pDeviceBuffer, pMean, pStdDev );
```

The channel-of-interest number can be either 1, 2, or 3.

4.5.3 Select-Channel Destination-Image Pointer

This is a pointer to the channel-of-interest within the first pixel of the destination image. E.g. if `pDst` is the pointer to the first pixel inside the ROI of a three channel image. Using the appropriate select-channel

copy primitive one could copy data into the second channel of this destination image from the first channel of a source image given by pSrc by offsetting the destination pointer by one:

```
nppiCopy_8u_C3CR(pSrc, nSrcStep, pDst + 1, nDstStep, oSizeROI);
```

4.6 Source-Image Sampling

A large number of NPP image-processing functions consume at least one source image and produce an output image (e.g. [nppiAddC_8u_C1RSfs\(\)](#) or [nppiFilterBox_8u_C1R\(\)](#)). All NPP functions falling into this category also operate on ROIs (see [Region-of-Interest \(ROI\)](#)) which for these functions should be considered to describe the destination ROI. In other words the ROI describes a rectangular region in the destination image and all pixels inside of this region are being written by the function in question.

In order to use such functions successfully it is important to understand how the user defined destination ROI affects which pixels in the input image(s) are being read by the algorithms. To simplify the discussion of ROI propagation (i.e. given a destination ROI, what are the ROIs in in the source(s)), it makes sense to distinguish two major cases:

1. Point-Wise Operations: These are primitives like [nppiAddC_8u_C1RSfs\(\)](#). Each output pixel requires exactly one input pixel to be read.
2. Neighborhood Operations: These are primitives like [nppiFilterBox_8u_C1R\(\)](#), which require a group of pixels from the source image(s) to be read in order to produce a single output.

4.6.1 Point-Wise Operations

As mentioned above, point-wise operations consume a single pixel from the input image (or a single pixel from each input image, if the operation in question has more than one input image) in order to produce a single output pixel.

4.6.2 Neighborhood Operations

In the case of neighborhood operations a number of input pixels (a "neighborhood" of pixels) is read in the input image (or images) in order to compute a single output pixel. All of the functions for [Filtering Functions](#) and [Morphological Operations](#) are neighborhood operations.

Most of these functions have parameters that affect the size and relative location of the neighborhood: a mask-size structure and an anchor-point structure. Both parameters are described in more detail in the next subsections.

4.6.2.1 Mask-Size Parameter

Many NPP neighborhood operations allow the user to specify the size of the neighborhood via a parameter usually named oMaskSize of type [NppiSize](#). In those cases the neighborhood of pixels read from the source(s) is exactly the size of the mask. Assuming the mask is anchored at location (0, 0) (see [Anchor-Point Parameter](#) below) and has a size of (w, h), i.e.

```
assert(oMaskSize.w == w);
assert(oMaskSize.h == h);
assert(oAnchor.x == 0);
assert(oAnchor.y == 0);
```

a neighborhood operation would read the following source pixels in order to compute destination pixel $D_{i,j}$:

$$\begin{array}{cccc} S_{i,j} & S_{i,j+1} & \dots & S_{i,j+w-1} \\ S_{i+1,j} & S_{i+1,j+1} & \dots & S_{i+1,j+w-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i+h-1,j} & S_{i+h-1,j+1} & \dots & S_{i+h-1,j+w-1} \end{array}$$

4.6.2.2 Anchor-Point Parameter

Many NPP primitives performing neighborhood operations allow the user to specify the relative location of the neighborhood via a parameter usually named `oAnchor` of type [NppiPoint](#). Using the anchor a developer can choose the position of the mask (see [Mask-Size Parameter](#)) relative to current pixel index.

Using the same example as in [Mask-Size Parameter](#), but this time with an anchor position of (a, b):

```
assert(oMaskSize.w == w);
assert(oMaskSize.h == h);
assert(oAnchor.x == a);
assert(oAnchor.y == b);
```

the following pixels from the source image would be read:

$$\begin{array}{cccc} S_{i-a,j-b} & S_{i-a,j-b+1} & \dots & S_{i-a,j-b+w-1} \\ S_{i-a+1,j-b} & S_{i-a+1,j-b+1} & \dots & S_{i-a+1,j-b+w-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i-a+h-1,j-b} & S_{i-a+h-1,j-b+1} & \dots & S_{i-a+h-1,j-b+w-1} \end{array}$$

4.6.2.3 Sampling Beyond Image Boundaries

NPP primitives in general and NPP neighborhood operations in particular require that all pixel locations read and written are valid and within the boundaries of the respective images. Sampling outside of the defined image data regions results in undefined behavior and may lead to system instability.

This poses a problem in practice: when processing full-size images one cannot choose the destination ROI to be the same size as the source image. Because neighborhood operations read pixels from an enlarged source ROI, the destination ROI must be shrunk so that the expanded source ROI does not exceed the source image's size.

For cases where this "shrinking" of the destination image size is unacceptable, NPP provides a set of border-expanding Copy primitives. E.g. [nppiCopyConstBorder_8u_C1R\(\)](#), [nppiCopyReplicateBorder_8u_C1R\(\)](#) and [nppiCopyWrapBorder_8u_C1R\(\)](#). The user can use these primitives to "expand" the source image's size using one of the three expansion modes. The expanded image can then be safely passed to a neighborhood operation producing a full-size result.

Chapter 5

Module Index

5.1 Modules

Here is a list of all modules:

NPP Core	31
NPP Type Definitions and Constants	34
Basic NPP Data Types	48
NPP Image Processing	52
Arithmetic and Logical Operations	53
Arithmetic Operations	54
AddC	56
MulC	82
MulCScale	108
SubC	115
DivC	141
AbsDiffC	167
Add	169
AddSquare	198
AddProduct	201
AddWeighted	205
Mul	209
MulScale	238
Sub	247
Div	277
Div_Round	306
Abs	321
AbsDiff	328
Sqr	331
Sqrt	345
Ln	357
Exp	364
Logical Operations	371
AndC	372
OrC	383
XorC	394
RShiftC	405
LShiftC	422

And	433
Or	445
Xor	457
Not	469
Alpha Composition	473
AlphaCompC	474
AlphaPremulC	482
AlphaComp	489
AlphaPremul	496
Color and Sampling Conversion	498
Color Model Conversion	499
Color Sampling Format Conversion	586
Color Gamma Correction	614
Complement Color Key	620
Color Processing	623
Compression	720
Quantization Functions	724
Labeling and Segmentation	730
GraphCut	731
Data Exchange and Initialization	738
Set	739
Copy	773
Convert	820
Scale	864
Copy Constant Border	879
Copy Replicate Border	892
Copy Wrap Border	904
Copy Sub-Pixel	917
Duplicate Channel	928
Transpose	935
Swap Channels	942
Filtering Functions	960
1D Linear Filter	1103
1D Window Sum	1197
1D Window Sum with Border Control	1208
Convolution	1221
2D Fixed Linear Filters	1280
Rank Filters	1298
Fixed Filters	1348
Geometry Transforms	1394
ResizeSqrPixel	1396
Resize	1419
Remap	1431
Rotate	1453
Mirror	1462
Affine Transforms	1479
Perspective Transform	1529
Linear Transforms	1575
Fourier Transforms	1576
Morphological Operations	1578
Dilation	1579
Dilation with border control	1586
Dilate3x3	1594

Dilate3x3Border	1600
Erode	1607
Erosion with border control	1614
Erode3x3	1622
Erode3x3Border	1628
Statistical Operations	1635
Sum	1702
Min	1717
MinIndx	1730
Max	1744
MaxIndx	1757
MinMax	1771
MinMaxIndx	1785
Mean	1802
Mean_StdDev	1823
Image Norms	1839
Norm_Inf	1841
Norm_L1	1863
Norm_L2	1884
NormDiff_Inf	1905
NormDiff_L1	1928
NormDiff_L2	1951
NormRel_Inf	1974
NormRel_L1	1997
NormRel_L2	2020
DotProd	2043
CountInRange.	2068
MaxEvery	2074
MinEvery	2081
Integral	2088
SqrIntegral	2090
RectStdDev	2093
HistogramEven	2096
HistogramRange	2109
Image Proximity	2125
SqrDistanceFull_Norm	2128
SqrDistanceSame_Norm	2139
SqrDistanceValid_Norm	2150
CrossCorrFull_Norm	2161
CrossCorrSame_Norm	2172
CrossCorrValid_Norm	2183
CrossCorrValid	2194
CrossCorrFull_NormLevel	2197
CrossCorrSame_NormLevel	2217
CrossCorrValid_NormLevel	2237
Image Quality Index	2257
MaximumError	2266
AverageError	2289
MaximumRelativeError	2312
AverageRelativeError	2336
Memory Management	2360
Threshold and Compare Operations	2372
Threshold Operations	2373

Compare Operations	2462
NPP Signal Processing	2485
Arithmetic and Logical Operations	2486
Arithmetic Operations	2487
AddC	2489
AddProductC	2498
MulC	2499
SubC	2509
SubCRev	2518
DivC	2527
DivCRev	2534
Add	2536
AddProduct	2548
Mul	2552
Sub	2565
Div	2575
Div_Round	2583
Abs	2586
Sqr	2589
Sqrt	2595
Cubrt	2603
Exp	2604
Ln	2608
10Log10	2612
SumLn	2613
Arctan	2617
Normalize	2619
Cauchy, CauchyD, and CauchyDD2	2622
Logical And Shift Operations	2624
AndC	2625
And	2628
OrC	2631
Or	2634
XorC	2637
Xor	2640
Not	2643
LShiftC	2646
RShiftC	2650
Conversion Functions	2654
Convert	2655
Threshold	2658
Filtering Functions	2683
Integral	2684
Initialization	2685
Set	2686
Zero	2691
Copy	2695
Statistical Functions	2699
MinEvery And MaxEvery Functions	2700
Sum	2704
Maximum	2711
Minimum	2721
Mean	2731

Standard Deviation	2737
Mean And Standard Deviation	2740
Minimum_Maximum	2744
Infinity Norm	2756
L1 Norm	2761
L2 Norm	2767
Infinity Norm Diff	2773
L1 Norm Diff	2778
L2 Norm Diff	2784
Dot Product	2790
Count In Range	2810
Count Zero Crossings	2811
MaximumError	2813
AverageError	2824
MaximumRelativeError	2835
AverageRelativeError	2847
Memory Management	2859
Malloc	2860
Free	2865

Chapter 6

Data Structure Index

6.1 Data Structures

Here are the data structures with brief descriptions:

NPP_ALIGN_16 (Complex Number This struct represents a long long complex number)	2867
NPP_ALIGN_8 (Complex Number This struct represents an unsigned int complex number) . .	2869
NppiHaarBuffer	2871
NppiHaarClassifier_32f	2872
NppiPoint (2D Point)	2873
NppiRect (2D Rectangle This struct contains position and size information of a rectangle in two space)	2874
NppiSize (2D Size This struct typically represents the size of a a rectangular region in two space)	2875
NppLibraryVersion	2876

Chapter 7

Module Documentation

7.1 NPP Core

Basic functions for library management, in particular library version and device property query functions.

Functions

- `const NppLibraryVersion * nppGetLibVersion (void)`
Get the NPP library version.
- `NppGpuComputeCapability nppGetGpuComputeCapability (void)`
What CUDA compute model is supported by the active CUDA device?
- `int nppGetGpuNumSMs (void)`
Get the number of Streaming Multiprocessors (SM) on the active CUDA device.
- `int nppGetMaxThreadsPerBlock (void)`
Get the maximum number of threads per block on the active CUDA device.
- `int nppGetMaxThreadsPerSM (void)`
Get the maximum number of threads per SM for the active GPU.
- `const char * nppGetGpuName (void)`
Get the name of the active CUDA device.
- `cudaStream_t nppGetStream (void)`
Get the NPP CUDA stream.
- `void nppSetStream (cudaStream_t hStream)`
Set the NPP CUDA stream.

7.1.1 Detailed Description

Basic functions for library management, in particular library version and device property query functions.

7.1.2 Function Documentation

7.1.2.1 `NppGpuComputeCapability nppGetGpuComputeCapability (void)`

What CUDA compute model is supported by the active CUDA device?

Before trying to call any NPP functions, the user should make a call this function to ensure that the current machine has a CUDA capable device.

Returns:

An enum value representing if a CUDA capable device was found and what level of compute capabilities it supports.

7.1.2.2 `const char* nppGetGpuName (void)`

Get the name of the active CUDA device.

Returns:

Name string of the active graphics-card/compute device in a system.

7.1.2.3 `int nppGetGpuNumSMs (void)`

Get the number of Streaming Multiprocessors (SM) on the active CUDA device.

Returns:

Number of SMs of the default CUDA device.

7.1.2.4 `const NppLibraryVersion* nppGetLibVersion (void)`

Get the NPP library version.

Returns:

A struct containing separate values for major and minor revision and build number.

7.1.2.5 `int nppGetMaxThreadsPerBlock (void)`

Get the maximum number of threads per block on the active CUDA device.

Returns:

Maximum number of threads per block on the active CUDA device.

7.1.2.6 int nppGetMaxThreadsPerSM (void)

Get the maximum number of threads per SM for the active GPU.

Returns:

Maximum number of threads per SM for the active GPU

7.1.2.7 cudaStream_t nppGetStream (void)

Get the NPP CUDA stream.

NPP enables concurrent device tasks via a global stream state variable. The NPP stream by default is set to stream 0, i.e. non-concurrent mode. A user can set the NPP stream to any valid CUDA stream. All CUDA commands issued by NPP (e.g. kernels launched by the NPP library) are then issued to that NPP stream.

7.1.2.8 void nppSetStream (cudaStream_t *hStream*)

Set the NPP CUDA stream.

See also:

[nppGetStream\(\)](#)

7.2 NPP Type Definitions and Constants

Data Structures

- struct [NppLibraryVersion](#)
- struct [NppiPoint](#)
2D Point
- struct [NppiSize](#)
2D Size This struct typically represents the size of a rectangular region in two space.
- struct [NppiRect](#)
2D Rectangle This struct contains position and size information of a rectangle in two space.
- struct [NppiHaarClassifier_32f](#)
- struct [NppiHaarBuffer](#)

Modules

- [Basic NPP Data Types](#)

Defines

- #define [NPP_MIN_8U](#) (0)
Minimum 8-bit unsigned integer.
- #define [NPP_MAX_8U](#) (255)
Maximum 8-bit unsigned integer.
- #define [NPP_MIN_16U](#) (0)
Minimum 16-bit unsigned integer.
- #define [NPP_MAX_16U](#) (65535)
Maximum 16-bit unsigned integer.
- #define [NPP_MIN_32U](#) (0)
Minimum 32-bit unsigned integer.
- #define [NPP_MAX_32U](#) (4294967295U)
Maximum 32-bit unsigned integer.
- #define [NPP_MIN_64U](#) (0)
Minimum 64-bit unsigned integer.
- #define [NPP_MAX_64U](#) (18446744073709551615ULL)
Maximum 64-bit unsigned integer.
- #define [NPP_MIN_8S](#) (-127 - 1)
Minimum 8-bit signed integer.

- #define `NPP_MAX_8S` (127)
Maximum 8-bit signed integer.
- #define `NPP_MIN_16S` (-32767 - 1)
Minimum 16-bit signed integer.
- #define `NPP_MAX_16S` (32767)
Maximum 16-bit signed integer.
- #define `NPP_MIN_32S` (-2147483647 - 1)
Minimum 32-bit signed integer.
- #define `NPP_MAX_32S` (2147483647)
Maximum 32-bit signed integer.
- #define `NPP_MAX_64S` (9223372036854775807LL)
Maximum 64-bit signed integer.
- #define `NPP_MIN_64S` (-9223372036854775807LL - 1)
Minimum 64-bit signed integer.
- #define `NPP_MINABS_32F` (1.175494351e-38f)
Smallest positive 32-bit floating point value.
- #define `NPP_MAXABS_32F` (3.402823466e+38f)
Largest positive 32-bit floating point value.
- #define `NPP_MINABS_64F` (2.2250738585072014e-308)
Smallest positive 64-bit floating point value.
- #define `NPP_MAXABS_64F` (1.7976931348623158e+308)
Largest positive 64-bit floating point value.

Enumerations

- enum `NppiInterpolationMode` {
`NPPI_INTER_UNDEFINED` = 0,
`NPPI_INTER_NN` = 1,
`NPPI_INTER_LINEAR` = 2,
`NPPI_INTER_CUBIC` = 4,
`NPPI_INTER_CUBIC2P_BSPLINE`,
`NPPI_INTER_CUBIC2P_CATMULLROM`,
`NPPI_INTER_CUBIC2P_B05C03`,
`NPPI_INTER_SUPER` = 8,
`NPPI_INTER_LANCZOS` = 16,
`NPPI_INTER_LANCZOS3_ADVANCED` = 17,
`NPPI_SMOOTH_EDGE` = (1 << 31) }

Filtering methods.

- enum `NppiBayerGridPosition` {
`NPPI_BAYER_BGGR` = 0,
`NPPI_BAYER_RGGB` = 1,
`NPPI_BAYER_GBRG` = 2,
`NPPI_BAYER_GRBG` = 3 }

Bayer Grid Position Registration.

- enum `NppiMaskSize` {
`NPP_MASK_SIZE_1_X_3`,
`NPP_MASK_SIZE_1_X_5`,
`NPP_MASK_SIZE_3_X_1` = 100,
`NPP_MASK_SIZE_5_X_1`,
`NPP_MASK_SIZE_3_X_3` = 200,
`NPP_MASK_SIZE_5_X_5`,
`NPP_MASK_SIZE_7_X_7` = 400,
`NPP_MASK_SIZE_9_X_9` = 500,
`NPP_MASK_SIZE_11_X_11` = 600,
`NPP_MASK_SIZE_13_X_13` = 700,
`NPP_MASK_SIZE_15_X_15` = 800 }

Fixed filter-kernel sizes.

- enum `NppStatus` {
`NPP_NOT_SUPPORTED_MODE_ERROR` = -9999,
`NPP_INVALID_HOST_POINTER_ERROR` = -1032,
`NPP_INVALID_DEVICE_POINTER_ERROR` = -1031,
`NPP_LUT_PALETTE_BITSIZE_ERROR` = -1030,
`NPP_ZC_MODE_NOT_SUPPORTED_ERROR` = -1028,
`NPP_NOT_SUFFICIENT_COMPUTE_CAPABILITY` = -1027,
`NPP_TEXTURE_BIND_ERROR` = -1024,
`NPP_WRONG_INTERSECTION_ROI_ERROR` = -1020,
`NPP_HAAR_CLASSIFIER_PIXEL_MATCH_ERROR` = -1006,
`NPP_MEMFREE_ERROR` = -1005,
`NPP_MEMSET_ERROR` = -1004,
`NPP_MEMCPY_ERROR` = -1003,
`NPP_ALIGNMENT_ERROR` = -1002,
`NPP_CUDA_KERNEL_EXECUTION_ERROR` = -1000,
`NPP_ROUND_MODE_NOT_SUPPORTED_ERROR` = -213,
`NPP_QUALITY_INDEX_ERROR` = -210,
`NPP_RESIZE_NO_OPERATION_ERROR` = -201,
`NPP_OVERFLOW_ERROR` = -109,

NPP_NOT_EVEN_STEP_ERROR = -108,
NPP_HISTOGRAM_NUMBER_OF_LEVELS_ERROR = -107,
NPP_LUT_NUMBER_OF_LEVELS_ERROR = -106,
NPP_CORRUPTED_DATA_ERROR = -61,
NPP_CHANNEL_ORDER_ERROR = -60,
NPP_ZERO_MASK_VALUE_ERROR = -59,
NPP_QUADRANGLE_ERROR = -58,
NPP_RECTANGLE_ERROR = -57,
NPP_COEFFICIENT_ERROR = -56,
NPP_NUMBER_OF_CHANNELS_ERROR = -53,
NPP_COI_ERROR = -52,
NPP_DIVISOR_ERROR = -51,
NPP_CHANNEL_ERROR = -47,
NPP_STRIDE_ERROR = -37,
NPP_ANCHOR_ERROR = -34,
NPP_MASK_SIZE_ERROR = -33,
NPP_RESIZE_FACTOR_ERROR = -23,
NPP_INTERPOLATION_ERROR = -22,
NPP_MIRROR_FLIP_ERROR = -21,
NPP_MOMENT_00_ZERO_ERROR = -20,
NPP_THRESHOLD_NEGATIVE_LEVEL_ERROR = -19,
NPP_THRESHOLD_ERROR = -18,
NPP_CONTEXT_MATCH_ERROR = -17,
NPP_FFT_FLAG_ERROR = -16,
NPP_FFT_ORDER_ERROR = -15,
NPP_STEP_ERROR = -14,
NPP_SCALE_RANGE_ERROR = -13,
NPP_DATA_TYPE_ERROR = -12,
NPP_OUT_OFF_RANGE_ERROR = -11,
NPP_DIVIDE_BY_ZERO_ERROR = -10,
NPP_MEMORY_ALLOCATION_ERR = -9,
NPP_NULL_POINTER_ERROR = -8,
NPP_RANGE_ERROR = -7,
NPP_SIZE_ERROR = -6,
NPP_BAD_ARGUMENT_ERROR = -5,
NPP_NO_MEMORY_ERROR = -4,
NPP_NOT_IMPLEMENTED_ERROR = -3,
NPP_ERROR = -2,
NPP_ERROR_RESERVED = -1,
NPP_NO_ERROR = 0,
NPP_SUCCESS = NPP_NO_ERROR,

```

NPP_NO_OPERATION_WARNING = 1,
NPP_DIVIDE_BY_ZERO_WARNING = 6,
NPP_AFFINE_QUAD_INCORRECT_WARNING = 28,
NPP_WRONG_INTERSECTION_ROI_WARNING = 29,
NPP_WRONG_INTERSECTION_QUAD_WARNING = 30,
NPP_DOUBLE_SIZE_WARNING = 35,
NPP_MISALIGNED_DST_ROI_WARNING = 10000 }

```

Error Status Codes.

- enum NppGpuComputeCapability {


```

NPP_CUDA_UNKNOWN_VERSION = -1,
NPP_CUDA_NOT_CAPABLE = 0,
NPP_CUDA_1_0 = 100,
NPP_CUDA_1_1 = 110,
NPP_CUDA_1_2 = 120,
NPP_CUDA_1_3 = 130,
NPP_CUDA_2_0 = 200,
NPP_CUDA_2_1 = 210,
NPP_CUDA_3_0 = 300,
NPP_CUDA_3_2 = 320,
NPP_CUDA_3_5 = 350,
NPP_CUDA_3_7 = 370,
NPP_CUDA_5_0 = 500,
NPP_CUDA_5_2 = 520,
NPP_CUDA_5_3 = 530,
NPP_CUDA_6_0 = 600 }

```
- enum NppiAxis {


```

NPP_HORIZONTAL_AXIS,
NPP_VERTICAL_AXIS,
NPP_BOTH_AXIS }

```
- enum NppCmpOp {


```

NPP_CMP_LESS,
NPP_CMP_LESS_EQ,
NPP_CMP_EQ,
NPP_CMP_GREATER_EQ,
NPP_CMP_GREATER }

```
- enum NppRoundMode {


```

NPP_RND_NEAR,
NPP_ROUND_NEAREST_TIES_TO_EVEN = NPP_RND_NEAR,
NPP_RND_FINANCIAL,
NPP_ROUND_NEAREST_TIES_AWAY_FROM_ZERO = NPP_RND_FINANCIAL,
NPP_RND_ZERO,
NPP_ROUND_TOWARD_ZERO = NPP_RND_ZERO }

```

Rounding Modes.

- enum `NppiBorderType` {
`NPP_BORDER_UNDEFINED` = 0,
`NPP_BORDER_NONE` = `NPP_BORDER_UNDEFINED`,
`NPP_BORDER_CONSTANT` = 1,
`NPP_BORDER_REPLICATE` = 2,
`NPP_BORDER_WRAP` = 3,
`NPP_BORDER_MIRROR` = 4 }
- enum `NppHintAlgorithm` {
`NPP_ALG_HINT_NONE`,
`NPP_ALG_HINT_FAST`,
`NPP_ALG_HINT_ACCURATE` }
- enum `NppiAlphaOp` {
`NPPI_OP_ALPHA_OVER`,
`NPPI_OP_ALPHA_IN`,
`NPPI_OP_ALPHA_OUT`,
`NPPI_OP_ALPHA_ATOP`,
`NPPI_OP_ALPHA_XOR`,
`NPPI_OP_ALPHA_PLUS`,
`NPPI_OP_ALPHA_OVER_PREMUL`,
`NPPI_OP_ALPHA_IN_PREMUL`,
`NPPI_OP_ALPHA_OUT_PREMUL`,
`NPPI_OP_ALPHA_ATOP_PREMUL`,
`NPPI_OP_ALPHA_XOR_PREMUL`,
`NPPI_OP_ALPHA_PLUS_PREMUL`,
`NPPI_OP_ALPHA_PREMUL` }
- enum `NppsZCType` {
`nppZCR`,
`nppZCXor`,
`nppZCC` }
- enum `NppiHuffmanTableType` {
`nppiDCTable`,
`nppiACTable` }

7.2.1 Define Documentation

7.2.1.1 #define NPP_MAX_16S (32767)

Maximum 16-bit signed integer.

7.2.1.2 #define NPP_MAX_16U (65535)

Maximum 16-bit unsigned integer.

7.2.1.3 #define NPP_MAX_32S (2147483647)

Maximum 32-bit signed integer.

7.2.1.4 #define NPP_MAX_32U (4294967295U)

Maximum 32-bit unsigned integer.

7.2.1.5 #define NPP_MAX_64S (9223372036854775807LL)

Maximum 64-bit signed integer.

7.2.1.6 #define NPP_MAX_64U (18446744073709551615ULL)

Maximum 64-bit unsigned integer.

7.2.1.7 #define NPP_MAX_8S (127)

Maximum 8-bit signed integer.

7.2.1.8 #define NPP_MAX_8U (255)

Maximum 8-bit unsigned integer.

7.2.1.9 #define NPP_MAXABS_32F (3.402823466e+38f)

Largest positive 32-bit floating point value.

7.2.1.10 #define NPP_MAXABS_64F (1.7976931348623158e+308)

Largest positive 64-bit floating point value.

7.2.1.11 #define NPP_MIN_16S (-32767 - 1)

Minimum 16-bit signed integer.

7.2.1.12 #define NPP_MIN_16U (0)

Minimum 16-bit unsigned integer.

7.2.1.13 #define NPP_MIN_32S (-2147483647 - 1)

Minimum 32-bit signed integer.

7.2.1.14 #define NPP_MIN_32U (0)

Minimum 32-bit unsigned integer.

7.2.1.15 #define NPP_MIN_64S (-9223372036854775807LL - 1)

Minimum 64-bit signed integer.

7.2.1.16 #define NPP_MIN_64U (0)

Minimum 64-bit unsigned integer.

7.2.1.17 #define NPP_MIN_8S (-127 - 1)

Minimum 8-bit signed integer.

7.2.1.18 #define NPP_MIN_8U (0)

Minimum 8-bit unsigned integer.

7.2.1.19 #define NPP_MINABS_32F (1.175494351e-38f)

Smallest positive 32-bit floating point value.

7.2.1.20 #define NPP_MINABS_64F (2.2250738585072014e-308)

Smallest positive 64-bit floating point value.

7.2.2 Enumeration Type Documentation**7.2.2.1 enum NppCmpOp**

Enumerator:

NPP_CMP_LESS

NPP_CMP_LESS_EQ

NPP_CMP_EQ

NPP_CMP_GREATER_EQ

NPP_CMP_GREATER

7.2.2.2 enum NppGpuComputeCapability

Enumerator:

NPP_CUDA_UNKNOWN_VERSION Indicates that the compute-capability query failed.

NPP_CUDA_NOT_CAPABLE Indicates that no CUDA capable device was found.

NPP_CUDA_1_0 Indicates that CUDA 1.0 capable device is machine's default device.
NPP_CUDA_1_1 Indicates that CUDA 1.1 capable device is machine's default device.
NPP_CUDA_1_2 Indicates that CUDA 1.2 capable device is machine's default device.
NPP_CUDA_1_3 Indicates that CUDA 1.3 capable device is machine's default device.
NPP_CUDA_2_0 Indicates that CUDA 2.0 capable device is machine's default device.
NPP_CUDA_2_1 Indicates that CUDA 2.1 capable device is machine's default device.
NPP_CUDA_3_0 Indicates that CUDA 3.0 capable device is machine's default device.
NPP_CUDA_3_2 Indicates that CUDA 3.2 capable device is machine's default device.
NPP_CUDA_3_5 Indicates that CUDA 3.5 capable device is machine's default device.
NPP_CUDA_3_7 Indicates that CUDA 3.7 capable device is machine's default device.
NPP_CUDA_5_0 Indicates that CUDA 5.0 capable device is machine's default device.
NPP_CUDA_5_2 Indicates that CUDA 5.2 capable device is machine's default device.
NPP_CUDA_5_3 Indicates that CUDA 5.3 capable device is machine's default device.
NPP_CUDA_6_0 Indicates that CUDA 6.0 or better is machine's default device.

7.2.2.3 enum NppHintAlgorithm

Enumerator:

NPP_ALG_HINT_NONE
NPP_ALG_HINT_FAST
NPP_ALG_HINT_ACCURATE

7.2.2.4 enum NppiAlphaOp

Enumerator:

NPPI_OP_ALPHA_OVER
NPPI_OP_ALPHA_IN
NPPI_OP_ALPHA_OUT
NPPI_OP_ALPHA_ATOP
NPPI_OP_ALPHA_XOR
NPPI_OP_ALPHA_PLUS
NPPI_OP_ALPHA_OVER_PREMUL
NPPI_OP_ALPHA_IN_PREMUL
NPPI_OP_ALPHA_OUT_PREMUL
NPPI_OP_ALPHA_ATOP_PREMUL
NPPI_OP_ALPHA_XOR_PREMUL
NPPI_OP_ALPHA_PLUS_PREMUL
NPPI_OP_ALPHA_PREMUL

7.2.2.5 enum NppiAxis**Enumerator:***NPP_HORIZONTAL_AXIS**NPP_VERTICAL_AXIS**NPP_BOTH_AXIS***7.2.2.6 enum NppiBayerGridPosition**

Bayer Grid Position Registration.

Enumerator:*NPPI_BAYER_BGGR* Default registration position.*NPPI_BAYER_RGGB**NPPI_BAYER_GBRG**NPPI_BAYER_GRBG***7.2.2.7 enum NppiBorderType****Enumerator:***NPP_BORDER_UNDEFINED**NPP_BORDER_NONE**NPP_BORDER_CONSTANT**NPP_BORDER_REPLICATE**NPP_BORDER_WRAP**NPP_BORDER_MIRROR***7.2.2.8 enum NppiHuffmanTableType****Enumerator:***nppiDCTable* DC Table.*nppiACTable* AC Table.**7.2.2.9 enum NppiInterpolationMode**

Filtering methods.

Enumerator:*NPPI_INTER_UNDEFINED**NPPI_INTER_NN* Nearest neighbor filtering.*NPPI_INTER_LINEAR* Linear interpolation.

NPPI_INTER_CUBIC Cubic interpolation.

NPPI_INTER_CUBIC2P_BSPLINE Two-parameter cubic filter (B=1, C=0).

NPPI_INTER_CUBIC2P_CATMULLROM Two-parameter cubic filter (B=0, C=1/2).

NPPI_INTER_CUBIC2P_B05C03 Two-parameter cubic filter (B=1/2, C=3/10).

NPPI_INTER_SUPER Super sampling.

NPPI_INTER_LANCZOS Lanczos filtering.

NPPI_INTER_LANCZOS3_ADVANCED Generic Lanczos filtering with order 3.

NPPI_SMOOTH_EDGE Smooth edge filtering.

7.2.2.10 enum NppiMaskSize

Fixed filter-kernel sizes.

Enumerator:

NPP_MASK_SIZE_1_X_3

NPP_MASK_SIZE_1_X_5

NPP_MASK_SIZE_3_X_1

NPP_MASK_SIZE_5_X_1

NPP_MASK_SIZE_3_X_3

NPP_MASK_SIZE_5_X_5

NPP_MASK_SIZE_7_X_7

NPP_MASK_SIZE_9_X_9

NPP_MASK_SIZE_11_X_11

NPP_MASK_SIZE_13_X_13

NPP_MASK_SIZE_15_X_15

7.2.2.11 enum NppRoundMode

Rounding Modes.

The enumerated rounding modes are used by a large number of NPP primitives to allow the user to specify the method by which fractional values are converted to integer values. Also see [Rounding Modes](#).

For NPP release 5.5 new names for the three rounding modes are introduced that are based on the naming conventions for rounding modes set forth in the IEEE-754 floating-point standard. Developers are encouraged to use the new, longer names to be future proof as the legacy names will be deprecated in subsequent NPP releases.

Enumerator:

NPP_RND_NEAR Round to the nearest even integer.

All fractional numbers are rounded to their nearest integer. The ambiguous cases (i.e. $\langle \text{integer} \rangle.5$) are rounded to the closest even integer. E.g.

- `roundNear(0.5) = 0`
- `roundNear(0.6) = 1`
- `roundNear(1.5) = 2`

- `roundNear(-1.5) = -2`

NPP_ROUND_NEAREST_TIES_TO_EVEN Alias name for [NPP_RND_NEAR](#).

NPP_RND_FINANCIAL Round according to financial rule.

All fractional numbers are rounded to their nearest integer. The ambiguous cases (i.e. `<integer>.5`) are rounded away from zero. E.g.

- `roundFinancial(0.4) = 0`
- `roundFinancial(0.5) = 1`
- `roundFinancial(-1.5) = -2`

NPP_ROUND_NEAREST_TIES_AWAY_FROM_ZERO Alias name for [NPP_RND_FINANCIAL](#).

NPP_RND_ZERO Round towards zero (truncation).

All fractional numbers of the form `<integer>.<decimals>` are truncated to `<integer>`.

- `roundZero(1.5) = 1`
- `roundZero(1.9) = 1`
- `roundZero(-2.5) = -2`

NPP_ROUND_TOWARD_ZERO Alias name for [NPP_RND_ZERO](#).

7.2.2.12 enum NppStatus

Error Status Codes.

Almost all NPP function return error-status information using these return codes. Negative return codes indicate errors, positive return codes indicate warnings, a return code of 0 indicates success.

Enumerator:

NPP_NOT_SUPPORTED_MODE_ERROR

NPP_INVALID_HOST_POINTER_ERROR

NPP_INVALID_DEVICE_POINTER_ERROR

NPP_LUT_PALETTE_BITSIZE_ERROR

NPP_ZC_MODE_NOT_SUPPORTED_ERROR ZeroCrossing mode not supported.

NPP_NOT_SUFFICIENT_COMPUTE_CAPABILITY

NPP_TEXTURE_BIND_ERROR

NPP_WRONG_INTERSECTION_ROI_ERROR

NPP_HAAR_CLASSIFIER_PIXEL_MATCH_ERROR

NPP_MEMFREE_ERROR

NPP_MEMSET_ERROR

NPP_MEMCPY_ERROR

NPP_ALIGNMENT_ERROR

NPP_CUDA_KERNEL_EXECUTION_ERROR

NPP_ROUND_MODE_NOT_SUPPORTED_ERROR Unsupported round mode.

NPP_QUALITY_INDEX_ERROR Image pixels are constant for quality index.

NPP_RESIZE_NO_OPERATION_ERROR One of the output image dimensions is less than 1 pixel.

NPP_OVERFLOW_ERROR Number overflows the upper or lower limit of the data type.

NPP_NOT_EVEN_STEP_ERROR Step value is not pixel multiple.

NPP_HISTOGRAM_NUMBER_OF_LEVELS_ERROR Number of levels for histogram is less than 2.

NPP_LUT_NUMBER_OF_LEVELS_ERROR Number of levels for LUT is less than 2.

NPP_CORRUPTED_DATA_ERROR Processed data is corrupted.

NPP_CHANNEL_ORDER_ERROR Wrong order of the destination channels.

NPP_ZERO_MASK_VALUE_ERROR All values of the mask are zero.

NPP_QUADRANGLE_ERROR The quadrangle is nonconvex or degenerates into triangle, line or point.

NPP_RECTANGLE_ERROR Size of the rectangle region is less than or equal to 1.

NPP_COEFFICIENT_ERROR Unallowable values of the transformation coefficients.

NPP_NUMBER_OF_CHANNELS_ERROR Bad or unsupported number of channels.

NPP_COI_ERROR Channel of interest is not 1, 2, or 3.

NPP_DIVISOR_ERROR Divisor is equal to zero.

NPP_CHANNEL_ERROR Illegal channel index.

NPP_STRIDE_ERROR Stride is less than the row length.

NPP_ANCHOR_ERROR Anchor point is outside mask.

NPP_MASK_SIZE_ERROR Lower bound is larger than upper bound.

NPP_RESIZE_FACTOR_ERROR

NPP_INTERPOLATION_ERROR

NPP_MIRROR_FLIP_ERROR

NPP_MOMENT_00_ZERO_ERROR

NPP_THRESHOLD_NEGATIVE_LEVEL_ERROR

NPP_THRESHOLD_ERROR

NPP_CONTEXT_MATCH_ERROR

NPP_FFT_FLAG_ERROR

NPP_FFT_ORDER_ERROR

NPP_STEP_ERROR Step is less or equal zero.

NPP_SCALE_RANGE_ERROR

NPP_DATA_TYPE_ERROR

NPP_OUT_OFF_RANGE_ERROR

NPP_DIVIDE_BY_ZERO_ERROR

NPP_MEMORY_ALLOCATION_ERR

NPP_NULL_POINTER_ERROR

NPP_RANGE_ERROR

NPP_SIZE_ERROR

NPP_BAD_ARGUMENT_ERROR

NPP_NO_MEMORY_ERROR

NPP_NOT_IMPLEMENTED_ERROR

NPP_ERROR

NPP_ERROR_RESERVED

NPP_NO_ERROR Error free operation.

NPP_SUCCESS Successful operation (same as ***NPP_NO_ERROR***).

NPP_NO_OPERATION_WARNING Indicates that no operation was performed.

NPP_DIVIDE_BY_ZERO_WARNING Divisor is zero however does not terminate the execution.

NPP_AFFINE_QUAD_INCORRECT_WARNING Indicates that the quadrangle passed to one of affine warping functions doesn't have necessary properties.

First 3 vertices are used, the fourth vertex discarded.

NPP_WRONG_INTERSECTION_ROI_WARNING The given ROI has no intersection with either the source or destination ROI.

Thus no operation was performed.

NPP_WRONG_INTERSECTION_QUAD_WARNING The given quadrangle has no intersection with either the source or destination ROI.

Thus no operation was performed.

NPP_DOUBLE_SIZE_WARNING Image size isn't multiple of two.

Indicates that in case of 422/411/420 sampling the ROI width/height was modified for proper processing.

NPP_MISALIGNED_DST_ROI_WARNING Speed reduction due to uncoalesced memory accesses warning.

7.2.2.13 enum NppsZCType

Enumerator:

nppZCR sign change

nppZCXor sign change XOR

nppZCC sign change count_0

7.3 Basic NPP Data Types

Data Structures

- struct [NPP_ALIGN_8](#)
Complex Number This struct represents an unsigned int complex number.
- struct [NPP_ALIGN_16](#)
Complex Number This struct represents a long long complex number.

Typedefs

- typedef unsigned char [Npp8u](#)
8-bit unsigned chars
- typedef signed char [Npp8s](#)
8-bit signed chars
- typedef unsigned short [Npp16u](#)
16-bit unsigned integers
- typedef short [Npp16s](#)
16-bit signed integers
- typedef unsigned int [Npp32u](#)
32-bit unsigned integers
- typedef int [Npp32s](#)
32-bit signed integers
- typedef unsigned long long [Npp64u](#)
64-bit unsigned integers
- typedef long long [Npp64s](#)
64-bit signed integers
- typedef float [Npp32f](#)
32-bit (IEEE) floating-point numbers
- typedef double [Npp64f](#)
64-bit floating-point numbers
- typedef struct [NPP_ALIGN_8](#) [Npp32uc](#)
Complex Number This struct represents an unsigned int complex number.
- typedef struct [NPP_ALIGN_8](#) [Npp32sc](#)
Complex Number This struct represents a signed int complex number.

- typedef struct [NPP_ALIGN_8 Npp32fc](#)
Complex Number This struct represents a single floating-point complex number.
- typedef struct [NPP_ALIGN_16 Npp64sc](#)
Complex Number This struct represents a long long complex number.
- typedef struct [NPP_ALIGN_16 Npp64fc](#)
Complex Number This struct represents a double floating-point complex number.

Functions

- struct [__align__](#) (2)
Complex Number This struct represents an unsigned char complex number.
- struct [__align__](#) (4)
Complex Number This struct represents an unsigned short complex number.

Variables

- [Npp8uc](#)
- [Npp16uc](#)
- [Npp16sc](#)

7.3.1 Typedef Documentation

7.3.1.1 typedef short Npp16s

16-bit signed integers

7.3.1.2 typedef unsigned short Npp16u

16-bit unsigned integers

7.3.1.3 typedef float Npp32f

32-bit (IEEE) floating-point numbers

7.3.1.4 typedef struct NPP_ALIGN_8 Npp32fc

Complex Number This struct represents a single floating-point complex number.

7.3.1.5 typedef int Npp32s

32-bit signed integers

7.3.1.6 `typedef struct NPP_ALIGN_8 Npp32sc`

Complex Number This struct represents a signed int complex number.

7.3.1.7 `typedef unsigned int Npp32u`

32-bit unsigned integers

7.3.1.8 `typedef struct NPP_ALIGN_8 Npp32uc`

Complex Number This struct represents an unsigned int complex number.

7.3.1.9 `typedef double Npp64f`

64-bit floating-point numbers

7.3.1.10 `typedef struct NPP_ALIGN_16 Npp64fc`

Complex Number This struct represents a double floating-point complex number.

7.3.1.11 `typedef long long Npp64s`

64-bit signed integers

7.3.1.12 `typedef struct NPP_ALIGN_16 Npp64sc`

Complex Number This struct represents a long long complex number.

7.3.1.13 `typedef unsigned long long Npp64u`

64-bit unsigned integers

7.3.1.14 `typedef signed char Npp8s`

8-bit signed chars

7.3.1.15 `typedef unsigned char Npp8u`

8-bit unsigned chars

7.3.2 Function Documentation

7.3.2.1 `struct __align__ (4) [read]`

Complex Number This struct represents an unsigned short complex number.

Complex Number This struct represents a short complex number.

< Real part
< Imaginary part
< Real part
< Imaginary part

7.3.2.2 `struct __align__(2) [read]`

Complex Number This struct represents an unsigned char complex number.

< Real part
< Imaginary part

7.3.3 Variable Documentation

7.3.3.1 `Npp16sc`

7.3.3.2 `Npp16uc`

7.3.3.3 `Npp8uc`

7.4 NPP Image Processing

Modules

- [Arithmetic and Logical Operations](#)
• [Color and Sampling Conversion](#)
Routines manipulating an image's color model and sampling format.
- [Compression](#)
Image compression primitives.
- [Labeling and Segmentation](#)
Pixel labeling and image segmentation operations.
- [Data Exchange and Initialization](#)
Primitives for initializing, copying and converting image data.
- [Filtering Functions](#)
Linear and non-linear image filtering functions.
- [Geometry Transforms](#)
Routines manipulating an image's geometry.
- [Linear Transforms](#)
Linear image transformations.
- [Morphological Operations](#)
Morphological image operations.
- [Statistical Operations](#)
Primitives for computing the statistical properties of an image.
- [Memory Management](#)
Routines for allocating and deallocating pitched image storage.
- [Threshold and Compare Operations](#)
Methods for pixel-wise threshold and compare operations.

7.5 Arithmetic and Logical Operations

Modules

- [Arithmetic Operations](#)
- [Logical Operations](#)
- [Alpha Composition](#)

7.6 Arithmetic Operations

Modules

- [AddC](#)
Adds a constant value to each pixel of an image.
- [MulC](#)
Multiplies each pixel of an image by a constant value.
- [MulCScale](#)
Multiplies each pixel of an image by a constant value then scales the result by the maximum value for the data bit width.
- [SubC](#)
Subtracts a constant value from each pixel of an image.
- [DivC](#)
Divides each pixel of an image by a constant value.
- [AbsDiffC](#)
Determines absolute difference between each pixel of an image and a constant value.
- [Add](#)
Pixel by pixel addition of two images.
- [AddSquare](#)
Pixel by pixel addition of squared pixels from source image to floating point pixel values of destination image.
- [AddProduct](#)
Pixel by pixel addition of product of pixels from two source images to floating point pixel values of destination image.
- [AddWeighted](#)
Pixel by pixel addition of alpha weighted pixel values from a source image to floating point pixel values of destination image.
- [Mul](#)
Pixel by pixel multiply of two images.
- [MulScale](#)
Pixel by pixel multiplies each pixel of two images then scales the result by the maximum value for the data bit width.
- [Sub](#)
Pixel by pixel subtraction of two images.
- [Div](#)
Pixel by pixel division of two images.

- [Div_Round](#)

Pixel by pixel division of two images using result rounding modes.

- [Abs](#)

Absolute value of each pixel value in an image.

- [AbsDiff](#)

Pixel by pixel absolute difference between two images.

- [Sqr](#)

Square each pixel in an image.

- [Sqrt](#)

Pixel by pixel square root of each pixel in an image.

- [Ln](#)

Pixel by pixel natural logarithm of each pixel in an image.

- [Exp](#)

Exponential value of each pixel in an image.

7.7 AddC

Adds a constant value to each pixel of an image.

Functions

- **NppStatus** **nppiAddC_8u_C1RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.
- **NppStatus** **nppiAddC_8u_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.
- **NppStatus** **nppiAddC_8u_C3RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.
- **NppStatus** **nppiAddC_8u_C3IRSfs** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel 8-bit unsigned char in place image add constant, scale, then clamp to saturated value.
- **NppStatus** **nppiAddC_8u_AC4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha image add constant, scale, then clamp to saturated value.
- **NppStatus** **nppiAddC_8u_AC4IRSfs** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.
- **NppStatus** **nppiAddC_8u_C4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.
- **NppStatus** **nppiAddC_8u_C4IRSfs** (const **Npp8u** aConstants[4], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.
- **NppStatus** **nppiAddC_16u_C1RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.
- **NppStatus** **nppiAddC_16u_C1IRSfs** (const **Npp16u** nConstant, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.
- **NppStatus** **nppiAddC_16u_C3RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16u_C3IRSfs` (const `Npp16u` aConstants[3], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16u_AC4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16u_AC4IRSfs` (const `Npp16u` aConstants[3], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16u_C4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16u_C4IRSfs` (const `Npp16u` aConstants[4], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16s_C1RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16s_C3RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short channel image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16s_C3IRSfs` (const `Npp16s` aConstants[3], `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16s_AC4RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16s_AC4IRSfs` (const `Npp16s` aConstants[3], `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16s_C4RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` aConstants[4], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16s_C4IRSfs` (const `Npp16s` aConstants[4], `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16sc_C1RSfs` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16sc_C1RSfs` (const `Npp16sc` nConstant, `Npp16sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16sc_C3RSfs` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16sc_C3IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16sc_AC4RSfs` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_16sc_AC4IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_32s_C1RSfs` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_32s_C3RSfs` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

- `NppStatus nppiAddC_32s_C3IRSfs` (const `Npp32s` aConstants[3], `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

- **NppStatus nppiAddC_32sc_C1RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** nConstant, **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.
- **NppStatus nppiAddC_32sc_C1IRSfs** (const **Npp32sc** nConstant, **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.
- **NppStatus nppiAddC_32sc_C3RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** aConstants[3], **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.
- **NppStatus nppiAddC_32sc_C3IRSfs** (const **Npp32sc** aConstants[3], **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.
- **NppStatus nppiAddC_32sc_AC4RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** aConstants[3], **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.
- **NppStatus nppiAddC_32sc_AC4IRSfs** (const **Npp32sc** aConstants[3], **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.
- **NppStatus nppiAddC_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** nConstant, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image add constant.
- **NppStatus nppiAddC_32f_C1IR** (const **Npp32f** nConstant, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image add constant.
- **NppStatus nppiAddC_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** aConstants[3], **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel image add constant.
- **NppStatus nppiAddC_32f_C3IR** (const **Npp32f** aConstants[3], **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel in place image add constant.
- **NppStatus nppiAddC_32f_AC4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** aConstants[3], **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel with unmodified alpha image add constant.
- **NppStatus nppiAddC_32f_AC4IR** (const **Npp32f** aConstants[3], **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 32-bit floating point channel with unmodified alpha in place image add constant.

- `NppStatus nppiAddC_32f_C4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` aConstants[4], `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four 32-bit floating point channel image add constant.

- `NppStatus nppiAddC_32f_C4IR` (const `Npp32f` aConstants[4], `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Four 32-bit floating point channel in place image add constant.

- `NppStatus nppiAddC_32fc_C1R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` nConstant, `Npp32fc` *pDst, int nDstStep, `NppiSize` oSizeROI)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

- `NppStatus nppiAddC_32fc_C1IR` (const `Npp32fc` nConstant, `Npp32fc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

- `NppStatus nppiAddC_32fc_C3R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` aConstants[3], `Npp32fc` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

- `NppStatus nppiAddC_32fc_C3IR` (const `Npp32fc` aConstants[3], `Npp32fc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

- `NppStatus nppiAddC_32fc_AC4R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` aConstants[3], `Npp32fc` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image add constant.

- `NppStatus nppiAddC_32fc_AC4IR` (const `Npp32fc` aConstants[3], `Npp32fc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image add constant.

- `NppStatus nppiAddC_32fc_C4R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` aConstants[4], `Npp32fc` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

- `NppStatus nppiAddC_32fc_C4IR` (const `Npp32fc` aConstants[4], `Npp32fc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

7.7.1 Detailed Description

Adds a constant value to each pixel of an image.

7.7.2 Function Documentation

7.7.2.1 `NppStatus nppiAddC_16s_AC4IRSfs (const Npp16s aConstants[3], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.2 `NppStatus nppiAddC_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s aConstants[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.3 `NppStatus nppiAddC_16s_C1IRSfs (const Npp16s nConstant, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.4 NppStatus nppiAddC_16s_C1RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s *nConstant*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.5 NppStatus nppiAddC_16s_C3IRSfs (const Npp16s *aConstants*[3], Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.6 NppStatus nppiAddC_16s_C3RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[3], Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.7 NppStatus nppiAddC_16s_C4IRSfs (const Npp16s *aConstants*[4], Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.8 NppStatus nppiAddC_16s_C4RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[4], Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.9 NppStatus nppiAddC_16sc_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.10 NppStatus nppiAddC_16sc_AC4RSfs (const Npp16sc * pSrcI, int nSrcIStep, const Npp16sc aConstants[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters:

pSrcI [Source-Image Pointer](#).

nSrcIStep [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.11 **NppStatus nppiAddC_16sc_C1IRSfs** (const Npp16sc *nConstant*, Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.12 **NppStatus nppiAddC_16sc_C1RSfs** (const Npp16sc * *pSrc1*, int *nSrc1Step*, const Npp16sc *nConstant*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.13 **NppStatus nppiAddC_16sc_C3IRSfs** (const Npp16sc *aConstants*[3], Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.14 NppStatus npAddC_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.15 NppStatus npAddC_16u_AC4IRSfs (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.16 **NppStatus nppiAddC_16u_AC4RSfs** (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[3], Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.17 **NppStatus nppiAddC_16u_C1IRSfs** (const Npp16u *nConstant*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

Parameters:

nConstant [Constant](#).
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.18 **NppStatus nppiAddC_16u_C1RSfs** (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u *nConstant*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nConstant [Constant](#).
pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.19 NppStatus nppiAddC_16u_C3IRSfs (const Npp16u aConstants[3], Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.20 NppStatus nppiAddC_16u_C3RSfs (const Npp16u *pSrcI, int nSrcIStep, const Npp16u aConstants[3], Npp16u *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrcI Source-Image Pointer.

nSrcIStep Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.21 **NppStatus nppiAddC_16u_C4IRSfs** (const Npp16u *aConstants*[4], Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.22 **NppStatus nppiAddC_16u_C4RSfs** (const Npp16u * *pSrcI*, int *nSrcIStep*, const Npp16u *aConstants*[4], Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrcI [Source-Image Pointer](#).
nSrcIStep [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.23 **NppStatus nppiAddC_32f_AC4IR** (const Npp32f *aConstants*[3], Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha in place image add constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.24 **NppStatus nppiAddC_32f_AC4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image add constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.25 **NppStatus nppiAddC_32f_C1IR** (const Npp32f *nConstant*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image add constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.26 **NppStatus nppiAddC_32f_C1R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *nConstant*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image add constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.27 **NppStatus nppiAddC_32f_C3IR** (const Npp32f *aConstants*[3], Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel in place image add constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.28 **NppStatus nppiAddC_32f_C3R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel image add constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.29 **NppStatus nppiAddC_32f_C4IR** (const Npp32f *aConstants*[4], Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image add constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.30 **NppStatus nppiAddC_32f_C4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[4], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image add constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.31 **NppStatus nppiAddC_32fc_AC4IR** (const Npp32fc *aConstants*[3], Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image add constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.32 **NppStatus nppiAddC_32fc_AC4R** (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image add constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.33 **NppStatus nppiAddC_32fc_C1IR** (const Npp32fc *nConstant*, Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.34 **NppStatus nppiAddC_32fc_C1R** (const Npp32fc * *pSrcI*, int *nSrcIStep*, const Npp32fc *nConstant*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

Parameters:

pSrcI Source-Image Pointer.
nSrcIStep Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.35 **NppStatus nppiAddC_32fc_C3IR** (const Npp32fc *aConstants*[3], Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.36 NppStatus nppiAddC_32fc_C3R (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.37 NppStatus nppiAddC_32fc_C4IR (const Npp32fc *aConstants*[4], Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.38 NppStatus nppiAddC_32fc_C4R (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[4], Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.39 NppStatus nppiAddC_32s_C1IRSfs (const Npp32s *nConstant*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

Parameters:

nConstant [Constant](#).

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.40 NppStatus nppiAddC_32s_C1RSfs (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *nConstant*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

nConstant [Constant](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.41 NppStatus nppiAddC_32s_C3IRSfs (const Npp32s *aConstants*[3], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants [fixed size array of constant values, one per channel](#).

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.42 **NppStatus nppiAddC_32s_C3RSfs** (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.43 **NppStatus nppiAddC_32sc_AC4IRSfs** (const Npp32sc *aConstants*[3], Npp32sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.44 NppStatus nppiAddC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.45 NppStatus nppiAddC_32sc_C1IRSfs (const Npp32sc nConstant, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.46 NppStatus nppiAddC_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc nConstant, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.

nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.47 **NppStatus nppiAddC_32sc_C3IRSfs** (const Npp32sc *aConstants*[3], Npp32sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.48 **NppStatus nppiAddC_32sc_C3RSfs** (const Npp32sc * *pSrc1*, int *nSrc1Step*, const Npp32sc *aConstants*[3], Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.49 NppStatus nppiAddC_8u_AC4IRSfs (const Npp8u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel..

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.50 NppStatus nppiAddC_8u_AC4RSfs (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel..

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.7.2.51 NppStatus nppiAddC_8u_C1IRSfs (const Npp8u *nConstant*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.52 `NppStatus nppiAddC_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nConstant Constant.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.53 `NppStatus nppiAddC_8u_C3IRSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel..

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.54 `NppStatus nppiAddC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel..
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.55 NppStatus nppiAddC_8u_C4IRSfs (const Npp8u *aConstants*[4], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.7.2.56 NppStatus nppiAddC_8u_C4RSfs (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel..
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8 MulC

Multiplies each pixel of an image by a constant value.

Functions

- **NppStatus nppiMulC_8u_C1RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.
- **NppStatus nppiMulC_8u_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.
- **NppStatus nppiMulC_8u_C3RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.
- **NppStatus nppiMulC_8u_C3IRSfs** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant, scale, then clamp to saturated value.
- **NppStatus nppiMulC_8u_AC4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.
- **NppStatus nppiMulC_8u_AC4IRSfs** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.
- **NppStatus nppiMulC_8u_C4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.
- **NppStatus nppiMulC_8u_C4IRSfs** (const **Npp8u** aConstants[4], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.
- **NppStatus nppiMulC_16u_C1RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.
- **NppStatus nppiMulC_16u_C1IRSfs** (const **Npp16u** nConstant, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.
- **NppStatus nppiMulC_16u_C3RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16u_C3IRSfs` (const `Npp16u` aConstants[3], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16u_AC4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16u_AC4IRSfs` (const `Npp16u` aConstants[3], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16u_C4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16u_C4IRSfs` (const `Npp16u` aConstants[4], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16s_C1RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16s_C3RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16s_C3IRSfs` (const `Npp16s` aConstants[3], `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16s_AC4RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16s_AC4IRSfs` (const `Npp16s` aConstants[3], `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_16s_C4RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` aConstants[4], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.
- `NppStatus nppiMulC_16s_C4IRSfs` (const `Npp16s` aConstants[4], `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.
- `NppStatus nppiMulC_16sc_C1RSfs` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.
- `NppStatus nppiMulC_16sc_C1IRSfs` (const `Npp16sc` nConstant, `Npp16sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.
- `NppStatus nppiMulC_16sc_C3RSfs` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.
- `NppStatus nppiMulC_16sc_C3IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.
- `NppStatus nppiMulC_16sc_AC4RSfs` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.
- `NppStatus nppiMulC_16sc_AC4IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.
- `NppStatus nppiMulC_32s_C1RSfs` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.
- `NppStatus nppiMulC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.
- `NppStatus nppiMulC_32s_C3RSfs` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Three 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.
- `NppStatus nppiMulC_32s_C3IRSfs` (const `Npp32s` aConstants[3], `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_32sc_C1RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` nConstant, `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_32sc_C1IRSfs` (const `Npp32sc` nConstant, `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_32sc_C3RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_32sc_C3IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_32sc_AC4RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_32sc_AC4IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

- `NppStatus nppiMulC_32f_C1R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` nConstant, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

One 32-bit floating point channel image multiply by constant.

- `NppStatus nppiMulC_32f_C1IR` (const `Npp32f` nConstant, `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

One 32-bit floating point channel in place image multiply by constant.

- `NppStatus nppiMulC_32f_C3R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three 32-bit floating point channel image multiply by constant.

- `NppStatus nppiMulC_32f_C3IR` (const `Npp32f` aConstants[3], `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Three 32-bit floating point channel in place image multiply by constant.

- `NppStatus nppiMulC_32f_AC4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four 32-bit floating point channel with unmodified alpha image multiply by constant.

- **NppStatus nppiMulC_32f_AC4IR** (const **Npp32f** aConstants[3], **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel with unmodified alpha in place image multiply by constant.
- **NppStatus nppiMulC_32f_C4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** aConstants[4], **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel image multiply by constant.
- **NppStatus nppiMulC_32f_C4IR** (const **Npp32f** aConstants[4], **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel in place image multiply by constant.
- **NppStatus nppiMulC_32fc_C1R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** nConstant, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.
- **NppStatus nppiMulC_32fc_C1IR** (const **Npp32fc** nConstant, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.
- **NppStatus nppiMulC_32fc_C3R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.
- **NppStatus nppiMulC_32fc_C3IR** (const **Npp32fc** aConstants[3], **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.
- **NppStatus nppiMulC_32fc_AC4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image multiply by constant.
- **NppStatus nppiMulC_32fc_AC4IR** (const **Npp32fc** aConstants[3], **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image multiply by constant.
- **NppStatus nppiMulC_32fc_C4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** aConstants[4], **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.
- **NppStatus nppiMulC_32fc_C4IR** (const **Npp32fc** aConstants[4], **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

7.8.1 Detailed Description

Multiplies each pixel of an image by a constant value.

7.8.2 Function Documentation

7.8.2.1 NppStatus nppiMulC_16s_AC4IRSfs (const Npp16s *aConstants*[3], Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.2 NppStatus nppiMulC_16s_AC4RSfs (const Npp16s * *pSrcI*, int *nSrcIStep*, const Npp16s *aConstants*[3], Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrcI [Source-Image Pointer](#).

nSrcIStep [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.3 `NppStatus nppiMulC_16s_C1IRSfs (const Npp16s nConstant, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.4 `NppStatus nppiMulC_16s_C1RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s nConstant, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.5 `NppStatus nppiMulC_16s_C3IRSfs (const Npp16s aConstants[3], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.6 NppStatus nppiMulC_16s_C3RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[3], Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.7 NppStatus nppiMulC_16s_C4IRSfs (const Npp16s *aConstants*[4], Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.8 NppStatus nppiMulC_16s_C4RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[4], Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.9 NppStatus nppiMulC_16sc_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.10 NppStatus nppiMulC_16sc_AC4RSfs (const Npp16sc * pSrcI, int nSrcIStep, const Npp16sc aConstants[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrcI [Source-Image Pointer](#).

nSrcIStep [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.11 **NppStatus nppiMulC_16sc_C1IRSfs** (const Npp16sc *nConstant*, Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.12 **NppStatus nppiMulC_16sc_C1RSfs** (const Npp16sc * *pSrc1*, int *nSrc1Step*, const Npp16sc *nConstant*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.13 **NppStatus nppiMulC_16sc_C3IRSfs** (const Npp16sc *aConstants*[3], Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.14 NppStatus nppiMulC_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.15 NppStatus nppiMulC_16u_AC4IRSfs (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.16 NppStatus nppiMulC_16u_AC4RSfs (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[3], Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.17 NppStatus nppiMulC_16u_C1IRSfs (const Npp16u *nConstant*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.18 NppStatus nppiMulC_16u_C1RSfs (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u *nConstant*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nConstant Constant.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.19 NppStatus nppiMulC_16u_C3IRSfs (const Npp16u aConstants[3], Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.20 NppStatus nppiMulC_16u_C3RSfs (const Npp16u *pSrcI, int nSrcIStep, const Npp16u aConstants[3], Npp16u *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrcI Source-Image Pointer.

nSrcIStep Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.21 **NppStatus nppiMulC_16u_C4IRSfs** (const Npp16u *aConstants*[4], Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.22 **NppStatus nppiMulC_16u_C4RSfs** (const Npp16u * *pSrcI*, int *nSrcIStep*, const Npp16u *aConstants*[4], Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrcI Source-Image Pointer.
nSrcIStep Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.23 **NppStatus nppiMulC_32f_AC4IR** (const Npp32f *aConstants*[3], Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha in place image multiply by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.24 **NppStatus nppiMulC_32f_AC4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image multiply by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.25 **NppStatus nppiMulC_32f_C1IR** (const Npp32f *nConstant*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image multiply by constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.26 **NppStatus nppiMulC_32f_C1R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *nConstant*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image multiply by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.27 NppStatus nppiMulC_32f_C3IR (const Npp32f aConstants[3], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three 32-bit floating point channel in place image multiply by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.28 NppStatus nppiMulC_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f aConstants[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)

Three 32-bit floating point channel image multiply by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.29 NppStatus nppiMulC_32f_C4IR (const Npp32f aConstants[4], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 32-bit floating point channel in place image multiply by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.30 **NppStatus nppiMulC_32f_C4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[4], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image multiply by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.31 **NppStatus nppiMulC_32fc_AC4IR** (const Npp32fc *aConstants*[3], Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image multiply by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.32 **NppStatus nppiMulC_32fc_AC4R** (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image multiply by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.33 NppStatus nppiMulC_32fc_C1IR (const Npp32fc *nConstant*, Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.34 NppStatus nppiMulC_32fc_C1R (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc *nConstant*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.35 NppStatus nppiMulC_32fc_C3IR (const Npp32fc *aConstants*[3], Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.36 **NppStatus nppiMulC_32fc_C3R** (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.37 **NppStatus nppiMulC_32fc_C4IR** (const Npp32fc *aConstants*[4], Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.38 **NppStatus nppiMulC_32fc_C4R** (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[4], Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.39 `NppStatus nppiMulC_32s_C1IRSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.40 `NppStatus nppiMulC_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nConstant Constant.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.41 `NppStatus nppiMulC_32s_C3IRSfs (const Npp32s aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.42 NppStatus nppiMulC_32s_C3RSfs (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.43 NppStatus nppiMulC_32sc_AC4IRSfs (const Npp32sc *aConstants*[3], Npp32sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.44 `NppStatus nppiMulC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.45 `NppStatus nppiMulC_32sc_C1IRSfs (const Npp32sc nConstant, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.46 `NppStatus nppiMulC_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc nConstant, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.

nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.47 **NppStatus nppiMulC_32sc_C3IRSfs** (const Npp32sc *aConstants*[3], Npp32sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.48 **NppStatus nppiMulC_32sc_C3RSfs** (const Npp32sc * *pSrc1*, int *nSrc1Step*, const Npp32sc *aConstants*[3], Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.49 **NppStatus nppiMulC_8u_AC4IRSfs** (const Npp8u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.50 **NppStatus nppiMulC_8u_AC4RSfs** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.8.2.51 **NppStatus nppiMulC_8u_C1IRSfs** (const Npp8u *nConstant*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.52 `NppStatus nppiMulC_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.53 `NppStatus nppiMulC_8u_C3IRSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.54 `NppStatus nppiMulC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.55 **NppStatus nppiMulC_8u_C4IRSfs** (const Npp8u *aConstants*[4], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.8.2.56 **NppStatus nppiMulC_8u_C4RSfs** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9 MulCScale

Multiplies each pixel of an image by a constant value then scales the result by the maximum value for the data bit width.

Functions

- `NppStatus nppiMulCScale_8u_C1R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` nConstant, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

One 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

- `NppStatus nppiMulCScale_8u_C1IR` (const `Npp8u` nConstant, `Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

One 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

- `NppStatus nppiMulCScale_8u_C3R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` aConstants[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

- `NppStatus nppiMulCScale_8u_C3IR` (const `Npp8u` aConstants[3], `Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant and scale by max bit width value.

- `NppStatus nppiMulCScale_8u_AC4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` aConstants[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant and scale by max bit width value.

- `NppStatus nppiMulCScale_8u_AC4IR` (const `Npp8u` aConstants[3], `Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale and scale by max bit width value.

- `NppStatus nppiMulCScale_8u_C4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` aConstants[4], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

- `NppStatus nppiMulCScale_8u_C4IR` (const `Npp8u` aConstants[4], `Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Four 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

- `NppStatus nppiMulCScale_16u_C1R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` nConstant, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)

One 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

- `NppStatus nppiMulCScale_16u_C1IR` (const `Npp16u` nConstant, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

One 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

- **NppStatus** **nppiMulCScale_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Three 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

- **NppStatus** **nppiMulCScale_16u_C3IR** (const **Npp16u** aConstants[3], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Three 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

- **NppStatus** **nppiMulCScale_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant and scale by max bit width value.

- **NppStatus** **nppiMulCScale_16u_AC4IR** (const **Npp16u** aConstants[3], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant and scale by max bit width value.

- **NppStatus** **nppiMulCScale_16u_C4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[4], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

- **NppStatus** **nppiMulCScale_16u_C4IR** (const **Npp16u** aConstants[4], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

7.9.1 Detailed Description

Multiplies each pixel of an image by a constant value then scales the result by the maximum value for the data bit width.

7.9.2 Function Documentation

7.9.2.1 **NppStatus** **nppiMulCScale_16u_AC4IR** (const **Npp16u** aConstants[3], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant and scale by max bit width value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.2 NppStatus nppiMulCScale_16u_AC4R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[3], Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant and scale by max bit width value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.9.2.3 NppStatus nppiMulCScale_16u_C1IR (const Npp16u *nConstant*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.9.2.4 NppStatus nppiMulCScale_16u_C1R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u *nConstant*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.9.2.5 NppStatus nppiMulCScale_16u_C3IR (const Npp16u *aConstants*[3], Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.6 NppStatus nppiMulCScale_16u_C3R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[3], Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.7 NppStatus nppiMulCScale_16u_C4IR (const Npp16u *aConstants*[4], Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.8 NppStatus nppiMulCScale_16u_C4R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[4], Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.9 NppStatus nppiMulCScale_8u_AC4IR (const Npp8u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale and scale by max bit width value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.10 NppStatus nppiMulCScale_8u_AC4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant and scale by max bit width value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.11 NppStatus nppiMulCScale_8u_C1IR (const Npp8u *nConstant*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.12 NppStatus nppiMulCScale_8u_C1R (const Npp8u * *pSrcI*, int *nSrcIStep*, const Npp8u *nConstant*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

Parameters:

pSrcI Source-Image Pointer.
nSrcIStep Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.13 NppStatus nppiMulCScale_8u_C3IR (const Npp8u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant and scale by max bit width value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.14 **NppStatus nppiMulCScale_8u_C3R** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.15 **NppStatus nppiMulCScale_8u_C4IR** (const Npp8u *aConstants*[4], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.9.2.16 **NppStatus nppiMulCScale_8u_C4R** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10 SubC

Subtracts a constant value from each pixel of an image.

Functions

- **NppStatus nppiSubC_8u_C1RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.
- **NppStatus nppiSubC_8u_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.
- **NppStatus nppiSubC_8u_C3RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.
- **NppStatus nppiSubC_8u_C3IRSfs** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel 8-bit unsigned char in place image subtract constant, scale, then clamp to saturated value.
- **NppStatus nppiSubC_8u_AC4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.
- **NppStatus nppiSubC_8u_AC4IRSfs** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.
- **NppStatus nppiSubC_8u_C4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.
- **NppStatus nppiSubC_8u_C4IRSfs** (const **Npp8u** aConstants[4], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.
- **NppStatus nppiSubC_16u_C1RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.
- **NppStatus nppiSubC_16u_C1IRSfs** (const **Npp16u** nConstant, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.
- **NppStatus nppiSubC_16u_C3RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16u_C3IRSfs` (const `Npp16u` aConstants[3], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16u_AC4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16u_AC4IRSfs` (const `Npp16u` aConstants[3], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16u_C4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16u_C4IRSfs` (const `Npp16u` aConstants[4], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16s_C1RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16s_C3RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16s_C3IRSfs` (const `Npp16s` aConstants[3], `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16s_AC4RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16s_AC4IRSfs` (const `Npp16s` aConstants[3], `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16s_C4RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` aConstants[4], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16s_C4IRSfs` (const `Npp16s` aConstants[4], `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16sc_C1RSfs` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16sc_C1IRSfs` (const `Npp16sc` nConstant, `Npp16sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16sc_C3RSfs` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16sc_C3IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16sc_AC4RSfs` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_16sc_AC4IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_32s_C1RSfs` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_32s_C3RSfs` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_32s_C3IRSfs` (const `Npp32s` aConstants[3], `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

- `NppStatus nppiSubC_32sc_C1RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` nConstant, `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.
- `NppStatus nppiSubC_32sc_C1IRSfs` (const `Npp32sc` nConstant, `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.
- `NppStatus nppiSubC_32sc_C3RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.
- `NppStatus nppiSubC_32sc_C3IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.
- `NppStatus nppiSubC_32sc_AC4RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.
- `NppStatus nppiSubC_32sc_AC4IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.
- `NppStatus nppiSubC_32f_C1R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` nConstant, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)
One 32-bit floating point channel image subtract constant.
- `NppStatus nppiSubC_32f_C1IR` (const `Npp32f` nConstant, `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
One 32-bit floating point channel in place image subtract constant.
- `NppStatus nppiSubC_32f_C3R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three 32-bit floating point channel image subtract constant.
- `NppStatus nppiSubC_32f_C3IR` (const `Npp32f` aConstants[3], `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Three 32-bit floating point channel in place image subtract constant.
- `NppStatus nppiSubC_32f_AC4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 32-bit floating point channel with unmodified alpha image subtract constant.
- `NppStatus nppiSubC_32f_AC4IR` (const `Npp32f` aConstants[3], `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Four 32-bit floating point channel with unmodified alpha in place image subtract constant.

- `NppStatus nppiSubC_32f_C4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f aConstants[4]`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

Four 32-bit floating point channel image subtract constant.

- `NppStatus nppiSubC_32f_C4IR` (const `Npp32f aConstants[4]`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

Four 32-bit floating point channel in place image subtract constant.

- `NppStatus nppiSubC_32fc_C1R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc nConstant`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

- `NppStatus nppiSubC_32fc_C1IR` (const `Npp32fc nConstant`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

- `NppStatus nppiSubC_32fc_C3R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc aConstants[3]`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

- `NppStatus nppiSubC_32fc_C3IR` (const `Npp32fc aConstants[3]`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

- `NppStatus nppiSubC_32fc_AC4R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc aConstants[3]`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image subtract constant.

- `NppStatus nppiSubC_32fc_AC4IR` (const `Npp32fc aConstants[3]`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image subtract constant.

- `NppStatus nppiSubC_32fc_C4R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc aConstants[4]`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

- `NppStatus nppiSubC_32fc_C4IR` (const `Npp32fc aConstants[4]`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

7.10.1 Detailed Description

Subtracts a constant value from each pixel of an image.

7.10.2 Function Documentation

7.10.2.1 **NppStatus nppiSubC_16s_AC4IRSfs** (const Npp16s *aConstants*[3], Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.2 **NppStatus nppiSubC_16s_AC4RSfs** (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[3], Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.3 **NppStatus nppiSubC_16s_C1IRSfs** (const Npp16s *nConstant*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.4 NppStatus nppiSubC_16s_C1RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s *nConstant*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.5 NppStatus nppiSubC_16s_C3IRSfs (const Npp16s *aConstants*[3], Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.6 `NppStatus nppiSubC_16s_C3RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s aConstants[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.7 `NppStatus nppiSubC_16s_C4IRSfs (const Npp16s aConstants[4], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.8 `NppStatus nppiSubC_16s_C4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s aConstants[4], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.9 NppStatus nppiSubC_16sc_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.10 NppStatus nppiSubC_16sc_AC4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.11 **NppStatus nppiSubC_16sc_C1IRSfs** (const Npp16sc *nConstant*, Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.12 **NppStatus nppiSubC_16sc_C1RSfs** (const Npp16sc * *pSrcI*, int *nSrcIStep*, const Npp16sc *nConstant*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrcI Source-Image Pointer.
nSrcIStep Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.13 **NppStatus nppiSubC_16sc_C3IRSfs** (const Npp16sc *aConstants*[3], Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.14 `NppStatus nppiSubC_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.15 `NppStatus nppiSubC_16u_AC4IRSfs (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.16 `NppStatus nppiSubC_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.17 `NppStatus nppiSubC_16u_C1IRSfs (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

nConstant [Constant](#).
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.18 `NppStatus nppiSubC_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nConstant [Constant](#).
pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.19 NppStatus nppiSubC_16u_C3IRSfs (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.20 NppStatus nppiSubC_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.21 `NppStatus nppiSubC_16u_C4IRSfs (const Npp16u aConstants[4], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.22 `NppStatus nppiSubC_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.23 `NppStatus nppiSubC_32f_AC4IR (const Npp32f aConstants[3], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image subtract constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.24 NppStatus nppiSubC_32f_AC4R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image subtract constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.25 NppStatus nppiSubC_32f_C1IR (const Npp32f *nConstant*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image subtract constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.26 NppStatus nppiSubC_32f_C1R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *nConstant*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image subtract constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.27 `NppStatus nppiSubC_32f_C3IR (const Npp32f aConstants[3], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel in place image subtract constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.28 `NppStatus nppiSubC_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f aConstants[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image subtract constant.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.29 `NppStatus nppiSubC_32f_C4IR (const Npp32f aConstants[4], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel in place image subtract constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.30 **NppStatus nppiSubC_32f_C4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[4], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image subtract constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.31 **NppStatus nppiSubC_32fc_AC4IR** (const Npp32fc *aConstants*[3], Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image subtract constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.32 **NppStatus nppiSubC_32fc_AC4R** (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image subtract constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.33 `NppStatus nppiSubC_32fc_C1IR` (const `Npp32fc nConstant`, `Npp32fc * pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.34 `NppStatus nppiSubC_32fc_C1R` (const `Npp32fc * pSrc1`, int `nSrc1Step`, const `Npp32fc nConstant`, `Npp32fc * pDst`, int `nDstStep`, `NppiSize oSizeROI`)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.35 `NppStatus nppiSubC_32fc_C3IR` (const `Npp32fc aConstants[3]`, `Npp32fc * pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.36 `NppStatus nppiSubC_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.37 `NppStatus nppiSubC_32fc_C4IR (const Npp32fc aConstants[4], Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.38 `NppStatus nppiSubC_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc aConstants[4], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.39 `NppStatus nppiSubC_32s_C1IRSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

nConstant [Constant](#).
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.40 `NppStatus nppiSubC_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nConstant [Constant](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.41 `NppStatus nppiSubC_32s_C3IRSfs (const Npp32s aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants [fixed size array of constant values, one per channel](#).

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.42 NppStatus nppiSubC_32s_C3RSfs (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.43 NppStatus nppiSubC_32sc_AC4IRSfs (const Npp32sc *aConstants*[3], Npp32sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.44 NppStatus nppiSubC_32sc_AC4RSfs (const Npp32sc * *pSrc1*, int *nSrc1Step*, const Npp32sc *aConstants*[3], Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.45 NppStatus nppiSubC_32sc_C1IRSfs (const Npp32sc *nConstant*, Npp32sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

nConstant [Constant](#).
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.46 NppStatus nppiSubC_32sc_C1RSfs (const Npp32sc * *pSrc1*, int *nSrc1Step*, const Npp32sc *nConstant*, Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.

nConstant Constant.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.47 `NppStatus nppiSubC_32sc_C3IRSfs (const Npp32sc aConstants[3], Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.48 `NppStatus nppiSubC_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.49 NppStatus nppiSubC_8u_AC4IRSfs (const Npp8u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.50 NppStatus nppiSubC_8u_AC4RSfs (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.51 NppStatus nppiSubC_8u_C1IRSfs (const Npp8u *nConstant*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.52 `NppStatus nppiSubC_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nConstant Constant.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.53 `NppStatus nppiSubC_8u_C3IRSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.10.2.54 `NppStatus nppiSubC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.55 NppStatus nppiSubC_8u_C4IRSfs (const Npp8u *aConstants*[4], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.10.2.56 NppStatus nppiSubC_8u_C4RSfs (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11 DivC

Divides each pixel of an image by a constant value.

Functions

- **NppStatus** **nppiDivC_8u_C1RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_8u_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_8u_C3RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_8u_C3IRSfs** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel 8-bit unsigned char in place image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_8u_AC4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_8u_AC4IRSfs** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_8u_C4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_8u_C4IRSfs** (const **Npp8u** aConstants[4], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_16u_C1RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_16u_C1IRSfs** (const **Npp16u** nConstant, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_16u_C3RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_16u_C3IRSfs` (const `Npp16u` aConstants[3], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_16u_AC4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_16u_AC4IRSfs` (const `Npp16u` aConstants[3], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_16u_C4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_16u_C4IRSfs` (const `Npp16u` aConstants[4], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_16s_C1RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_16s_C3RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_16s_C3IRSfs` (const `Npp16s` aConstants[3], `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_16s_AC4RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_16s_AC4IRSfs` (const `Npp16s` aConstants[3], `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

- **NppStatus** **nppiDivC_16s_C4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** aConstants[4], **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_16s_C4IRSfs** (const **Npp16s** aConstants[4], **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_16sc_C1RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** nConstant, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_16sc_C1IRSfs** (const **Npp16sc** nConstant, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_16sc_C3RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** aConstants[3], **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_16sc_C3IRSfs** (const **Npp16sc** aConstants[3], **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_16sc_AC4RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** aConstants[3], **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_16sc_AC4IRSfs** (const **Npp16sc** aConstants[3], **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_32s_C1RSfs** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** nConstant, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_32s_C1IRSfs** (const **Npp32s** nConstant, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_32s_C3RSfs** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** aConstants[3], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.
- **NppStatus** **nppiDivC_32s_C3IRSfs** (const **Npp32s** aConstants[3], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Three 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_32sc_C1RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` nConstant, `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_32sc_C1IRSfs` (const `Npp32sc` nConstant, `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_32sc_C3RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_32sc_C3IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_32sc_AC4RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_32sc_AC4IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_32f_C1R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` nConstant, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

One 32-bit floating point channel image divided by constant.

- `NppStatus nppiDivC_32f_C1IR` (const `Npp32f` nConstant, `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

One 32-bit floating point channel in place image divided by constant.

- `NppStatus nppiDivC_32f_C3R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three 32-bit floating point channel image divided by constant.

- `NppStatus nppiDivC_32f_C3IR` (const `Npp32f` aConstants[3], `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Three 32-bit floating point channel in place image divided by constant.

- `NppStatus nppiDivC_32f_AC4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four 32-bit floating point channel with unmodified alpha image divided by constant.

- **NppStatus nppiDivC_32f_AC4IR** (const **Npp32f** aConstants[3], **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel with unmodified alpha in place image divided by constant.
- **NppStatus nppiDivC_32f_C4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** aConstants[4], **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel image divided by constant.
- **NppStatus nppiDivC_32f_C4IR** (const **Npp32f** aConstants[4], **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel in place image divided by constant.
- **NppStatus nppiDivC_32fc_C1R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** nConstant, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.
- **NppStatus nppiDivC_32fc_C1IR** (const **Npp32fc** nConstant, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.
- **NppStatus nppiDivC_32fc_C3R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.
- **NppStatus nppiDivC_32fc_C3IR** (const **Npp32fc** aConstants[3], **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.
- **NppStatus nppiDivC_32fc_AC4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image divided by constant.
- **NppStatus nppiDivC_32fc_AC4IR** (const **Npp32fc** aConstants[3], **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image divided by constant.
- **NppStatus nppiDivC_32fc_C4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** aConstants[4], **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.
- **NppStatus nppiDivC_32fc_C4IR** (const **Npp32fc** aConstants[4], **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

7.11.1 Detailed Description

Divides each pixel of an image by a constant value.

7.11.2 Function Documentation

7.11.2.1 **NppStatus nppiDivC_16s_AC4IRSfs** (const Npp16s *aConstants*[3], Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.2 **NppStatus nppiDivC_16s_AC4RSfs** (const Npp16s * *pSrcI*, int *nSrcIStep*, const Npp16s *aConstants*[3], Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrcI [Source-Image Pointer](#).

nSrcIStep [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.3 NppStatus nppiDivC_16s_C1RSfs (const Npp16s *nConstant*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.4 NppStatus nppiDivC_16s_C1RSfs (const Npp16s * *pSrcI*, int *nSrcIStep*, const Npp16s *nConstant*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrcI Source-Image Pointer.
nSrcIStep Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.5 NppStatus nppiDivC_16s_C3IRSfs (const Npp16s *aConstants*[3], Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.6 **NppStatus nppiDivC_16s_C3RSfs** (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[3], Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.7 **NppStatus nppiDivC_16s_C4IRSfs** (const Npp16s *aConstants*[4], Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.8 **NppStatus nppiDivC_16s_C4RSfs** (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[4], Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.9 NppStatus nppiDivC_16sc_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.10 NppStatus nppiDivC_16sc_AC4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.11 `NppStatus nppiDivC_16sc_C1IRSfs (const Npp16sc nConstant, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.12 `NppStatus nppiDivC_16sc_C1RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc nConstant, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.13 `NppStatus nppiDivC_16sc_C3IRSfs (const Npp16sc aConstants[3], Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.14 `NppStatus nppiDivC_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.15 `NppStatus nppiDivC_16u_AC4IRSfs (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.16 `NppStatus nppiDivC_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.17 `NppStatus nppiDivC_16u_C1IRSfs (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

nConstant [Constant](#).
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.18 `NppStatus nppiDivC_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nConstant [Constant](#).
pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.19 NppStatus nppiDivC_16u_C3IRSfs (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.20 NppStatus nppiDivC_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.21 `NppStatus nppiDivC_16u_C4IRSfs (const Npp16u aConstants[4], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.22 `NppStatus nppiDivC_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.23 `NppStatus nppiDivC_32f_AC4IR (const Npp32f aConstants[3], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image divided by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.24 **NppStatus nppiDivC_32f_AC4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image divided by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.25 **NppStatus nppiDivC_32f_C1IR** (const Npp32f *nConstant*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image divided by constant.

Parameters:

nConstant Constant.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.26 **NppStatus nppiDivC_32f_C1R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *nConstant*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image divided by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nConstant Constant.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.27 `NppStatus nppiDivC_32f_C3IR (const Npp32f aConstants[3], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel in place image divided by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.28 `NppStatus nppiDivC_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f aConstants[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image divided by constant.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.29 `NppStatus nppiDivC_32f_C4IR (const Npp32f aConstants[4], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel in place image divided by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.30 **NppStatus nppiDivC_32f_C4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[4], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image divided by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.31 **NppStatus nppiDivC_32fc_AC4IR** (const Npp32fc *aConstants*[3], Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image divided by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.32 **NppStatus nppiDivC_32fc_AC4R** (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image divided by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.33 `NppStatus nppiDivC_32fc_C1IR` (const `Npp32fc nConstant`, `Npp32fc * pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.34 `NppStatus nppiDivC_32fc_C1R` (const `Npp32fc * pSrcI`, int `nSrcIStep`, const `Npp32fc nConstant`, `Npp32fc * pDst`, int `nDstStep`, `NppiSize oSizeROI`)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

Parameters:

pSrcI Source-Image Pointer.
nSrcIStep Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.35 `NppStatus nppiDivC_32fc_C3IR` (const `Npp32fc aConstants[3]`, `Npp32fc * pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.36 `NppStatus nppiDivC_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.37 `NppStatus nppiDivC_32fc_C4IR (const Npp32fc aConstants[4], Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.38 `NppStatus nppiDivC_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc aConstants[4], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.39 `NppStatus nppiDivC_32s_C1IRSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

nConstant [Constant](#).

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.40 `NppStatus nppiDivC_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

nConstant [Constant](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.41 `NppStatus nppiDivC_32s_C3IRSfs (const Npp32s aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.42 NppStatus nppiDivC_32s_C3RSfs (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.43 NppStatus nppiDivC_32sc_AC4IRSfs (const Npp32sc *aConstants*[3], Npp32sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.44 `NppStatus nppiDivC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.45 `NppStatus nppiDivC_32sc_C1IRSfs (const Npp32sc nConstant, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.46 `NppStatus nppiDivC_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc nConstant, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nConstant Constant.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.47 NppStatus nppiDivC_32sc_C3IRSfs (const Npp32sc aConstants[3], Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.48 NppStatus nppiDivC_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.49 **NppStatus nppiDivC_8u_AC4IRSfs** (const Npp8u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.50 **NppStatus nppiDivC_8u_AC4RSfs** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.11.2.51 **NppStatus nppiDivC_8u_C1IRSfs** (const Npp8u *nConstant*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

nConstant Constant.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.52 `NppStatus nppiDivC_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nConstant Constant.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.53 `NppStatus nppiDivC_8u_C3IRSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.54 `NppStatus nppiDivC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.55 NppStatus nppiDivC_8u_C4IRSfs (const Npp8u *aConstants*[4], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.11.2.56 NppStatus nppiDivC_8u_C4RSfs (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.12 AbsDiffC

Determines absolute difference between each pixel of an image and a constant value.

Functions

- **NppStatus nppiAbsDiffC_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nConstant)
One 8-bit unsigned char channel image absolute difference with constant.
- **NppStatus nppiAbsDiffC_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp16u** nConstant)
One 16-bit unsigned short channel image absolute difference with constant.
- **NppStatus nppiAbsDiffC_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nConstant)
One 32-bit floating point channel image absolute difference with constant.

7.12.1 Detailed Description

Determines absolute difference between each pixel of an image and a constant value.

7.12.2 Function Documentation

7.12.2.1 **NppStatus nppiAbsDiffC_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp16u** nConstant)

One 16-bit unsigned short channel image absolute difference with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.12.2.2 **NppStatus nppiAbsDiffC_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nConstant)

One 32-bit floating point channel image absolute difference with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.12.2.3 NppStatus nppiAbsDiffC_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp8u nConstant)

One 8-bit unsigned char channel image absolute difference with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13 Add

Pixel by pixel addition of two images.

Functions

- **NppStatus** **nppiAdd_8u_C1RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiAdd_8u_C1IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiAdd_8u_C3RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiAdd_8u_C3IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiAdd_8u_AC4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiAdd_8u_AC4IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiAdd_8u_C4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiAdd_8u_C4IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiAdd_16u_C1RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiAdd_16u_C1IRSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiAdd_16u_C3RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Three 16-bit unsigned short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiAdd_16u_C3IRSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiAdd_16u_AC4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiAdd_16u_AC4IRSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiAdd_16u_C4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiAdd_16u_C4IRSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiAdd_16s_C1RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 16-bit signed short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiAdd_16s_C1IRSfs` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 16-bit signed short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiAdd_16s_C3RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Three 16-bit signed short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiAdd_16s_C3IRSfs` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus** **nppiAdd_16s_AC4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus** **nppiAdd_16s_AC4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus** **nppiAdd_16s_C4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus** **nppiAdd_16s_C4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus** **nppiAdd_16sc_C1RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus** **nppiAdd_16sc_C1IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus** **nppiAdd_16sc_C3RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus** **nppiAdd_16sc_C3IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus** **nppiAdd_16sc_AC4RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus** **nppiAdd_16sc_AC4IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiAdd_32s_C1RSfs** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiAdd_32s_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.
- **NppStatus nppiAdd_32s_C1IRSfs** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiAdd_32s_C3RSfs** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 32-bit signed integer channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiAdd_32s_C3IRSfs** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiAdd_32sc_C1RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiAdd_32sc_C1IRSfs** (const **Npp32sc** *pSrc, int nSrcStep, **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiAdd_32sc_C3RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiAdd_32sc_C3IRSfs** (const **Npp32sc** *pSrc, int nSrcStep, **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiAdd_32sc_AC4RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiAdd_32sc_AC4IRSfs** (const **Npp32sc** *pSrc, int nSrcStep, **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiAdd_32f_C1R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

One 32-bit floating point channel image addition.

- `NppStatus nppiAdd_32f_C1IR` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

One 32-bit floating point channel in place image addition.

- `NppStatus nppiAdd_32f_C3R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three 32-bit floating point channel image addition.

- `NppStatus nppiAdd_32f_C3IR` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

One 32-bit floating point channel in place image addition.

- `NppStatus nppiAdd_32f_AC4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four 32-bit floating point channel with unmodified alpha image addition.

- `NppStatus nppiAdd_32f_AC4IR` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Four 32-bit floating point channel with unmodified alpha in place image addition.

- `NppStatus nppiAdd_32f_C4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four 32-bit floating point channel image addition.

- `NppStatus nppiAdd_32f_C4IR` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Four 32-bit floating point channel in place image addition.

- `NppStatus nppiAdd_32fc_C1R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `Npp32fc` *pDst, int nDstStep, `NppiSize` oSizeROI)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

- `NppStatus nppiAdd_32fc_C1IR` (const `Npp32fc` *pSrc, int nSrcStep, `Npp32fc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

- `NppStatus nppiAdd_32fc_C3R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `Npp32fc` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

- `NppStatus nppiAdd_32fc_C3IR` (const `Npp32fc` *pSrc, int nSrcStep, `Npp32fc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

- **NppStatus nppiAdd_32fc_AC4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition.
- **NppStatus nppiAdd_32fc_AC4IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition.
- **NppStatus nppiAdd_32fc_C4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.
- **NppStatus nppiAdd_32fc_C4IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

7.13.1 Detailed Description

Pixel by pixel addition of two images.

7.13.2 Function Documentation

7.13.2.1 **NppStatus nppiAdd_16s_AC4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.13.2.2 **NppStatus nppiAdd_16s_AC4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.3 NppStatus nppiAdd_16s_C1IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.4 NppStatus nppiAdd_16s_C1RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.5 NppStatus npAdd_16s_C3IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.6 NppStatus npAdd_16s_C3RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit signed short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.7 **NppStatus nppiAdd_16s_C4IRSfs** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.8 **NppStatus nppiAdd_16s_C4RSfs** (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.9 **NppStatus nppiAdd_16sc_AC4IRSfs** (const Npp16sc * *pSrc*, int *nSrcStep*, Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.10 `NppStatus nppiAdd_16sc_AC4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.11 `NppStatus nppiAdd_16sc_C1IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.12 NppStatus npAdd_16sc_C1RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.13 NppStatus npAdd_16sc_C3RSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.14 NppStatus npAdd_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.15 **NppStatus nppiAdd_16u_AC4IRSfs** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.16 **NppStatus nppiAdd_16u_AC4RSfs** (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.17 NppStatus npAdd_16u_C1RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.18 NppStatus npAdd_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.19 `NppStatus nppiAdd_16u_C3IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.13.2.20 `NppStatus nppiAdd_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.13.2.21 `NppStatus nppiAdd_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.22 `NppStatus nppiAdd_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.23 `NppStatus nppiAdd_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image addition.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.24 **NppStatus nppiAdd_32f_AC4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image addition.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.13.2.25 **NppStatus nppiAdd_32f_C1IR** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image addition.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.13.2.26 **NppStatus nppiAdd_32f_C1R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image addition.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.27 NppStatus npAdd_32f_C3IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One 32-bit floating point channel in place image addition.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.28 NppStatus npAdd_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)

Three 32-bit floating point channel image addition.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.29 **NppStatus nppiAdd_32f_C4IR** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image addition.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.30 **NppStatus nppiAdd_32f_C4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image addition.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.31 **NppStatus nppiAdd_32fc_AC4IR** (const Npp32fc * *pSrc*, int *nSrcStep*, Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.32 `NppStatus nppiAdd_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.33 `NppStatus nppiAdd_32fc_C1IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.34 `NppStatus nppiAdd_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.35 NppStatus npAdd_32fc_C3IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.36 NppStatus npAdd_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.37 NppStatus nppiAdd_32fc_C4IR (const Npp32fc * *pSrc*, int *nSrcStep*, Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.38 NppStatus nppiAdd_32fc_C4R (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc * *pSrc2*, int *nSrc2Step*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.39 NppStatus nppiAdd_32s_C1IRSfs (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.40 NppStatus npAdd_32s_C1R (const Npp32s *pSrc1, int nSrc1Step, const Npp32s *pSrc2, int nSrc2Step, Npp32s *pDst, int nDstStep, NppiSize oSizeROI)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image add. Add the pixel values of corresponding pixels in the ROI and write them to the output image.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.41 NppStatus npAdd_32s_C1RSfs (const Npp32s *pSrc1, int nSrc1Step, const Npp32s *pSrc2, int nSrc2Step, Npp32s *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 32-bit signed integer channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.42 `NppStatus nppiAdd_32s_C3IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.43 `NppStatus nppiAdd_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.44 `NppStatus nppiAdd_32sc_AC4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.45 `NppStatus npplAdd_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.46 `NppStatus npplAdd_32sc_C1IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.47 NppStatus nppiAdd_32sc_C1RSfs (const Npp32sc * *pSrc1*, int *nSrc1Step*, const Npp32sc * *pSrc2*, int *nSrc2Step*, Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.48 NppStatus nppiAdd_32sc_C3RSfs (const Npp32sc * *pSrc*, int *nSrcStep*, Npp32sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.49 NppStatus nppiAdd_32sc_C3RSfs (const Npp32sc * *pSrc1*, int *nSrc1Step*, const Npp32sc * *pSrc2*, int *nSrc2Step*, Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.50 NppStatus nppiAdd_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 8-bit unsigned char channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.51 NppStatus nppiAdd_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 8-bit unsigned char channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.52 NppStatus nppiAdd_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 8-bit unsigned char channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.53 NppStatus nppiAdd_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 8-bit unsigned char channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.54 `NppStatus nppiAdd_8u_C3IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.55 `NppStatus nppiAdd_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.56 `NppStatus nppiAdd_8u_C4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.13.2.57 `NppStatus nppiAdd_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.14 AddSquare

Pixel by pixel addition of squared pixels from source image to floating point pixel values of destination image.

Functions

- **NppStatus nppiAddSquare_8u32f_C1IMR** (const **Npp8u** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).
- **NppStatus nppiAddSquare_8u32f_C1IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image squared then added to in place floating point destination image.
- **NppStatus nppiAddSquare_16u32f_C1IMR** (const **Npp16u** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).
- **NppStatus nppiAddSquare_16u32f_C1IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image squared then added to in place floating point destination image.
- **NppStatus nppiAddSquare_32f_C1IMR** (const **Npp32f** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).
- **NppStatus nppiAddSquare_32f_C1IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image squared then added to in place floating point destination image.

7.14.1 Detailed Description

Pixel by pixel addition of squared pixels from source image to floating point pixel values of destination image.

7.14.2 Function Documentation

7.14.2.1 **NppStatus nppiAddSquare_16u32f_C1IMR** (const **Npp16u** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

One 16-bit unsigned short channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.14.2.2 NppStatus nppiAddSquare_16u32f_C1IR (const Npp16u * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image squared then added to in place floating point destination image.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.14.2.3 NppStatus nppiAddSquare_32f_C1IMR (const Npp32f * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.14.2.4 **NppStatus nppiAddSquare_32f_C1IR** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image squared then added to in place floating point destination image.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.14.2.5 **NppStatus nppiAddSquare_8u32f_C1IMR** (const Npp8u * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.14.2.6 **NppStatus nppiAddSquare_8u32f_C1IR** (const Npp8u * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image squared then added to in place floating point destination image.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.15 AddProduct

Pixel by pixel addition of product of pixels from two source images to floating point pixel values of destination image.

Functions

- **NppStatus nppiAddProduct_8u32f_C1IMR** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).
- **NppStatus nppiAddProduct_8u32f_C1IR** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image product added to in place floating point destination image.
- **NppStatus nppiAddProduct_16u32f_C1IMR** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).
- **NppStatus nppiAddProduct_16u32f_C1IR** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image product added to in place floating point destination image.
- **NppStatus nppiAddProduct_32f_C1IMR** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).
- **NppStatus nppiAddProduct_32f_C1IR** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image product added to in place floating point destination image.

7.15.1 Detailed Description

Pixel by pixel addition of product of pixels from two source images to floating point pixel values of destination image.

7.15.2 Function Documentation

7.15.2.1 NppStatus nppiAddProduct_16u32f_C1IMR (const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One 16-bit unsigned short channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.15.2.2 **NppStatus nppiAddProduct_16u32f_C11R** (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image product added to in place floating point destination image.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.15.2.3 **NppStatus nppiAddProduct_32f_C11MR** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, const Npp8u * *pMask*, int *nMaskStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.15.2.4 NppStatus npAddProduct_32f_C1IR (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image product added to in place floating point destination image.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.15.2.5 NppStatus npAddProduct_8u32f_C1IMR (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, const Npp8u * *pMask*, int *nMaskStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.15.2.6 NppStatus nppiAddProduct_8u32f_C1IR (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image product added to in place floating point destination image.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.16 AddWeighted

Pixel by pixel addition of alpha weighted pixel values from a source image to floating point pixel values of destination image.

Functions

- **NppStatus** **nppiAddWeighted_8u32f_C1IMR** (const **Npp8u** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)
One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).
- **NppStatus** **nppiAddWeighted_8u32f_C1IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)
One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image.
- **NppStatus** **nppiAddWeighted_16u32f_C1IMR** (const **Npp16u** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)
One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).
- **NppStatus** **nppiAddWeighted_16u32f_C1IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)
One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image.
- **NppStatus** **nppiAddWeighted_32f_C1IMR** (const **Npp32f** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)
One 32-bit floating point channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).
- **NppStatus** **nppiAddWeighted_32f_C1IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)
One 32-bit floating point channel alpha weighted image added to in place floating point destination image.

7.16.1 Detailed Description

Pixel by pixel addition of alpha weighted pixel values from a source image to floating point pixel values of destination image.

7.16.2 Function Documentation

7.16.2.1 **NppStatus** **nppiAddWeighted_16u32f_C1IMR** (const **Npp16u** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)

One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nAlpha Alpha weight to be applied to source image pixels (0.0F to 1.0F)

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.16.2.2 NppStatus nppiAddWeighted_16u32f_C11R (const Npp16u * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)

One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nAlpha Alpha weight to be applied to source image pixels (0.0F to 1.0F)

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.16.2.3 NppStatus nppiAddWeighted_32f_C11MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)

One 32-bit floating point channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

nAlpha Alpha weight to be applied to source image pixels (0.0F to 1.0F)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.16.2.4 NppStatus npAddWeighted_32f_C1IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)

One 32-bit floating point channel alpha weighted image added to in place floating point destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nAlpha Alpha weight to be applied to source image pixels (0.0F to 1.0F)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.16.2.5 NppStatus npAddWeighted_8u32f_C1IMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)

One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nAlpha Alpha weight to be applied to source image pixels (0.0F to 1.0F)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.16.2.6 NppStatus nppiAddWeighted_8u32f_C1IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, Npp32f *nAlpha*)

One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nAlpha Alpha weight to be applied to source image pixels (0.0F to 1.0F)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.17 Mul

Pixel by pixel multiply of two images.

Functions

- **NppStatus nppiMul_8u_C1RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_8u_C1RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_8u_C3RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_8u_C3RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_8u_AC4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_8u_AC4RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_8u_C4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_8u_C4RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_16u_C1RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiMul_16u_C1IRSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiMul_16u_C3RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Three 16-bit unsigned short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiMul_16u_C3IRSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiMul_16u_AC4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel with unmodified alpha image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiMul_16u_AC4IRSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel with unmodified alpha in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiMul_16u_C4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiMul_16u_C4IRSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiMul_16s_C1RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 16-bit signed short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiMul_16s_C1IRSfs` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
One 16-bit signed short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiMul_16s_C3RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
Three 16-bit signed short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- `NppStatus nppiMul_16s_C3IRSfs` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_16s_AC4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_16s_AC4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_16s_C4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_16s_C4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_16sc_C1RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_16sc_C1IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_16sc_C3RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_16sc_C3IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_16sc_AC4RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_16sc_AC4IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_32s_C1RSfs** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_32s_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.
- **NppStatus nppiMul_32s_C1IRSfs** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_32s_C3RSfs** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 32-bit signed integer channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_32s_C3IRSfs** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_32sc_C1RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_32sc_C1IRSfs** (const **Npp32sc** *pSrc, int nSrcStep, **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_32sc_C3RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_32sc_C3IRSfs** (const **Npp32sc** *pSrc, int nSrcStep, **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_32sc_AC4RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiMul_32sc_AC4IRSfs** (const **Npp32sc** *pSrc, int nSrcStep, **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiMul_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image multiplication.
- **NppStatus nppiMul_32f_C1IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image multiplication.
- **NppStatus nppiMul_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel image multiplication.
- **NppStatus nppiMul_32f_C3IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image multiplication.
- **NppStatus nppiMul_32f_AC4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel with unmodified alpha image multiplication.
- **NppStatus nppiMul_32f_AC4IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel with unmodified alpha in place image multiplication.
- **NppStatus nppiMul_32f_C4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel image multiplication.
- **NppStatus nppiMul_32f_C4IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel in place image multiplication.
- **NppStatus nppiMul_32fc_C1R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.
- **NppStatus nppiMul_32fc_C1IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.
- **NppStatus nppiMul_32fc_C3R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

- **NppStatus nppiMul_32fc_C3IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.
- **NppStatus nppiMul_32fc_AC4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication.
- **NppStatus nppiMul_32fc_AC4IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication.
- **NppStatus nppiMul_32fc_C4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.
- **NppStatus nppiMul_32fc_C4IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

7.17.1 Detailed Description

Pixel by pixel multiply of two images.

7.17.2 Function Documentation

7.17.2.1 **NppStatus nppiMul_16s_AC4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

- pSrc** Source-Image Pointer.
- nSrcStep** Source-Image Line Step.
- pSrcDst** In-Place Image Pointer.
- nSrcDstStep** In-Place-Image Line Step.
- oSizeROI** Region-of-Interest (ROI).
- nScaleFactor** Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.2 NppStatus nppiMul_16s_AC4RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.3 NppStatus nppiMul_16s_C1IRSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.4 NppStatus nppiMul_16s_C1RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.5 NppStatus nppiMul_16s_C3IRSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.6 NppStatus nppiMul_16s_C3RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.7 NppStatus nppiMul_16s_C4IRSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.8 NppStatus nppiMul_16s_C4RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.9 NppStatus nppiMul_16sc_AC4IRSfs (const Npp16sc * *pSrc*, int *nSrcStep*, Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.10 `NppStatus nppiMul_16sc_AC4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.11 `NppStatus nppiMul_16sc_C1IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.12 NppStatus nppiMul_16sc_C1RSfs (const Npp16sc * *pSrc1*, int *nSrc1Step*, const Npp16sc * *pSrc2*, int *nSrc2Step*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.13 NppStatus nppiMul_16sc_C3IRSfs (const Npp16sc * *pSrc*, int *nSrcStep*, Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.14 NppStatus nppiMul_16sc_C3RSfs (const Npp16sc * *pSrc1*, int *nSrc1Step*, const Npp16sc * *pSrc2*, int *nSrc2Step*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.15 **NppStatus nppiMul_16u_AC4IRSfs** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.16 **NppStatus nppiMul_16u_AC4RSfs** (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.17 NppStatus nppiMul_16u_C1IRSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.18 NppStatus nppiMul_16u_C1IRSfs (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.19 NppStatus nppiMul_16u_C3IRSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.20 NppStatus nppiMul_16u_C3RSfs (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.21 NppStatus nppiMul_16u_C4IRSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.22 `NppStatus nppiMul_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.23 `NppStatus nppiMul_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image multiplication.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.24 **NppStatus nppiMul_32f_AC4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image multiplication.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.17.2.25 **NppStatus nppiMul_32f_C1IR** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image multiplication.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.17.2.26 **NppStatus nppiMul_32f_C1R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image multiplication.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.27 NppStatus nppiMul_32f_C3IR (const Npp32f *pSrc, int nSrcStep, Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One 32-bit floating point channel in place image multiplication.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.28 NppStatus nppiMul_32f_C3R (const Npp32f *pSrc1, int nSrc1Step, const Npp32f *pSrc2, int nSrc2Step, Npp32f *pDst, int nDstStep, NppiSize oSizeROI)

Three 32-bit floating point channel image multiplication.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.29 **NppStatus nppiMul_32f_C4IR** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image multiplication.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.17.2.30 **NppStatus nppiMul_32f_C4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image multiplication.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.17.2.31 **NppStatus nppiMul_32fc_AC4IR** (const Npp32fc * *pSrc*, int *nSrcStep*, Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.17.2.32 **NppStatus nppiMul_32fc_AC4R** (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc * *pSrc2*, int *nSrc2Step*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.33 **NppStatus nppiMul_32fc_C1IR** (const Npp32fc * *pSrc*, int *nSrcStep*, Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.34 **NppStatus nppiMul_32fc_C1R** (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc * *pSrc2*, int *nSrc2Step*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.35 NppStatus nppiMul_32fc_C3IR (const Npp32fc * *pSrc*, int *nSrcStep*, Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.36 NppStatus nppiMul_32fc_C3R (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc * *pSrc2*, int *nSrc2Step*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.37 `NppStatus nppiMul_32fc_C4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.38 `NppStatus nppiMul_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.39 `NppStatus nppiMul_32s_C1IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.17.2.40 **NppStatus nppiMul_32s_C1R** (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

1 channel 32-bit image multiplication. Multiply corresponding pixels in ROI.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.17.2.41 **NppStatus nppiMul_32s_C1RSfs** (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.17.2.42 `NppStatus nppiMul_32s_C3IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.43 `NppStatus nppiMul_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.44 `NppStatus nppiMul_32sc_AC4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.45 `NppStatus nppiMul_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.46 `NppStatus nppiMul_32sc_C1IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.47 NppStatus nppiMul_32sc_C1RSfs (const Npp32sc * *pSrc1*, int *nSrc1Step*, const Npp32sc * *pSrc2*, int *nSrc2Step*, Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.48 NppStatus nppiMul_32sc_C3IRSfs (const Npp32sc * *pSrc*, int *nSrcStep*, Npp32sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.49 NppStatus nppiMul_32sc_C3RSfs (const Npp32sc * *pSrc1*, int *nSrc1Step*, const Npp32sc * *pSrc2*, int *nSrc2Step*, Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.50 NppStatus nppiMul_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 8-bit unsigned char channel with unmodified alpha in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.51 NppStatus nppiMul_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 8-bit unsigned char channel with unmodified alpha image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.52 `NppStatus nppiMul_8u_C1RSfs (const Npp8u *pSrc, int nSrcStep, Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.53 `NppStatus nppiMul_8u_C1RSfs (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.54 **NppStatus nppiMul_8u_C3IRSfs** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.55 **NppStatus nppiMul_8u_C3RSfs** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.56 **NppStatus nppiMul_8u_C4IRSfs** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.17.2.57 `NppStatus nppiMul_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.18 MulScale

Pixel by pixel multiplies each pixel of two images then scales the result by the maximum value for the data bit width.

Functions

- **NppStatus nppiMulScale_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.
- **NppStatus nppiMulScale_8u_C1IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.
- **NppStatus nppiMulScale_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.
- **NppStatus nppiMulScale_8u_C3IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.
- **NppStatus nppiMulScale_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.
- **NppStatus nppiMulScale_8u_AC4IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.
- **NppStatus nppiMulScale_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.
- **NppStatus nppiMulScale_8u_C4IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.
- **NppStatus nppiMulScale_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.
- **NppStatus nppiMulScale_16u_C1IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

- **NppStatus nppiMulScale_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Three 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

- **NppStatus nppiMulScale_16u_C3IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

- **NppStatus nppiMulScale_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.

- **NppStatus nppiMulScale_16u_AC4IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.

- **NppStatus nppiMulScale_16u_C4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

- **NppStatus nppiMulScale_16u_C4IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

7.18.1 Detailed Description

Pixel by pixel multiplies each pixel of two images then scales the result by the maximum value for the data bit width.

7.18.2 Function Documentation

7.18.2.1 **NppStatus nppiMulScale_16u_AC4IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.18.2.2 NppStatus nppiMulScale_16u_AC4R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.18.2.3 NppStatus nppiMulScale_16u_C11R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.18.2.4 NppStatus nppiMulScale_16u_C1R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.18.2.5 NppStatus nppiMulScale_16u_C3IR (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.18.2.6 NppStatus nppiMulScale_16u_C3R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.18.2.7 NppStatus nppiMulScale_16u_C4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.18.2.8 NppStatus nppiMulScale_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.18.2.9 NppStatus nppiMulScale_8u_AC4IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.18.2.10 NppStatus nppiMulScale_8u_AC4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.18.2.11 NppStatus nppiMulScale_8u_C1IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.18.2.12 NppStatus nppiMulScale_8u_C1R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.18.2.13 NppStatus nppiMulScale_8u_C3IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.18.2.14 **NppStatus nppiMulScale_8u_C3R** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.18.2.15 **NppStatus nppiMulScale_8u_C4IR** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.18.2.16 **NppStatus nppiMulScale_8u_C4R** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19 Sub

Pixel by pixel subtraction of two images.

Functions

- **NppStatus nppiSub_8u_C1RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_8u_C1IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_8u_C3RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_8u_C3IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_8u_AC4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_8u_AC4IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_8u_C4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_8u_C4IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_16u_C1RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_16u_C1RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_16u_C3RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit unsigned short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_16u_C3IRSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_16u_AC4RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel with unmodified alpha image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_16u_AC4IRSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel with unmodified alpha in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_16u_C4RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_16u_C4IRSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_16s_C1RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_16s_C1IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_16s_C3RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit signed short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_16s_C3IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_16s_AC4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_16s_AC4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_16s_C4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_16s_C4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_16sc_C1RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_16sc_C1IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_16sc_C3RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_16sc_C3IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_16sc_AC4RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_16sc_AC4IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_32s_C1RSfs** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_32s_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.
- **NppStatus nppiSub_32s_C1IRSfs** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_32s_C3RSfs** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 32-bit signed integer channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_32s_C3IRSfs** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_32s_C4RSfs** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 32-bit signed integer channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_32s_C4IRSfs** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 32-bit signed integer channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_32sc_C1RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_32sc_C1IRSfs** (const **Npp32sc** *pSrc, int nSrcStep, **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_32sc_C3RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSub_32sc_C3IRSfs** (const **Npp32sc** *pSrc, int nSrcStep, **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_32sc_AC4RSfs** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_32sc_AC4IRSfs** (const **Npp32sc** *pSrc, int nSrcStep, **Npp32sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSub_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image subtraction.
- **NppStatus nppiSub_32f_C1IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image subtraction.
- **NppStatus nppiSub_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel image subtraction.
- **NppStatus nppiSub_32f_C3IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image subtraction.
- **NppStatus nppiSub_32f_AC4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel with unmodified alpha image subtraction.
- **NppStatus nppiSub_32f_AC4IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel with unmodified alpha in place image subtraction.
- **NppStatus nppiSub_32f_C4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel image subtraction.
- **NppStatus nppiSub_32f_C4IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel in place image subtraction.
- **NppStatus nppiSub_32fc_C1R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

- **NppStatus nppiSub_32fc_C1IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.
- **NppStatus nppiSub_32fc_C3R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.
- **NppStatus nppiSub_32fc_C3IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.
- **NppStatus nppiSub_32fc_AC4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction.
- **NppStatus nppiSub_32fc_AC4IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction.
- **NppStatus nppiSub_32fc_C4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.
- **NppStatus nppiSub_32fc_C4IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

7.19.1 Detailed Description

Pixel by pixel subtraction of two images.

7.19.2 Function Documentation

7.19.2.1 **NppStatus nppiSub_16s_AC4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

- pSrc** Source-Image Pointer.
- nSrcStep** Source-Image Line Step.
- pSrcDst** In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.2 `NppStatus nppiSub_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.3 `NppStatus nppiSub_16s_C1IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.4 NppStatus nppiSub_16s_C1RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.5 NppStatus nppiSub_16s_C3IRSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.6 NppStatus nppiSub_16s_C3RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.7 NppStatus nppiSub_16s_C4IRSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.8 NppStatus nppiSub_16s_C4RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.9 NppStatus nppiSub_16sc_AC4IRSfs (const Npp16sc * *pSrc*, int *nSrcStep*, Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.10 NppStatus nppiSub_16sc_AC4RSfs (const Npp16sc * *pSrc1*, int *nSrc1Step*, const Npp16sc * *pSrc2*, int *nSrc2Step*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.11 NppStatus nppiSub_16sc_C1IRSfs (const Npp16sc * *pSrc*, int *nSrcStep*, Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.12 NppStatus npipiSub_16sc_C1RSfs (const Npp16sc * *pSrc1*, int *nSrc1Step*, const Npp16sc * *pSrc2*, int *nSrc2Step*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.13 NppStatus npipiSub_16sc_C3IRSfs (const Npp16sc * *pSrc*, int *nSrcStep*, Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.14 **NppStatus nppiSub_16sc_C3RSfs** (const Npp16sc * *pSrc1*, int *nSrc1Step*, const Npp16sc * *pSrc2*, int *nSrc2Step*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.15 **NppStatus nppiSub_16u_AC4IRSfs** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.16 **NppStatus nppiSub_16u_AC4RSfs** (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.17 NppStatus nppiSub_16u_C1RSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.18 NppStatus nppiSub_16u_C1RSfs (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.19 NppStatus nppiSub_16u_C3IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.20 NppStatus nppiSub_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.21 `NppStatus nppiSub_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.22 `NppStatus nppiSub_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.23 `NppStatus nppiSub_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image subtraction.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.24 NppStatus nppiSub_32f_AC4R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image subtraction.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.25 NppStatus nppiSub_32f_C1IR (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image subtraction.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.26 `NppStatus nppiSub_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image subtraction.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.19.2.27 `NppStatus nppiSub_32f_C3IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image subtraction.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.19.2.28 `NppStatus nppiSub_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image subtraction.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.29 NppStatus npSub_32f_C4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 32-bit floating point channel in place image subtraction.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.30 NppStatus npSub_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)

Four 32-bit floating point channel image subtraction.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.31 `NppStatus nppiSub_32fc_AC4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.32 `NppStatus nppiSub_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.33 `NppStatus nppiSub_32fc_C1IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.34 NppStatus nppiSub_32fc_C1R (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc * *pSrc2*, int *nSrc2Step*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.35 NppStatus nppiSub_32fc_C3IR (const Npp32fc * *pSrc*, int *nSrcStep*, Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.36 `NppStatus nppiSub_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.37 `NppStatus nppiSub_32fc_C4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.38 `NppStatus nppiSub_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.39 NppStatus nppiSub_32s_C1IRSfs (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.40 NppStatus nppiSub_32s_C1R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image subtraction. Subtract *pSrc1*'s pixels from corresponding pixels in *pSrc2*.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.41 `NppStatus nppiSub_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.42 `NppStatus nppiSub_32s_C3IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.43 `NppStatus nppiSub_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.44 NppStatus nppiSub_32s_C4IRSfs (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.45 NppStatus nppiSub_32s_C4RSfs (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.46 NppStatus nppiSub_32sc_AC4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.47 NppStatus nppiSub_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.48 **NppStatus nppiSub_32sc_C1IRSfs** (const Npp32sc * *pSrc*, int *nSrcStep*, Npp32sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.19.2.49 **NppStatus nppiSub_32sc_C1RSfs** (const Npp32sc * *pSrc1*, int *nSrc1Step*, const Npp32sc * *pSrc2*, int *nSrc2Step*, Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.19.2.50 **NppStatus nppiSub_32sc_C3IRSfs** (const Npp32sc * *pSrc*, int *nSrcStep*, Npp32sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.51 NppStatus nppiSub_32sc_C3RSfs (const Npp32sc * *pSrc1*, int *nSrc1Step*, const Npp32sc * *pSrc2*, int *nSrc2Step*, Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.52 NppStatus nppiSub_8u_AC4IRSfs (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.53 `NppStatus nppiSub_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.19.2.54 `NppStatus nppiSub_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.19.2.55 `NppStatus nppiSub_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.56 NppStatus nppiSub_8u_C3IRSfs (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image subtraction, scale by $2^{-(nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.57 NppStatus nppiSub_8u_C3RSfs (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image subtraction, scale by $2^{-(nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.58 NppStatus nppiSub_8u_C4IRSfs (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.19.2.59 NppStatus nppiSub_8u_C4RSfs (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20 Div

Pixel by pixel division of two images.

Functions

- **NppStatus** **nppiDiv_8u_C1RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiDiv_8u_C1IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiDiv_8u_C3RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiDiv_8u_C3IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiDiv_8u_AC4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiDiv_8u_AC4IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel with unmodified alpha in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiDiv_8u_C4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiDiv_8u_C4IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiDiv_16u_C1RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiDiv_16u_C1IRSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_16u_C3RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_16u_C3IRSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_16u_AC4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_16u_AC4IRSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_16u_C4RSfs` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_16u_C4IRSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_16s_C1RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_16s_C1IRSfs` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_16s_C3RSfs` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_16s_C3IRSfs` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_16s_AC4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel with unmodified alpha image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_16s_AC4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel with unmodified alpha in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_16s_C4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_16s_C4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_16sc_C1RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_16sc_C1IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_16sc_C3RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_16sc_C3IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_16sc_AC4RSfs** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_16sc_AC4IRSfs** (const **Npp16sc** *pSrc, int nSrcStep, **Npp16sc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_32s_C1RSfs** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 32-bit signed integer channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_32s_C1R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

- `NppStatus nppiDiv_32s_C1RSfs` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_32s_C3RSfs` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed integer channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_32s_C3IRSfs` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_32sc_C1RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_32sc_C1IRSfs` (const `Npp32sc` *pSrc, int nSrcStep, `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_32sc_C3RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_32sc_C3IRSfs` (const `Npp32sc` *pSrc, int nSrcStep, `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_32sc_AC4RSfs` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiDiv_32sc_AC4IRSfs` (const `Npp32sc` *pSrc, int nSrcStep, `Npp32sc` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image division.
- **NppStatus nppiDiv_32f_C1IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image division.
- **NppStatus nppiDiv_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel image division.
- **NppStatus nppiDiv_32f_C3IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image division.
- **NppStatus nppiDiv_32f_AC4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel with unmodified alpha image division.
- **NppStatus nppiDiv_32f_AC4IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel with unmodified alpha in place image division.
- **NppStatus nppiDiv_32f_C4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel image division.
- **NppStatus nppiDiv_32f_C4IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel in place image division.
- **NppStatus nppiDiv_32fc_C1R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.
- **NppStatus nppiDiv_32fc_C1IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.
- **NppStatus nppiDiv_32fc_C3R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.
- **NppStatus nppiDiv_32fc_C3IR** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32fc** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.
- **NppStatus nppiDiv_32fc_AC4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **Npp32fc** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division.

- **NppStatus nppiDiv_32fc_AC4IR** (const [Npp32fc](#) *pSrc, int nSrcStep, [Npp32fc](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division.

- **NppStatus nppiDiv_32fc_C4R** (const [Npp32fc](#) *pSrc1, int nSrc1Step, const [Npp32fc](#) *pSrc2, int nSrc2Step, [Npp32fc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

- **NppStatus nppiDiv_32fc_C4IR** (const [Npp32fc](#) *pSrc, int nSrcStep, [Npp32fc](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

7.20.1 Detailed Description

Pixel by pixel division of two images.

7.20.2 Function Documentation

7.20.2.1 **NppStatus nppiDiv_16s_AC4IRSfs** (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

[pSrc](#) [Source-Image Pointer](#).
[nSrcStep](#) [Source-Image Line Step](#).
[pSrcDst](#) [In-Place Image Pointer](#).
[nSrcDstStep](#) [In-Place-Image Line Step](#).
[oSizeROI](#) [Region-of-Interest \(ROI\)](#).
[nScaleFactor](#) [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.20.2.2 **NppStatus nppiDiv_16s_AC4RSfs** (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [Npp16s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

[pSrc1](#) [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.3 NppStatus nppiDiv_16s_C1IRSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.4 NppStatus nppiDiv_16s_C1RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.5 NppStatus nppiDiv_16s_C3IRSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.6 NppStatus nppiDiv_16s_C3RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.7 NppStatus nppiDiv_16s_C4IRSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.8 NppStatus nppiDiv_16s_C4RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.9 NppStatus nppiDiv_16sc_AC4IRSfs (const Npp16sc * *pSrc*, int *nSrcStep*, Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.10 `NppStatus nppiDiv_16sc_AC4RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.11 `NppStatus nppiDiv_16sc_C1IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.12 `NppStatus nppiDiv_16sc_C1RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.13 NppStatus nppiDiv_16sc_C3IRSfs (const Npp16sc * *pSrc*, int *nSrcStep*, Npp16sc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by $2^{-(nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.14 NppStatus nppiDiv_16sc_C3RSfs (const Npp16sc * *pSrc1*, int *nSrc1Step*, const Npp16sc * *pSrc2*, int *nSrc2Step*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by $2^{-(nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.15 NppStatus nppiDiv_16u_AC4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.16 NppStatus nppiDiv_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.17 NppStatus nppiDiv_16u_C1IRSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.18 NppStatus nppiDiv_16u_C1RSfs (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.19 NppStatus nppiDiv_16u_C3IRSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.20 `NppStatus npDiv_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.21 `NppStatus npDiv_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.22 `NppStatus nppiDiv_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.23 `NppStatus nppiDiv_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image division.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.24 `NppStatus nppiDiv_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha image division.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.25 NppStatus nppiDiv_32f_C1IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One 32-bit floating point channel in place image division.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.26 NppStatus nppiDiv_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)

One 32-bit floating point channel image division.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.27 NppStatus nppiDiv_32f_C3IR (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image division.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.20.2.28 NppStatus nppiDiv_32f_C3R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel image division.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.20.2.29 NppStatus nppiDiv_32f_C4IR (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image division.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.20.2.30 **NppStatus nppiDiv_32f_C4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image division.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.31 **NppStatus nppiDiv_32fc_AC4IR** (const Npp32fc * *pSrc*, int *nSrcStep*, Npp32fc * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.32 **NppStatus nppiDiv_32fc_AC4R** (const Npp32fc * *pSrc1*, int *nSrc1Step*, const Npp32fc * *pSrc2*, int *nSrc2Step*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.33 NppStatus nppiDiv_32fc_C1IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.34 NppStatus nppiDiv_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.35 `NppStatus nppiDiv_32fc_C3IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.20.2.36 `NppStatus nppiDiv_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.20.2.37 `NppStatus nppiDiv_32fc_C4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.20.2.38 `NppStatus nppiDiv_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.39 `NppStatus nppiDiv_32s_C1IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.40 `NppStatus nppiDiv_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image division. Divide pixels in pSrc2 by pSrc1's pixels.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.41 `NppStatus nppiDiv_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.42 `NppStatus nppiDiv_32s_C3IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.43 `NppStatus nppiDiv_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.44 `NppStatus nppiDiv_32sc_AC4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.45 `NppStatus nppiDiv_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.46 `NppStatus nppiDiv_32sc_C1IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by $2^{-(nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.47 `NppStatus nppiDiv_32sc_C1IRSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by $2^{-(nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.48 NppStatus nppiDiv_32sc_C3IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.49 NppStatus nppiDiv_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.50 `NppStatus nppiDiv_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.20.2.51 `NppStatus nppiDiv_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.20.2.52 `NppStatus nppiDiv_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.53 `NppStatus nppiDiv_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.54 `NppStatus nppiDiv_8u_C3IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.20.2.55 `NppStatus nppiDiv_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.20.2.56 `NppStatus nppiDiv_8u_C4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.20.2.57 `NppStatus nppiDiv_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21 Div_Round

Pixel by pixel division of two images using result rounding modes.

Functions

- **NppStatus nppiDiv_Round_8u_C1RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
One 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_8u_C1IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
One 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_8u_C3RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Three 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_8u_C3IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Three 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_8u_AC4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Four 8-bit unsigned char channel image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_8u_AC4IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Four 8-bit unsigned char channel in place image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_8u_C4RSfs** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Four 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_8u_C4IRSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Four 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_16u_C1RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

One 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_Round_16u_C1RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

One 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_Round_16u_C3RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

Three 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_Round_16u_C3RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

Three 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_Round_16u_AC4RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

Four 16-bit unsigned short channel image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_Round_16u_AC4RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

Four 16-bit unsigned short channel in place image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_Round_16u_C4RSfs** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

Four 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_Round_16u_C4RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

Four 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_Round_16s_C1RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

One 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_Round_16s_C1RSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

One 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiDiv_Round_16s_C3RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Three 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_16s_C3IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Three 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_16s_AC4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Four 16-bit signed short channel image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_16s_AC4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Four 16-bit signed short channel in place image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_16s_C4RSfs** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Four 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiDiv_Round_16s_C4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)
Four 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

7.21.1 Detailed Description

Pixel by pixel division of two images using result rounding modes.

7.21.2 Function Documentation

7.21.2.1 **NppStatus nppiDiv_Round_16s_AC4IRSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

Four 16-bit signed short channel in place image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

- pSrc** Source-Image Pointer.
- nSrcStep** Source-Image Line Step.
- pSrcDst** In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.2 NppStatus nppiDiv_Round_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)

Four 16-bit signed short channel image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.3 NppStatus nppiDiv_Round_16s_C1IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)

One 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.21.2.4 NppStatus nppiDiv_Round_16s_C1RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)

One 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.21.2.5 NppStatus nppiDiv_Round_16s_C3IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)

Three 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.21.2.6 NppStatus nppiDiv_Round_16s_C3RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Three 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.7 NppStatus nppiDiv_Round_16s_C4IRSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Four 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.8 NppStatus nppiDiv_Round_16s_C4RSfs (const Npp16s * *pSrc1*, int *nSrc1Step*, const Npp16s * *pSrc2*, int *nSrc2Step*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Four 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.9 NppStatus nppiDiv_Round_16u_AC4IRSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.10 NppStatus nppiDiv_Round_16u_AC4RSfs (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Four 16-bit unsigned short channel image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.11 NppStatus nppiDiv_Round_16u_C1IRSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.12 `NppStatus nppiDiv_Round_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.21.2.13 `NppStatus nppiDiv_Round_16u_C3IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Three 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.21.2.14 NppStatus nppiDiv_Round_16u_C3RSfs (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Three 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.15 NppStatus nppiDiv_Round_16u_C4IRSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.16 `NppStatus nppiDiv_Round_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.17 `NppStatus nppiDiv_Round_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel in place image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.18 `NppStatus nppiDiv_Round_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.19 `NppStatus nppiDiv_Round_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.20 NppStatus nppiDiv_Round_8u_C1RSfs (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

One 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.21.2.21 NppStatus nppiDiv_Round_8u_C3IRSfs (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Three 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.21.2.22 NppStatus nppiDiv_Round_8u_C3RSfs (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Three 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.23 NppStatus nppiDiv_Round_8u_C4IRSfs (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.21.2.24 `NppStatus nppiDiv_Round_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rndMode Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.22 Abs

Absolute value of each pixel value in an image.

Functions

- **NppStatus** **nppiAbs_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit signed short channel image absolute value.
- **NppStatus** **nppiAbs_16s_C1IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit signed short channel in place image absolute value.
- **NppStatus** **nppiAbs_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 16-bit signed short channel image absolute value.
- **NppStatus** **nppiAbs_16s_C3IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 16-bit signed short channel in place image absolute value.
- **NppStatus** **nppiAbs_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit signed short channel image absolute value with unmodified alpha.
- **NppStatus** **nppiAbs_16s_AC4IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit signed short channel in place image absolute value with unmodified alpha.
- **NppStatus** **nppiAbs_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit signed short channel image absolute value.
- **NppStatus** **nppiAbs_16s_C4IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit signed short channel in place image absolute value.
- **NppStatus** **nppiAbs_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image absolute value.
- **NppStatus** **nppiAbs_32f_C1IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image absolute value.
- **NppStatus** **nppiAbs_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel image absolute value.
- **NppStatus** **nppiAbs_32f_C3IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel in place image absolute value.
- **NppStatus** **nppiAbs_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 32-bit floating point channel image absolute value with unmodified alpha.

- **NppStatus nppiAbs_32f_AC4IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel in place image absolute value with unmodified alpha.
- **NppStatus nppiAbs_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel image absolute value.
- **NppStatus nppiAbs_32f_C4IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel in place image absolute value.

7.22.1 Detailed Description

Absolute value of each pixel value in an image.

7.22.2 Function Documentation

7.22.2.1 NppStatus nppiAbs_16s_AC4IR (Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 16-bit signed short channel in place image absolute value with unmodified alpha.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.2 NppStatus nppiAbs_16s_AC4R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit signed short channel image absolute value with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.3 NppStatus nppiAbs_16s_C1IR (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit signed short channel in place image absolute value.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.4 NppStatus nppiAbs_16s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit signed short channel image absolute value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.5 NppStatus nppiAbs_16s_C3IR (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit signed short channel in place image absolute value.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.6 NppStatus nppiAbs_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit signed short channel image absolute value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.7 NppStatus nppiAbs_16s_C4IR (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit signed short channel in place image absolute value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.8 NppStatus nppiAbs_16s_C4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit signed short channel image absolute value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.9 NppStatus nppiAbs_32f_AC4IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image absolute value with unmodified alpha.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.10 NppStatus nppiAbs_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image absolute value with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.11 NppStatus nppiAbs_32f_C1IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image absolute value.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.12 NppStatus nppiAbs_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image absolute value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.13 NppStatus nppiAbs_32f_C3IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel in place image absolute value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.14 NppStatus nppiAbs_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel image absolute value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.15 NppStatus nppiAbs_32f_C4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 32-bit floating point channel in place image absolute value.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.22.2.16 NppStatus nppiAbs_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)

Four 32-bit floating point channel image absolute value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.23 AbsDiff

Pixel by pixel absolute difference between two images.

Functions

- **NppStatus** **nppiAbsDiff_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel absolute difference of image1 minus image2.
- **NppStatus** **nppiAbsDiff_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channels absolute difference of image1 minus image2.
- **NppStatus** **nppiAbsDiff_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channels absolute difference of image1 minus image2.
- **NppStatus** **nppiAbsDiff_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel absolute difference of image1 minus image2.
- **NppStatus** **nppiAbsDiff_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel absolute difference of image1 minus image2.

7.23.1 Detailed Description

Pixel by pixel absolute difference between two images.

7.23.2 Function Documentation

7.23.2.1 **NppStatus** **nppiAbsDiff_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

One 16-bit unsigned short channel absolute difference of image1 minus image2.

Parameters:

- pSrc1** Source-Image Pointer.
- nSrc1Step** Source-Image Line Step.
- pSrc2** Source-Image Pointer.
- nSrc2Step** Source-Image Line Step.
- pDst** Destination-Image Pointer.
- nDstStep** Destination-Image Line Step.
- oSizeROI** Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.23.2.2 **NppStatus nppiAbsDiff_32f_C1R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel absolute difference of image1 minus image2.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.23.2.3 **NppStatus nppiAbsDiff_8u_C1R** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel absolute difference of image1 minus image2.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.23.2.4 **NppStatus nppiAbsDiff_8u_C3R** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channels absolute difference of image1 minus image2.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.23.2.5 `NppStatus nppiAbsDiff_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channels absolute difference of image1 minus image2.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24 Sqr

Square each pixel in an image.

Functions

- **NppStatus nppiSqr_8u_C1RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_8u_C1IRSfs** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_8u_C3RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_8u_C3IRSfs** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_8u_AC4RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_8u_AC4IRSfs** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel in place image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_8u_C4RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_8u_C4IRSfs** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_16u_C1RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_16u_C1IRSfs** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiSqr_16u_C3RSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiSqr_16u_C3IRSfs` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiSqr_16u_AC4RSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiSqr_16u_AC4IRSfs` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel in place image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiSqr_16u_C4RSfs` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiSqr_16u_C4IRSfs` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiSqr_16s_C1RSfs` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiSqr_16s_C1IRSfs` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiSqr_16s_C3RSfs` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- `NppStatus nppiSqr_16s_C3IRSfs` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 16-bit signed short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSqr_16s_AC4RSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_16s_AC4IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel in place image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_16s_C4RSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_16s_C4IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqr_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image squared.
- **NppStatus nppiSqr_32f_C1IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image squared.
- **NppStatus nppiSqr_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel image squared.
- **NppStatus nppiSqr_32f_C3IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel in place image squared.
- **NppStatus nppiSqr_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel image squared with unmodified alpha.
- **NppStatus nppiSqr_32f_AC4IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel in place image squared with unmodified alpha.
- **NppStatus nppiSqr_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel image squared.
- **NppStatus nppiSqr_32f_C4IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit floating point channel in place image squared.

7.24.1 Detailed Description

Square each pixel in an image.

7.24.2 Function Documentation

7.24.2.1 NppStatus nppiSqr_16s_AC4IRSfs (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.24.2.2 NppStatus nppiSqr_16s_AC4RSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.24.2.3 NppStatus nppiSqr_16s_C1IRSfs (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.24.2.4 NppStatus nppiSqr_16s_C1RSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.5 NppStatus nppiSqr_16s_C3IRSfs (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.6 NppStatus nppiSqr_16s_C3RSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.7 **NppStatus nppiSqr_16s_C4IRSfs** (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.24.2.8 **NppStatus nppiSqr_16s_C4RSfs** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.24.2.9 **NppStatus nppiSqr_16u_AC4IRSfs** (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.24.2.10 NppStatus nppiSqr_16u_AC4RSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.11 NppStatus nppiSqr_16u_C1IRSfs (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.12 NppStatus nppiSqr_16u_C1RSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.13 NppStatus nppiSqr_16u_C3IRSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.14 NppStatus nppiSqr_16u_C3RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.15 NppStatus nppiSqr_16u_C4IRSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.16 NppStatus nppiSqr_16u_C4RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.17 NppStatus nppiSqr_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 32-bit floating point channel in place image squared with unmodified alpha.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.18 NppStatus nppiSqr_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)

Four 32-bit floating point channel image squared with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.19 NppStatus nppiSqr_32f_C1IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image squared.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.20 NppStatus nppiSqr_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image squared.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.21 NppStatus nppiSqr_32f_C3IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel in place image squared.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.22 NppStatus nppiSqr_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel image squared.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.23 NppStatus nppiSqr_32f_C4IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image squared.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.24 NppStatus nppiSqr_32f_C4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image squared.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.25 `NppStatus nppiSqr_8u_AC4IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.24.2.26 `NppStatus nppiSqr_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image squared with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.24.2.27 `NppStatus nppiSqr_8u_C1IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.24.2.28 NppStatus nppiSqr_8u_C1RSfs (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.29 NppStatus nppiSqr_8u_C3IRSfs (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.30 NppStatus nppiSqr_8u_C3RSfs (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.24.2.31 NppStatus nppiSqr_8u_C4IRSfs (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.24.2.32 NppStatus nppiSqr_8u_C4RSfs (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image squared, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.25 Sqrt

Pixel by pixel square root of each pixel in an image.

Functions

- **NppStatus** **nppiSqrt_8u_C1RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiSqrt_8u_C1IRSfs** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiSqrt_8u_C3RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiSqrt_8u_C3IRSfs** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiSqrt_8u_AC4RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiSqrt_8u_AC4IRSfs** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 8-bit unsigned char channel in place image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiSqrt_16u_C1RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiSqrt_16u_C1IRSfs** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiSqrt_16u_C3RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit unsigned short channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiSqrt_16u_C3IRSfs** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit unsigned short channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqrt_16u_AC4RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqrt_16u_AC4IRSfs** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit unsigned short channel in place image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqrt_16s_C1RSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqrt_16s_C1IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqrt_16s_C3RSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit signed short channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqrt_16s_C3IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit signed short channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqrt_16s_AC4RSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqrt_16s_AC4IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Four 16-bit signed short channel in place image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiSqrt_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image square root.
- **NppStatus nppiSqrt_32f_C1IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image square root.

- **NppStatus nppiSqrt_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)

Three 32-bit floating point channel image square root.

- **NppStatus nppiSqrt_32f_C3IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Three 32-bit floating point channel in place image square root.

- **NppStatus nppiSqrt_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 32-bit floating point channel image square root with unmodified alpha.

- **NppStatus nppiSqrt_32f_AC4IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 32-bit floating point channel in place image square root with unmodified alpha.

- **NppStatus nppiSqrt_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 32-bit floating point channel image square root.

- **NppStatus nppiSqrt_32f_C4IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 32-bit floating point channel in place image square root.

7.25.1 Detailed Description

Pixel by pixel square root of each pixel in an image.

7.25.2 Function Documentation

7.25.2.1 **NppStatus nppiSqrt_16s_AC4IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel in place image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst **In-Place Image Pointer.**

nSrcDstStep **In-Place-Image Line Step.**

oSizeROI **Region-of-Interest (ROI).**

nScaleFactor **Integer Result Scaling.**

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.2 **NppStatus nppiSqrt_16s_AC4RSfs** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.3 **NppStatus nppiSqrt_16s_C1IRSfs** (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.4 **NppStatus nppiSqrt_16s_C1RSfs** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.5 NppStatus nppiSqrt_16s_C3IRSfs (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit signed short channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.6 NppStatus nppiSqrt_16s_C3RSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit signed short channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.7 NppStatus nppiSqrt_16u_AC4IRSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel in place image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.8 NppStatus nppiSqrt_16u_AC4RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.9 NppStatus nppiSqrt_16u_C1IRSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.10 **NppStatus nppiSqrt_16u_C1RSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit unsigned short channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.11 **NppStatus nppiSqrt_16u_C3IRSfs (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit unsigned short channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.12 **NppStatus nppiSqrt_16u_C3RSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit unsigned short channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.13 NppStatus nppiSqrt_32f_AC4IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image square root with unmodified alpha.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.14 NppStatus nppiSqrt_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image square root with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.15 NppStatus nppiSqrt_32f_C1IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image square root.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.16 NppStatus nppiSqrt_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image square root.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.17 NppStatus nppiSqrt_32f_C3IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel in place image square root.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.18 NppStatus nppiSqrt_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel image square root.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.19 NppStatus nppiSqrt_32f_C4IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image square root.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.20 NppStatus nppiSqrt_32f_C4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image square root.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.21 NppStatus nppiSqrt_8u_AC4IRSfs (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.22 NppStatus nppiSqrt_8u_AC4RSfs (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image square root with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.25.2.23 NppStatus nppiSqrt_8u_C1IRSfs (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.25.2.24 NppStatus nppiSqrt_8u_C1RSfs (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.25 `NppStatus nppiSqrt_8u_C3IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel in place image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.25.2.26 `NppStatus nppiSqrt_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image square root, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26 Ln

Pixel by pixel natural logarithm of each pixel in an image.

Functions

- **NppStatus nppiLn_8u_C1RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiLn_8u_C1IRSfs** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiLn_8u_C3RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiLn_8u_C3IRSfs** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiLn_16u_C1RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiLn_16u_C1IRSfs** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiLn_16u_C3RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit unsigned short channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiLn_16u_C3IRSfs** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit unsigned short channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiLn_16s_C1RSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiLn_16s_C1IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiLn_16s_C3RSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit signed short channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiLn_16s_C3IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit signed short channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiLn_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image natural logarithm.
- **NppStatus nppiLn_32f_C1IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image natural logarithm.
- **NppStatus nppiLn_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel image natural logarithm.
- **NppStatus nppiLn_32f_C3IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel in place image natural logarithm.

7.26.1 Detailed Description

Pixel by pixel natural logarithm of each pixel in an image.

7.26.2 Function Documentation

7.26.2.1 **NppStatus nppiLn_16s_C1IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.2 NppStatus nppiLn_16s_C1RSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.3 NppStatus nppiLn_16s_C3IRSfs (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.4 NppStatus nppiLn_16s_C3RSfs (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.5 NppStatus nppiLn_16u_C1IRSfs (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.26.2.6 NppStatus nppiLn_16u_C1RSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.26.2.7 NppStatus nppiLn_16u_C3IRSfs (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.26.2.8 NppStatus nppiLn_16u_C3RSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.9 NppStatus nppiLn_32f_C1IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image natural logarithm.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.10 NppStatus nppiLn_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image natural logarithm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.11 NppStatus nppiLn_32f_C3IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel in place image natural logarithm.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.12 NppStatus nppiLn_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel image natural logarithm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.13 NppStatus nppiLn_8u_C1IRSfs (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.14 **NppStatus nppiLn_8u_C1RSfs** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.15 **NppStatus nppiLn_8u_C3IRSfs** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.26.2.16 **NppStatus nppiLn_8u_C3RSfs** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27 Exp

Exponential value of each pixel in an image.

Functions

- **NppStatus** **nppiExp_8u_C1RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiExp_8u_C1IRSfs** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 8-bit unsigned char channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiExp_8u_C3RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiExp_8u_C3IRSfs** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 8-bit unsigned char channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiExp_16u_C1RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiExp_16u_C1IRSfs** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit unsigned short channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiExp_16u_C3RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit unsigned short channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiExp_16u_C3IRSfs** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit unsigned short channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus** **nppiExp_16s_C1RSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

- **NppStatus nppiExp_16s_C1IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
One 16-bit signed short channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiExp_16s_C3RSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit signed short channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiExp_16s_C3IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
Three 16-bit signed short channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.
- **NppStatus nppiExp_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel image exponential.
- **NppStatus nppiExp_32f_C1IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit floating point channel in place image exponential.
- **NppStatus nppiExp_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel image exponential.
- **NppStatus nppiExp_32f_C3IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 32-bit floating point channel in place image exponential.

7.27.1 Detailed Description

Exponential value of each pixel in an image.

7.27.2 Function Documentation

7.27.2.1 **NppStatus nppiExp_16s_C1IRSfs** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.2 **NppStatus nppiExp_16s_C1RSfs** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.3 **NppStatus nppiExp_16s_C3IRSfs** (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.4 **NppStatus nppiExp_16s_C3RSfs** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.5 NppStatus nppiExp_16u_C1IRSfs (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.27.2.6 NppStatus nppiExp_16u_C1RSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.27.2.7 NppStatus nppiExp_16u_C3IRSfs (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.27.2.8 NppStatus nppiExp_16u_C3RSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.9 NppStatus nppiExp_32f_C1IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image exponential.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.10 NppStatus nppiExp_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image exponential.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.11 NppStatus nppiExp_32f_C3IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel in place image exponential.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.12 NppStatus nppiExp_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel image exponential.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.13 NppStatus nppiExp_8u_C1IRSfs (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.14 **NppStatus nppiExp_8u_C1RSfs** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.15 **NppStatus nppiExp_8u_C3IRSfs** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.27.2.16 **NppStatus nppiExp_8u_C3RSfs** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.28 Logical Operations

Modules

- [AndC](#)

Pixel by pixel logical and of an image with a constant.

- [OrC](#)

Pixel by pixel logical or of an image with a constant.

- [XorC](#)

Pixel by pixel logical exclusive or of an image with a constant.

- [RShiftC](#)

Pixel by pixel right shift of an image by a constant value.

- [LShiftC](#)

Pixel by pixel left shift of an image by a constant value.

- [And](#)

Pixel by pixel logical and of images.

- [Or](#)

Pixel by pixel logical or of images.

- [Xor](#)

Pixel by pixel logical exclusive or of images.

- [Not](#)

Pixel by pixel logical not of image.

7.29 AndC

Pixel by pixel logical and of an image with a constant.

Functions

- **NppStatus** **nppiAndC_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image logical and with constant.
- **NppStatus** **nppiAndC_8u_C1IR** (const **Npp8u** nConstant, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel in place image logical and with constant.
- **NppStatus** **nppiAndC_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel image logical and with constant.
- **NppStatus** **nppiAndC_8u_C3IR** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel in place image logical and with constant.
- **NppStatus** **nppiAndC_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical and with constant with unmodified alpha.
- **NppStatus** **nppiAndC_8u_AC4IR** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical and with constant with unmodified alpha.
- **NppStatus** **nppiAndC_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical and with constant.
- **NppStatus** **nppiAndC_8u_C4IR** (const **Npp8u** aConstants[4], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical and with constant.
- **NppStatus** **nppiAndC_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image logical and with constant.
- **NppStatus** **nppiAndC_16u_C1IR** (const **Npp16u** nConstant, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel in place image logical and with constant.
- **NppStatus** **nppiAndC_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel image logical and with constant.

- **NppStatus nppiAndC_16u_C3IR** (const **Npp16u** aConstants[3], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel in place image logical and with constant.
- **NppStatus nppiAndC_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel image logical and with constant with unmodified alpha.
- **NppStatus nppiAndC_16u_AC4IR** (const **Npp16u** aConstants[3], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel in place image logical and with constant with unmodified alpha.
- **NppStatus nppiAndC_16u_C4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[4], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel image logical and with constant.
- **NppStatus nppiAndC_16u_C4IR** (const **Npp16u** aConstants[4], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel in place image logical and with constant.
- **NppStatus nppiAndC_32s_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** nConstant, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit signed integer channel image logical and with constant.
- **NppStatus nppiAndC_32s_C1IR** (const **Npp32s** nConstant, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit signed integer channel in place image logical and with constant.
- **NppStatus nppiAndC_32s_C3R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** aConstants[3], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit signed integer channel image logical and with constant.
- **NppStatus nppiAndC_32s_C3IR** (const **Npp32s** aConstants[3], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 32-bit signed integer channel in place image logical and with constant.
- **NppStatus nppiAndC_32s_AC4R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** aConstants[3], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel image logical and with constant with unmodified alpha.
- **NppStatus nppiAndC_32s_AC4IR** (const **Npp32s** aConstants[3], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel in place image logical and with constant with unmodified alpha.
- **NppStatus nppiAndC_32s_C4R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** aConstants[4], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel image logical and with constant.
- **NppStatus nppiAndC_32s_C4IR** (const **Npp32s** aConstants[4], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel in place image logical and with constant.

7.29.1 Detailed Description

Pixel by pixel logical and of an image with a constant.

7.29.2 Function Documentation

7.29.2.1 `NppStatus nppiAndC_16u_AC4IR (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical and with constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.29.2.2 `NppStatus nppiAndC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical and with constant with unmodified alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.29.2.3 `NppStatus nppiAndC_16u_C1IR (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical and with constant.

Parameters:

nConstant Constant.
pSrcDst [In-Place Image Pointer](#).

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.4 NppStatus nppiAndC_16u_C1R (const Npp16u *pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u *pDst, int nDstStep, NppiSize oSizeROI)

One 16-bit unsigned short channel image logical and with constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nConstant Constant.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.5 NppStatus nppiAndC_16u_C3IR (const Npp16u aConstants[3], Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three 16-bit unsigned short channel in place image logical and with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.6 NppStatus nppiAndC_16u_C3R (const Npp16u *pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u *pDst, int nDstStep, NppiSize oSizeROI)

Three 16-bit unsigned short channel image logical and with constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.7 NppStatus nppiAndC_16u_C4IR (const Npp16u aConstants[4], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel in place image logical and with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.8 NppStatus nppiAndC_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel image logical and with constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.9 NppStatus nppiAndC_32s_AC4IR (const Npp32s *aConstants*[3], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical and with constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.10 NppStatus nppiAndC_32s_AC4R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical and with constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.11 NppStatus nppiAndC_32s_C1IR (const Npp32s *nConstant*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image logical and with constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.12 **NppStatus nppiAndC_32s_C1R** (**const Npp32s * pSrc1**, **int nSrc1Step**, **const Npp32s nConstant**, **Npp32s * pDst**, **int nDstStep**, **NppiSize oSizeROI**)

One 32-bit signed integer channel image logical and with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.13 **NppStatus nppiAndC_32s_C3IR** (**const Npp32s aConstants[3]**, **Npp32s * pSrcDst**, **int nSrcDstStep**, **NppiSize oSizeROI**)

Three 32-bit signed integer channel in place image logical and with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.14 **NppStatus nppiAndC_32s_C3R** (**const Npp32s * pSrc1**, **int nSrc1Step**, **const Npp32s aConstants[3]**, **Npp32s * pDst**, **int nDstStep**, **NppiSize oSizeROI**)

Three 32-bit signed integer channel image logical and with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.15 NppStatus nppiAndC_32s_C4IR (const Npp32s *aConstants*[4], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical and with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.16 NppStatus nppiAndC_32s_C4R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[4], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical and with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.17 NppStatus nppiAndC_8u_AC4IR (const Npp8u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical and with constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.18 **NppStatus nppiAndC_8u_AC4R** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical and with constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.19 **NppStatus nppiAndC_8u_C1IR** (const Npp8u *nConstant*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image logical and with constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.20 **NppStatus nppiAndC_8u_C1R** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *nConstant*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image logical and with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.21 NppStatus nppiAndC_8u_C3IR (const Npp8u *aConstants*[3], Npp8u **pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image logical and with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.22 NppStatus nppiAndC_8u_C3R (const Npp8u **pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image logical and with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.23 NppStatus nppiAndC_8u_C4IR (const Npp8u *aConstants*[4], Npp8u **pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical and with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.29.2.24 NppStatus nppiAndC_8u_C4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical and with constant.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.30 OrC

Pixel by pixel logical or of an image with a constant.

Functions

- **NppStatus nppiOrC_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image logical or with constant.
- **NppStatus nppiOrC_8u_C1IR** (const **Npp8u** nConstant, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel in place image logical or with constant.
- **NppStatus nppiOrC_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel image logical or with constant.
- **NppStatus nppiOrC_8u_C3IR** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel in place image logical or with constant.
- **NppStatus nppiOrC_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical or with constant with unmodified alpha.
- **NppStatus nppiOrC_8u_AC4IR** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical or with constant with unmodified alpha.
- **NppStatus nppiOrC_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical or with constant.
- **NppStatus nppiOrC_8u_C4IR** (const **Npp8u** aConstants[4], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical or with constant.
- **NppStatus nppiOrC_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image logical or with constant.
- **NppStatus nppiOrC_16u_C1IR** (const **Npp16u** nConstant, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel in place image logical or with constant.
- **NppStatus nppiOrC_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel image logical or with constant.

- `NppStatus nppiOrC_16u_C3IR` (const `Npp16u` aConstants[3], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Three 16-bit unsigned short channel in place image logical or with constant.
- `NppStatus nppiOrC_16u_AC4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel image logical or with constant with unmodified alpha.
- `NppStatus nppiOrC_16u_AC4IR` (const `Npp16u` aConstants[3], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel in place image logical or with constant with unmodified alpha.
- `NppStatus nppiOrC_16u_C4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel image logical or with constant.
- `NppStatus nppiOrC_16u_C4IR` (const `Npp16u` aConstants[4], `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel in place image logical or with constant.
- `NppStatus nppiOrC_32s_C1R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
One 32-bit signed integer channel image logical or with constant.
- `NppStatus nppiOrC_32s_C1IR` (const `Npp32s` nConstant, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
One 32-bit signed integer channel in place image logical or with constant.
- `NppStatus nppiOrC_32s_C3R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three 32-bit signed integer channel image logical or with constant.
- `NppStatus nppiOrC_32s_C3IR` (const `Npp32s` aConstants[3], `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Three 32-bit signed integer channel in place image logical or with constant.
- `NppStatus nppiOrC_32s_AC4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel image logical or with constant with unmodified alpha.
- `NppStatus nppiOrC_32s_AC4IR` (const `Npp32s` aConstants[3], `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel in place image logical or with constant with unmodified alpha.
- `NppStatus nppiOrC_32s_C4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` aConstants[4], `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel image logical or with constant.
- `NppStatus nppiOrC_32s_C4IR` (const `Npp32s` aConstants[4], `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel in place image logical or with constant.

7.30.1 Detailed Description

Pixel by pixel logical or of an image with a constant.

7.30.2 Function Documentation

7.30.2.1 `NppStatus nppiOrC_16u_AC4IR (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical or with constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.2 `NppStatus nppiOrC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical or with constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.3 `NppStatus nppiOrC_16u_C1IR (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical or with constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.4 NppStatus nppiOrC_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

One 16-bit unsigned short channel image logical or with constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nConstant Constant.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.5 NppStatus nppiOrC_16u_C3IR (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three 16-bit unsigned short channel in place image logical or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.6 NppStatus nppiOrC_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

Three 16-bit unsigned short channel image logical or with constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.7 NppStatus nppiOrC_16u_C4IR (const Npp16u aConstants[4], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel in place image logical or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.8 NppStatus nppiOrC_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel image logical or with constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.9 NppStatus nppiOrC_32s_AC4IR (const Npp32s *aConstants*[3], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical or with constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.30.2.10 NppStatus nppiOrC_32s_AC4R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical or with constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.30.2.11 NppStatus nppiOrC_32s_C1IR (const Npp32s *nConstant*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image logical or with constant.

Parameters:

nConstant Constant.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.30.2.12 NppStatus nppiOrC_32s_C1R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *nConstant*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image logical or with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.13 NppStatus nppiOrC_32s_C3IR (const Npp32s *aConstants*[3], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image logical or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.14 NppStatus nppiOrC_32s_C3R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image logical or with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.15 `NppStatus nppiOrC_32s_C4IR (const Npp32s aConstants[4], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.30.2.16 `NppStatus nppiOrC_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s aConstants[4], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical or with constant.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.30.2.17 `NppStatus nppiOrC_8u_AC4IR (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical or with constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.30.2.18 NppStatus nppiOrC_8u_AC4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical or with constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.19 NppStatus nppiOrC_8u_C1IR (const Npp8u *nConstant*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image logical or with constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.20 NppStatus nppiOrC_8u_C1R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *nConstant*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image logical or with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.30.2.21 `NppStatus nppiOrC_8u_C3IR (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image logical or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.30.2.22 `NppStatus nppiOrC_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical or with constant.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.30.2.23 `NppStatus nppiOrC_8u_C4IR (const Npp8u aConstants[4], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.30.2.24 `NppStatus nppiOrC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical or with constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31 XorC

Pixel by pixel logical exclusive or of an image with a constant.

Functions

- **NppStatus nppiXorC_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image logical exclusive or with constant.
- **NppStatus nppiXorC_8u_C1IR** (const **Npp8u** nConstant, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel in place image logical exclusive or with constant.
- **NppStatus nppiXorC_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel image logical exclusive or with constant.
- **NppStatus nppiXorC_8u_C3IR** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel in place image logical exclusive or with constant.
- **NppStatus nppiXorC_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical exclusive or with constant with unmodified alpha.
- **NppStatus nppiXorC_8u_AC4IR** (const **Npp8u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical exclusive or with constant with unmodified alpha.
- **NppStatus nppiXorC_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical exclusive or with constant.
- **NppStatus nppiXorC_8u_C4IR** (const **Npp8u** aConstants[4], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical exclusive or with constant.
- **NppStatus nppiXorC_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image logical exclusive or with constant.
- **NppStatus nppiXorC_16u_C1IR** (const **Npp16u** nConstant, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel in place image logical exclusive or with constant.
- **NppStatus nppiXorC_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel image logical exclusive or with constant.

- **NppStatus nppiXorC_16u_C3IR** (const **Npp16u** aConstants[3], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel in place image logical exclusive or with constant.
- **NppStatus nppiXorC_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel image logical exclusive or with constant with unmodified alpha.
- **NppStatus nppiXorC_16u_AC4IR** (const **Npp16u** aConstants[3], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel in place image logical exclusive or with constant with unmodified alpha.
- **NppStatus nppiXorC_16u_C4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** aConstants[4], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel image logical exclusive or with constant.
- **NppStatus nppiXorC_16u_C4IR** (const **Npp16u** aConstants[4], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel in place image logical exclusive or with constant.
- **NppStatus nppiXorC_32s_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** nConstant, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit signed integer channel image logical exclusive or with constant.
- **NppStatus nppiXorC_32s_C1IR** (const **Npp32s** nConstant, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit signed integer channel in place image logical exclusive or with constant.
- **NppStatus nppiXorC_32s_C3R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** aConstants[3], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit signed integer channel image logical exclusive or with constant.
- **NppStatus nppiXorC_32s_C3IR** (const **Npp32s** aConstants[3], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 32-bit signed integer channel in place image logical exclusive or with constant.
- **NppStatus nppiXorC_32s_AC4R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** aConstants[3], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel image logical exclusive or with constant with unmodified alpha.
- **NppStatus nppiXorC_32s_AC4IR** (const **Npp32s** aConstants[3], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel in place image logical exclusive or with constant with unmodified alpha.
- **NppStatus nppiXorC_32s_C4R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** aConstants[4], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel image logical exclusive or with constant.
- **NppStatus nppiXorC_32s_C4IR** (const **Npp32s** aConstants[4], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel in place image logical exclusive or with constant.

7.31.1 Detailed Description

Pixel by pixel logical exclusive or of an image with a constant.

7.31.2 Function Documentation

7.31.2.1 `NppStatus nppiXorC_16u_AC4IR (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical exclusive or with constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.31.2.2 `NppStatus nppiXorC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical exclusive or with constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.31.2.3 `NppStatus nppiXorC_16u_C1IR (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical exclusive or with constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.4 NppStatus nppiXorC_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

One 16-bit unsigned short channel image logical exclusive or with constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nConstant Constant.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.5 NppStatus nppiXorC_16u_C3IR (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three 16-bit unsigned short channel in place image logical exclusive or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.6 NppStatus nppiXorC_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

Three 16-bit unsigned short channel image logical exclusive or with constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.7 NppStatus nppiXorC_16u_C4IR (const Npp16u aConstants[4], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel in place image logical exclusive or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.8 NppStatus nppiXorC_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel image logical exclusive or with constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.9 NppStatus nppiXorC_32s_AC4IR (const Npp32s *aConstants*[3], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical exclusive or with constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.10 NppStatus nppiXorC_32s_AC4R (const Npp32s * *pSrcI*, int *nSrcIStep*, const Npp32s *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical exclusive or with constant with unmodified alpha.

Parameters:

pSrcI Source-Image Pointer.
nSrcIStep Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.11 NppStatus nppiXorC_32s_C1IR (const Npp32s *nConstant*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image logical exclusive or with constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.12 NppStatus nppiXorC_32s_C1R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *nConstant*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image logical exclusive or with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.13 NppStatus nppiXorC_32s_C3IR (const Npp32s *aConstants*[3], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image logical exclusive or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.14 NppStatus nppiXorC_32s_C3R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image logical exclusive or with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.15 NppStatus nppiXorC_32s_C4IR (const Npp32s *aConstants*[4], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical exclusive or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.16 NppStatus nppiXorC_32s_C4R (const Npp32s * *pSrcI*, int *nSrcIStep*, const Npp32s *aConstants*[4], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical exclusive or with constant.

Parameters:

pSrcI Source-Image Pointer.
nSrcIStep Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.17 NppStatus nppiXorC_8u_AC4IR (const Npp8u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical exclusive or with constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.18 `NppStatus nppiXorC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical exclusive or with constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.19 `NppStatus nppiXorC_8u_C1IR (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical exclusive or with constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.20 `NppStatus nppiXorC_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical exclusive or with constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.31.2.21 NppStatus nppiXorC_8u_C3IR (const Npp8u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image logical exclusive or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.31.2.22 NppStatus nppiXorC_8u_C3R (const Npp8u * *pSrcI*, int *nSrcIStep*, const Npp8u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image logical exclusive or with constant.

Parameters:

pSrcI Source-Image Pointer.
nSrcIStep Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.31.2.23 NppStatus nppiXorC_8u_C4IR (const Npp8u *aConstants*[4], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical exclusive or with constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.31.2.24 NppStatus nppiXorC_8u_C4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical exclusive or with constant.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32 RShiftC

Pixel by pixel right shift of an image by a constant value.

Functions

- **NppStatus nppiRShiftC_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image right shift by constant.
- **NppStatus nppiRShiftC_8u_C1IR** (const **Npp32u** nConstant, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel in place image right shift by constant.
- **NppStatus nppiRShiftC_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel image right shift by constant.
- **NppStatus nppiRShiftC_8u_C3IR** (const **Npp32u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel in place image right shift by constant.
- **NppStatus nppiRShiftC_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image right shift by constant with unmodified alpha.
- **NppStatus nppiRShiftC_8u_AC4IR** (const **Npp32u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image right shift by constant with unmodified alpha.
- **NppStatus nppiRShiftC_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image right shift by constant.
- **NppStatus nppiRShiftC_8u_C4IR** (const **Npp32u** aConstants[4], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image right shift by constant.
- **NppStatus nppiRShiftC_8s_C1R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit signed char channel image right shift by constant.
- **NppStatus nppiRShiftC_8s_C1IR** (const **Npp32u** nConstant, **Npp8s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit signed char channel in place image right shift by constant.
- **NppStatus nppiRShiftC_8s_C3R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit signed char channel image right shift by constant.

- **NppStatus nppiRShiftC_8s_C3IR** (const **Npp32u** aConstants[3], **Npp8s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 8-bit signed char channel in place image right shift by constant.
- **NppStatus nppiRShiftC_8s_AC4R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit signed char channel image right shift by constant with unmodified alpha.
- **NppStatus nppiRShiftC_8s_AC4IR** (const **Npp32u** aConstants[3], **Npp8s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit signed char channel in place image right shift by constant with unmodified alpha.
- **NppStatus nppiRShiftC_8s_C4R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit signed char channel image right shift by constant.
- **NppStatus nppiRShiftC_8s_C4IR** (const **Npp32u** aConstants[4], **Npp8s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit signed char channel in place image right shift by constant.
- **NppStatus nppiRShiftC_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image right shift by constant.
- **NppStatus nppiRShiftC_16u_C1IR** (const **Npp32u** nConstant, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel in place image right shift by constant.
- **NppStatus nppiRShiftC_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel image right shift by constant.
- **NppStatus nppiRShiftC_16u_C3IR** (const **Npp32u** aConstants[3], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel in place image right shift by constant.
- **NppStatus nppiRShiftC_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel image right shift by constant with unmodified alpha.
- **NppStatus nppiRShiftC_16u_AC4IR** (const **Npp32u** aConstants[3], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel in place image right shift by constant with unmodified alpha.
- **NppStatus nppiRShiftC_16u_C4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel image right shift by constant.
- **NppStatus nppiRShiftC_16u_C4IR** (const **Npp32u** aConstants[4], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel in place image right shift by constant.

- **NppStatus nppiRShiftC_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit signed short channel image right shift by constant.
- **NppStatus nppiRShiftC_16s_C1IR** (const **Npp32u** nConstant, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit signed short channel in place image right shift by constant.
- **NppStatus nppiRShiftC_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 16-bit signed short channel image right shift by constant.
- **NppStatus nppiRShiftC_16s_C3IR** (const **Npp32u** aConstants[3], **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 16-bit signed short channel in place image right shift by constant.
- **NppStatus nppiRShiftC_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit signed short channel image right shift by constant with unmodified alpha.
- **NppStatus nppiRShiftC_16s_AC4IR** (const **Npp32u** aConstants[3], **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit signed short channel in place image right shift by constant with unmodified alpha.
- **NppStatus nppiRShiftC_16s_C4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit signed short channel image right shift by constant.
- **NppStatus nppiRShiftC_16s_C4IR** (const **Npp32u** aConstants[4], **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit signed short channel in place image right shift by constant.
- **NppStatus nppiRShiftC_32s_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit signed integer channel image right shift by constant.
- **NppStatus nppiRShiftC_32s_C1IR** (const **Npp32u** nConstant, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit signed integer channel in place image right shift by constant.
- **NppStatus nppiRShiftC_32s_C3R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit signed integer channel image right shift by constant.
- **NppStatus nppiRShiftC_32s_C3IR** (const **Npp32u** aConstants[3], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 32-bit signed integer channel in place image right shift by constant.
- **NppStatus nppiRShiftC_32s_AC4R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 32-bit signed integer channel image right shift by constant with unmodified alpha.

- **NppStatus nppiRShiftC_32s_AC4IR** (const **Npp32u** aConstants[3], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 32-bit signed integer channel in place image right shift by constant with unmodified alpha.

- **NppStatus nppiRShiftC_32s_C4R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 32-bit signed integer channel image right shift by constant.

- **NppStatus nppiRShiftC_32s_C4IR** (const **Npp32u** aConstants[4], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 32-bit signed integer channel in place image right shift by constant.

7.32.1 Detailed Description

Pixel by pixel right shift of an image by a constant value.

7.32.2 Function Documentation

7.32.2.1 **NppStatus nppiRShiftC_16s_AC4IR** (const **Npp32u** aConstants[3], **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit signed short channel in place image right shift by constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.2 **NppStatus nppiRShiftC_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 16-bit signed short channel image right shift by constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.3 NppStatus nppiRShiftC_16s_C1IR (const Npp32u *nConstant*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit signed short channel in place image right shift by constant.

Parameters:

nConstant Constant.

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.4 NppStatus nppiRShiftC_16s_C1R (const Npp16s * *pSrcI*, int *nSrcIStep*, const Npp32u *nConstant*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit signed short channel image right shift by constant.

Parameters:

pSrcI [Source-Image Pointer](#).

nSrcIStep [Source-Image Line Step](#).

nConstant Constant.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.5 NppStatus nppiRShiftC_16s_C3IR (const Npp32u *aConstants*[3], Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit signed short channel in place image right shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.6 NppStatus nppiRShiftC_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI)

Three 16-bit signed short channel image right shift by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.7 NppStatus nppiRShiftC_16s_C4IR (const Npp32u aConstants[4], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 16-bit signed short channel in place image right shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.8 NppStatus nppiRShiftC_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp16s * pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit signed short channel image right shift by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.9 NppStatus nppiRShiftC_16u_AC4IR (const Npp32u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel in place image right shift by constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.10 NppStatus nppiRShiftC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel image right shift by constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.11 NppStatus nppiRShiftC_16u_C1IR (const Npp32u *nConstant*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image right shift by constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.12 NppStatus nppiRShiftC_16u_C1R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp32u *nConstant*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image right shift by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.13 NppStatus nppiRShiftC_16u_C3IR (const Npp32u *aConstants*[3], Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image right shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.14 NppStatus nppiRShiftC_16u_C3R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image right shift by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.15 NppStatus nppiRShiftC_16u_C4IR (const Npp32u *aConstants*[4], Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image right shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.16 NppStatus nppiRShiftC_16u_C4R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[4], Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image right shift by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.17 `NppStatus nppiRShiftC_32s_AC4IR (const Npp32u aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image right shift by constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.18 `NppStatus nppiRShiftC_32s_AC4R (const Npp32s * pSrcI, int nSrcIStep, const Npp32u aConstants[3], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image right shift by constant with unmodified alpha.

Parameters:

pSrcI [Source-Image Pointer](#).
nSrcIStep [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.19 `NppStatus nppiRShiftC_32s_C1IR (const Npp32u nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel in place image right shift by constant.

Parameters:

nConstant Constant.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.20 NppStatus nppiRShiftC_32s_C1R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32u *nConstant*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image right shift by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.21 NppStatus nppiRShiftC_32s_C3IR (const Npp32u *aConstants*[3], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image right shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.22 NppStatus nppiRShiftC_32s_C3R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image right shift by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.23 `NppStatus nppiRShiftC_32s_C4IR (const Npp32u aConstants[4], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image right shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.24 `NppStatus nppiRShiftC_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image right shift by constant.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.25 `NppStatus nppiRShiftC_8s_AC4IR (const Npp32u aConstants[3], Npp8s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit signed char channel in place image right shift by constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.26 NppStatus nppiRShiftC_8s_AC4R (const Npp8s * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit signed char channel image right shift by constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.27 NppStatus nppiRShiftC_8s_C11R (const Npp32u *nConstant*, Npp8s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit signed char channel in place image right shift by constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.28 NppStatus nppiRShiftC_8s_C1R (const Npp8s * *pSrc1*, int *nSrc1Step*, const Npp32u *nConstant*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit signed char channel image right shift by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.29 `NppStatus nppiRShiftC_8s_C3IR (const Npp32u aConstants[3], Npp8s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit signed char channel in place image right shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.30 `NppStatus nppiRShiftC_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit signed char channel image right shift by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.31 `NppStatus nppiRShiftC_8s_C4IR (const Npp32u aConstants[4], Npp8s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit signed char channel in place image right shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.32 NppStatus nppiRShiftC_8s_C4R (const Npp8s * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[4], Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit signed char channel image right shift by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.33 NppStatus nppiRShiftC_8u_AC4IR (const Npp32u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image right shift by constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.34 NppStatus nppiRShiftC_8u_AC4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image right shift by constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.32.2.35 `NppStatus nppiRShiftC_8u_C1IR (const Npp32u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image right shift by constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.36 `NppStatus nppiRShiftC_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image right shift by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.37 `NppStatus nppiRShiftC_8u_C3IR (const Npp32u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image right shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.38 NppStatus nppiRShiftC_8u_C3R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image right shift by constant.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.39 NppStatus nppiRShiftC_8u_C4IR (const Npp32u *aConstants*[4], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image right shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.32.2.40 NppStatus nppiRShiftC_8u_C4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[4], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image right shift by constant.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

aConstants fixed size array of constant values, one per channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.33 LShiftC

Pixel by pixel left shift of an image by a constant value.

Functions

- **NppStatus nppiLShiftC_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image left shift by constant.
- **NppStatus nppiLShiftC_8u_C1IR** (const **Npp32u** nConstant, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel in place image left shift by constant.
- **NppStatus nppiLShiftC_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel image left shift by constant.
- **NppStatus nppiLShiftC_8u_C3IR** (const **Npp32u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel in place image left shift by constant.
- **NppStatus nppiLShiftC_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image left shift by constant with unmodified alpha.
- **NppStatus nppiLShiftC_8u_AC4IR** (const **Npp32u** aConstants[3], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image left shift by constant with unmodified alpha.
- **NppStatus nppiLShiftC_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image left shift by constant.
- **NppStatus nppiLShiftC_8u_C4IR** (const **Npp32u** aConstants[4], **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image left shift by constant.
- **NppStatus nppiLShiftC_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image left shift by constant.
- **NppStatus nppiLShiftC_16u_C1IR** (const **Npp32u** nConstant, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel in place image left shift by constant.
- **NppStatus nppiLShiftC_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel image left shift by constant.

- **NppStatus** **nppiLShiftC_16u_C3IR** (const **Npp32u** aConstants[3], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel in place image left shift by constant.
- **NppStatus** **nppiLShiftC_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel image left shift by constant with unmodified alpha.
- **NppStatus** **nppiLShiftC_16u_AC4IR** (const **Npp32u** aConstants[3], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel in place image left shift by constant with unmodified alpha.
- **NppStatus** **nppiLShiftC_16u_C4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel image left shift by constant.
- **NppStatus** **nppiLShiftC_16u_C4IR** (const **Npp32u** aConstants[4], **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel in place image left shift by constant.
- **NppStatus** **nppiLShiftC_32s_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 32-bit signed integer channel image left shift by constant.
- **NppStatus** **nppiLShiftC_32s_C1IR** (const **Npp32u** nConstant, **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 32-bit signed integer channel in place image left shift by constant.
- **NppStatus** **nppiLShiftC_32s_C3R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 32-bit signed integer channel image left shift by constant.
- **NppStatus** **nppiLShiftC_32s_C3IR** (const **Npp32u** aConstants[3], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 32-bit signed integer channel in place image left shift by constant.
- **NppStatus** **nppiLShiftC_32s_AC4R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel image left shift by constant with unmodified alpha.
- **NppStatus** **nppiLShiftC_32s_AC4IR** (const **Npp32u** aConstants[3], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel in place image left shift by constant with unmodified alpha.
- **NppStatus** **nppiLShiftC_32s_C4R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel image left shift by constant.
- **NppStatus** **nppiLShiftC_32s_C4IR** (const **Npp32u** aConstants[4], **Npp32s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 32-bit signed integer channel in place image left shift by constant.

7.33.1 Detailed Description

Pixel by pixel left shift of an image by a constant value.

7.33.2 Function Documentation

7.33.2.1 `NppStatus nppiLShiftC_16u_AC4IR (const Npp32u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image left shift by constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.33.2.2 `NppStatus nppiLShiftC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image left shift by constant with unmodified alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.33.2.3 `NppStatus nppiLShiftC_16u_C1IR (const Npp32u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image left shift by constant.

Parameters:

nConstant Constant.
pSrcDst [In-Place Image Pointer](#).

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.4 NppStatus nppiLShiftC_16u_C1R (const Npp16u *pSrc1, int nSrc1Step, const Npp32u nConstant, Npp16u *pDst, int nDstStep, NppiSize oSizeROI)

One 16-bit unsigned short channel image left shift by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nConstant Constant

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.5 NppStatus nppiLShiftC_16u_C3IR (const Npp32u aConstants[3], Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three 16-bit unsigned short channel in place image left shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.6 NppStatus nppiLShiftC_16u_C3R (const Npp16u *pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp16u *pDst, int nDstStep, NppiSize oSizeROI)

Three 16-bit unsigned short channel image left shift by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.7 NppStatus nppiLShiftC_16u_C4IR (const Npp32u aConstants[4], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel in place image left shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.8 NppStatus nppiLShiftC_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel image left shift by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.9 NppStatus nppiLShiftC_32s_AC4IR (const Npp32u *aConstants*[3], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image left shift by constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.10 NppStatus nppiLShiftC_32s_AC4R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image left shift by constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.11 NppStatus nppiLShiftC_32s_C1IR (const Npp32u *nConstant*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image left shift by constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.12 **NppStatus nppiLShiftC_32s_C1R** (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32u *nConstant*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image left shift by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.13 **NppStatus nppiLShiftC_32s_C3IR** (const Npp32u *aConstants*[3], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image left shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.14 **NppStatus nppiLShiftC_32s_C3R** (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image left shift by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.15 NppStatus nppiLShiftC_32s_C4IR (const Npp32u *aConstants*[4], Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image left shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.33.2.16 NppStatus nppiLShiftC_32s_C4R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[4], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image left shift by constant.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
aConstants fixed size array of constant values, one per channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.33.2.17 NppStatus nppiLShiftC_8u_AC4IR (const Npp32u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image left shift by constant with unmodified alpha.

Parameters:

aConstants fixed size array of constant values, one per channel.
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.33.2.18 **NppStatus nppiLShiftC_8u_AC4R** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image left shift by constant with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
aConstants fixed size array of constant values, one per channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.19 **NppStatus nppiLShiftC_8u_C11R** (const Npp32u *nConstant*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image left shift by constant.

Parameters:

nConstant Constant.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.20 **NppStatus nppiLShiftC_8u_C1R** (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp32u *nConstant*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image left shift by constant.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nConstant Constant.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.33.2.21 NppStatus nppiLShiftC_8u_C3IR (const Npp32u *aConstants*[3], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image left shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.33.2.22 NppStatus nppiLShiftC_8u_C3R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image left shift by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.33.2.23 NppStatus nppiLShiftC_8u_C4IR (const Npp32u *aConstants*[4], Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image left shift by constant.

Parameters:

aConstants fixed size array of constant values, one per channel.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.33.2.24 `NppStatus nppiLShiftC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image left shift by constant.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

aConstants fixed size array of constant values, one per channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34 And

Pixel by pixel logical and of images.

Functions

- **NppStatus** **nppiAnd_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image logical and.
- **NppStatus** **nppiAnd_8u_C1IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel in place image logical and.
- **NppStatus** **nppiAnd_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel image logical and.
- **NppStatus** **nppiAnd_8u_C3IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel in place image logical and.
- **NppStatus** **nppiAnd_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical and with unmodified alpha.
- **NppStatus** **nppiAnd_8u_AC4IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical and with unmodified alpha.
- **NppStatus** **nppiAnd_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical and.
- **NppStatus** **nppiAnd_8u_C4IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical and.
- **NppStatus** **nppiAnd_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image logical and.
- **NppStatus** **nppiAnd_16u_C1IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel in place image logical and.
- **NppStatus** **nppiAnd_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel image logical and.

- `NppStatus nppiAnd_16u_C3IR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Three 16-bit unsigned short channel in place image logical and.
- `NppStatus nppiAnd_16u_AC4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel image logical and with unmodified alpha.
- `NppStatus nppiAnd_16u_AC4IR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel in place image logical and with unmodified alpha.
- `NppStatus nppiAnd_16u_C4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel image logical and.
- `NppStatus nppiAnd_16u_C4IR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel in place image logical and.
- `NppStatus nppiAnd_32s_C1R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
One 32-bit signed integer channel image logical and.
- `NppStatus nppiAnd_32s_C1IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
One 32-bit signed integer channel in place image logical and.
- `NppStatus nppiAnd_32s_C3R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three 32-bit signed integer channel image logical and.
- `NppStatus nppiAnd_32s_C3IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Three 32-bit signed integer channel in place image logical and.
- `NppStatus nppiAnd_32s_AC4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel image logical and with unmodified alpha.
- `NppStatus nppiAnd_32s_AC4IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel in place image logical and with unmodified alpha.
- `NppStatus nppiAnd_32s_C4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel image logical and.
- `NppStatus nppiAnd_32s_C4IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel in place image logical and.

7.34.1 Detailed Description

Pixel by pixel logical and of images.

7.34.2 Function Documentation

7.34.2.1 `NppStatus nppiAnd_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical and with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.2 `NppStatus nppiAnd_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical and with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.3 `NppStatus nppiAnd_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical and.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.4 NppStatus nppiAnd_16u_C1R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image logical and.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.5 NppStatus nppiAnd_16u_C3IR (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image logical and.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.6 NppStatus nppiAnd_16u_C3R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image logical and.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.7 NppStatus nppiAnd_16u_C4IR (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image logical and.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.8 NppStatus nppiAnd_16u_C4R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image logical and.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.9 NppStatus nppiAnd_32s_AC4IR (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical and with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.10 NppStatus nppiAnd_32s_AC4R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical and with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.11 NppStatus nppiAnd_32s_C1IR (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image logical and.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.12 NppStatus nppiAnd_32s_C1R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image logical and.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.13 NppStatus nppiAnd_32s_C3IR (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image logical and.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.14 **NppStatus nppiAnd_32s_C3R** (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image logical and.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.15 **NppStatus nppiAnd_32s_C4IR** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical and.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.16 **NppStatus nppiAnd_32s_C4R** (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical and.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.17 NppStatus nppiAnd_8u_AC4IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical and with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.18 NppStatus nppiAnd_8u_AC4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical and with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.34.2.19 NppStatus nppiAnd_8u_C1IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image logical and.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.20 NppStatus nppiAnd_8u_C1R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image logical and.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.21 NppStatus nppiAnd_8u_C3IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image logical and.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.22 `NppStatus nppiAnd_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical and.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.23 `NppStatus nppiAnd_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical and.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.34.2.24 `NppStatus nppiAnd_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical and.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35 Or

Pixel by pixel logical or of images.

Functions

- **NppStatus nppiOr_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image logical or.
- **NppStatus nppiOr_8u_C1IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel in place image logical or.
- **NppStatus nppiOr_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel image logical or.
- **NppStatus nppiOr_8u_C3IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel in place image logical or.
- **NppStatus nppiOr_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical or with unmodified alpha.
- **NppStatus nppiOr_8u_AC4IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical or with unmodified alpha.
- **NppStatus nppiOr_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical or.
- **NppStatus nppiOr_8u_C4IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical or.
- **NppStatus nppiOr_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image logical or.
- **NppStatus nppiOr_16u_C1IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel in place image logical or.
- **NppStatus nppiOr_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel image logical or.

- `NppStatus nppiOr_16u_C3IR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Three 16-bit unsigned short channel in place image logical or.
- `NppStatus nppiOr_16u_AC4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel image logical or with unmodified alpha.
- `NppStatus nppiOr_16u_AC4IR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel in place image logical or with unmodified alpha.
- `NppStatus nppiOr_16u_C4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel image logical or.
- `NppStatus nppiOr_16u_C4IR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel in place image logical or.
- `NppStatus nppiOr_32s_C1R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
One 32-bit signed integer channel image logical or.
- `NppStatus nppiOr_32s_C1IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
One 32-bit signed integer channel in place image logical or.
- `NppStatus nppiOr_32s_C3R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three 32-bit signed integer channel image logical or.
- `NppStatus nppiOr_32s_C3IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Three 32-bit signed integer channel in place image logical or.
- `NppStatus nppiOr_32s_AC4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel image logical or with unmodified alpha.
- `NppStatus nppiOr_32s_AC4IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel in place image logical or with unmodified alpha.
- `NppStatus nppiOr_32s_C4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel image logical or.
- `NppStatus nppiOr_32s_C4IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel in place image logical or.

7.35.1 Detailed Description

Pixel by pixel logical or of images.

7.35.2 Function Documentation

7.35.2.1 `NppStatus nppiOr_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical or with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.2 `NppStatus nppiOr_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical or with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.3 `NppStatus nppiOr_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical or.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.4 NppStatus nppiOr_16u_C1R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image logical or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.5 NppStatus nppiOr_16u_C3IR (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image logical or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.6 NppStatus nppiOr_16u_C3R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image logical or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.7 NppStatus nppiOr_16u_C4IR (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image logical or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.8 NppStatus nppiOr_16u_C4R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image logical or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.9 NppStatus nppiOr_32s_AC4IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 32-bit signed integer channel in place image logical or with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.10 NppStatus nppiOr_32s_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)

Four 32-bit signed integer channel image logical or with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.11 `NppStatus nppiOr_32s_C1IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel in place image logical or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.12 `NppStatus nppiOr_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel image logical or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.13 `NppStatus nppiOr_32s_C3IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel in place image logical or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.14 NppStatus nppiOr_32s_C3R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image logical or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.15 NppStatus nppiOr_32s_C4IR (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.16 NppStatus nppiOr_32s_C4R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.17 NppStatus nppiOr_8u_AC4IR (const Npp8u *pSrc, int nSrcStep, Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 8-bit unsigned char channel in place image logical or with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.18 NppStatus nppiOr_8u_AC4R (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

Four 8-bit unsigned char channel image logical or with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.35.2.19 NppStatus nppiOr_8u_C1IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image logical or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.20 NppStatus nppiOr_8u_C1R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image logical or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.21 NppStatus nppiOr_8u_C3IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image logical or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.22 NppStatus nppiOr_8u_C3R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image logical or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.23 NppStatus nppiOr_8u_C4IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.35.2.24 NppStatus nppiOr_8u_C4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36 Xor

Pixel by pixel logical exclusive or of images.

Functions

- `NppStatus nppiXor_8u_C1R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
One 8-bit unsigned char channel image logical exclusive or.
- `NppStatus nppiXor_8u_C1IR` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
One 8-bit unsigned char channel in place image logical exclusive or.
- `NppStatus nppiXor_8u_C3R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three 8-bit unsigned char channel image logical exclusive or.
- `NppStatus nppiXor_8u_C3IR` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Three 8-bit unsigned char channel in place image logical exclusive or.
- `NppStatus nppiXor_8u_AC4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 8-bit unsigned char channel image logical exclusive or with unmodified alpha.
- `NppStatus nppiXor_8u_AC4IR` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 8-bit unsigned char channel in place image logical exclusive or with unmodified alpha.
- `NppStatus nppiXor_8u_C4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 8-bit unsigned char channel image logical exclusive or.
- `NppStatus nppiXor_8u_C4IR` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 8-bit unsigned char channel in place image logical exclusive or.
- `NppStatus nppiXor_16u_C1R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
One 16-bit unsigned short channel image logical exclusive or.
- `NppStatus nppiXor_16u_C1IR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
One 16-bit unsigned short channel in place image logical exclusive or.
- `NppStatus nppiXor_16u_C3R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three 16-bit unsigned short channel image logical exclusive or.

- `NppStatus nppiXor_16u_C3IR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Three 16-bit unsigned short channel in place image logical exclusive or.
- `NppStatus nppiXor_16u_AC4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel image logical exclusive or with unmodified alpha.
- `NppStatus nppiXor_16u_AC4IR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel in place image logical exclusive or with unmodified alpha.
- `NppStatus nppiXor_16u_C4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel image logical exclusive or.
- `NppStatus nppiXor_16u_C4IR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 16-bit unsigned short channel in place image logical exclusive or.
- `NppStatus nppiXor_32s_C1R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
One 32-bit signed integer channel image logical exclusive or.
- `NppStatus nppiXor_32s_C1IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
One 32-bit signed integer channel in place image logical exclusive or.
- `NppStatus nppiXor_32s_C3R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three 32-bit signed integer channel image logical exclusive or.
- `NppStatus nppiXor_32s_C3IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Three 32-bit signed integer channel in place image logical exclusive or.
- `NppStatus nppiXor_32s_AC4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel image logical exclusive or with unmodified alpha.
- `NppStatus nppiXor_32s_AC4IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel in place image logical exclusive or with unmodified alpha.
- `NppStatus nppiXor_32s_C4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel image logical exclusive or.
- `NppStatus nppiXor_32s_C4IR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)
Four 32-bit signed integer channel in place image logical exclusive or.

7.36.1 Detailed Description

Pixel by pixel logical exclusive or of images.

7.36.2 Function Documentation

7.36.2.1 `NppStatus nppiXor_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical exclusive or with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.2 `NppStatus nppiXor_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical exclusive or with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.3 `NppStatus nppiXor_16u_C11R (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical exclusive or.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.4 NppStatus nppiXor_16u_C1R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image logical exclusive or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.5 NppStatus nppiXor_16u_C3IR (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image logical exclusive or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.6 NppStatus nppiXor_16u_C3R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image logical exclusive or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.7 NppStatus nppiXor_16u_C4IR (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image logical exclusive or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.8 NppStatus nppiXor_16u_C4R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image logical exclusive or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.9 NppStatus nppiXor_32s_AC4IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 32-bit signed integer channel in place image logical exclusive or with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.10 NppStatus nppiXor_32s_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)

Four 32-bit signed integer channel image logical exclusive or with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.11 **NppStatus nppiXor_32s_C1IR** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image logical exclusive or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.12 **NppStatus nppiXor_32s_C1R** (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image logical exclusive or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.13 **NppStatus nppiXor_32s_C3IR** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image logical exclusive or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.14 `NppStatus nppiXor_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel image logical exclusive or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.15 `NppStatus nppiXor_32s_C4IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical exclusive or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.16 `NppStatus nppiXor_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical exclusive or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.17 NppStatus nppiXor_8u_AC4IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical exclusive or with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.18 NppStatus nppiXor_8u_AC4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical exclusive or with unmodified alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.36.2.19 `NppStatus nppiXor_8u_C1IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical exclusive or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.20 `NppStatus nppiXor_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical exclusive or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.21 `NppStatus nppiXor_8u_C3IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image logical exclusive or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.22 NppStatus nppiXor_8u_C3R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image logical exclusive or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.23 NppStatus nppiXor_8u_C4IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical exclusive or.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.36.2.24 NppStatus nppiXor_8u_C4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical exclusive or.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37 Not

Pixel by pixel logical not of image.

Functions

- **NppStatus nppiNot_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image logical not.
- **NppStatus nppiNot_8u_C1IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel in place image logical not.
- **NppStatus nppiNot_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel image logical not.
- **NppStatus nppiNot_8u_C3IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel in place image logical not.
- **NppStatus nppiNot_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical not with unmodified alpha.
- **NppStatus nppiNot_8u_AC4IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical not with unmodified alpha.
- **NppStatus nppiNot_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image logical not.
- **NppStatus nppiNot_8u_C4IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image logical not.

7.37.1 Detailed Description

Pixel by pixel logical not of image.

7.37.2 Function Documentation

7.37.2.1 NppStatus nppiNot_8u_AC4IR (Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 8-bit unsigned char channel in place image logical not with unmodified alpha.

Parameters:

pSrcDst **In-Place Image Pointer.**

nSrcDstStep **In-Place-Image Line Step.**

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.2 `NppStatus nppiNot_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical not with unmodified alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.3 `NppStatus nppiNot_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical not.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.4 `NppStatus nppiNot_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical not.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.5 NppStatus nppiNot_8u_C3IR (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image logical not.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.6 NppStatus nppiNot_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image logical not.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.7 NppStatus nppiNot_8u_C4IR (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical not.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.37.2.8 NppStatus nppiNot_8u_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical not.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.38 Alpha Composition

Modules

- [AlphaCompC](#)
Composite two images using constant alpha values.
- [AlphaPremulC](#)
Premultiplies pixels of an image using a constant alpha value.
- [AlphaComp](#)
Composite two images using alpha opacity values contained in each image.
- [AlphaPremul](#)
Premultiplies image pixels by image alpha opacity values.

7.39 AlphaCompC

Composite two images using constant alpha values.

Functions

- `NppStatus nppiAlphaCompC_8u_C1R` (const `Npp8u` *pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

One 8-bit unsigned char channel image composition using constant alpha.

- `NppStatus nppiAlphaCompC_8u_C3R` (const `Npp8u` *pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

Three 8-bit unsigned char channel image composition using constant alpha.

- `NppStatus nppiAlphaCompC_8u_C4R` (const `Npp8u` *pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

Four 8-bit unsigned char channel image composition using constant alpha.

- `NppStatus nppiAlphaCompC_8u_AC4R` (const `Npp8u` *pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

Four 8-bit unsigned char channel image composition with alpha using constant source alpha.

- `NppStatus nppiAlphaCompC_8s_C1R` (const `Npp8s` *pSrc1, int nSrc1Step, `Npp8s` nAlpha1, const `Npp8s` *pSrc2, int nSrc2Step, `Npp8s` nAlpha2, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

One 8-bit signed char channel image composition using constant alpha.

- `NppStatus nppiAlphaCompC_16u_C1R` (const `Npp16u` *pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

One 16-bit unsigned short channel image composition using constant alpha.

- `NppStatus nppiAlphaCompC_16u_C3R` (const `Npp16u` *pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

Three 16-bit unsigned short channel image composition using constant alpha.

- `NppStatus nppiAlphaCompC_16u_C4R` (const `Npp16u` *pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

Four 16-bit unsigned short channel image composition using constant alpha.

- `NppStatus nppiAlphaCompC_16u_AC4R` (const `Npp16u` *pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` *pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

Four 16-bit unsigned short channel image composition with alpha using constant source alpha.

- `NppStatus nppiAlphaCompC_16s_C1R` (const `Npp16s` *pSrc1, int nSrc1Step, `Npp16s` nAlpha1, const `Npp16s` *pSrc2, int nSrc2Step, `Npp16s` nAlpha2, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

One 16-bit signed short channel image composition using constant alpha.

- `NppStatus nppiAlphaCompC_32u_C1R` (const `Npp32u` *pSrc1, int nSrc1Step, `Npp32u` nAlpha1, const `Npp32u` *pSrc2, int nSrc2Step, `Npp32u` nAlpha2, `Npp32u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

One 32-bit unsigned integer channel image composition using constant alpha.

- `NppStatus nppiAlphaCompC_32s_C1R` (const `Npp32s` *pSrc1, int nSrc1Step, `Npp32s` nAlpha1, const `Npp32s` *pSrc2, int nSrc2Step, `Npp32s` nAlpha2, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

One 32-bit signed integer channel image composition using constant alpha.

- `NppStatus nppiAlphaCompC_32f_C1R` (const `Npp32f` *pSrc1, int nSrc1Step, `Npp32f` nAlpha1, const `Npp32f` *pSrc2, int nSrc2Step, `Npp32f` nAlpha2, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

One 32-bit floating point channel image composition using constant alpha.

7.39.1 Detailed Description

Composite two images using constant alpha values.

7.39.2 Function Documentation

7.39.2.1 `NppStatus nppiAlphaCompC_16s_C1R` (const `Npp16s` *pSrc1, int nSrc1Step, `Npp16s` nAlpha1, const `Npp16s` *pSrc2, int nSrc2Step, `Npp16s` nAlpha2, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

One 16-bit signed short channel image composition using constant alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nAlpha1 Image alpha opacity (0 - max channel pixel value).

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

nAlpha2 Image alpha opacity (0 - max channel pixel value).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eAlphaOp alpha-blending operation..

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.39.2.2 **NppStatus nppiAlphaCompC_16u_AC4R** (const Npp16u * *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u *nAlpha2*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Four 16-bit unsigned short channel image composition with alpha using constant source alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.3 **NppStatus nppiAlphaCompC_16u_C1R** (const Npp16u * *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u *nAlpha2*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 16-bit unsigned short channel image composition using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.4 NppStatus nppiAlphaCompC_16u_C3R (const Npp16u * *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u *nAlpha2*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Three 16-bit unsigned short channel image composition using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.5 NppStatus nppiAlphaCompC_16u_C4R (const Npp16u * *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp16u *nAlpha2*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Four 16-bit unsigned short channel image composition using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.6 NppStatus nppiAlphaCompC_32f_C1R (const Npp32f * *pSrc1*, int *nSrc1Step*, Npp32f *nAlpha1*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f *nAlpha2*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 32-bit floating point channel image composition using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0.0 - 1.0).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0.0 - 1.0).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.7 NppStatus nppiAlphaCompC_32s_C1R (const Npp32s * *pSrc1*, int *nSrc1Step*, Npp32s *nAlpha1*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s *nAlpha2*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 32-bit signed integer channel image composition using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.8 NppStatus nppiAlphaCompC_32u_C1R (const Npp32u * *pSrc1*, int *nSrc1Step*, Npp32u *nAlpha1*, const Npp32u * *pSrc2*, int *nSrc2Step*, Npp32u *nAlpha2*, Npp32u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 32-bit unsigned integer channel image composition using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.9 NppStatus nppiAlphaCompC_8s_C1R (const Npp8s * *pSrc1*, int *nSrc1Step*, Npp8s *nAlpha1*, const Npp8s * *pSrc2*, int *nSrc2Step*, Npp8s *nAlpha2*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 8-bit signed char channel image composition using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.10 `NppStatus nppiAlphaCompC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 8-bit unsigned char channel image composition with alpha using constant source alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.11 `NppStatus nppiAlphaCompC_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

One 8-bit unsigned char channel image composition using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.12 `NppStatus nppiAlphaCompC_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Three 8-bit unsigned char channel image composition using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.39.2.13 `NppStatus nppiAlphaCompC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 8-bit unsigned char channel image composition using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
nAlpha2 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40 AlphaPremulC

Premultiplies pixels of an image using a constant alpha value.

Functions

- **NppStatus** **nppiAlphaPremulC_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel image premultiplication using constant alpha.
- **NppStatus** **nppiAlphaPremulC_8u_C1IR** (**Npp8u** nAlpha1, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 8-bit unsigned char channel in place image premultiplication using constant alpha.
- **NppStatus** **nppiAlphaPremulC_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel image premultiplication using constant alpha.
- **NppStatus** **nppiAlphaPremulC_8u_C3IR** (**Npp8u** nAlpha1, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 8-bit unsigned char channel in place image premultiplication using constant alpha.
- **NppStatus** **nppiAlphaPremulC_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image premultiplication using constant alpha.
- **NppStatus** **nppiAlphaPremulC_8u_C4IR** (**Npp8u** nAlpha1, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image premultiplication using constant alpha.
- **NppStatus** **nppiAlphaPremulC_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel image premultiplication with alpha using constant alpha.
- **NppStatus** **nppiAlphaPremulC_8u_AC4IR** (**Npp8u** nAlpha1, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 8-bit unsigned char channel in place image premultiplication with alpha using constant alpha.
- **NppStatus** **nppiAlphaPremulC_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, **Npp16u** nAlpha1, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel image premultiplication using constant alpha.
- **NppStatus** **nppiAlphaPremulC_16u_C1IR** (**Npp16u** nAlpha1, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
One 16-bit unsigned short channel in place image premultiplication using constant alpha.
- **NppStatus** **nppiAlphaPremulC_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, **Npp16u** nAlpha1, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel image premultiplication using constant alpha.

- **NppStatus nppiAlphaPremulC_16u_C3IR** (**Npp16u** nAlpha1, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Three 16-bit unsigned short channel in place image premultiplication using constant alpha.
- **NppStatus nppiAlphaPremulC_16u_C4R** (const **Npp16u** *pSrc1, int nSrc1Step, **Npp16u** nAlpha1, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel image premultiplication using constant alpha.
- **NppStatus nppiAlphaPremulC_16u_C4IR** (**Npp16u** nAlpha1, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel in place image premultiplication using constant alpha.
- **NppStatus nppiAlphaPremulC_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, **Npp16u** nAlpha1, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel image premultiplication with alpha using constant alpha.
- **NppStatus nppiAlphaPremulC_16u_AC4IR** (**Npp16u** nAlpha1, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
Four 16-bit unsigned short channel in place image premultiplication with alpha using constant alpha.

7.40.1 Detailed Description

Premultiplies pixels of an image using a constant alpha value.

7.40.2 Function Documentation

7.40.2.1 **NppStatus nppiAlphaPremulC_16u_AC4IR** (**Npp16u** nAlpha1, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel in place image premultiplication with alpha using constant alpha.

Parameters:

nAlpha1 Image alpha opacity (0 - max channel pixel value).

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.40.2.2 **NppStatus nppiAlphaPremulC_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, **Npp16u** nAlpha1, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel image premultiplication with alpha using constant alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nAlpha1 Image alpha opacity (0 - max channel pixel value).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.40.2.3 NppStatus nppiAlphaPremulC_16u_C1IR (Npp16u *nAlpha1*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image premultiplication using constant alpha.

Parameters:

nAlpha1 Image alpha opacity (0 - max channel pixel value).

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.40.2.4 NppStatus nppiAlphaPremulC_16u_C1R (const Npp16u * *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image premultiplication using constant alpha.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

nAlpha1 Image alpha opacity (0 - max channel pixel value).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.40.2.5 NppStatus nppiAlphaPremulC_16u_C3IR (Npp16u *nAlpha1*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image premultiplication using constant alpha.

Parameters:

nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.6 NppStatus nppiAlphaPremulC_16u_C3R (const Npp16u * *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image premultiplication using constant alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.7 NppStatus nppiAlphaPremulC_16u_C4IR (Npp16u *nAlpha1*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image premultiplication using constant alpha.

Parameters:

nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.8 NppStatus nppiAlphaPremulC_16u_C4R (const Npp16u * *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image premultiplication using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.9 NppStatus nppiAlphaPremulC_8u_AC4IR (Npp8u *nAlpha1*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image premultiplication with alpha using constant alpha.

Parameters:

nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.10 NppStatus nppiAlphaPremulC_8u_AC4R (const Npp8u * *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image premultiplication with alpha using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.11 NppStatus nppiAlphaPremulC_8u_C1IR (Npp8u *nAlpha1*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image premultiplication using constant alpha.

Parameters:

nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.12 NppStatus nppiAlphaPremulC_8u_C1R (const Npp8u * *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image premultiplication using constant alpha.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.13 NppStatus nppiAlphaPremulC_8u_C3IR (Npp8u *nAlpha1*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image premultiplication using constant alpha.

Parameters:

nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.14 NppStatus nppiAlphaPremulC_8u_C3R (const Npp8u * *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image premultiplication using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.15 NppStatus nppiAlphaPremulC_8u_C4IR (Npp8u *nAlpha1*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image premultiplication using constant alpha.

Parameters:

nAlpha1 Image alpha opacity (0 - max channel pixel value).
pSrcDst [In-Place Image Pointer](#).
nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.40.2.16 NppStatus nppiAlphaPremulC_8u_C4R (const Npp8u * *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image premultiplication using constant alpha.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
nAlpha1 Image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41 AlphaComp

Composite two images using alpha opacity values contained in each image.

Functions

- **NppStatus** **nppiAlphaComp_8u_AC1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)
One 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).
- **NppStatus** **nppiAlphaComp_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)
Four 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).
- **NppStatus** **nppiAlphaComp_8s_AC1R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)
One 8-bit signed char channel image composition using image alpha values (0 - max channel pixel value).
- **NppStatus** **nppiAlphaComp_16u_AC1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)
One 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).
- **NppStatus** **nppiAlphaComp_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)
Four 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).
- **NppStatus** **nppiAlphaComp_16s_AC1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)
One 16-bit signed short channel image composition using image alpha values (0 - max channel pixel value).
- **NppStatus** **nppiAlphaComp_32u_AC1R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)
One 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).
- **NppStatus** **nppiAlphaComp_32u_AC4R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)
Four 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).
- **NppStatus** **nppiAlphaComp_32s_AC1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)
One 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).

- **NppStatus nppiAlphaComp_32s_AC4R** (const [Npp32s](#) *pSrc1, int nSrc1Step, const [Npp32s](#) *pSrc2, int nSrc2Step, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiAlphaOp](#) eAlphaOp)
Four 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).
- **NppStatus nppiAlphaComp_32f_AC1R** (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiAlphaOp](#) eAlphaOp)
One 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).
- **NppStatus nppiAlphaComp_32f_AC4R** (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiAlphaOp](#) eAlphaOp)
Four 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).

7.41.1 Detailed Description

Composite two images using alpha opacity values contained in each image.

7.41.2 Function Documentation

7.41.2.1 **NppStatus nppiAlphaComp_16s_AC1R** (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [Npp16s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiAlphaOp](#) eAlphaOp)

One 16-bit signed short channel image composition using image alpha values (0 - max channel pixel value).

Parameters:

[pSrc1](#) Source-Image Pointer.
[nSrc1Step](#) Source-Image Line Step.
[pSrc2](#) Source-Image Pointer.
[nSrc2Step](#) Source-Image Line Step.
[pDst](#) Destination-Image Pointer.
[nDstStep](#) Destination-Image Line Step.
[oSizeROI](#) Region-of-Interest (ROI).
[eAlphaOp](#) alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.2 **NppStatus nppiAlphaComp_16u_AC1R** (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [Npp16u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiAlphaOp](#) eAlphaOp)

One 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eAlphaOp alpha-blending operation..

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.41.2.3 `NppStatus nppiAlphaComp_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eAlphaOp alpha-blending operation..

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.41.2.4 `NppStatus nppiAlphaComp_32f_AC1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

One 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.5 **NppStatus nppiAlphaComp_32f_AC4R** (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Four 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.6 **NppStatus nppiAlphaComp_32s_AC1R** (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eAlphaOp alpha-blending operation..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.7 NppStatus nppiAlphaComp_32s_AC4R (const Npp32s * *pSrc1*, int *nSrc1Step*, const Npp32s * *pSrc2*, int *nSrc2Step*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Four 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eAlphaOp alpha-blending operation..

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.41.2.8 NppStatus nppiAlphaComp_32u_AC1R (const Npp32u * *pSrc1*, int *nSrc1Step*, const Npp32u * *pSrc2*, int *nSrc2Step*, Npp32u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eAlphaOp alpha-blending operation..

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.41.2.9 NppStatus nppiAlphaComp_32u_AC4R (const Npp32u * *pSrc1*, int *nSrc1Step*, const Npp32u * *pSrc2*, int *nSrc2Step*, Npp32u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Four 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp [alpha-blending operation..](#)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.10 NppStatus nppiAlphaComp_8s_AC1R (const Npp8s * *pSrc1*, int *nSrc1Step*, const Npp8s * *pSrc2*, int *nSrc2Step*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 8-bit signed char channel image composition using image alpha values (0 - max channel pixel value).

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eAlphaOp [alpha-blending operation..](#)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.41.2.11 NppStatus nppiAlphaComp_8u_AC1R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eAlphaOp alpha-blending operation..

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.41.2.12 `NppStatus nppiAlphaComp_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eAlphaOp alpha-blending operation..

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.42 AlphaPremul

Premultiplies image pixels by image alpha opacity values.

Functions

- **NppStatus nppiAlphaPremul_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 8-bit unsigned char channel image premultiplication with pixel alpha (0 - max channel pixel value).

- **NppStatus nppiAlphaPremul_8u_AC4IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 8-bit unsigned char channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

- **NppStatus nppiAlphaPremul_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel image premultiplication with pixel alpha (0 - max channel pixel value).

- **NppStatus nppiAlphaPremul_16u_AC4IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

7.42.1 Detailed Description

Premultiplies image pixels by image alpha opacity values.

7.42.2 Function Documentation

7.42.2.1 **NppStatus nppiAlphaPremul_16u_AC4IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.42.2.2 NppStatus nppiAlphaPremul_16u_AC4R (const Npp16u * *pSrc1*, int *nSrc1Step*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image premultiplication with pixel alpha (0 - max channel pixel value).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.42.2.3 NppStatus nppiAlphaPremul_8u_AC4IR (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.42.2.4 NppStatus nppiAlphaPremul_8u_AC4R (const Npp8u * *pSrc1*, int *nSrc1Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image premultiplication with pixel alpha (0 - max channel pixel value).

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.43 Color and Sampling Conversion

Routines manipulating an image's color model and sampling format.

Modules

- [Color Model Conversion](#)

Routines for converting between various image color models.

- [Color Sampling Format Conversion](#)

Routines for converting between various image color sampling formats.

- [Color Gamma Correction](#)

Routines for correcting image color gamma.

- [Complement Color Key](#)

Routines for performing complement color key replacement.

- [Color Processing](#)

Routines for performing image color manipulation.

7.43.1 Detailed Description

Routines manipulating an image's color model and sampling format.

7.44 Color Model Conversion

Routines for converting between various image color models.

RGBToYUV

RGB to YUV color conversion.

Here is how NPP converts gamma corrected RGB or BGR to YUV. For digital RGB values in the range [0..255], Y has the range [0..255], U varies in the range [-112..+112], and V in the range [-157..+157]. To fit in the range of [0..255], a constant value of 128 is added to computed U and V values, and V is then saturated.

```
Npp32f nY = 0.299F * R + 0.587F * G + 0.114F * B;
Npp32f nU = (0.492F * ((Npp32f)nB - nY)) + 128.0F;
Npp32f nV = (0.877F * ((Npp32f)nR - nY)) + 128.0F;
if (nV > 255.0F)
    nV = 255.0F;
```

- **NppStatus nppiRGBToYUV_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YUV color conversion.

- **NppStatus nppiRGBToYUV_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha.

- **NppStatus nppiRGBToYUV_8u_P3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV color conversion.

- **NppStatus nppiRGBToYUV_8u_C3P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV color conversion.

- **NppStatus nppiRGBToYUV_8u_AC4P4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[4], int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha.

BGRToYUV

BGR to YUV color conversion.

Here is how NPP converts gamma corrected RGB or BGR to YUV. For digital RGB values in the range [0..255], Y has the range [0..255], U varies in the range [-112..+112], and V in the range [-157..+157]. To fit in the range of [0..255], a constant value of 128 is added to computed U and V values, and V is then saturated.

```

Npp32f nY = 0.299F * R + 0.587F * G + 0.114F * B;
Npp32f nU = (0.492F * ((Npp32f)nB - nY)) + 128.0F;
Npp32f nV = (0.877F * ((Npp32f)nR - nY)) + 128.0F;
if (nV > 255.0F)
    nV = 255.0F;

```

- **NppStatus nppiBGRToYUV_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed YUV color conversion.

- **NppStatus nppiBGRToYUV_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha.

- **NppStatus nppiBGRToYUV_8u_P3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar YUV color conversion.

- **NppStatus nppiBGRToYUV_8u_C3P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YUV color conversion.

- **NppStatus nppiBGRToYUV_8u_AC4P4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[4], int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha.

YUVToRGB

YUV to RGB color conversion.

Here is how NPP converts YUV to gamma corrected RGB or BGR.

```

Npp32f nY = (Npp32f)Y;
Npp32f nU = (Npp32f)U - 128.0F;
Npp32f nV = (Npp32f)V - 128.0F;
Npp32f nR = nY + 1.140F * nV;
if (nR < 0.0F)
    nR = 0.0F;
if (nR > 255.0F)
    nR = 255.0F;
Npp32f nG = nY - 0.394F * nU - 0.581F * nV;
if (nG < 0.0F)
    nG = 0.0F;
if (nG > 255.0F)
    nG = 255.0F;
Npp32f nB = nY + 2.032F * nU;
if (nB < 0.0F)
    nB = 0.0F;
if (nB > 255.0F)
    nB = 255.0F;

```

- **NppStatus nppiYUVToRGB_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed RGB color conversion.

- **NppStatus nppiYUVToRGB_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed RGB color conversion with alpha, not affecting alpha.

- **NppStatus nppiYUVToRGB_8u_P3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar RGB color conversion.

- **NppStatus nppiYUVToRGB_8u_P3C3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed RGB color conversion.

YUVToBGR

YUV to BGR color conversion.

Here is how NPP converts YUV to gamma corrected RGB or BGR.

```
Npp32f nY = (Npp32f) Y;
Npp32f nU = (Npp32f) U - 128.0F;
Npp32f nV = (Npp32f) V - 128.0F;
Npp32f nR = nY + 1.140F * nV;
if (nR < 0.0F)
    nR = 0.0F;
if (nR > 255.0F)
    nR = 255.0F;
Npp32f nG = nY - 0.394F * nU - 0.581F * nV;
if (nG < 0.0F)
    nG = 0.0F;
if (nG > 255.0F)
    nG = 255.0F;
Npp32f nB = nY + 2.032F * nU;
if (nB < 0.0F)
    nB = 0.0F;
if (nB > 255.0F)
    nB = 255.0F;
```

- **NppStatus nppiYUVToBGR_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed BGR color conversion.

- **NppStatus nppiYUVToBGR_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed BGR color conversion with alpha, not affecting alpha.

- **NppStatus nppiYUVToBGR_8u_P3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar BGR color conversion.

- **NppStatus nppiYUVToBGR_8u_P3C3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed BGR color conversion.

RGBToYUV422

RGB to YUV422 color conversion.

- `NppStatus nppiRGBToYUV422_8u_C3C2R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YUV422 color conversion.
- `NppStatus nppiRGBToYUV422_8u_P3R` (const `Npp8u` *const pSrc[3], int nSrcStep, `Npp8u` *pDst[3], int rDstStep[3], `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV422 color conversion.
- `NppStatus nppiRGBToYUV422_8u_C3P3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst[3], int rDstStep[3], `NppiSize` oSizeROI)
3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV422 color conversion.

YUV422ToRGB

YUV422 to RGB color conversion.

- `NppStatus nppiYUV422ToRGB_8u_C2C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
2 channel 8-bit unsigned packed YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.
- `NppStatus nppiYUV422ToRGB_8u_P3R` (const `Npp8u` *const pSrc[3], int rSrcStep[3], `Npp8u` *pDst[3], int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned planar RGB color conversion.
- `NppStatus nppiYUV422ToRGB_8u_P3C3R` (const `Npp8u` *const pSrc[3], int rSrcStep[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.
- `NppStatus nppiYUV422ToRGB_8u_P3AC4R` (const `Npp8u` *const pSrc[3], int rSrcStep[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar YUV422 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.

RGBToYUV420

RGB to YUV420 color conversion.

- `NppStatus nppiRGBToYUV420_8u_P3R` (const `Npp8u` *const pSrc[3], int nSrcStep, `Npp8u` *pDst[3], int rDstStep[3], `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV420 color conversion.

- **NppStatus nppiRGBToYUV420_8u_C3P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV420 color conversion.

YUV420ToRGB

YUV420 to RGB color conversion.

- **NppStatus nppiYUV420ToRGB_8u_P3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned planar RGB color conversion.

- **NppStatus nppiYUV420ToRGB_8u_P3C3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed RGB color conversion.

- **NppStatus nppiYUV420ToRGB_8u_P3C4R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha (0xFF).

- **NppStatus nppiYUV420ToRGB_8u_P3AC4R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.

NV21ToRGB

NV21 to RGB color conversion.

- **NppStatus nppiNV21ToRGB_8u_P2C4R** (const **Npp8u** *const pSrc[2], int rSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

2 channel 8-bit unsigned planar NV21 to 4 channel 8-bit unsigned packed ARGB color conversion with constant alpha (0xFF).

BGRToYUV420

BGR to YUV420 color conversion.

- **NppStatus nppiBGRToYUV420_8u_AC4P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YUV420 color conversion.

YUV420ToBGR

YUV420 to BGR color conversion.

- **NppStatus** **nppiYUV420ToBGR_8u_P3C3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed BGR color conversion.
- **NppStatus** **nppiYUV420ToBGR_8u_P3C4R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha (0xFF).

NV21ToBGR

NV21 to BGR color conversion.

- **NppStatus** **nppiNV21ToBGR_8u_P2C4R** (const **Npp8u** *const pSrc[2], int rSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
2 channel 8-bit unsigned planar NV21 to 4 channel 8-bit unsigned packed BGRA color conversion with constant alpha (0xFF).

RGBToYCbCr

RGB to YCbCr color conversion.

Here is how NPP converts gamma corrected RGB or BGR to YCbCr. In the YCbCr model, Y is defined to have a nominal range [16..235], while Cb and Cr are defined to have a range [16..240], with the value of 128 as corresponding to zero.

```
Npp32f nY  = 0.257F * R + 0.504F * G + 0.098F * B + 16.0F;
Npp32f nCb = -0.148F * R - 0.291F * G + 0.439F * B + 128.0F;
Npp32f nCr = 0.439F * R - 0.368F * G - 0.071F * B + 128.0F;
```

- **NppStatus** **nppiRGBToYCbCr_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit packed YCbCr color conversion.
- **NppStatus** **nppiRGBToYCbCr_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed RGB with alpha to 4 channel unsigned 8-bit packed YCbCr with alpha color conversion, not affecting alpha.
- **NppStatus** **nppiRGBToYCbCr_8u_P3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)
3 channel planar 8-bit unsigned RGB to 3 channel planar 8-bit YCbCr color conversion.
- **NppStatus** **nppiRGBToYCbCr_8u_C3P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit planar YCbCr color conversion.

- **NppStatus nppiRGBToYCbCr_8u_AC4P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion.

YCbCrToRGB

YCbCr to RGB color conversion.

Here is how NPP converts YCbCr to gamma corrected RGB or BGR. The output RGB values are saturated to the range [0..255].

```
Npp32f nY = 1.164F * ((Npp32f)Y - 16.0F);
Npp32f nR = ((Npp32f)Cr - 128.0F);
Npp32f nB = ((Npp32f)Cb - 128.0F);
Npp32f nG = nY - 0.813F * nR - 0.392F * nB;
if (nG > 255.0F)
    nG = 255.0F;
nR = nY + 1.596F * nR;
if (nR > 255.0F)
    nR = 255.0F;
nB = nY + 2.017F * nB;
if (nB > 255.0F)
    nB = 255.0F;
```

- **NppStatus nppiYCbCrToRGB_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.

- **NppStatus nppiYCbCrToRGB_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed YCbCr with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion, not affecting alpha.

- **NppStatus nppiYCbCrToRGB_8u_P3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned planar RGB color conversion.

- **NppStatus nppiYCbCrToRGB_8u_P3C3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.

- **NppStatus nppiYCbCrToRGB_8u_P3C4R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.

YCbCrToBGR

YCbCr to BGR color conversion.

- `NppStatus nppiYCbCrToBGR_8u_P3C3R` (const `Npp8u` *const pSrc[3], int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR color conversion.
- `NppStatus nppiYCbCrToBGR_8u_P3C4R` (const `Npp8u` *const pSrc[3], int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp8u` nAval)
3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

YCbCrToBGR_709CSC

YCbCr to BGR_709CSC color conversion.

- `NppStatus nppiYCbCrToBGR_709CSC_8u_P3C3R` (const `Npp8u` *const pSrc[3], int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR_709CSC color conversion.
- `NppStatus nppiYCbCrToBGR_709CSC_8u_P3C4R` (const `Npp8u` *const pSrc[3], int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp8u` nAval)
3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR_709CSC color conversion with constant alpha.

RGBToYCbCr422

RGB to YCbCr422 color conversion.

- `NppStatus nppiRGBToYCbCr422_8u_C3C2R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion.
- `NppStatus nppiRGBToYCbCr422_8u_C3P3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst[3], int nDstStep[3], `NppiSize` oSizeROI)
3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr422 color conversion.
- `NppStatus nppiRGBToYCbCr422_8u_P3C2R` (const `Npp8u` *const pSrc[3], int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion.

YCbCr422ToRGB

YCbCr422 to RGB color conversion.

- `NppStatus nppiYCbCr422ToRGB_8u_C2C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion.

- **NppStatus nppiYCbCr422ToRGB_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)
2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar RGB color conversion.
- **NppStatus nppiYCbCr422ToRGB_8u_P3C3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion.

RGBToYCrCb422

RGB to YCrCb422 color conversion.

- **NppStatus nppiRGBToYCrCb422_8u_C3C2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion.
- **NppStatus nppiRGBToYCrCb422_8u_P3C2R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion.

YCrCb422ToRGB

YCrCb422 to RGB color conversion.

- **NppStatus nppiYCrCb422ToRGB_8u_C2C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed RGB color conversion.
- **NppStatus nppiYCrCb422ToRGB_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)
2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar RGB color conversion.

BGRToYCbCr422

BGR to YCbCr422 color conversion.

- **NppStatus nppiBGRToYCbCr422_8u_C3C2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed YCrCb422 color conversion.
- **NppStatus nppiBGRToYCbCr422_8u_AC4C2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed YCrCb422 color conversion.
- **NppStatus nppiBGRToYCbCr422_8u_C3P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr422 color conversion.

- `NppStatus nppiBGRToYCbCr422_8u_AC4P3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst[3], int rDstStep[3], `NppiSize` oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr422 color conversion.

YCbCr422ToBGR

YCbCr422 to BGR color conversion.

- `NppStatus nppiYCbCr422ToBGR_8u_C2C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed BGR color conversion.

- `NppStatus nppiYCbCr422ToBGR_8u_C2C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp8u` nAval)

2 channel 8-bit unsigned packed YCrCb422 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

- `NppStatus nppiYCbCr422ToBGR_8u_P3C3R` (const `Npp8u` *const pSrc[3], int rSrcStep[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed BGR color conversion.

RGBToCbYCr422

RGB to CbYCr422 color conversion.

- `NppStatus nppiRGBToCbYCr422_8u_C3C2R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed CbYCr422 color conversion.

- `NppStatus nppiRGBToCbYCr422Gamma_8u_C3C2R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

3 channel 8-bit unsigned packed RGB first gets forward gamma corrected then converted to 2 channel 8-bit unsigned packed CbYCr422 color conversion.

CbYCr422ToRGB

CbYCr422 to RGB color conversion.

- `NppStatus nppiCbYCr422ToRGB_8u_C2C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

2 channel 8-bit unsigned packed CbYCrC22 to 3 channel 8-bit unsigned packed RGB color conversion.

BGRToCbYCr422

BGR to CbYCr422 color conversion.

- **NppStatus nppiBGRToCbYCr422_8u_AC4C2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422 color conversion.

BGRToCbYCr422_709HDTV

BGR to CbYCr422_709HDTV color conversion.

- **NppStatus nppiBGRToCbYCr422_709HDTV_8u_C3C2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed CbYCr422_709HDTV color conversion.
- **NppStatus nppiBGRToCbYCr422_709HDTV_8u_AC4C2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422_709HDTV color conversion.

CbYCr422ToBGR

CbYCr422 to BGR color conversion.

- **NppStatus nppiCbYCr422ToBGR_8u_C2C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)
2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR color conversion with alpha.

CbYCr422ToBGR_709HDTV

CbYCr422 to BGR_709HDTV color conversion.

- **NppStatus nppiCbYCr422ToBGR_709HDTV_8u_C2C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned packed BGR_709HDTV color conversion.
- **NppStatus nppiCbYCr422ToBGR_709HDTV_8u_C2C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)
2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR_709HDTV color conversion with constant alpha.

RGBToYCbCr420

RGB to YCbCr420 color conversion.

- `NppStatus nppiRGBToYCbCr420_8u_C3P3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst[3], int rDstStep[3], `NppiSize` oSizeROI)
3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

YCbCr420ToRGB

YCbCr420 to RGB color conversion.

- `NppStatus nppiYCbCr420ToRGB_8u_P3C3R` (const `Npp8u` *const pSrc[3], int rSrcStep[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed RGB color conversion.

RGBToYCrCb420

RGB to YCrCb420 color conversion.

- `NppStatus nppiRGBToYCrCb420_8u_AC4P3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst[3], int rDstStep[3], `NppiSize` oSizeROI)
4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

YCrCb420ToRGB

YCrCb420 to RGB color conversion.

- `NppStatus nppiYCrCb420ToRGB_8u_P3C4R` (const `Npp8u` *const pSrc[3], int rSrcStep[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp8u` nAval)
3 channel 8-bit unsigned planar YCrCb420 to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.

BGRToYCbCr420

BGR to YCbCr420 color conversion.

- `NppStatus nppiBGRToYCbCr420_8u_C3P3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst[3], int rDstStep[3], `NppiSize` oSizeROI)
3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420 color conversion.
- `NppStatus nppiBGRToYCbCr420_8u_AC4P3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst[3], int rDstStep[3], `NppiSize` oSizeROI)
4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

BGRToYCbCr420_709CSC

BGR to YCbCr420_709CSC color conversion.

- **NppStatus** **nppiBGRToYCbCr420_709CSC_8u_C3P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420_709CSC color conversion.
- **NppStatus** **nppiBGRToYCbCr420_709CSC_8u_AC4P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420_709CSC color conversion.

BGRToYCbCr420_709HDTV

BGR to YCbCr420_709HDTV color conversion.

- **NppStatus** **nppiBGRToYCbCr420_709HDTV_8u_AC4P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420_709HDTV color conversion.

BGRToYCrCb420_709CSC

BGR to YCrCb420_709CSC color conversion.

- **NppStatus** **nppiBGRToYCrCb420_709CSC_8u_C3P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420_709CSC color conversion.
- **NppStatus** **nppiBGRToYCrCb420_709CSC_8u_AC4P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420_709CSC color conversion.

YCbCr420ToBGR

YCbCr420 to BGR color conversion.

- **NppStatus** **nppiYCbCr420ToBGR_8u_P3C3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR color conversion.
- **NppStatus** **nppiYCbCr420ToBGR_8u_P3C4R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)

3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

YCbCr420ToBGR_709CSC

YCbCr420_709CSC to BGR color conversion.

- `NppStatus nppiYCbCr420ToBGR_709CSC_8u_P3C3R` (const `Npp8u` *const `pSrc[3]`, int `rSrcStep[3]`, `Npp8u` *`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR_709CSC color conversion.

YCbCr420ToBGR_709HDTV

YCbCr420_709HDTV to BGR color conversion.

- `NppStatus nppiYCbCr420ToBGR_709HDTV_8u_P3C4R` (const `Npp8u` *const `pSrc[3]`, int `rSrcStep[3]`, `Npp8u` *`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`, `Npp8u` `nAval`)

3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR_709HDTV color conversion with constant alpha.

BGRToYCrCb420

BGR to YCrCb420 color conversion.

- `NppStatus nppiBGRToYCrCb420_8u_C3P3R` (const `Npp8u` *`pSrc`, int `nSrcStep`, `Npp8u` *`pDst[3]`, int `rDstStep[3]`, `NppiSize` `oSizeROI`)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

- `NppStatus nppiBGRToYCrCb420_8u_AC4P3R` (const `Npp8u` *`pSrc`, int `nSrcStep`, `Npp8u` *`pDst[3]`, int `rDstStep[3]`, `NppiSize` `oSizeROI`)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

BGRToYCbCr411

BGR to YCbCr411 color conversion.

- `NppStatus nppiBGRToYCbCr411_8u_C3P3R` (const `Npp8u` *`pSrc`, int `nSrcStep`, `Npp8u` *`pDst[3]`, int `rDstStep[3]`, `NppiSize` `oSizeROI`)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

- `NppStatus nppiBGRToYCbCr411_8u_AC4P3R` (const `Npp8u` *`pSrc`, int `nSrcStep`, `Npp8u` *`pDst[3]`, int `rDstStep[3]`, `NppiSize` `oSizeROI`)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

BGRToYCbCr

BGR to YCbCr color conversion.

- `NppStatus nppiBGRToYCbCr_8u_C3P3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst[3], int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr color conversion.
- `NppStatus nppiBGRToYCbCr_8u_AC4P3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst[3], int nDstStep, `NppiSize` oSizeROI)
4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion.
- `NppStatus nppiBGRToYCbCr_8u_AC4P4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst[4], int nDstStep, `NppiSize` oSizeROI)
4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar YCbCr color conversion.

YCbCr411ToBGR

YCbCr411 to BGR color conversion.

- `NppStatus nppiYCbCr411ToBGR_8u_P3C3R` (const `Npp8u` *const pSrc[3], int rSrcStep[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned packed BGR color conversion.
- `NppStatus nppiYCbCr411ToBGR_8u_P3C4R` (const `Npp8u` *const pSrc[3], int rSrcStep[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp8u` nAval)
3 channel 8-bit unsigned planar YCbCr411 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

RGBToXYZ

RGB to XYZ color conversion.

Here is how NPP converts gamma corrected RGB or BGR to XYZ.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F * nNormalizedB;
if (nX > 1.0F)
    nX = 1.0F;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F * nNormalizedB;
if (nY > 1.0F)
    nY = 1.0F;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F * nNormalizedB;
if (nZ > 1.0F)
    nZ = 1.0F;
X = (Npp8u) (nX * 255.0F);
Y = (Npp8u) (nY * 255.0F);
Z = (Npp8u) (nZ * 255.0F);
```

- **NppStatus nppiRGBToXYZ_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed XYZ color conversion.

- **NppStatus nppiRGBToXYZ_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed XYZ with alpha color conversion.

XYZToRGB

XYZ to RGB color conversion.

Here is how NPP converts XYZ to gamma corrected RGB or BGR. The code assumes that X,Y, and Z values are in the range [0..1].

```
Npp32f nNormalizedX = (Npp32f)X * 0.003921569F; // / 255.0F
Npp32f nNormalizedY = (Npp32f)Y * 0.003921569F;
Npp32f nNormalizedZ = (Npp32f)Z * 0.003921569F;
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F * nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F * nNormalizedZ;
if (nG > 1.0F)
    nG = 1.0F;
Npp32f nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F * nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u) (nR * 255.0F);
G = (Npp8u) (nG * 255.0F);
B = (Npp8u) (nB * 255.0F);
```

- **NppStatus nppiXYZToRGB_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed XYZ to 3 channel 8-bit unsigned packed RGB color conversion.

- **NppStatus nppiXYZToRGB_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed XYZ with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

RGBToLUV

RGB to LUV color conversion.

Here is how NPP converts gamma corrected RGB or BGR to CIE LUV using the CIE XYZ D65 white point with a Y luminance of 1.0. The computed values of the L component are in the range [0..100], U component in the range [-134..220], and V component in the range [-140..122]. The code uses cbrtf() the 32 bit floating point cube root math function.

```
// use CIE D65 chromaticity coordinates
#define nCIE_XYZ_D65_xn 0.312713F
#define nCIE_XYZ_D65_yn 0.329016F
```

```

#define nn_DIVISOR (-2.0F * nCIE_XYZ_D65_xn + 12.0F * nCIE_XYZ_D65_yn + 3.0F)
#define nun (4.0F * nCIE_XYZ_D65_xn / nn_DIVISOR)
#define nvN (9.0F * nCIE_XYZ_D65_yn / nn_DIVISOR)

// First convert to XYZ
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F * nNormalizedB;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F * nNormalizedB;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F * nNormalizedB;
// Now calculate LUV from the XYZ value
Npp32f nTemp = nX + 15.0F * nY + 3.0F * nZ;
Npp32f nu = 4.0F * nX / nTemp;
Npp32f nv = 9.0F * nY / nTemp;
Npp32f nL = 116.0F * cbtrf(nY) - 16.0F;
if (nL < 0.0F)
    nL = 0.0F;
if (nL > 100.0F)
    nL = 100.0F;
nTemp = 13.0F * nL;
Npp32f nU = nTemp * (nu - nun);
if (nU < -134.0F)
    nU = -134.0F;
if (nU > 220.0F)
    nU = 220.0F;
Npp32f nV = nTemp * (nv - nvN);
if (nV < -140.0F)
    nV = -140.0F;
if (nV > 122.0F)
    nV = 122.0F;
L = (Npp8u)(nL * 255.0F * 0.01F); // / 100.0F
U = (Npp8u)((nU + 134.0F) * 255.0F * 0.0028249F); // / 354.0F
V = (Npp8u)((nV + 140.0F) * 255.0F * 0.0038168F); // / 262.0F

```

- **NppStatus nppiRGBToLUV_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed LUV color conversion.

- **NppStatus nppiRGBToLUV_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed LUV with alpha color conversion.

LUVToRGB

LUV to RGB color conversion.

Here is how NPP converts CIE LUV to gamma corrected RGB or BGR using the CIE XYZ D65 white point with a Y luminance of 1.0. The code uses powf() the 32 bit floating point power math function.

```

// use CIE D65 chromaticity coordinates
#define nCIE_XYZ_D65_xn 0.312713F
#define nCIE_XYZ_D65_yn 0.329016F
#define nn_DIVISOR (-2.0F * nCIE_XYZ_D65_xn + 12.0F * nCIE_XYZ_D65_yn + 3.0F)
#define nun (4.0F * nCIE_XYZ_D65_xn / nn_DIVISOR)
#define nvN (9.0F * nCIE_XYZ_D65_yn / nn_DIVISOR)

// First convert normalized LUV back to original CIE LUV range
Npp32f nL = (Npp32f)L * 100.0F * 0.003921569F; // / 255.0F
Npp32f nU = ((Npp32f)U * 354.0F * 0.003921569F) - 134.0F;

```

```

Npp32f nV = ((Npp32f)V * 262.0F * 0.003921569F) - 140.0F;
// Now convert LUV to CIE XYZ
Npp32f nTemp = 13.0F * nL;
Npp32f nu = nU / nTemp + nun;
Npp32f nv = nV / nTemp + nvu;
Npp32f nNormalizedY;
if (nL > 7.9996248F)
{
    nNormalizedY = (nL + 16.0F) * 0.008621F; // / 116.0F
    nNormalizedY = powf(nNormalizedY, 3.0F);
}
else
{
    nNormalizedY = nL * 0.001107F; // / 903.3F
}
Npp32f nNormalizedX = (-9.0F * nNormalizedY * nu) / ((nu - 4.0F) * nv - nu * nv);
Npp32f nNormalizedZ = (9.0F * nNormalizedY - 15.0F * nv * nNormalizedY - nv * nNormalizedX) / (3.0F * nv + nNormalizedX + 15.0F * nv);
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F * nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
if (nR < 0.0F)
    nR = 0.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F * nNormalizedZ;
if (nG > 1.0F)
    nG = 1.0F;
if (nG < 0.0F)
    nG = 0.0F;
Npp32f nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F * nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;
if (nB < 0.0F)
    nB = 0.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);

```

- **NppStatus nppiLUVToRGB_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed LUV to 3 channel 8-bit unsigned packed RGB color conversion.

- **NppStatus nppiLUVToRGB_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed LUV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

BGRToLab

BGR to Lab color conversion.

This is how NPP converts gamma corrected BGR or RGB to Lab using the CIE Lab D65 white point with a Y luminance of 1.0. The computed values of the L component are in the range [0..100], a and b component values are in the range [-128..127]. The code uses `cbrtf()` the 32 bit floating point cube root math function.

```

// use CIE Lab chromaticity coordinates
#define nCIE_LAB_D65_xn 0.950455F
#define nCIE_LAB_D65_yn 1.0F
#define nCIE_LAB_D65_zn 1.088753F
// First convert to XYZ
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;

```

```

Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F * nNormalizedB;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F * nNormalizedB;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F * nNormalizedB;
Npp32f nL = cbrtf(nY);
Npp32f nA;
Npp32f nB;
Npp32f nfX = nX * 1.052128F; // / nCIE_LAB_D65_xn;
Npp32f nfY = nY;
Npp32f nfZ = nZ * 0.918482F; // / nCIE_LAB_D65_zn;
nfY = nL - 16.0F;
nL = 116.0F * nL - 16.0F;
nA = cbrtf(nfX) - 16.0F;
nA = 500.0F * (nA - nfY);
nB = cbrtf(nfZ) - 16.0F;
nB = 200.0F * (nfY - nB);
// Now scale Lab range
nL = nL * 255.0F * 0.01F; // / 100.0F
nA = nA + 128.0F;
nB = nB + 128.0F;
L = (Npp8u)nL;
a = (Npp8u)nA;
b = (Npp8u)nB;

```

- **NppStatus nppiBGRToLab_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed Lab color conversion.

LabToBGR

Lab to BGR color conversion.

This is how NPP converts Lab to gamma corrected BGR or RGB using the CIE Lab D65 white point with a Y luminance of 1.0. The code uses `powf()` the 32 bit floating point power math function.

```

// use CIE Lab chromaticity coordinates
#define nCIE_LAB_D65_xn 0.950455F
#define nCIE_LAB_D65_yn 1.0F
#define nCIE_LAB_D65_zn 1.088753F
// First convert Lab back to original range then to CIE XYZ
Npp32f nL = (Npp32f)L * 100.0F * 0.003921569F; // / 255.0F
Npp32f nA = (Npp32f)a - 128.0F;
Npp32f nB = (Npp32f)b - 128.0F;
Npp32f nP = (nL + 16.0F) * 0.008621F; // / 116.0F
Npp32f nNormalizedY = nP * nP * nP; // powf(nP, 3.0F);
Npp32f nNormalizedX = nCIE_LAB_D65_xn * powf((nP + nA * 0.002F), 3.0F); // / 500.0F
Npp32f nNormalizedZ = nCIE_LAB_D65_zn * powf((nP - nB * 0.005F), 3.0F); // / 200.0F
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F * nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F * nNormalizedZ;
if (nG > 1.0F)
    nG = 1.0F;
nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F * nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);

```

- **NppStatus nppiLabToBGR_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed Lab to 3 channel 8-bit unsigned packed BGR color conversion.

RGBToYCC

RGB to PhotoYCC color conversion.

This is how NPP converts gamma corrected BGR or RGB to PhotoYCC. The computed Y, C1, C2 values are then quantized and converted to fit in the range [0..1] before expanding to 8 bits.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nY = 0.299F * nNormalizedR + 0.587F * nNormalizedG + 0.114F * nNormalizedB;
Npp32f nC1 = nNormalizedB - nY;
nC1 = 111.4F * 0.003921569F * nC1 + 156.0F * 0.003921569F;
Npp32f nC2 = nNormalizedR - nY;
nC2 = 135.64F * 0.003921569F * nC2 + 137.0F * 0.003921569F;
nY = 1.0F * 0.713267F * nY; // / 1.402F
Y = (Npp8u)(nY * 255.0F);
C1 = (Npp8u)(nC1 * 255.0F);
C2 = (Npp8u)(nC2 * 255.0F);
```

- [NppStatus npapiRGBToYCC_8u_C3R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YCC color conversion.

- [NppStatus npapiRGBToYCC_8u_AC4R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YCC with alpha color conversion.

YCCToRGB

PhotoYCC to RGB color conversion.

This is how NPP converts PhotoYCC to gamma corrected RGB or BGR.

```
Npp32f nNormalizedY = ((Npp32f)Y * 0.003921569F) * 1.3584F; // / 255.0F
Npp32f nNormalizedC1 = (((Npp32f)C1 * 0.003921569F) - 156.0F * 0.003921569F) * 2.2179F;
Npp32f nNormalizedC2 = (((Npp32f)C2 * 0.003921569F) - 137.0F * 0.003921569F) * 1.8215F;
Npp32f nR = nNormalizedY + nNormalizedC2;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = nNormalizedY - 0.194F * nNormalizedC1 - 0.509F * nNormalizedC2;
if (nG > 1.0F)
    nG = 1.0F;
Npp32f nB = nNormalizedY + nNormalizedC1;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);
```

- [NppStatus npapiYCCToRGB_8u_C3R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)

3 channel 8-bit unsigned packed YCC to 3 channel 8-bit unsigned packed RGB color conversion.

- **NppStatus nppiYCCToRGB_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed YCC with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

RGBToHLS

RGB to HLS color conversion.

This is how NPP converts gamma corrected RGB or BGR to HLS. This code uses the fmaxf() and fminf() 32 bit floating point math functions.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nS;
Npp32f nH;
// Lightness
Npp32f nMax = fmaxf(nNormalizedR, nNormalizedG);
nMax = fmaxf(nMax, nNormalizedB);
Npp32f nMin = fminf(nNormalizedR, nNormalizedG);
nMin = fminf(nMin, nNormalizedB);
Npp32f nL = (nMax + nMin) * 0.5F;
Npp32f nDivisor = nMax - nMin;
// Saturation
if (nDivisor == 0.0F) // achromatics case
{
    nS = 0.0F;
    nH = 0.0F;
}
else // chromatics case
{
    if (nL > 0.5F)
        nS = nDivisor / (1.0F - (nMax + nMin - 1.0F));
    else
        nS = nDivisor / (nMax + nMin);
}
// Hue
Npp32f nCr = (nMax - nNormalizedR) / nDivisor;
Npp32f nCg = (nMax - nNormalizedG) / nDivisor;
Npp32f nCb = (nMax - nNormalizedB) / nDivisor;
if (nNormalizedR == nMax)
    nH = nCb - nCg;
else if (nNormalizedG == nMax)
    nH = 2.0F + nCr - nCb;
else if (nNormalizedB == nMax)
    nH = 4.0F + nCg - nCr;
nH = nH * 0.1666667F; // / 6.0F
if (nH < 0.0F)
    nH = nH + 1.0F;
H = (Npp8u) (nH * 255.0F);
L = (Npp8u) (nL * 255.0F);
S = (Npp8u) (nS * 255.0F);
```

- **NppStatus nppiRGBToHLS_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HLS color conversion.

- `NppStatus nppiRGBToHLS_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

HLSToRGB

HLS to RGB color conversion.

This is how NPP converts HLS to gamma corrected RGB or BGR.

```
Npp32f nNormalizedH = (Npp32f)H * 0.003921569F; // / 255.0F
Npp32f nNormalizedL = (Npp32f)L * 0.003921569F;
Npp32f nNormalizedS = (Npp32f)S * 0.003921569F;
Npp32f nM1;
Npp32f nM2;
Npp32f nR;
Npp32f nG;
Npp32f nB;
Npp32f nh = 0.0F;
if (nNormalizedL <= 0.5F)
    nM2 = nNormalizedL * (1.0F + nNormalizedS);
else
    nM2 = nNormalizedL + nNormalizedS - nNormalizedL * nNormalizedS;
nM1 = 2.0F * nNormalizedL - nM2;
if (nNormalizedS == 0.0F)
    nR = nG = nB = nNormalizedL;
else
{
    nh = nNormalizedH + 0.3333F;
    if (nh > 1.0F)
        nh -= 1.0F;
}
Npp32f nMDiff = nM2 - nM1;
if (0.6667F < nh)
    nR = nM1;
else
{
    if (nh < 0.1667F)
        nR = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
    else if (nh < 0.5F)
        nR = nM2;
    else
        nR = nM1 + nMDiff * (0.6667F - nh) * 6.0F; // / 0.1667F
}
if (nR > 1.0F)
    nR = 1.0F;
nh = nNormalizedH;
if (0.6667F < nh)
    nG = nM1;
else
{
    if (nh < 0.1667F)
        nG = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
    else if (nh < 0.5F)
        nG = nM2;
    else
        nG = nM1 + nMDiff * (0.6667F - nh) * 6.0F; // / 0.1667F
}
if (nG > 1.0F)
    nG = 1.0F;
nh = nNormalizedH - 0.3333F;
if (nh < 0.0F)
```



```

    nh += 1.0F;
    if (0.6667F < nh)
        nB = nM1;
    else
    {
        if (nh < 0.1667F)
            nB = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
        else if (nh < 0.5F)
            nB = nM2;
        else
            nB = nM1 + nMDiff * (0.6667F - nh) * 6.0F; // / 0.1667F
    }
    if (nB > 1.0F)
        nB = 1.0F;
    R = (Npp8u) (nR * 255.0F);
    G = (Npp8u) (nG * 255.0F);
    B = (Npp8u) (nB * 255.0F);

```

- **NppStatus nppiHLSToRGB_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned packed RGB color conversion.
- **NppStatus nppiHLSToRGB_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

BGRToHLS

BGR to HLS color conversion.

- **NppStatus nppiBGRToHLS_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.
- **NppStatus nppiBGRToHLS_8u_C3P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar HLS color conversion.
- **NppStatus nppiBGRToHLS_8u_AC4P4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[4], int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.
- **NppStatus nppiBGRToHLS_8u_P3C3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned packed HLS color conversion.
- **NppStatus nppiBGRToHLS_8u_AP4C4R** (const **Npp8u** *const pSrc[4], int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

- **NppStatus nppiBGRToHLS_8u_P3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar HLS color conversion.
- **NppStatus nppiBGRToHLS_8u_AP4R** (const **Npp8u** *const pSrc[4], int nSrcStep, **Npp8u** *pDst[4], int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.

HLSToBGR

HLS to BGR color conversion.

- **NppStatus nppiHLSToBGR_8u_C3P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned planar BGR color conversion.
- **NppStatus nppiHLSToBGR_8u_AC4P4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[4], int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.
- **NppStatus nppiHLSToBGR_8u_P3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned planar BGR color conversion.
- **NppStatus nppiHLSToBGR_8u_AP4R** (const **Npp8u** *const pSrc[4], int nSrcStep, **Npp8u** *pDst[4], int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.
- **NppStatus nppiHLSToBGR_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.
- **NppStatus nppiHLSToBGR_8u_P3C3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned packed BGR color conversion.
- **NppStatus nppiHLSToBGR_8u_AP4C4R** (const **Npp8u** *const pSrc[4], int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.

RGBToHSV

RGB to HSV color conversion.

This is how NPP converts gamma corrected RGB or BGR to HSV. This code uses the `fmaxf()` and `fminf()` 32 bit floating point math functions.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nS;
Npp32f nH;
// Value
Npp32f nV = fmaxf(nNormalizedR, nNormalizedG);
nV = fmaxf(nV, nNormalizedB);
// Saturation
Npp32f nTemp = fminf(nNormalizedR, nNormalizedG);
nTemp = fminf(nTemp, nNormalizedB);
Npp32f nDivisor = nV - nTemp;
if (nV == 0.0F) // achromatics case
{
    nS = 0.0F;
    nH = 0.0F;
}
else // chromatics case
    nS = nDivisor / nV;
// Hue:
Npp32f nCr = (nV - nNormalizedR) / nDivisor;
Npp32f nCg = (nV - nNormalizedG) / nDivisor;
Npp32f nCb = (nV - nNormalizedB) / nDivisor;
if (nNormalizedR == nV)
    nH = nCb - nCg;
else if (nNormalizedG == nV)
    nH = 2.0F + nCr - nCb;
else if (nNormalizedB == nV)
    nH = 4.0F + nCg - nCr;
nH = nH * 0.1666667F; // / 6.0F
if (nH < 0.0F)
    nH = nH + 1.0F;
H = (Npp8u) (nH * 255.0F);
S = (Npp8u) (nS * 255.0F);
V = (Npp8u) (nV * 255.0F);
```

- **NppStatus nppiRGBToHSV_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HSV color conversion.
- **NppStatus nppiRGBToHSV_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HSV with alpha color conversion.

HSVTToRGB

HSV to RGB color conversion.

This is how NPP converts HSV to gamma corrected RGB or BGR. This code uses the `floorf()` 32 bit floating point math function.

```
Npp32f nNormalizedH = (Npp32f)H * 0.003921569F; // / 255.0F
Npp32f nNormalizedS = (Npp32f)S * 0.003921569F;
Npp32f nNormalizedV = (Npp32f)V * 0.003921569F;
Npp32f nR;
Npp32f nG;
```

```

Npp32f nB;
if (nNormalizedS == 0.0F)
{
    nR = nG = nB = nNormalizedV;
}
else
{
    if (nNormalizedH == 1.0F)
        nNormalizedH = 0.0F;
    else
        nNormalizedH = nNormalizedH * 6.0F; // / 0.1667F
}
Npp32f nI = floorf(nNormalizedH);
Npp32f nF = nNormalizedH - nI;
Npp32f nM = nNormalizedV * (1.0F - nNormalizedS);
Npp32f nN = nNormalizedV * (1.0F - nNormalizedS * nF);
Npp32f nK = nNormalizedV * (1.0F - nNormalizedS * (1.0F - nF));
if (nI == 0.0F)
{ nR = nNormalizedV; nG = nK; nB = nM; }
else if (nI == 1.0F)
{ nR = nN; nG = nNormalizedV; nB = nM; }
else if (nI == 2.0F)
{ nR = nM; nG = nNormalizedV; nB = nK; }
else if (nI == 3.0F)
{ nR = nM; nG = nN; nB = nNormalizedV; }
else if (nI == 4.0F)
{ nR = nK; nG = nM; nB = nNormalizedV; }
else if (nI == 5.0F)
{ nR = nNormalizedV; nG = nM; nB = nN; }
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);

```

- **NppStatus nppiHSVToRGB_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed HSV to 3 channel 8-bit unsigned packed RGB color conversion.
- **NppStatus nppiHSVToRGB_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed HSV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

RGBToGray

RGB to CCIR601 Gray conversion.

Here is how NPP converts gamma corrected RGB to CCIR601 Gray.

```
nGray = 0.299F * R + 0.587F * G + 0.114F * B;
```

- **NppStatus nppiRGBToGray_8u_C3C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed RGB to 1 channel 8-bit unsigned packed Gray conversion.
- **NppStatus nppiRGBToGray_8u_AC4C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed RGB with alpha to 1 channel 8-bit unsigned packed Gray conversion.

- `NppStatus nppiRGBToGray_16u_C3C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 16-bit unsigned packed RGB to 1 channel 16-bit unsigned packed Gray conversion.
- `NppStatus nppiRGBToGray_16u_AC4C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 16-bit unsigned packed RGB with alpha to 1 channel 16-bit unsigned packed Gray conversion.
- `NppStatus nppiRGBToGray_16s_C3C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 16-bit signed packed RGB to 1 channel 16-bit signed packed Gray conversion.
- `NppStatus nppiRGBToGray_16s_AC4C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 16-bit signed packed RGB with alpha to 1 channel 16-bit signed packed Gray conversion.
- `NppStatus nppiRGBToGray_32f_C3C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 32-bit floating point packed RGB to 1 channel 32-bit floating point packed Gray conversion.
- `NppStatus nppiRGBToGray_32f_AC4C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 32-bit floating point packed RGB with alpha to 1 channel 32-bit floating point packed Gray conversion.

ColorToGray

RGB Color to Gray conversion using user supplied conversion coefficients.

Here is how NPP converts gamma corrected RGB Color to Gray using user supplied conversion coefficients.

```
nGray = aCoeffs[0] * R + aCoeffs[1] * G + aCoeffs[2] * B;
```

For the C4C1R versions of the functions the calculations are as follows. For BGRA or other formats with alpha just rearrange the coefficients accordingly.

```
nGray = aCoeffs[0] * R + aCoeffs[1] * G + aCoeffs[2] * B + aCoeffs[3] * A;
```

- `NppStatus nppiColorToGray_8u_C3C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])
3 channel 8-bit unsigned packed RGB to 1 channel 8-bit unsigned packed Gray conversion.
- `NppStatus nppiColorToGray_8u_AC4C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])
4 channel 8-bit unsigned packed RGB with alpha to 1 channel 8-bit unsigned packed Gray conversion.
- `NppStatus nppiColorToGray_8u_C4C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[4])
4 channel 8-bit unsigned packed RGBA to 1 channel 8-bit unsigned packed Gray conversion.

- `NppStatus nppiColorToGray_16u_C3C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])
3 channel 16-bit unsigned packed RGB to 1 channel 16-bit unsigned packed Gray conversion.
- `NppStatus nppiColorToGray_16u_AC4C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])
4 channel 16-bit unsigned packed RGB with alpha to 1 channel 16-bit unsigned packed Gray conversion.
- `NppStatus nppiColorToGray_16u_C4C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[4])
4 channel 16-bit unsigned packed RGBA to 1 channel 16-bit unsigned packed Gray conversion.
- `NppStatus nppiColorToGray_16s_C3C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])
3 channel 16-bit signed packed RGB to 1 channel 16-bit signed packed Gray conversion.
- `NppStatus nppiColorToGray_16s_AC4C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])
4 channel 16-bit signed packed RGB with alpha to 1 channel 16-bit signed packed Gray conversion.
- `NppStatus nppiColorToGray_16s_C4C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[4])
4 channel 16-bit signed packed RGBA to 1 channel 16-bit signed packed Gray conversion.
- `NppStatus nppiColorToGray_32f_C3C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])
3 channel 32-bit floating point packed RGB to 1 channel 32-bit floating point packed Gray conversion.
- `NppStatus nppiColorToGray_32f_AC4C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])
4 channel 32-bit floating point packed RGB with alpha to 1 channel 32-bit floating point packed Gray conversion.
- `NppStatus nppiColorToGray_32f_C4C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[4])
4 channel 32-bit floating point packed RGBA to 1 channel 32-bit floating point packed Gray conversion.

ColorDebayer

Grayscale Color Filter Array to RGB Color Debayer conversion.

Generates one RGB color pixel for every grayscale source pixel. Source and destination images must have even width and height. Missing pixel colors are generated using bilinear interpolation with chroma correlation of generated green values (eInterpolation MUST be set to 0). eGrid allows the user to specify the Bayer grid registration position at source image location oSrcROI.x, oSrcROI.y relative to pSrc. Possible registration positions are:

NPPI_BAYER_BGGR	NPPI_BAYER_RGGB	NPPI_BAYER_GBRG	NPPI_BAYER_GRGB
B G	R G	G B	G R
G R	G B	R G	B G

If it becomes necessary to access source pixels outside source image then the source image borders are mirrored.

Here is how the algorithm works. R, G, and B base pixels from the source image are used unmodified. To generate R values for those G pixels, the average of $R(x - 1, y)$ and $R(x + 1, y)$ or $R(x, y - 1)$ and $R(x, y + 1)$ is used depending on whether the left and right or top and bottom pixels are R base pixels. To generate B values for those G pixels, the same algorithm is used using nearest B values. For an R base pixel, if there are no B values in the upper, lower, left, or right adjacent pixels then B is the average of B values in the 4 diagonal (G base) pixels. The same algorithm is used using R values to generate the R value of a B base pixel. Chroma correlation is applied to generated G values only, for a B base pixel $G(x - 1, y)$ and $G(x + 1, y)$ are averaged or $G(x, y - 1)$ and $G(x, y + 1)$ are averaged depending on whether the absolute difference between $B(x, y)$ and the average of $B(x - 2, y)$ and $B(x + 2, y)$ is smaller than the absolute difference between $B(x, y)$ and the average of $B(x, y - 2)$ and $B(x, y + 2)$. For an R base pixel the same algorithm is used testing against the surrounding R values at those offsets. If the horizontal and vertical differences are the same at one of those pixels then the average of the four left, right, upper and lower G values is used instead.

- **NppStatus nppiCFAToRGB_8u_C1C3R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiRect** oSrcROI, **Npp8u** *pDst, int nDstStep, **NppiBayerGridPosition** eGrid, **NppiInterpolationMode** eInterpolation)
1 channel 8-bit unsigned packed CFA grayscale Bayer pattern to 3 channel 8-bit unsigned packed RGB conversion.
- **NppStatus nppiCFAToRGBA_8u_C1AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiRect** oSrcROI, **Npp8u** *pDst, int nDstStep, **NppiBayerGridPosition** eGrid, **NppiInterpolationMode** eInterpolation, **Npp8u** nAlpha)
1 channel 8-bit unsigned packed CFA grayscale Bayer pattern to 4 channel 8-bit unsigned packed RGB conversion with alpha.
- **NppStatus nppiCFAToRGB_16u_C1C3R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiRect** oSrcROI, **Npp16u** *pDst, int nDstStep, **NppiBayerGridPosition** eGrid, **NppiInterpolationMode** eInterpolation)
1 channel 16-bit unsigned packed CFA grayscale Bayer pattern to 3 channel 16-bit unsigned packed RGB conversion.
- **NppStatus nppiCFAToRGBA_16u_C1AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiRect** oSrcROI, **Npp16u** *pDst, int nDstStep, **NppiBayerGridPosition** eGrid, **NppiInterpolationMode** eInterpolation, **Npp16u** nAlpha)
1 channel 16-bit unsigned packed CFA grayscale Bayer pattern to 4 channel 16-bit unsigned packed RGB conversion with alpha.

7.44.1 Detailed Description

Routines for converting between various image color models.

7.44.2 Function Documentation

7.44.2.1 **NppStatus nppiBGRToCbYCr422_709HDTV_8u_AC4C2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422_709HDTV color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.2 NppStatus nppiBGRToCbYCr422_709HDTV_8u_C3C2R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed CbYCr422_709HDTV color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.3 NppStatus nppiBGRToCbYCr422_8u_AC4C2R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422 color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.4 NppStatus nppiBGRToHLS_8u_AC4P4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[4], int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.5 NppStatus nppiBGRToHLS_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.6 NppStatus nppiBGRToHLS_8u_AP4C4R (const Npp8u *const *pSrc*[4], int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.7 **NppStatus nppiBGRToHLS_8u_AP4R** (const Npp8u *const *pSrc*[4], int *nSrcStep*, Npp8u **pDst*[4], int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.8 **NppStatus nppiBGRToHLS_8u_C3P3R** (const Npp8u **pSrc*, int *nSrcStep*, Npp8u **pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar HLS color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.9 **NppStatus nppiBGRToHLS_8u_P3C3R** (const Npp8u *const *pSrc*[3], int *nSrcStep*, Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned packed HLS color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.10 NppStatus nppiBGRToHLS_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar HLS color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.11 NppStatus nppiBGRToLab_8u_C3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed Lab color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.12 NppStatus nppiBGRToYCbCr411_8u_AC4P3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.13 **NppStatus nppiBGRToYCbCr411_8u_C3P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr411 color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.14 **NppStatus nppiBGRToYCbCr420_709CSC_8u_AC4P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420_709CSC color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.15 **NppStatus nppiBGRToYCbCr420_709CSC_8u_C3P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420_709CSC color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.16 NppStatus nppiBGRToYCbCr420_709HDTV_8u_AC4P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420_709HDTV color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.17 NppStatus nppiBGRToYCbCr420_8u_AC4P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.18 **NppStatus nppiBGRToYCbCr420_8u_C3P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420 color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.19 **NppStatus nppiBGRToYCbCr422_8u_AC4C2R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed YCrCb422 color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.20 **NppStatus nppiBGRToYCbCr422_8u_AC4P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr422 color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.21 NppStatus nppiBGRToYCbCr422_8u_C3C2R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed YCrCb422 color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.22 NppStatus nppiBGRToYCbCr422_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr422 color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.23 **NppStatus nppiBGRToYCbCr_8u_AC4P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.24 **NppStatus nppiBGRToYCbCr_8u_AC4P4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[4], int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar YCbCr color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.25 **NppStatus nppiBGRToYCbCr_8u_C3P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.26 NppStatus nppiBGRToYCrCb420_709CSC_8u_AC4P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420_709CSC color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.27 NppStatus nppiBGRToYCrCb420_709CSC_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420_709CSC color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.28 **NppStatus nppiBGRToYCrCb420_8u_AC4P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.29 **NppStatus nppiBGRToYCrCb420_8u_C3P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.30 **NppStatus nppiBGRToYUV420_8u_AC4P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YUV420 color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.31 NppStatus nppiBGRToYUV_8u_AC4P4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[4], int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.32 NppStatus nppiBGRToYUV_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.33 **NppStatus nppiBGRToYUV_8u_C3P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YUV color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.34 **NppStatus nppiBGRToYUV_8u_C3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed YUV color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.35 **NppStatus nppiBGRToYUV_8u_P3R** (const Npp8u *const *pSrc*[3], int *nSrcStep*, Npp8u * *pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar YUV color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.36 NppStatus nppiCbYCr422ToBGR_709HDTV_8u_C2C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned packed BGR_709HDTV color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.37 NppStatus nppiCbYCr422ToBGR_709HDTV_8u_C2C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)

2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR_709HDTV color conversion with constant alpha.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nAval 8-bit unsigned alpha constant.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.38 NppStatus nppiCbYCr422ToBGR_8u_C2C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)

2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR color conversion with alpha.

images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nAval 8-bit unsigned alpha constant.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.39 `NppStatus nppiCbYCr422ToRGB_8u_C2C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned packed CbYCrC22 to 3 channel 8-bit unsigned packed RGB color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.40 `NppStatus nppiCFAToRGB_16u_C1C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiBayerGridPosition eGrid, NppiInterpolationMode eInterpolation)`

1 channel 16-bit unsigned packed CFA grayscale Bayer pattern to 3 channel 16-bit unsigned packed RGB conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize full source image width and height relative to pSrc.
oSrcROI rectangle specifying starting source image pixel x and y location relative to pSrc and ROI width and height.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
eGrid enumeration value specifying bayer grid registration position at location oSrcROI.x, oSrcROI.y relative to pSrc.
eInterpolation MUST be NPPI_INTER_UNDEFINED

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.41 NppStatus nppiCFAToRGB_8u_C1C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiBayerGridPosition eGrid, NppiInterpolationMode eInterpolation)

1 channel 8-bit unsigned packed CFA grayscale Bayer pattern to 3 channel 8-bit unsigned packed RGB conversion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize full source image width and height relative to pSrc.

oSrcROI rectangle specifying starting source image pixel x and y location relative to pSrc and ROI width and height.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

eGrid enumeration value specifying bayer grid registration position at location oSrcROI.x, oSrcROI.y relative to pSrc.

eInterpolation MUST be NPPI_INTER_UNDEFINED

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.42 NppStatus nppiCFAToRGBA_16u_C1AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiBayerGridPosition eGrid, NppiInterpolationMode eInterpolation, Npp16u nAlpha)

1 channel 16-bit unsigned packed CFA grayscale Bayer pattern to 4 channel 16-bit unsigned packed RGB conversion with alpha.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize full source image width and height relative to pSrc.

oSrcROI rectangle specifying starting source image pixel x and y location relative to pSrc and ROI width and height.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

eGrid enumeration value specifying bayer grid registration position at location oSrcROI.x, oSrcROI.y relative to pSrc.

eInterpolation MUST be NPPI_INTER_UNDEFINED

nAlpha constant alpha value to be written to each destination pixel

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.43 `NppStatus nppiCFAToRGBA_8u_C1AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiBayerGridPosition eGrid, NppiInterpolationMode eInterpolation, Npp8u nAlpha)`

1 channel 8-bit unsigned packed CFA grayscale Bayer pattern to 4 channel 8-bit unsigned packed RGB conversion with alpha.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize full source image width and height relative to pSrc.

oSrcROI rectangle specifying starting source image pixel x and y location relative to pSrc and ROI width and height.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

eGrid enumeration value specifying bayer grid registration position at location oSrcROI.x, oSrcROI.y relative to pSrc.

eInterpolation MUST be NPPI_INTER_UNDEFINED

nAlpha constant alpha value to be written to each destination pixel

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.44 `NppStatus nppiColorToGray_16s_AC4C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])`

4 channel 16-bit signed packed RGB with alpha to 1 channel 16-bit signed packed Gray conversion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.45 `NppStatus nppiColorToGray_16s_C3C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])`

3 channel 16-bit signed packed RGB to 1 channel 16-bit signed packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.46 NppStatus nppiColorToGray_16s_C4C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[4])

4 channel 16-bit signed packed RGBA to 1 channel 16-bit signed packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.47 NppStatus nppiColorToGray_16u_AC4C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])

4 channel 16-bit unsigned packed RGB with alpha to 1 channel 16-bit unsigned packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.48 `NppStatus nppiColorToGray_16u_C3C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])`

3 channel 16-bit unsigned packed RGB to 1 channel 16-bit unsigned packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.49 `NppStatus nppiColorToGray_16u_C4C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[4])`

4 channel 16-bit unsigned packed RGBA to 1 channel 16-bit unsigned packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.50 `NppStatus nppiColorToGray_32f_AC4C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])`

4 channel 32-bit floating point packed RGB with alpha to 1 channel 32-bit floating point packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.51 `NppStatus nppiColorToGray_32f_C3C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])`

3 channel 32-bit floating point packed RGB to 1 channel 32-bit floating point packed Gray conversion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.52 `NppStatus nppiColorToGray_32f_C4C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[4])`

4 channel 32-bit floating point packed RGBA to 1 channel 32-bit floating point packed Gray conversion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.53 `NppStatus nppiColorToGray_8u_AC4C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])`

4 channel 8-bit unsigned packed RGB with alpha to 1 channel 8-bit unsigned packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.54 `NppStatus nppiColorToGray_8u_C3C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])`

3 channel 8-bit unsigned packed RGB to 1 channel 8-bit unsigned packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.55 `NppStatus nppiColorToGray_8u_C4C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[4])`

4 channel 8-bit unsigned packed RGBA to 1 channel 8-bit unsigned packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aCoeffs fixed size array of constant floating point conversion coefficient values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.56 NppStatus nppiHLSToBGR_8u_AC4P4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[4], int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.57 NppStatus nppiHLSToBGR_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.58 NppStatus nppiHLSToBGR_8u_AP4C4R (const Npp8u *const *pSrc*[4], int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.59 NppStatus nppiHLSToBGR_8u_AP4R (const Npp8u *const pSrc[4], int nSrcStep, Npp8u *pDst[4], int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.60 NppStatus nppiHLSToBGR_8u_C3P3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned planar BGR color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.61 NppStatus nppiHLSToBGR_8u_P3C3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.62 NppStatus nppiHLSToBGR_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned planar BGR color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.63 NppStatus nppiHLSToRGB_8u_AC4R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.64 NppStatus nppiHLSToRGB_8u_C3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.65 `NppStatus nppiHSVToRGB_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed HSV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.66 `NppStatus nppiHSVToRGB_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed HSV to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.67 `NppStatus nppiLabToBGR_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed Lab to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.68 NppStatus nppiLUVToRGB_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed LUV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.69 NppStatus nppiLUVToRGB_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed LUV to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.70 NppStatus nppiNV21ToBGR_8u_P2C4R (const Npp8u *const *pSrc*[2], int *rSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned planar NV21 to 4 channel 8-bit unsigned packed BGRA color conversion with constant alpha (0xFF).

Parameters:

pSrc Source-Planar-Image Pointer Array (one for Y plane, one for VU plane).
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.71 **NppStatus nppiNV21ToRGB_8u_P2C4R** (const Npp8u *const *pSrc*[2], int *rSrcStep*, Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned planar NV21 to 4 channel 8-bit unsigned packed ARGB color conversion with constant alpha (0xFF).

Parameters:

pSrc Source-Planar-Image Pointer Array (one for Y plane, one for VU plane).

rSrcStep Source-Planar-Image Line Step Array.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.72 **NppStatus nppiRGBToCbYCr422_8u_C3C2R** (const Npp8u **pSrc*, int *nSrcStep*, Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed CbYCr422 color conversion. images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.73 **NppStatus nppiRGBToCbYCr422Gamma_8u_C3C2R** (const Npp8u **pSrc*, int *nSrcStep*, Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB first gets forward gamma corrected then converted to 2 channel 8-bit unsigned packed CbYCr422 color conversion. images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.74 `NppStatus nppiRGBToGray_16s_AC4C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit signed packed RGB with alpha to 1 channel 16-bit signed packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.75 `NppStatus nppiRGBToGray_16s_C3C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 16-bit signed packed RGB to 1 channel 16-bit signed packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.76 `NppStatus nppiRGBToGray_16u_AC4C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned packed RGB with alpha to 1 channel 16-bit unsigned packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.77 NppStatus nppiRGBToGray_16u_C3C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 16-bit unsigned packed RGB to 1 channel 16-bit unsigned packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.78 NppStatus nppiRGBToGray_32f_AC4C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 32-bit floating point packed RGB with alpha to 1 channel 32-bit floating point packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.79 NppStatus nppiRGBToGray_32f_C3C1R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 32-bit floating point packed RGB to 1 channel 32-bit floating point packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.80 NppStatus nppiRGBToGray_8u_AC4C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed RGB with alpha to 1 channel 8-bit unsigned packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.81 NppStatus nppiRGBToGray_8u_C3C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 1 channel 8-bit unsigned packed Gray conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.82 **NppStatus nppiRGBToHLS_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.83 **NppStatus nppiRGBToHLS_8u_C3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HLS color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.84 **NppStatus nppiRGBToHSV_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HSV with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.85 NppStatus nppiRGBToHSV_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HSV color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.86 NppStatus nppiRGBToLUV_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed LUV with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.87 NppStatus nppiRGBToLUV_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed LUV color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.88 **NppStatus nppiRGBToXYZ_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed XYZ with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.89 **NppStatus nppiRGBToXYZ_8u_C3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed XYZ color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.90 **NppStatus nppiRGBToYCbCr420_8u_C3P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr420 color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.91 NppStatus nppiRGBToYCbCr422_8u_C3C2R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.92 NppStatus nppiRGBToYCbCr422_8u_C3P3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr422 color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.93 NppStatus nppiRGBToYCbCr422_8u_P3C2R (const Npp8u *const *pSrc*[3], int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion. images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.94 `NppStatus nppiRGBToYCbCr_8u_AC4P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Planar-Image Pointer Array.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.95 `NppStatus nppiRGBToYCbCr_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed RGB with alpha to 4 channel unsigned 8-bit packed YCbCr with alpha color conversion, not affecting alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.96 `NppStatus nppiRGBToYCbCr_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit planar YCbCr color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.97 NppStatus nppiRGBToYCbCr_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit packed YCbCr color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.98 NppStatus nppiRGBToYCbCr_8u_P3R (const Npp8u *const *pSrc*[3], int *nSrcStep*, Npp8u * *pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

3 channel planar 8-bit unsigned RGB to 3 channel planar 8-bit YCbCr color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.99 **NppStatus nppiRGBToYCC_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YCC with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.100 **NppStatus nppiRGBToYCC_8u_C3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YCC color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.101 **NppStatus nppiRGBToYCrCb420_8u_AC4P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.102 NppStatus nppiRGBToYCrCb422_8u_C3C2R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.103 NppStatus nppiRGBToYCrCb422_8u_P3C2R (const Npp8u *const *pSrc*[3], int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion. images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.104 NppStatus nppiRGBToYUV420_8u_C3P3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV420 color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.105 NppStatus nppiRGBToYUV420_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV420 color conversion.
images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.106 NppStatus nppiRGBToYUV422_8u_C3C2R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YUV422 color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.107 NppStatus nppiRGBToYUV422_8u_C3P3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV422 color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.108 NppStatus nppiRGBToYUV422_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV422 color conversion. images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.109 NppStatus nppiRGBToYUV_8u_AC4P4R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[4], int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.110 NppStatus nppiRGBToYUV_8u_AC4R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.111 `NppStatus nppiRGBToYUV_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV color conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Planar-Image Pointer Array.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.112 `NppStatus nppiRGBToYUV_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YUV color conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.113 `NppStatus nppiRGBToYUV_8u_P3R (const Npp8u * const pSrc[3], int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.

nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.114 NppStatus nppiXYZToRGB_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed XYZ with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.115 NppStatus nppiXYZToRGB_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned packed XYZ to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.116 **NppStatus nppiYCbCr411ToBGR_8u_P3C3R** (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.117 **NppStatus nppiYCbCr411ToBGR_8u_P3C4R** (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp8u *nAval*)

3 channel 8-bit unsigned planar YCbCr411 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nAval 8-bit unsigned alpha constant.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.118 **NppStatus nppiYCbCr420ToBGR_709CSC_8u_P3C3R** (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR_709CSC color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.119 NppStatus nppiYCbCr420ToBGR_709HDTV_8u_P3C4R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)

3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR_709HDTV color conversion with constant alpha.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nAval 8-bit unsigned alpha constant.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.120 NppStatus nppiYCbCr420ToBGR_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.121 NppStatus nppiYCbCr420ToBGR_8u_P3C4R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)

3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

nAval 8-bit unsigned alpha constant.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.122 `NppStatus nppiYCbCr420ToRGB_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

rSrcStep [Source-Planar-Image Line Step Array](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.123 `NppStatus nppiYCbCr422ToBGR_8u_C2C3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed BGR color conversion. images.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.124 `NppStatus nppiYCbCr422ToBGR_8u_C2C4R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)`

2 channel 8-bit unsigned packed YCrCb422 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nAval 8-bit unsigned alpha constant.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.125 `NppStatus nppiYCbCr422ToBGR_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed BGR color conversion. images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.126 `NppStatus nppiYCbCr422ToRGB_8u_C2C3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.127 NppStatus nppiYCbCr422ToRGB_8u_C2P3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar RGB color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.128 NppStatus nppiYCbCr422ToRGB_8u_P3C3R (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion. images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.129 NppStatus nppiYCbCrToBGR_709CSC_8u_P3C3R (const Npp8u *const *pSrc*[3], int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR_709CSC color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.130 NppStatus nppiYCbCrToBGR_709CSC_8u_P3C4R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR_709CSC color conversion with constant alpha.

Parameters:

pSrc Source-Planar-Image Pointer Array.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nAval 8-bit unsigned alpha constant.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.131 NppStatus nppiYCbCrToBGR_8u_P3C3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.132 NppStatus nppiYCbCrToBGR_8u_P3C4R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

Parameters:

pSrc Source-Planar-Image Pointer Array.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nAval 8-bit unsigned alpha constant.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.133 NppStatus nppiYCbCrToRGB_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed YCbCr with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion, not affecting alpha.

Alpha channel is the last channel and is not processed.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.134 NppStatus nppiYCbCrToRGB_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned packed YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.135 NppStatus nppiYCbCrToRGB_8u_P3C3R (const Npp8u * const pSrc[3], int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.136 NppStatus nppiYCbCrToRGB_8u_P3C4R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nAval 8-bit unsigned alpha constant.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.137 NppStatus nppiYCbCrToRGB_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.138 **NppStatus nppiYCCToRGB_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed YCC with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.139 **NppStatus nppiYCCToRGB_8u_C3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed YCC to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.140 **NppStatus nppiYCrCb420ToRGB_8u_P3C4R** (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp8u *nAval*)

3 channel 8-bit unsigned planar YCrCb420 to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nAval 8-bit unsigned alpha constant.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.141 NppStatus nppiYCrCb422ToRGB_8u_C2C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed RGB color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.142 NppStatus nppiYCrCb422ToRGB_8u_C2P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar RGB color conversion. images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.143 NppStatus nppiYUV420ToBGR_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.144 NppStatus nppiYUV420ToBGR_8u_P3C4R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha (0xFF).

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.145 NppStatus nppiYUV420ToRGB_8u_P3AC4R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.146 NppStatus nppiYUV420ToRGB_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.147 NppStatus nppiYUV420ToRGB_8u_P3C4R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha (0xFF).

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.148 NppStatus nppiYUV420ToRGB_8u_P3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Planar-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.149 NppStatus nppiYUV422ToRGB_8u_C2C3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

2 channel 8-bit unsigned packed YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.150 NppStatus nppiYUV422ToRGB_8u_P3AC4R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YUV422 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.151 NppStatus nppiYUV422ToRGB_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.152 NppStatus nppiYUV422ToRGB_8u_P3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Planar-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.153 NppStatus nppiYUVToBGR_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed BGR color conversion with alpha, not affecting alpha.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.154 NppStatus nppiYUVToBGR_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.155 NppStatus nppiYUVToBGR_8u_P3C3R (const Npp8u *const *pSrc*[3], int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.156 `NppStatus nppiYUVToBGR_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar BGR color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.157 `NppStatus nppiYUVToRGB_8u_AC4R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed RGB color conversion with alpha, not affecting alpha.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.158 `NppStatus nppiYUVToRGB_8u_C3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.44.2.159 `NppStatus nppiYUVToRGB_8u_P3C3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.44.2.160 `NppStatus nppiYUVToRGB_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.

nSrcStep Source-Image Line Step.

pDst Destination-Planar-Image Pointer Array.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45 Color Sampling Format Conversion

Routines for converting between various image color sampling formats.

YCbCr420ToYCbCr411

YCbCr420 to YCbCr411 sampling format conversion.

- **NppStatus** **nppiYCbCr420ToYCbCr411_8u_P3P2R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDstY, int nDstYStep, **Npp8u** *pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.
- **NppStatus** **nppiYCbCr420ToYCbCr411_8u_P2P3R** (const **Npp8u** *pSrcY, int nSrcYStep, const **Npp8u** *pSrcCbCr, int nSrcCbCrStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

YCbCr422ToYCbCr422

YCbCr422 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCbCr422_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.
- **NppStatus** **nppiYCbCr422_8u_P3C2R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

YCbCr422ToYCrCb422

YCbCr422 to YCrCb422 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCrCb422_8u_C2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.
- **NppStatus** **nppiYCbCr422ToYCrCb422_8u_P3C2R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

YCbCr422ToCbYCr422

YCbCr422 to CbYCr422 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToCbYCr422_8u_C2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

CbYCr422ToYCbCr411

CbYCr422 to YCbCr411 sampling format conversion.

- **NppStatus** **nppiCbYCr422ToYCbCr411_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

YCbCr422ToYCbCr420

YCbCr422 to YCbCr420 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCbCr420_8u_P3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst[3], int nDstStep[3], **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.
- **NppStatus** **nppiYCbCr422ToYCbCr420_8u_P3P2R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDstY, int nDstYStep, **Npp8u** *pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.
- **NppStatus** **nppiYCbCr422ToYCbCr420_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.
- **NppStatus** **nppiYCbCr422ToYCbCr420_8u_C2P2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDstY, int nDstYStep, **Npp8u** *pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)
2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

YCrCb420ToYCbCr422

YCrCb420 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCrCb420ToYCbCr422_8u_P3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YCrCb420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCrCb420ToYCbCr422_8u_P3C2R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

YCbCr422ToYCrCb420

YCbCr422 to YCrCb420 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCrCb420_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

YCbCr422ToYCbCr411

YCbCr422 to YCbCr411 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCbCr411_8u_P3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCbCr411_8u_P3P2R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDstY, int nDstYStep, **Npp8u** *pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCbCr411_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCbCr411_8u_C2P2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDstY, int nDstYStep, **Npp8u** *pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

YCrCb422ToYCbCr422

YCrCb422 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCrCb422ToYCbCr422_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

YCrCb422ToYCbCr420

YCrCb422 to YCbCr420 sampling format conversion.

- **NppStatus** **nppiYCrCb422ToYCbCr420_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

YCrCb422ToYCbCr411

YCrCb422 to YCbCr411 sampling format conversion.

- **NppStatus** **nppiYCrCb422ToYCbCr411_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

CbYCr422ToYCbCr422

CbYCr422 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiCbYCr422ToYCbCr422_8u_C2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

- **NppStatus** **nppiCbYCr422ToYCbCr422_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

CbYCr422ToYCbCr420

CbYCr422 to YCbCr420 sampling format conversion.

- **NppStatus** **nppiCbYCr422ToYCbCr420_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

- **NppStatus** **nppiCbYCr422ToYCbCr420_8u_C2P2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDstY, int nDstYStep, **Npp8u** *pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

CbYCr422ToYCrCb420

CbYCr422 to YCrCb420 sampling format conversion.

- **NppStatus** **nppiCbYCr422ToYCrCb420_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

YCbCr420ToYCbCr420

YCbCr420 to YCbCr420 sampling format conversion.

- **NppStatus** **nppiYCbCr420_8u_P3P2R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDstY, int nDstYStep, **Npp8u** *pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

- **NppStatus** **nppiYCbCr420_8u_P2P3R** (const **Npp8u** *const pSrcY, int nSrcYStep, const **Npp8u** *pSrcCbCr, int nSrcCbCrStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

YCbCr420ToYCbCr422

YCbCr420 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCbCr420ToYCbCr422_8u_P3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst[3], int nDstStep[3], **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCbCr420ToYCbCr422_8u_P2P3R** (const **Npp8u** *pSrcY, int nSrcYStep, const **Npp8u** *pSrcCbCr, int nSrcCbCrStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCbCr420ToYCbCr422_8u_P2C2R** (const **Npp8u** *pSrcY, int nSrcYStep, const **Npp8u** *pSrcCbCr, int nSrcCbCrStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

YCbCr420ToCbYCr422

YCbCr420 to CbYCr422 sampling format conversion.

- `NppStatus nppiYCbCr420ToCbYCr422_8u_P2C2R` (const `Npp8u` *pSrcY, int nSrcYStep, const `Npp8u` *pSrcCbCr, int nSrcCbCrStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

YCbCr420ToYCrCb420

YCbCr420 to YCrCb420 sampling format conversion.

- `NppStatus nppiYCbCr420ToYCrCb420_8u_P2P3R` (const `Npp8u` *pSrcY, int nSrcYStep, const `Npp8u` *pSrcCbCr, int nSrcCbCrStep, `Npp8u` *pDst[3], int rDstStep[3], `NppiSize` oSizeROI)
2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

YCrCb420ToCbYCr422

YCrCb420 to CbYCr422 sampling format conversion.

- `NppStatus nppiYCrCb420ToCbYCr422_8u_P3C2R` (const `Npp8u` *const pSrc[3], int rSrcStep[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

YCrCb420ToYCbCr420

YCrCb420 to YCbCr420 sampling format conversion.

- `NppStatus nppiYCrCb420ToYCbCr420_8u_P3P2R` (const `Npp8u` *const pSrc[3], int rSrcStep[3], `Npp8u` *pDstY, int nDstYStep, `Npp8u` *pDstCbCr, int nDstCbCrStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

YCrCb420ToYCbCr411

YCrCb420 to YCbCr411 sampling format conversion.

- `NppStatus nppiYCrCb420ToYCbCr411_8u_P3P2R` (const `Npp8u` *const pSrc[3], int rSrcStep[3], `Npp8u` *pDstY, int nDstYStep, `Npp8u` *pDstCbCr, int nDstCbCrStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

YCbCr411ToYCbCr411

YCbCr411 to YCbCr411 sampling format conversion.

- **NppStatus** **nppiYCbCr411_8u_P3P2R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDstY, int nDstYStep, **Npp8u** *pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.
- **NppStatus** **nppiYCbCr411_8u_P2P3R** (const **Npp8u** *pSrcY, int nSrcYStep, const **Npp8u** *pSrcCbCr, int nSrcCbCrStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

YCbCr411ToYCbCr422

YCbCr411 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCbCr411ToYCbCr422_8u_P3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst[3], int nDstStep[3], **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.
- **NppStatus** **nppiYCbCr411ToYCbCr422_8u_P3C2R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.
- **NppStatus** **nppiYCbCr411ToYCbCr422_8u_P2P3R** (const **Npp8u** *const pSrcY, int nSrcYStep, const **Npp8u** *pSrcCbCr, int nSrcCbCrStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.
- **NppStatus** **nppiYCbCr411ToYCbCr422_8u_P2C2R** (const **Npp8u** *pSrcY, int nSrcYStep, const **Npp8u** *pSrcCbCr, int nSrcCbCrStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
2 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

YCbCr411ToYCrCb422

YCbCr411 to YCrCb422 sampling format conversion.

- **NppStatus** **nppiYCbCr411ToYCrCb422_8u_P3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst[3], int nDstStep[3], **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb422 sampling format conversion.
- **NppStatus** **nppiYCbCr411ToYCrCb422_8u_P3C2R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

YCbCr411ToYCbCr420

YCbCr411 to YCbCr420 sampling format conversion.

- **NppStatus nppiYCbCr411ToYCbCr420_8u_P3R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDst[3], int nDstStep[3], **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.
- **NppStatus nppiYCbCr411ToYCbCr420_8u_P3P2R** (const **Npp8u** *const pSrc[3], int rSrcStep[3], **Npp8u** *pDstY, int nDstYStep, **Npp8u** *pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.
- **NppStatus nppiYCbCr411ToYCbCr420_8u_P2P3R** (const **Npp8u** *pSrcY, int nSrcYStep, const **Npp8u** *pSrcCbCr, int nSrcCbCrStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

YCbCr411ToYCrCb420

YCbCr411 to YCrCb420 sampling format conversion.

- **NppStatus nppiYCbCr411ToYCrCb420_8u_P2P3R** (const **Npp8u** *pSrcY, int nSrcYStep, const **Npp8u** *pSrcCbCr, int nSrcCbCrStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)
2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

7.45.1 Detailed Description

Routines for converting between various image color sampling formats.

7.45.2 Function Documentation

7.45.2.1 **NppStatus nppiCbYCr422ToYCbCr411_8u_C2P3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.2 **NppStatus nppiCbYCr422ToYCbCr420_8u_C2P2R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDstY*, int *nDstYStep*, Npp8u * *pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDstY Destination-Planar-Image Pointer.
nDstYStep Destination-Planar-Image Line Step.
pDstCbCr Destination-Planar-Image Pointer.
nDstCbCrStep Destination-Planar-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.3 **NppStatus nppiCbYCr422ToYCbCr420_8u_C2P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.4 NppStatus nppiCbYCr422ToYCbCr422_8u_C2P3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Planar-Image Pointer Array.

rDstStep Destination-Planar-Image Line Step Array.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.5 NppStatus nppiCbYCr422ToYCbCr422_8u_C2R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.6 NppStatus nppiCbYCr422ToYCrCb420_8u_C2P3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.7 `NppStatus nppiYCbCr411_8u_P2P3R (const Npp8u *pSrcY, int nSrcYStep, const Npp8u *pSrcCbCr, int nSrcCbCrStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

Parameters:

pSrcY Source-Planar-Image Pointer.
nSrcYStep Source-Planar-Image Line Step.
pSrcCbCr Source-Planar-Image Pointer.
nSrcCbCrStep Source-Planar-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.8 `NppStatus nppiYCbCr411_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDstY, int nDstYStep, Npp8u *pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDstY Destination-Planar-Image Pointer.
nDstYStep Destination-Planar-Image Line Step.
pDstCbCr Destination-Planar-Image Pointer.
nDstCbCrStep Destination-Planar-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.9 NppStatus nppiYCbCr411ToYCbCr420_8u_P2P3R (const Npp8u * *pSrcY*, int *nSrcYStep*, const Npp8u * *pSrcCbCr*, int *nSrcCbCrStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

Parameters:

pSrcY Source-Planar-Image Pointer.
nSrcYStep Source-Planar-Image Line Step.
pSrcCbCr Source-Planar-Image Pointer.
nSrcCbCrStep Source-Planar-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.10 NppStatus nppiYCbCr411ToYCbCr420_8u_P3P2R (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u * *pDstY*, int *nDstYStep*, Npp8u * *pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDstY Destination-Planar-Image Pointer.
nDstYStep Destination-Planar-Image Line Step.
pDstCbCr Destination-Planar-Image Pointer.
nDstCbCrStep Destination-Planar-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.11 NppStatus nppiYCbCr411ToYCbCr420_8u_P3R (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u * *pDst*[3], int *nDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.12 `NppStatus nppiYCbCr411ToYCbCr422_8u_P2C2R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

Parameters:

pSrcY Source-Planar-Image Pointer.
nSrcYStep Source-Planar-Image Line Step.
pSrcCbCr Source-Planar-Image Pointer.
nSrcCbCrStep Source-Planar-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.13 `NppStatus nppiYCbCr411ToYCbCr422_8u_P2P3R (const Npp8u * const pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

Parameters:

pSrcY Source-Planar-Image Pointer.
nSrcYStep Source-Planar-Image Line Step.
pSrcCbCr Source-Planar-Image Pointer.
nSrcCbCrStep Source-Planar-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.14 NppStatus nppiYCbCr411ToYCbCr422_8u_P3C2R (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.15 NppStatus nppiYCbCr411ToYCbCr422_8u_P3R (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u **pDst*[3], int *nDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.16 NppStatus nppiYCbCr411ToYCrCb420_8u_P2P3R (const Npp8u **pSrcY*, int *nSrcYStep*, const Npp8u **pSrcCbCr*, int *nSrcCbCrStep*, Npp8u **pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

Parameters:

pSrcY Source-Planar-Image Pointer.
nSrcYStep Source-Planar-Image Line Step.

pSrcCbCr Source-Planar-Image Pointer.
nSrcCbCrStep Source-Planar-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.17 NppStatus nppiYCbCr411ToYCrCb422_8u_P3C2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.18 NppStatus nppiYCbCr411ToYCrCb422_8u_P3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u * pDst[3], int nDstStep[3], NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb422 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.19 `NppStatus nppiYCbCr420_8u_P2P3R (const Npp8u *const pSrcY, int nSrcYStep, const Npp8u *pSrcCbCr, int nSrcCbCrStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

Parameters:

pSrcY Source-Planar-Image Pointer.
nSrcYStep Source-Planar-Image Line Step.
pSrcCbCr Source-Planar-Image Pointer.
nSrcCbCrStep Source-Planar-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.20 `NppStatus nppiYCbCr420_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDstY, int nDstYStep, Npp8u *pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDstY Destination-Planar-Image Pointer.
nDstYStep Destination-Planar-Image Line Step.
pDstCbCr Destination-Planar-Image Pointer.
nDstCbCrStep Destination-Planar-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.21 `NppStatus nppiYCbCr420ToCbYCr422_8u_P2C2R (const Npp8u *pSrcY, int nSrcYStep, const Npp8u *pSrcCbCr, int nSrcCbCrStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

Parameters:

pSrcY Source-Planar-Image Pointer.
nSrcYStep Source-Planar-Image Line Step.
pSrcCbCr Source-Planar-Image Pointer.
nSrcCbCrStep Source-Planar-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.22 `NppStatus nppiYCbCr420ToYCbCr411_8u_P2P3R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

Parameters:

pSrcY Source-Planar-Image Pointer.
nSrcYStep Source-Planar-Image Line Step.
pSrcCbCr Source-Planar-Image Pointer.
nSrcCbCrStep Source-Planar-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.23 `NppStatus nppiYCbCr420ToYCbCr411_8u_P3P2R (const Npp8u * const pSrc[3], int rSrcStep[3], Npp8u * pDstY, int nDstYStep, Npp8u * pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDstY Destination-Planar-Image Pointer.
nDstYStep Destination-Planar-Image Line Step.

pDstCbCr Destination-Planar-Image Pointer.
nDstCbCrStep Destination-Planar-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.24 `NppStatus nppiYCbCr420ToYCbCr422_8u_P2C2R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

Parameters:

pSrcY Source-Planar-Image Pointer.
nSrcYStep Source-Planar-Image Line Step.
pSrcCbCr Source-Planar-Image Pointer.
nSrcCbCrStep Source-Planar-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.25 `NppStatus nppiYCbCr420ToYCbCr422_8u_P2P3R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

Parameters:

pSrcY Source-Planar-Image Pointer.
nSrcYStep Source-Planar-Image Line Step.
pSrcCbCr Source-Planar-Image Pointer.
nSrcCbCrStep Source-Planar-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.26 `NppStatus nppiYCbCr420ToYCbCr422_8u_P3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst[3], int nDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.27 `NppStatus nppiYCbCr420ToYCrCb420_8u_P2P3R (const Npp8u *pSrcY, int nSrcYStep, const Npp8u *pSrcCbCr, int nSrcCbCrStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

Parameters:

pSrcY Source-Planar-Image Pointer.
nSrcYStep Source-Planar-Image Line Step.
pSrcCbCr Source-Planar-Image Pointer.
nSrcCbCrStep Source-Planar-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.28 `NppStatus nppiYCbCr422_8u_C2P3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.29 **NppStatus nppiYCbCr422_8u_P3C2R** (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.30 **NppStatus nppiYCbCr422ToCbYCr422_8u_C2R** (const Npp8u **pSrc*, int *nSrcStep*, Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.31 NppStatus nppiYCbCr422ToYCbCr411_8u_C2P2R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDstY*, int *nDstYStep*, Npp8u * *pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDstY Destination-Planar-Image Pointer.
nDstYStep Destination-Planar-Image Line Step.
pDstCbCr Destination-Planar-Image Pointer.
nDstCbCrStep Destination-Planar-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.32 NppStatus nppiYCbCr422ToYCbCr411_8u_C2P3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.33 NppStatus nppiYCbCr422ToYCbCr411_8u_P3P2R (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u * *pDstY*, int *nDstYStep*, Npp8u * *pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDstY Destination-Planar-Image Pointer.
nDstYStep Destination-Planar-Image Line Step.
pDstCbCr Destination-Planar-Image Pointer.
nDstCbCrStep Destination-Planar-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.34 NppStatus nppiYCbCr422ToYCbCr411_8u_P3R (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u **pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.35 NppStatus nppiYCbCr422ToYCbCr420_8u_C2P2R (const Npp8u **pSrc*, int *nSrcStep*, Npp8u **pDstY*, int *nDstYStep*, Npp8u **pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDstY Destination-Planar-Image Pointer.
nDstYStep Destination-Planar-Image Line Step.
pDstCbCr Destination-Planar-Image Pointer.

nDstCbCrStep Destination-Planar-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.36 `NppStatus nppiYCbCr422ToYCbCr420_8u_C2P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Planar-Image Pointer Array.

rDstStep Destination-Planar-Image Line Step Array.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.37 `NppStatus nppiYCbCr422ToYCbCr420_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u * pDstY, int nDstYStep, Npp8u * pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.

rSrcStep Source-Planar-Image Line Step Array.

pDstY Destination-Planar-Image Pointer.

nDstYStep Destination-Planar-Image Line Step.

pDstCbCr Destination-Planar-Image Pointer.

nDstCbCrStep Destination-Planar-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.38 NppStatus nppiYCbCr422ToYCbCr420_8u_P3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst[3], int nDstStep[3], NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.

rSrcStep Source-Planar-Image Line Step Array.

pDst Destination-Planar-Image Pointer Array.

nDstStep Destination-Planar-Image Line Step Array.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.39 NppStatus nppiYCbCr422ToYCrCb420_8u_C2P3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Planar-Image Pointer Array.

rDstStep Destination-Planar-Image Line Step Array.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.40 NppStatus nppiYCbCr422ToYCrCb422_8u_C2R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.41 **NppStatus nppiYCbCr422ToYCrCb422_8u_P3C2R** (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.42 **NppStatus nppiYCrCb420ToCbYCr422_8u_P3C2R** (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.43 `NppStatus nppiYCrCb420ToYCbCr411_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDstY, int nDstYStep, Npp8u *pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDstY Destination-Planar-Image Pointer.
nDstYStep Destination-Planar-Image Line Step.
pDstCbCr Destination-Planar-Image Pointer.
nDstCbCrStep Destination-Planar-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.44 `NppStatus nppiYCrCb420ToYCbCr420_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDstY, int nDstYStep, Npp8u *pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDstY Destination-Planar-Image Pointer.
nDstYStep Destination-Planar-Image Line Step.
pDstCbCr Destination-Planar-Image Pointer.
nDstCbCrStep Destination-Planar-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.45 **NppStatus nppiYCrCb420ToYCbCr422_8u_P3C2R** (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.46 **NppStatus nppiYCrCb420ToYCbCr422_8u_P3R** (const Npp8u *const *pSrc*[3], int *rSrcStep*[3], Npp8u **pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCrCb420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Planar-Image Pointer Array.
rSrcStep Source-Planar-Image Line Step Array.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.47 **NppStatus nppiYCrCb422ToYCbCr411_8u_C2P3R** (const Npp8u **pSrc*, int *nSrcStep*, Npp8u **pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.48 NppStatus nppiYCrCb422ToYCbCr420_8u_C2P3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.45.2.49 NppStatus nppiYCrCb422ToYCbCr422_8u_C2P3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Planar-Image Pointer Array.
rDstStep Destination-Planar-Image Line Step Array.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.46 Color Gamma Correction

Routines for correcting image color gamma.

GammaFwd

Forward gamma correction.

- **NppStatus nppiGammaFwd_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed color not in place forward gamma correction.
- **NppStatus nppiGammaFwd_8u_C3IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed color in place forward gamma correction.
- **NppStatus nppiGammaFwd_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed color with alpha not in place forward gamma correction.
- **NppStatus nppiGammaFwd_8u_AC4IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed color with alpha in place forward gamma correction.
- **NppStatus nppiGammaFwd_8u_P3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar color not in place forward gamma correction.
- **NppStatus nppiGammaFwd_8u_IP3R** (**Npp8u** *const pSrcDst[3], int nSrcDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned planar color in place forward gamma correction.

GammaInv

Inverse gamma correction.

- **NppStatus nppiGammaInv_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed color not in place inverse gamma correction.
- **NppStatus nppiGammaInv_8u_C3IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned packed color in place inverse gamma correction.
- **NppStatus nppiGammaInv_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed color with alpha not in place inverse gamma correction.
- **NppStatus nppiGammaInv_8u_AC4IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned packed color with alpha in place inverse gamma correction.

- **NppStatus nppiGammaInv_8u_P3R** (const **Npp8u** *const pSrc[3], int nSrcStep, **Npp8u** *pDst[3], int nDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar color not in place inverse gamma correction.

- **NppStatus nppiGammaInv_8u_IP3R** (**Npp8u** *const pSrcDst[3], int nSrcDstStep, **NppiSize** oSizeROI)

3 channel 8-bit unsigned planar color in place inverse gamma correction.

7.46.1 Detailed Description

Routines for correcting image color gamma.

7.46.2 Function Documentation

7.46.2.1 **NppStatus nppiGammaFwd_8u_AC4IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed color with alpha in place forward gamma correction.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.46.2.2 **NppStatus nppiGammaFwd_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed color with alpha not in place forward gamma correction.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.46.2.3 NppStatus nppiGammaFwd_8u_C3IR (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed color in place forward gamma correction.

Parameters:

pSrcDst in place packed pixel image pointer.
nSrcDstStep in place packed pixel format image line step.
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.46.2.4 NppStatus nppiGammaFwd_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed color not in place forward gamma correction.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.46.2.5 NppStatus nppiGammaFwd_8u_IP3R (Npp8u *const *pSrcDst*[3], int *nSrcDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar color in place forward gamma correction.

Parameters:

pSrcDst in place planar pixel format image pointer array.
nSrcDstStep in place planar pixel format image line step.
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.46.2.6 NppStatus nppiGammaFwd_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar color not in place forward gamma correction.

Parameters:

pSrc source planar pixel format image pointer array.

nSrcStep source planar pixel format image line step.

pDst destination planar pixel format image pointer array.

nDstStep destination planar pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.46.2.7 NppStatus nppiGammaInv_8u_AC4IR (Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed color with alpha in place inverse gamma correction.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.46.2.8 NppStatus nppiGammaInv_8u_AC4R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed color with alpha not in place inverse gamma correction.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.46.2.9 NppStatus nppiGammaInv_8u_C3IR (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed color in place inverse gamma correction.

Parameters:

pSrcDst in place packed pixel format image pointer.
nSrcDstStep in place packed pixel format image line step.
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.46.2.10 NppStatus nppiGammaInv_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed color not in place inverse gamma correction.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.46.2.11 NppStatus nppiGammaInv_8u_IP3R (Npp8u *const *pSrcDst*[3], int *nSrcDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar color in place inverse gamma correction.

Parameters:

pSrcDst in place planar pixel format image pointer array.
nSrcDstStep in place planar pixel format image line step.
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.46.2.12 NppStatus nppiGammaInv_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar color not in place inverse gamma correction.

Parameters:

pSrc source planar pixel format image pointer array.

nSrcStep source planar pixel format image line step.

pDst destination planar pixel format image pointer array.

nDstStep destination planar pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.47 Complement Color Key

Routines for performing complement color key replacement.

CompColorKey

Complement color key replacement.

- **NppStatus nppiCompColorKey_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nColorKeyConst)
1 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.
- **NppStatus nppiCompColorKey_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nColorKeyConst[3])
3 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.
- **NppStatus nppiCompColorKey_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nColorKeyConst[4])
4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.
- **NppStatus nppiAlphaCompColorKey_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, **Npp8u** nAlpha1, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** nAlpha2, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nColorKeyConst[4], **NppiAlphaOp** nppAlphaOp)
4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2 with alpha blending.

7.47.1 Detailed Description

Routines for performing complement color key replacement.

7.47.2 Function Documentation

7.47.2.1 NppStatus nppiAlphaCompColorKey_8u_AC4R (const **Npp8u** *pSrc1, int nSrc1Step, **Npp8u** nAlpha1, const **Npp8u** *pSrc2, int nSrc2Step, **Npp8u** nAlpha2, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nColorKeyConst[4], **NppiAlphaOp** nppAlphaOp)

4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2 with alpha blending.

Parameters:

- pSrc1** source1 packed pixel format image pointer.
- nSrc1Step** source1 packed pixel format image line step.
- nAlpha1** source1 image alpha opacity (0 - max channel pixel value).
- pSrc2** source2 packed pixel format image pointer.

nSrc2Step source2 packed pixel format image line step.
nAlpha2 source2 image alpha opacity (0 - max channel pixel value).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nColorKeyConst color key constant array
nppAlphaOp NppiAlphaOp alpha compositing operation selector (excluding premul ops).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.47.2.2 NppStatus nppiCompColorKey_8u_C1R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp8u *nColorKeyConst*)

1 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

Parameters:

pSrc1 source1 packed pixel format image pointer.
nSrc1Step source1 packed pixel format image line step.
pSrc2 source2 packed pixel format image pointer.
nSrc2Step source2 packed pixel format image line step.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nColorKeyConst color key constant

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.47.2.3 NppStatus nppiCompColorKey_8u_C3R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp8u *nColorKeyConst*[3])

3 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

Parameters:

pSrc1 source1 packed pixel format image pointer.
nSrc1Step source1 packed pixel format image line step.
pSrc2 source2 packed pixel format image pointer.
nSrc2Step source2 packed pixel format image line step.

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nColorKeyConst color key constant array

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.47.2.4 `NppStatus nppiCompColorKey_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp8u nColorKeyConst[4])`

4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

Parameters:

pSrc1 source1 packed pixel format image pointer.
nSrc1Step source1 packed pixel format image line step.
pSrc2 source2 packed pixel format image pointer.
nSrc2Step source2 packed pixel format image line step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nColorKeyConst color key constant array

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.48 Color Processing

Routines for performing image color manipulation.

ColorTwist

Perform color twist pixel processing.

Color twist consists of applying the following formula to each image pixel using coefficients from the user supplied color twist host matrix array as follows where dst[x] and src[x] represent destination pixel and source pixel channel or plane x. The full sized coefficient matrix should be sent for all pixel channel sizes, the function will process the appropriate coefficients and channels for the corresponding pixel size.

$$\begin{aligned} \text{dst}[0] &= \text{aTwist}[0][0] * \text{src}[0] + \text{aTwist}[0][1] * \text{src}[1] + \text{aTwist}[0][2] * \text{src}[2] + \text{aTwist}[0][3] \\ \text{dst}[1] &= \text{aTwist}[1][0] * \text{src}[0] + \text{aTwist}[1][1] * \text{src}[1] + \text{aTwist}[1][2] * \text{src}[2] + \text{aTwist}[1][3] \\ \text{dst}[2] &= \text{aTwist}[2][0] * \text{src}[0] + \text{aTwist}[2][1] * \text{src}[1] + \text{aTwist}[2][2] * \text{src}[2] + \text{aTwist}[2][3] \end{aligned}$$

- **NppStatus nppiColorTwist32f_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** aTwist[3][4])
1 channel 8-bit unsigned color twist.
- **NppStatus nppiColorTwist32f_8u_C1IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** aTwist[3][4])
1 channel 8-bit unsigned in place color twist.
- **NppStatus nppiColorTwist32f_8u_C2R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** aTwist[3][4])
2 channel 8-bit unsigned color twist.
- **NppStatus nppiColorTwist32f_8u_C2IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** aTwist[3][4])
2 channel 8-bit unsigned in place color twist.
- **NppStatus nppiColorTwist32f_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** aTwist[3][4])
3 channel 8-bit unsigned color twist.
- **NppStatus nppiColorTwist32f_8u_C3IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** aTwist[3][4])
3 channel 8-bit unsigned in place color twist.
- **NppStatus nppiColorTwist32f_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** aTwist[3][4])
4 channel 8-bit unsigned color twist, with alpha copy.
- **NppStatus nppiColorTwist32f_8u_C4IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** aTwist[3][4])
4 channel 8-bit unsigned in place color twist, not affecting Alpha.
- **NppStatus nppiColorTwist32f_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** aTwist[3][4])

4 channel 8-bit unsigned color twist, not affecting Alpha.

- `NppStatus nppiColorTwist32f_8u_AC4IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[3][4]`)

4 channel 8-bit unsigned in place color twist, not affecting Alpha.

- `NppStatus nppiColorTwist32fC_8u_C4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[4][4]`, `const Npp32f aConstants[4]`)

4 channel 8-bit unsigned color twist with 4x4 matrix and constant vector addition.

- `NppStatus nppiColorTwist32fC_8u_C4IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[4][4]`, `const Npp32f aConstants[4]`)

4 channel 8-bit unsigned in place color twist with 4x4 matrix and an additional constant vector addition.

- `NppStatus nppiColorTwist32f_8u_P3R` (`const Npp8u *const pSrc[3]`, `int nSrcStep`, `Npp8u *const pDst[3]`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[3][4]`)

3 channel 8-bit unsigned planar color twist.

- `NppStatus nppiColorTwist32f_8u_IP3R` (`Npp8u *const pSrcDst[3]`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[3][4]`)

3 channel 8-bit unsigned planar in place color twist.

- `NppStatus nppiColorTwist32f_8s_C1R` (`const Npp8s *pSrc`, `int nSrcStep`, `Npp8s *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[3][4]`)

1 channel 8-bit signed color twist.

- `NppStatus nppiColorTwist32f_8s_C1IR` (`Npp8s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[3][4]`)

1 channel 8-bit signed in place color twist.

- `NppStatus nppiColorTwist32f_8s_C2R` (`const Npp8s *pSrc`, `int nSrcStep`, `Npp8s *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[3][4]`)

2 channel 8-bit signed color twist.

- `NppStatus nppiColorTwist32f_8s_C2IR` (`Npp8s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[3][4]`)

2 channel 8-bit signed in place color twist.

- `NppStatus nppiColorTwist32f_8s_C3R` (`const Npp8s *pSrc`, `int nSrcStep`, `Npp8s *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[3][4]`)

3 channel 8-bit signed color twist.

- `NppStatus nppiColorTwist32f_8s_C3IR` (`Npp8s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[3][4]`)

3 channel 8-bit signed in place color twist.

- `NppStatus nppiColorTwist32f_8s_C4R` (`const Npp8s *pSrc`, `int nSrcStep`, `Npp8s *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[3][4]`)

4 channel 8-bit signed color twist, with alpha copy.

- `NppStatus nppiColorTwist32f_8s_C4IR` (`Npp8s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
4 channel 8-bit signed in place color twist, not affecting Alpha.
- `NppStatus nppiColorTwist32f_8s_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
4 channel 8-bit signed color twist, not affecting Alpha.
- `NppStatus nppiColorTwist32f_8s_AC4IR` (`Npp8s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
4 channel 8-bit signed in place color twist, not affecting Alpha.
- `NppStatus nppiColorTwist32f_8s_P3R` (const `Npp8s` *const pSrc[3], int nSrcStep, `Npp8s` *const pDst[3], int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
3 channel 8-bit signed planar color twist.
- `NppStatus nppiColorTwist32f_8s_IP3R` (`Npp8s` *const pSrcDst[3], int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
3 channel 8-bit signed planar in place color twist.
- `NppStatus nppiColorTwist32f_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
1 channel 16-bit unsigned color twist.
- `NppStatus nppiColorTwist32f_16u_C1IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
1 channel 16-bit unsigned in place color twist.
- `NppStatus nppiColorTwist32f_16u_C2R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
2 channel 16-bit unsigned color twist.
- `NppStatus nppiColorTwist32f_16u_C2IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
2 channel 16-bit unsigned in place color twist.
- `NppStatus nppiColorTwist32f_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
3 channel 16-bit unsigned color twist.
- `NppStatus nppiColorTwist32f_16u_C3IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
3 channel 16-bit unsigned in place color twist.
- `NppStatus nppiColorTwist32f_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
4 channel 16-bit unsigned color twist, not affecting Alpha.
- `NppStatus nppiColorTwist32f_16u_AC4IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])
4 channel 16-bit unsigned in place color twist, not affecting Alpha.

- `NppStatus nppiColorTwist32f_16u_P3R` (const `Npp16u` *const `pSrc[3]`, int `nSrcStep`, `Npp16u` *const `pDst[3]`, int `nDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
3 channel 16-bit unsigned planar color twist.
- `NppStatus nppiColorTwist32f_16u_IP3R` (`Npp16u` *const `pSrcDst[3]`, int `nSrcDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
3 channel 16-bit unsigned planar in place color twist.
- `NppStatus nppiColorTwist32f_16s_C1R` (const `Npp16s` *`pSrc`, int `nSrcStep`, `Npp16s` *`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
1 channel 16-bit signed color twist.
- `NppStatus nppiColorTwist32f_16s_C1IR` (`Npp16s` *`pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
1 channel 16-bit signed in place color twist.
- `NppStatus nppiColorTwist32f_16s_C2R` (const `Npp16s` *`pSrc`, int `nSrcStep`, `Npp16s` *`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
2 channel 16-bit signed color twist.
- `NppStatus nppiColorTwist32f_16s_C2IR` (`Npp16s` *`pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
2 channel 16-bit signed in place color twist.
- `NppStatus nppiColorTwist32f_16s_C3R` (const `Npp16s` *`pSrc`, int `nSrcStep`, `Npp16s` *`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
3 channel 16-bit signed color twist.
- `NppStatus nppiColorTwist32f_16s_C3IR` (`Npp16s` *`pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
3 channel 16-bit signed in place color twist.
- `NppStatus nppiColorTwist32f_16s_AC4R` (const `Npp16s` *`pSrc`, int `nSrcStep`, `Npp16s` *`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
4 channel 16-bit signed color twist, not affecting Alpha.
- `NppStatus nppiColorTwist32f_16s_AC4IR` (`Npp16s` *`pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
4 channel 16-bit signed in place color twist, not affecting Alpha.
- `NppStatus nppiColorTwist32f_16s_P3R` (const `Npp16s` *const `pSrc[3]`, int `nSrcStep`, `Npp16s` *const `pDst[3]`, int `nDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
3 channel 16-bit signed planar color twist.
- `NppStatus nppiColorTwist32f_16s_IP3R` (`Npp16s` *const `pSrcDst[3]`, int `nSrcDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)
3 channel 16-bit signed planar in place color twist.
- `NppStatus nppiColorTwist_32f_C1R` (const `Npp32f` *`pSrc`, int `nSrcStep`, `Npp32f` *`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`, const `Npp32f` `aTwist[3][4]`)

1 channel 32-bit floating point color twist.

- `NppStatus nppiColorTwist_32f_C1IR (Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

1 channel 32-bit floating point in place color twist.

- `NppStatus nppiColorTwist_32f_C2R (const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

2 channel 32-bit floating point color twist.

- `NppStatus nppiColorTwist_32f_C2IR (Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

2 channel 32-bit floating point in place color twist.

- `NppStatus nppiColorTwist_32f_C3R (const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 32-bit floating point color twist.

- `NppStatus nppiColorTwist_32f_C3IR (Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 32-bit floating point in place color twist.

- `NppStatus nppiColorTwist_32f_C4R (const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 32-bit floating point color twist, with alpha copy.

- `NppStatus nppiColorTwist_32f_C4IR (Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 32-bit floating point in place color twist, not affecting Alpha.

- `NppStatus nppiColorTwist_32f_AC4R (const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 32-bit floating point color twist, not affecting Alpha.

- `NppStatus nppiColorTwist_32f_AC4IR (Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 32-bit floating point in place color twist, not affecting Alpha.

- `NppStatus nppiColorTwist_32fC_C4R (const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[4][4], const Npp32f aConstants[4])`

4 channel 32-bit floating point color twist with 4x4 matrix and constant vector addition.

- `NppStatus nppiColorTwist_32fC_C4IR (Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[4][4], const Npp32f aConstants[4])`

4 channel 32-bit floating point in place color twist with 4x4 matrix and an additional constant vector addition.

- `NppStatus nppiColorTwist_32f_P3R (const Npp32f *const pSrc[3], int nSrcStep, Npp32f *const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 32-bit floating point planar color twist.

- `NppStatus nppiColorTwist_32f_IP3R` (`Npp32f *const pSrcDst[3]`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f aTwist[3][4]`)

3 channel 32-bit floating point planar in place color twist.

ColorLUT

Perform image color processing using members of various types of color look up tables.

- `NppStatus nppiLUT_8u_C1R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues`, `const Npp32s *pLevels`, `int nLevels`)

8-bit unsigned look-up-table color conversion.

- `NppStatus nppiLUT_8u_C1IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues`, `const Npp32s *pLevels`, `int nLevels`)

8-bit unsigned look-up-table in place color conversion.

- `NppStatus nppiLUT_8u_C3R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)

3 channel 8-bit unsigned look-up-table color conversion.

- `NppStatus nppiLUT_8u_C3IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)

3 channel 8-bit unsigned look-up-table in place color conversion.

- `NppStatus nppiLUT_8u_C4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[4]`, `const Npp32s *pLevels[4]`, `int nLevels[4]`)

4 channel 8-bit unsigned look-up-table color conversion.

- `NppStatus nppiLUT_8u_C4IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[4]`, `const Npp32s *pLevels[4]`, `int nLevels[4]`)

4 channel 8-bit unsigned look-up-table in place color conversion.

- `NppStatus nppiLUT_8u_AC4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)

4 channel 8-bit unsigned look-up-table color conversion, not affecting Alpha.

- `NppStatus nppiLUT_8u_AC4IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)

4 channel 8-bit unsigned look-up-table in place color conversion, not affecting Alpha.

- `NppStatus nppiLUT_16u_C1R` (`const Npp16u *pSrc`, `int nSrcStep`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues`, `const Npp32s *pLevels`, `int nLevels`)

16-bit unsigned look-up-table color conversion.

- `NppStatus nppiLUT_16u_C1IR` (`Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues`, `const Npp32s *pLevels`, `int nLevels`)

16-bit unsigned look-up-table in place color conversion.

- `NppStatus nppiLUT_16u_C3R` (`const Npp16u *pSrc`, `int nSrcStep`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)

3 channel 16-bit unsigned look-up-table color conversion.

- `NppStatus nppiLUT_16u_C3IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

3 channel 16-bit unsigned look-up-table in place color conversion.

- `NppStatus nppiLUT_16u_C4R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues[4], const Npp32s *pLevels[4], int nLevels[4])`

4 channel 16-bit unsigned look-up-table color conversion.

- `NppStatus nppiLUT_16u_C4IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues[4], const Npp32s *pLevels[4], int nLevels[4])`

4 channel 16-bit unsigned look-up-table in place color conversion.

- `NppStatus nppiLUT_16u_AC4R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

- `NppStatus nppiLUT_16u_AC4IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.

- `NppStatus nppiLUT_16s_C1R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues, const Npp32s *pLevels, int nLevels)`

16-bit signed look-up-table color conversion.

- `NppStatus nppiLUT_16s_C1IR (Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues, const Npp32s *pLevels, int nLevels)`

16-bit signed look-up-table in place color conversion.

- `NppStatus nppiLUT_16s_C3R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

3 channel 16-bit signed look-up-table color conversion.

- `NppStatus nppiLUT_16s_C3IR (Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

3 channel 16-bit signed look-up-table in place color conversion.

- `NppStatus nppiLUT_16s_C4R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues[4], const Npp32s *pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table color conversion.

- `NppStatus nppiLUT_16s_C4IR (Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues[4], const Npp32s *pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table in place color conversion.

- `NppStatus nppiLUT_16s_AC4R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.

- **NppStatus** **nppiLUT_16s_AC4IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[3], const **Npp32s** *pLevels[3], int nLevels[3])
4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.
- **NppStatus** **nppiLUT_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** *pValues, const **Npp32f** *pLevels, int nLevels)
32-bit floating point look-up-table color conversion.
- **NppStatus** **nppiLUT_32f_C1IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** *pValues, const **Npp32f** *pLevels, int nLevels)
32-bit floating point look-up-table in place color conversion.
- **NppStatus** **nppiLUT_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** *pValues[3], const **Npp32f** *pLevels[3], int nLevels[3])
3 channel 32-bit floating point look-up-table color conversion.
- **NppStatus** **nppiLUT_32f_C3IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** *pValues[3], const **Npp32f** *pLevels[3], int nLevels[3])
3 channel 32-bit floating point look-up-table in place color conversion.
- **NppStatus** **nppiLUT_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** *pValues[4], const **Npp32f** *pLevels[4], int nLevels[4])
4 channel 32-bit floating point look-up-table color conversion.
- **NppStatus** **nppiLUT_32f_C4IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** *pValues[4], const **Npp32f** *pLevels[4], int nLevels[4])
4 channel 32-bit floating point look-up-table in place color conversion.
- **NppStatus** **nppiLUT_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** *pValues[3], const **Npp32f** *pLevels[3], int nLevels[3])
4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.
- **NppStatus** **nppiLUT_32f_AC4IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** *pValues[3], const **Npp32f** *pLevels[3], int nLevels[3])
4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

ColorLUT_Linear

Perform image color processing using linear interpolation between members of various types of color look up tables.

- **NppStatus** **nppiLUT_Linear_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues, const **Npp32s** *pLevels, int nLevels)
8-bit unsigned linear interpolated look-up-table color conversion.
- **NppStatus** **nppiLUT_Linear_8u_C1IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues, const **Npp32s** *pLevels, int nLevels)
8-bit unsigned linear interpolated look-up-table in place color conversion.

- **NppStatus nppiLUT_Linear_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[3], const **Npp32s** *pLevels[3], int nLevels[3])
3 channel 8-bit unsigned linear interpolated look-up-table color conversion.
- **NppStatus nppiLUT_Linear_8u_C3IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[3], const **Npp32s** *pLevels[3], int nLevels[3])
3 channel 8-bit unsigned linear interpolated look-up-table in place color conversion.
- **NppStatus nppiLUT_Linear_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[4], const **Npp32s** *pLevels[4], int nLevels[4])
4 channel 8-bit unsigned linear interpolated look-up-table color conversion.
- **NppStatus nppiLUT_Linear_8u_C4IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[4], const **Npp32s** *pLevels[4], int nLevels[4])
4 channel 8-bit unsigned linear interpolated look-up-table in place color conversion.
- **NppStatus nppiLUT_Linear_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[3], const **Npp32s** *pLevels[3], int nLevels[3])
4 channel 8-bit unsigned linear interpolated look-up-table color conversion, not affecting Alpha.
- **NppStatus nppiLUT_Linear_8u_AC4IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[3], const **Npp32s** *pLevels[3], int nLevels[3])
4 channel 8-bit unsigned linear interpolated look-up-table in place color conversion, not affecting Alpha.
- **NppStatus nppiLUT_Linear_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues, const **Npp32s** *pLevels, int nLevels)
16-bit unsigned look-up-table color conversion.
- **NppStatus nppiLUT_Linear_16u_C1IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues, const **Npp32s** *pLevels, int nLevels)
16-bit unsigned look-up-table in place color conversion.
- **NppStatus nppiLUT_Linear_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[3], const **Npp32s** *pLevels[3], int nLevels[3])
3 channel 16-bit unsigned look-up-table color conversion.
- **NppStatus nppiLUT_Linear_16u_C3IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[3], const **Npp32s** *pLevels[3], int nLevels[3])
3 channel 16-bit unsigned look-up-table in place color conversion.
- **NppStatus nppiLUT_Linear_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[4], const **Npp32s** *pLevels[4], int nLevels[4])
4 channel 16-bit unsigned look-up-table color conversion.
- **NppStatus nppiLUT_Linear_16u_C4IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[4], const **Npp32s** *pLevels[4], int nLevels[4])
4 channel 16-bit unsigned look-up-table in place color conversion.
- **NppStatus nppiLUT_Linear_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pValues[3], const **Npp32s** *pLevels[3], int nLevels[3])
4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

- `NppStatus nppiLUT_Linear_16u_AC4IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[3], const `Npp32s` *pLevels[3], int nLevels[3])
4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.
- `NppStatus nppiLUT_Linear_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues, const `Npp32s` *pLevels, int nLevels)
16-bit signed look-up-table color conversion.
- `NppStatus nppiLUT_Linear_16s_C1IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues, const `Npp32s` *pLevels, int nLevels)
16-bit signed look-up-table in place color conversion.
- `NppStatus nppiLUT_Linear_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[3], const `Npp32s` *pLevels[3], int nLevels[3])
3 channel 16-bit signed look-up-table color conversion.
- `NppStatus nppiLUT_Linear_16s_C3IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[3], const `Npp32s` *pLevels[3], int nLevels[3])
3 channel 16-bit signed look-up-table in place color conversion.
- `NppStatus nppiLUT_Linear_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[4], const `Npp32s` *pLevels[4], int nLevels[4])
4 channel 16-bit signed look-up-table color conversion.
- `NppStatus nppiLUT_Linear_16s_C4IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[4], const `Npp32s` *pLevels[4], int nLevels[4])
4 channel 16-bit signed look-up-table in place color conversion.
- `NppStatus nppiLUT_Linear_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[3], const `Npp32s` *pLevels[3], int nLevels[3])
4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.
- `NppStatus nppiLUT_Linear_16s_AC4IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[3], const `Npp32s` *pLevels[3], int nLevels[3])
4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.
- `NppStatus nppiLUT_Linear_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pValues, const `Npp32f` *pLevels, int nLevels)
32-bit floating point look-up-table color conversion.
- `NppStatus nppiLUT_Linear_32f_C1IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` *pValues, const `Npp32f` *pLevels, int nLevels)
32-bit floating point look-up-table in place color conversion.
- `NppStatus nppiLUT_Linear_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pValues[3], const `Npp32f` *pLevels[3], int nLevels[3])
3 channel 32-bit floating point look-up-table color conversion.
- `NppStatus nppiLUT_Linear_32f_C3IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` *pValues[3], const `Npp32f` *pLevels[3], int nLevels[3])

3 channel 32-bit floating point look-up-table in place color conversion.

- `NppStatus nppiLUT_Linear_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pValues[4], const `Npp32f` *pLevels[4], int nLevels[4])

4 channel 32-bit floating point look-up-table color conversion.

- `NppStatus nppiLUT_Linear_32f_C4IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` *pValues[4], const `Npp32f` *pLevels[4], int nLevels[4])

4 channel 32-bit floating point look-up-table in place color conversion.

- `NppStatus nppiLUT_Linear_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pValues[3], const `Npp32f` *pLevels[3], int nLevels[3])

4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.

- `NppStatus nppiLUT_Linear_32f_AC4IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` *pValues[3], const `Npp32f` *pLevels[3], int nLevels[3])

4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

ColorLUT_Cubic

Perform image color processing using linear interpolation between members of various types of color look up tables.

- `NppStatus nppiLUT_Cubic_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues, const `Npp32s` *pLevels, int nLevels)

8-bit unsigned cubic interpolated look-up-table color conversion.

- `NppStatus nppiLUT_Cubic_8u_C1IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues, const `Npp32s` *pLevels, int nLevels)

8-bit unsigned cubic interpolated look-up-table in place color conversion.

- `NppStatus nppiLUT_Cubic_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[3], const `Npp32s` *pLevels[3], int nLevels[3])

3 channel 8-bit unsigned cubic interpolated look-up-table color conversion.

- `NppStatus nppiLUT_Cubic_8u_C3IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[3], const `Npp32s` *pLevels[3], int nLevels[3])

3 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion.

- `NppStatus nppiLUT_Cubic_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[4], const `Npp32s` *pLevels[4], int nLevels[4])

4 channel 8-bit unsigned cubic interpolated look-up-table color conversion.

- `NppStatus nppiLUT_Cubic_8u_C4IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[4], const `Npp32s` *pLevels[4], int nLevels[4])

4 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion.

- `NppStatus nppiLUT_Cubic_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pValues[3], const `Npp32s` *pLevels[3], int nLevels[3])

4 channel 8-bit unsigned cubic interpolated look-up-table color conversion, not affecting Alpha.

- `NppStatus nppiLUT_Cubic_8u_AC4IR (Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

4 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion, not affecting Alpha.

- `NppStatus nppiLUT_Cubic_16u_C1R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues, const Npp32s *pLevels, int nLevels)`

16-bit unsigned look-up-table color conversion.

- `NppStatus nppiLUT_Cubic_16u_C1IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues, const Npp32s *pLevels, int nLevels)`

16-bit unsigned look-up-table in place color conversion.

- `NppStatus nppiLUT_Cubic_16u_C3R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

3 channel 16-bit unsigned look-up-table color conversion.

- `NppStatus nppiLUT_Cubic_16u_C3IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

3 channel 16-bit unsigned look-up-table in place color conversion.

- `NppStatus nppiLUT_Cubic_16u_C4R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues[4], const Npp32s *pLevels[4], int nLevels[4])`

4 channel 16-bit unsigned look-up-table color conversion.

- `NppStatus nppiLUT_Cubic_16u_C4IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues[4], const Npp32s *pLevels[4], int nLevels[4])`

4 channel 16-bit unsigned look-up-table in place color conversion.

- `NppStatus nppiLUT_Cubic_16u_AC4R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

- `NppStatus nppiLUT_Cubic_16u_AC4IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.

- `NppStatus nppiLUT_Cubic_16s_C1R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues, const Npp32s *pLevels, int nLevels)`

16-bit signed look-up-table color conversion.

- `NppStatus nppiLUT_Cubic_16s_C1IR (Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s *pValues, const Npp32s *pLevels, int nLevels)`

16-bit signed look-up-table in place color conversion.

- `NppStatus nppiLUT_Cubic_16s_C3R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32s *pValues[3], const Npp32s *pLevels[3], int nLevels[3])`

3 channel 16-bit signed look-up-table color conversion.

- `NppStatus nppiLUT_Cubic_16s_C3IR` (`Npp16s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)
3 channel 16-bit signed look-up-table in place color conversion.
- `NppStatus nppiLUT_Cubic_16s_C4R` (`const Npp16s *pSrc`, `int nSrcStep`, `Npp16s *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[4]`, `const Npp32s *pLevels[4]`, `int nLevels[4]`)
4 channel 16-bit signed look-up-table color conversion.
- `NppStatus nppiLUT_Cubic_16s_C4IR` (`Npp16s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[4]`, `const Npp32s *pLevels[4]`, `int nLevels[4]`)
4 channel 16-bit signed look-up-table in place color conversion.
- `NppStatus nppiLUT_Cubic_16s_AC4R` (`const Npp16s *pSrc`, `int nSrcStep`, `Npp16s *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)
4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.
- `NppStatus nppiLUT_Cubic_16s_AC4IR` (`Npp16s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)
4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.
- `NppStatus nppiLUT_Cubic_32f_C1R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues`, `const Npp32f *pLevels`, `int nLevels`)
32-bit floating point look-up-table color conversion.
- `NppStatus nppiLUT_Cubic_32f_C1IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues`, `const Npp32f *pLevels`, `int nLevels`)
32-bit floating point look-up-table in place color conversion.
- `NppStatus nppiLUT_Cubic_32f_C3R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[3]`, `const Npp32f *pLevels[3]`, `int nLevels[3]`)
3 channel 32-bit floating point look-up-table color conversion.
- `NppStatus nppiLUT_Cubic_32f_C3IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[3]`, `const Npp32f *pLevels[3]`, `int nLevels[3]`)
3 channel 32-bit floating point look-up-table in place color conversion.
- `NppStatus nppiLUT_Cubic_32f_C4R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[4]`, `const Npp32f *pLevels[4]`, `int nLevels[4]`)
4 channel 32-bit floating point look-up-table color conversion.
- `NppStatus nppiLUT_Cubic_32f_C4IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[4]`, `const Npp32f *pLevels[4]`, `int nLevels[4]`)
4 channel 32-bit floating point look-up-table in place color conversion.
- `NppStatus nppiLUT_Cubic_32f_AC4R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[3]`, `const Npp32f *pLevels[3]`, `int nLevels[3]`)
4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.
- `NppStatus nppiLUT_Cubic_32f_AC4IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[3]`, `const Npp32f *pLevels[3]`, `int nLevels[3]`)
4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

ColorLUT_Trilinear

Perform image color processing using 3D trilinear interpolation between members of various types of color look up tables.

- **NppStatus nppiLUT_Trilinear_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32u** *pValues, **Npp8u** *pLevels[3], int aLevels[3])
Four channel 8-bit unsigned 3D trilinear interpolated look-up-table color conversion, with alpha copy.
- **NppStatus nppiLUT_Trilinear_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32u** *pValues, **Npp8u** *pLevels[3], int aLevels[3])
Four channel 8-bit unsigned 3D trilinear interpolated look-up-table color conversion, not affecting alpha.
- **NppStatus nppiLUT_Trilinear_8u_AC4IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32u** *pValues, **Npp8u** *pLevels[3], int aLevels[3])
Four channel 8-bit unsigned 3D trilinear interpolated look-up-table in place color conversion, not affecting alpha.

ColorLUTPalette

Perform image color processing using various types of bit range restricted palette color look up tables.

- **NppStatus nppiLUTPalette_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pTable, int nBitSize)
One channel 8-bit unsigned bit range restricted palette look-up-table color conversion.
- **NppStatus nppiLUTPalette_8u24u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pTable, int nBitSize)
One channel 8-bit unsigned bit range restricted 24-bit palette look-up-table color conversion with 24-bit destination output per pixel.
- **NppStatus nppiLUTPalette_8u32u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32u** *pTable, int nBitSize)
One channel 8-bit unsigned bit range restricted 32-bit palette look-up-table color conversion with 32-bit destination output per pixel.
- **NppStatus nppiLUTPalette_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pTables[3], int nBitSize)
Three channel 8-bit unsigned bit range restricted palette look-up-table color conversion.
- **NppStatus nppiLUTPalette_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pTables[4], int nBitSize)
Four channel 8-bit unsigned bit range restricted palette look-up-table color conversion.
- **NppStatus nppiLUTPalette_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pTables[3], int nBitSize)
Four channel 8-bit unsigned bit range restricted palette look-up-table color conversion, not affecting Alpha.
- **NppStatus nppiLUTPalette_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** *pTable, int nBitSize)

One channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

- **NppStatus nppiLUTPalette_16u8u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pTable, int nBitSize)

One channel 16-bit unsigned bit range restricted 8-bit unsigned palette look-up-table color conversion with 8-bit unsigned destination output per pixel.

- **NppStatus nppiLUTPalette_16u24u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pTable, int nBitSize)

One channel 16-bit unsigned bit range restricted 24-bit unsigned palette look-up-table color conversion with 24-bit unsigned destination output per pixel.

- **NppStatus nppiLUTPalette_16u32u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32u** *pTable, int nBitSize)

One channel 16-bit unsigned bit range restricted 32-bit palette look-up-table color conversion with 32-bit unsigned destination output per pixel.

- **NppStatus nppiLUTPalette_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** *pTables[3], int nBitSize)

Three channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

- **NppStatus nppiLUTPalette_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** *pTables[4], int nBitSize)

Four channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

- **NppStatus nppiLUTPalette_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** *pTables[3], int nBitSize)

Four channel 16-bit unsigned bit range restricted palette look-up-table color conversion, not affecting Alpha.

- **NppStatus nppiLUTPaletteSwap_8u_C3A0C4R** (const **Npp8u** *pSrc, int nSrcStep, int nAlphaValue, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pTables[3], int nBitSize)

Three channel 8-bit unsigned source bit range restricted palette look-up-table color conversion to four channel 8-bit unsigned destination output with alpha.

- **NppStatus nppiLUTPaletteSwap_16u_C3A0C4R** (const **Npp16u** *pSrc, int nSrcStep, int nAlphaValue, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** *pTables[3], int nBitSize)

Three channel 16-bit unsigned source bit range restricted palette look-up-table color conversion to four channel 16-bit unsigned destination output with alpha.

7.48.1 Detailed Description

Routines for performing image color manipulation.

7.48.2 Function Documentation

7.48.2.1 **NppStatus nppiColorTwist32f_16s_AC4IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** aTwist[3][4])

4 channel 16-bit signed in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst in place packed pixel format image pointer.
nSrcDstStep in place packed pixel format image line step.
oSizeROI [Region-of-Interest \(ROI\)](#).
aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.2 NppStatus nppiColorTwist32f_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

4 channel 16-bit signed color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.3 NppStatus nppiColorTwist32f_16s_C11R (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

1 channel 16-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.
nSrcDstStep in place packed pixel format image line step.
oSizeROI [Region-of-Interest \(ROI\)](#).
aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.4 NppStatus nppiColorTwist32f_16s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

1 channel 16-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.5 NppStatus nppiColorTwist32f_16s_C2IR (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

2 channel 16-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI Region-of-Interest (ROI).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.6 NppStatus nppiColorTwist32f_16s_C2R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

2 channel 16-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.7 NppStatus nppiColorTwist32f_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 16-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.8 NppStatus nppiColorTwist32f_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 16-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.9 NppStatus nppiColorTwist32f_16s_IP3R (Npp16s *const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 16-bit signed planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place planar pixel format image pointer array, one pointer per plane.

nSrcDstStep in place planar pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.10 NppStatus nppiColorTwist32f_16s_P3R (const Npp16s *const pSrc[3], int nSrcStep, Npp16s *const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 16-bit signed planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.11 NppStatus nppiColorTwist32f_16u_AC4IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

4 channel 16-bit unsigned in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.12 `NppStatus nppiColorTwist32f_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 16-bit unsigned color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.13 `NppStatus nppiColorTwist32f_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

1 channel 16-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.14 `NppStatus nppiColorTwist32f_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

1 channel 16-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.15 NppStatus nppiColorTwist32f_16u_C2IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

2 channel 16-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.
nSrcDstStep in place packed pixel format image line step.
oSizeROI [Region-of-Interest \(ROI\)](#).
aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.16 NppStatus nppiColorTwist32f_16u_C2R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

2 channel 16-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI [Region-of-Interest \(ROI\)](#).
aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.17 `NppStatus nppiColorTwist32f_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 16-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.18 `NppStatus nppiColorTwist32f_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 16-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.19 `NppStatus nppiColorTwist32f_16u_IP3R (Npp16u *const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 16-bit unsigned planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place planar pixel format image pointer array, one pointer per plane.

nSrcDstStep in place planar pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.20 `NppStatus nppiColorTwist32f_16u_P3R (const Npp16u *const pSrc[3], int nSrcStep, Npp16u *const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 16-bit unsigned planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.21 `NppStatus nppiColorTwist32f_8s_AC4IR (Npp8s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 8-bit signed in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI Region-of-Interest (ROI).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.22 `NppStatus nppiColorTwist32f_8s_AC4R (const Npp8s *pSrc, int nSrcStep, Npp8s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 8-bit signed color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.23 **NppStatus nppiColorTwist32f_8s_C1IR (Npp8s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])**

1 channel 8-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.
nSrcDstStep in place packed pixel format image line step.
oSizeROI Region-of-Interest (ROI).
aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.24 **NppStatus nppiColorTwist32f_8s_C1R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])**

1 channel 8-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.25 `NppStatus nppiColorTwist32f_8s_C2IR (Npp8s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

2 channel 8-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.26 `NppStatus nppiColorTwist32f_8s_C2R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

2 channel 8-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.27 `NppStatus nppiColorTwist32f_8s_C3IR (Npp8s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 8-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.28 **NppStatus nppiColorTwist32f_8s_C3R** (const Npp8s * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

3 channel 8-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.29 **NppStatus nppiColorTwist32f_8s_C4IR** (Npp8s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

4 channel 8-bit signed in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is unmodified.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI Region-of-Interest (ROI).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.30 **NppStatus nppiColorTwist32f_8s_C4R** (const Npp8s * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

4 channel 8-bit signed color twist, with alpha copy.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is copied unmodified from the source pixel to the destination pixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.31 NppStatus nppiColorTwist32f_8s_IP3R (Npp8s *const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 8-bit signed planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place planar pixel format image pointer array, one pointer per plane.

nSrcDstStep in place planar pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.32 NppStatus nppiColorTwist32f_8s_P3R (const Npp8s *const pSrc[3], int nSrcStep, Npp8s *const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 8-bit signed planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.33 `NppStatus nppiColorTwist32f_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 8-bit unsigned in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.34 `NppStatus nppiColorTwist32f_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 8-bit unsigned color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.35 `NppStatus nppiColorTwist32f_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

1 channel 8-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.36 `NppStatus nppiColorTwist32f_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

1 channel 8-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.37 `NppStatus nppiColorTwist32f_8u_C2IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

2 channel 8-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.38 `NppStatus nppiColorTwist32f_8u_C2R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

2 channel 8-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.39 `NppStatus nppiColorTwist32f_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 8-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI Region-of-Interest (ROI).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.40 `NppStatus nppiColorTwist32f_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 8-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.41 NppStatus nppiColorTwist32f_8u_C4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

4 channel 8-bit unsigned in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is unmodified.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.42 NppStatus nppiColorTwist32f_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

4 channel 8-bit unsigned color twist, with alpha copy.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is copied unmodified from the source pixel to the destination pixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.43 NppStatus nppiColorTwist32f_8u_IP3R (Npp8u *const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 8-bit unsigned planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place planar pixel format image pointer array, one pointer per plane.

nSrcDstStep in place planar pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.44 NppStatus nppiColorTwist32f_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 8-bit unsigned planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.45 NppStatus nppiColorTwist32fC_8u_C4IR (Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[4][4], const Npp32f aConstants[4])

4 channel 8-bit unsigned in place color twist with 4x4 matrix and an additional constant vector addition.

An input 4x4 color twist matrix with floating-point coefficient values with an additional constant vector addition is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] + aTwist[0][3] * src[3] + aTwist[0][4] * aConstants[0]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] + aTwist[1][3] * src[3] + aTwist[1][4] * aConstants[0]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] + aTwist[2][3] * src[3] + aTwist[2][4] * aConstants[0]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] + aTwist[3][3] * src[3] + aTwist[3][4] * aConstants[0]
```

Parameters:

pSrcDst in place packed pixel format image pointer.
nSrcDstStep in place packed pixel format image line step.
oSizeROI [Region-of-Interest \(ROI\)](#).
aTwist The color twist matrix with floating-point coefficient values.
aConstants fixed size array of constant values, one per channel..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.46 NppStatus nppiColorTwist32fC_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[4][4], const Npp32f aConstants[4])

4 channel 8-bit unsigned color twist with 4x4 matrix and constant vector addition.

An input 4x4 color twist matrix with floating-point coefficient values with an additional constant vector addition is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] + aTwist[0][3] * src[3] + aConstants[0]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] + aTwist[1][3] * src[3] + aConstants[1]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] + aTwist[2][3] * src[3] + aConstants[2]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] + aTwist[3][3] * src[3] + aConstants[3]
```

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aTwist The color twist matrix with floating-point coefficient values.
aConstants fixed size array of constant values, one per channel..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.47 NppStatus nppiColorTwist_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

4 channel 32-bit floating point in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.48 `NppStatus nppiColorTwist_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 32-bit floating point color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.49 `NppStatus nppiColorTwist_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

1 channel 32-bit floating point in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.50 NppStatus nppiColorTwist_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

1 channel 32-bit floating point color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.51 NppStatus nppiColorTwist_32f_C2IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

2 channel 32-bit floating point in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI Region-of-Interest (ROI).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.52 NppStatus nppiColorTwist_32f_C2R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

2 channel 32-bit floating point color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.53 `NppStatus nppiColorTwist_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 32-bit floating point in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.54 `NppStatus nppiColorTwist_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 32-bit floating point color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.55 **NppStatus nppiColorTwist_32f_C4IR** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

4 channel 32-bit floating point in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not modified.

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.56 **NppStatus nppiColorTwist_32f_C4R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

4 channel 32-bit floating point color twist, with alpha copy.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is copied unmodified from the source pixel to the destination pixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.57 **NppStatus nppiColorTwist_32f_IP3R** (Npp32f *const *pSrcDst*[3], int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

3 channel 32-bit floating point planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrcDst in place planar pixel format image pointer array, one pointer per plane.

nSrcDstStep in place planar pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.58 `NppStatus nppiColorTwist_32f_P3R (const Npp32f *const pSrc[3], int nSrcStep, Npp32f *const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 32-bit floating point planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.59 `NppStatus nppiColorTwist_32fC_C4IR (Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[4][4], const Npp32f aConstants[4])`

4 channel 32-bit floating point in place color twist with 4x4 matrix and an additional constant vector addition.

An input 4x4 color twist matrix with floating-point coefficient values with an additional constant vector addition is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] + aTwist[0][3] * src[3] + aTwist[0][4] * aConstants[0]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] + aTwist[1][3] * src[3] + aTwist[1][4] * aConstants[0]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] + aTwist[2][3] * src[3] + aTwist[2][4] * aConstants[0]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] + aTwist[3][3] * src[3] + aTwist[3][4] * aConstants[0]
```

Parameters:

pSrcDst in place packed pixel format image pointer.

nSrcDstStep in place packed pixel format image line step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aTwist The color twist matrix with floating-point coefficient values.

aConstants fixed size array of constant values, one per channel..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.60 NppStatus nppiColorTwist_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[4][4], const Npp32f aConstants[4])

4 channel 32-bit floating point color twist with 4x4 matrix and constant vector addition.

An input 4x4 color twist matrix with floating-point coefficient values with an additional constant vector addition is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] + aTwist[0][3] * src[3] + aConstants[0]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] + aTwist[1][3] * src[3] + aConstants[1]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] + aTwist[2][3] * src[3] + aConstants[2]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] + aTwist[3][3] * src[3] + aConstants[3]
```

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aTwist The color twist matrix with floating-point coefficient values.
aConstants fixed size array of constant values, one per channel..

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.48.2.61 NppStatus nppiLUT_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])

4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.62 `NppStatus nppiLUT_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.63 `NppStatus nppiLUT_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.64 `NppStatus nppiLUT_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.65 `NppStatus nppiLUT_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.66 `NppStatus nppiLUT_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.67 `NppStatus nppiLUT_16s_C4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.68 `NppStatus nppiLUT_16s_C4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.69 `NppStatus nppiLUT_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.70 `NppStatus nppiLUT_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.71 `NppStatus nppiLUT_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.72 `NppStatus nppiLUT_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.73 `NppStatus nppiLUT_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.74 `NppStatus nppiLUT_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.75 `NppStatus nppiLUT_16u_C4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.76 **NppStatus nppiLUT_16u_C4R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[4], const Npp32s * *pLevels*[4], int *nLevels*[4])

4 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.77 **NppStatus nppiLUT_32f_AC4IR** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f * *pValues*[3], const Npp32f * *pLevels*[3], int *nLevels*[3])

4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.78 `NppStatus nppiLUT_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pValues[3], const Npp32f * pLevels[3], int nLevels[3])`

4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.79 `NppStatus nppiLUT_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f * pValues, const Npp32f * pLevels, int nLevels)`

32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.80 `NppStatus nppiLUT_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pValues, const Npp32f * pLevels, int nLevels)`

32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.81 `NppStatus nppiLUT_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f * pValues[3], const Npp32f * pLevels[3], int nLevels[3])`

3 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.82 `NppStatus nppiLUT_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pValues[3], const Npp32f * pLevels[3], int nLevels[3])`

3 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.83 `NppStatus nppiLUT_32f_C4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f * pValues[4], const Npp32f * pLevels[4], int nLevels[4])`

4 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.84 **NppStatus nppiLUT_32f_C4R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pValues*[4], const Npp32f * *pLevels*[4], int *nLevels*[4])

4 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.85 **NppStatus nppiLUT_8u_AC4IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[3], const Npp32s * *pLevels*[3], int *nLevels*[3])

4 channel 8-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.86 `NppStatus nppiLUT_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 8-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 256.

7.48.2.87 `NppStatus nppiLUT_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

8-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 256.

7.48.2.88 `NppStatus nppiLUT_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.89 `NppStatus nppiLUT_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 8-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.90 `NppStatus nppiLUT_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 256.

7.48.2.91 `NppStatus nppiLUT_8u_C4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 8-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 256.

7.48.2.92 NppStatus nppiLUT_8u_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[4], const Npp32s * *pLevels*[4], int *nLevels*[4])

4 channel 8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.93 NppStatus nppiLUT_Cubic_16s_AC4IR (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[3], const Npp32s * *pLevels*[3], int *nLevels*[3])

4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.94 `NppStatus nppiLUT_Cubic_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.95 `NppStatus nppiLUT_Cubic_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.96 `NppStatus nppiLUT_Cubic_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.97 `NppStatus nppiLUT_Cubic_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.98 `NppStatus nppiLUT_Cubic_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.99 `NppStatus nppiLUT_Cubic_16s_C4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.100 `NppStatus nppiLUT_Cubic_16s_C4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.101 `NppStatus nppiLUT_Cubic_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.102 **NppStatus nppiLUT_Cubic_16u_AC4R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[3], const Npp32s * *pLevels*[3], int *nLevels*[3])

4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.103 **NppStatus nppiLUT_Cubic_16u_C1IR** (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*, const Npp32s * *pLevels*, int *nLevels*)

16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.104 NppStatus nppiLUT_Cubic_16u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*, const Npp32s * *pLevels*, int *nLevels*)

16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.105 NppStatus nppiLUT_Cubic_16u_C3IR (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[3], const Npp32s * *pLevels*[3], int *nLevels*[3])

3 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.106 NppStatus nppiLUT_Cubic_16u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[3], const Npp32s * *pLevels*[3], int *nLevels*[3])

3 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.107 NppStatus nppiLUT_Cubic_16u_C4IR (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[4], const Npp32s * *pLevels*[4], int *nLevels*[4])

4 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.108 NppStatus nppiLUT_Cubic_16u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[4], const Npp32s * *pLevels*[4], int *nLevels*[4])

4 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.109 NppStatus nppiLUT_Cubic_32f_AC4IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f * *pValues*[3], const Npp32f * *pLevels*[3], int *nLevels*[3])

4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.110 `NppStatus nppiLUT_Cubic_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pValues[3], const Npp32f * pLevels[3], int nLevels[3])`

4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.111 `NppStatus nppiLUT_Cubic_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f * pValues, const Npp32f * pLevels, int nLevels)`

32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.112 `NppStatus nppiLUT_Cubic_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pValues, const Npp32f * pLevels, int nLevels)`

32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.113 `NppStatus nppiLUT_Cubic_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f * pValues[3], const Npp32f * pLevels[3], int nLevels[3])`

3 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.114 `NppStatus nppiLUT_Cubic_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pValues[3], const Npp32f * pLevels[3], int nLevels[3])`

3 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.115 `NppStatus nppiLUT_Cubic_32f_C4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f * pValues[4], const Npp32f * pLevels[4], int nLevels[4])`

4 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.116 `NppStatus nppiLUT_Cubic_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pValues[4], const Npp32f * pLevels[4], int nLevels[4])`

4 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.117 `NppStatus nppiLUT_Cubic_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through cubic interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.118 NppStatus nppiLUT_Cubic_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[3], const Npp32s * *pLevels*[3], int *nLevels*[3])

4 channel 8-bit unsigned cubic interpolated look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through cubic interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 256.

7.48.2.119 NppStatus nppiLUT_Cubic_8u_C1IR (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*, const Npp32s * *pLevels*, int *nLevels*)

8-bit unsigned cubic interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 256.

7.48.2.120 `NppStatus nppiLUT_Cubic_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

8-bit unsigned cubic interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.121 `NppStatus nppiLUT_Cubic_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.122 `NppStatus nppiLUT_Cubic_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 8-bit unsigned cubic interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.123 `NppStatus nppiLUT_Cubic_8u_C4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.124 `NppStatus nppiLUT_Cubic_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 8-bit unsigned cubic interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.125 `NppStatus nppiLUT_Linear_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.126 NppStatus nppiLUT_Linear_16s_AC4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[3], const Npp32s * *pLevels*[3], int *nLevels*[3])

4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.127 NppStatus nppiLUT_Linear_16s_CIIR (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*, const Npp32s * *pLevels*, int *nLevels*)

16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.128 `NppStatus nppiLUT_Linear_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.129 `NppStatus nppiLUT_Linear_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.130 `NppStatus nppiLUT_Linear_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.131 `NppStatus nppiLUT_Linear_16s_C4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.132 `NppStatus nppiLUT_Linear_16s_C4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.133 `NppStatus nppiLUT_Linear_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.134 NppStatus nppiLUT_Linear_16u_AC4R (**const Npp16u * pSrc**, **int nSrcStep**, **Npp16u * pDst**, **int nDstStep**, **NppiSize oSizeROI**, **const Npp32s * pValues[3]**, **const Npp32s * pLevels[3]**, **int nLevels[3]**)

4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.135 NppStatus nppiLUT_Linear_16u_C11R (**Npp16u * pSrcDst**, **int nSrcDstStep**, **NppiSize oSizeROI**, **const Npp32s * pValues**, **const Npp32s * pLevels**, **int nLevels**)

16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.136 `NppStatus nppiLUT_Linear_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.137 `NppStatus nppiLUT_Linear_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.138 `NppStatus nppiLUT_Linear_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.139 `NppStatus nppiLUT_Linear_16u_C4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.140 `NppStatus nppiLUT_Linear_16u_C4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.141 `NppStatus nppiLUT_Linear_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f * pValues[3], const Npp32f * pLevels[3], int nLevels[3])`

4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.142 **NppStatus nppiLUT_Linear_32f_AC4R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pValues*[3], const Npp32f * *pLevels*[3], int *nLevels*[3])

4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.143 **NppStatus nppiLUT_Linear_32f_C11R** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f * *pValues*, const Npp32f * *pLevels*, int *nLevels*)

32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.144 `NppStatus nppiLUT_Linear_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pValues, const Npp32f * pLevels, int nLevels)`

32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.145 `NppStatus nppiLUT_Linear_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f * pValues[3], const Npp32f * pLevels[3], int nLevels[3])`

3 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.146 NppStatus nppiLUT_Linear_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pValues*[3], const Npp32f * *pLevels*[3], int *nLevels*[3])

3 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.147 NppStatus nppiLUT_Linear_32f_C4IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f * *pValues*[4], const Npp32f * *pLevels*[4], int *nLevels*[4])

4 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.148 `NppStatus nppiLUT_Linear_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pValues[4], const Npp32f * pLevels[4], int nLevels[4])`

4 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

7.48.2.149 `NppStatus nppiLUT_Linear_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 8-bit unsigned linear interpolated look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through linear interpolation. Alpha channel is the last channel and is not processed.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 256.

7.48.2.150 **NppStatus nppiLUT_Linear_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*[3], const Npp32s * *pLevels*[3], int *nLevels*[3])

4 channel 8-bit unsigned linear interpolated look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through linear interpolation. Alpha channel is the last channel and is not processed.

>>>>>> ATTENTION ATTENTION <<<<<<<

NOTE: As of the 5.0 release of NPP, the *pValues* and *pLevels* pointers need to be host memory pointers to arrays of device memory pointers.

>>>>>> <<<<<<<

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.151 **NppStatus nppiLUT_Linear_8u_C1IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s * *pValues*, const Npp32s * *pLevels*, int *nLevels*)

8-bit unsigned linear interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.152 `NppStatus nppiLUT_Linear_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

8-bit unsigned linear interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

>>>>>> ATTENTION ATTENTION <<<<<<<

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be device memory pointers.

>>>>>> <<<<<<<

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Pointer to an array of user defined OUTPUT values (this is now a device memory pointer)

pLevels Pointer to an array of user defined INPUT values (this is now a device memory pointer)

nLevels Number of user defined number of input/output mapping points (levels)

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.153 `NppStatus nppiLUT_Linear_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 8-bit unsigned linear interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.154 `NppStatus nppiLUT_Linear_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 8-bit unsigned linear interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

>>>>>> ATTENTION ATTENTION <<<<<<<

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be host memory pointers to arrays of device memory pointers.

>>>>>> <<<<<<<

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pValues Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.155 `NppStatus nppiLUT_Linear_8u_C4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 8-bit unsigned linear interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.156 `NppStatus nppiLUT_Linear_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 8-bit unsigned linear interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

>>>>>> ATTENTION ATTENTION <<<<<<<

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be host memory pointers to arrays of device memory pointers.

>>>>>> <<<<<<<

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

pLevels Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

nLevels Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.157 **NppStatus nppiLUT_Trilinear_8u_AC4IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, Npp32u * *pValues*, Npp8u * *pLevels*[3], int *aLevels*[3])

Four channel 8-bit unsigned 3D trilinear interpolated look-up-table in place color conversion, not affecting alpha.

Alpha channel is the last channel and is not processed.

The LUT is derived from a set of user defined mapping points through trilinear interpolation.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Device pointer *aLevels*[2] number of contiguous 2D x,y planes of 4-byte packed RGBX values containing the user defined base OUTPUT values at that x,y, and z (R,G,B) level location. Each level must contain x * y 4-byte packed pixel values (4th byte is used for alignment only and is ignored) in row (x) order.

pLevels Host pointer to an array of 3 host pointers, one per cube edge, pointing to user defined INPUT level values.

aLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per 3D cube edge. *aLevels*[0] represents the number of x axis levels (Red), *aLevels*[1] represents the number of y axis levels (Green), and *aLevels*[2] represents the number of z axis levels (Blue).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_NUMBER_OF_LEVELS_ERROR](#) if the number of levels is less than 2 or greater than 256.

7.48.2.158 **NppStatus nppiLUT_Trilinear_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32u * *pValues*, Npp8u * *pLevels*[3], int *aLevels*[3])

Four channel 8-bit unsigned 3D trilinear interpolated look-up-table color conversion, not affecting alpha.

Alpha channel is the last channel and is not processed.

The LUT is derived from a set of user defined mapping points through trilinear interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Device pointer to *aLevels*[2] number of contiguous 2D x,y planes of 4-byte packed RGBX values containing the user defined base OUTPUT values at that x,y, and z (R,G,B) level location. Each level must contain x * y 4-byte packed pixel values (4th byte is used for alignment only and is ignored) in row (x) order.

pLevels Host pointer to an array of 3 host pointers, one per cube edge, pointing to user defined INPUT level values.

aLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per 3D cube edge. *aLevels*[0] represents the number of x axis levels (Red), *aLevels*[1] represents the number of y axis levels (Green), and *aLevels*[2] represents the number of z axis levels (Blue).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- **[NPP_LUT_NUMBER_OF_LEVELS_ERROR](#)** if the number of levels is less than 2 or greater than 256.

7.48.2.159 NppStatus nppiLUT_Trilinear_8u_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32u * *pValues*, Npp8u * *pLevels*[3], int *aLevels*[3])

Four channel 8-bit unsigned 3D trilinear interpolated look-up-table color conversion, with alpha copy.

Alpha channel is the last channel and is copied to the destination unmodified.

The LUT is derived from a set of user defined mapping points through trilinear interpolation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pValues Device pointer to *aLevels*[2] number of contiguous 2D x,y planes of 4-byte packed RGBX values containing the user defined base OUTPUT values at that x,y, and z (R,G,B) level location. Each level must contain x * y 4-byte packed pixel values (4th byte is used for alignment only and is ignored) in row (x) order.

pLevels Host pointer to an array of 3 host pointers, one per cube edge, pointing to user defined INPUT level values.

aLevels Host pointer to an array of 3 user defined number of input/output mapping points, one per 3D cube edge. *aLevels*[0] represents the number of x axis levels (Red), *aLevels*[1] represents the number of y axis levels (Green), and *aLevels*[2] represents the number of z axis levels (Blue).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- **[NPP_LUT_NUMBER_OF_LEVELS_ERROR](#)** if the number of levels is less than 2 or greater than 256.

7.48.2.160 NppStatus nppiLUTPalette_16u24u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pTable*, int *nBitSize*)

One channel 16-bit unsigned bit range restricted 24-bit unsigned palette look-up-table color conversion with 24-bit unsigned destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step (3 unsigned bytes per pixel).
oSizeROI Region-of-Interest (ROI).
pTable Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
nBitSize Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_PALETTE_BITSIZE_ERROR** if *nBitSize* is < 1 or > 16.

7.48.2.161 **NppStatus nppiLUTPalette_16u32u_C1R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp32u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32u * *pTable*, int *nBitSize*)

One channel 16-bit unsigned bit range restricted 32-bit palette look-up-table color conversion with 32-bit unsigned destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step (4 bytes per pixel).
oSizeROI Region-of-Interest (ROI).
pTable Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
nBitSize Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_PALETTE_BITSIZE_ERROR** if *nBitSize* is < 1 or > 16.

7.48.2.162 **NppStatus nppiLUTPalette_16u8u_C1R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pTable*, int *nBitSize*)

One channel 16-bit unsigned bit range restricted 8-bit unsigned palette look-up-table color conversion with 8-bit unsigned destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step (1 unsigned byte per pixel).
- oSizeROI* Region-of-Interest (ROI).
- pTable* Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
- nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_PALETTE_BITSIZE_ERROR** if *nBitSize* is < 1 or > 16.

7.48.2.163 **NppStatus nppiLUTPalette_16u_AC4R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u * *pTables*[3], int *nBitSize*)

Four channel 16-bit unsigned bit range restricted palette look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. Alpha channel is the last channel and is not processed.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pTables* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.
- nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_PALETTE_BITSIZE_ERROR** if *nBitSize* is < 1 or > 16.

7.48.2.164 **NppStatus nppiLUTPalette_16u_C1R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u * *pTable*, int *nBitSize*)

One channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pTable* Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
- nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

Returns:

- Image Data Related Error Codes, ROI Related Error Codes
 - [NPP_LUT_PALETTE_BITSIZE_ERROR](#) if *nBitSize* is < 1 or > 16.

7.48.2.165 **NppStatus nppiLUTPalette_16u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u * *pTables*[3], int *nBitSize*)**

Three channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pTables* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.
- nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

Returns:

- Image Data Related Error Codes, ROI Related Error Codes
 - [NPP_LUT_PALETTE_BITSIZE_ERROR](#) if *nBitSize* is < 1 or > 16.

7.48.2.166 **NppStatus nppiLUTPalette_16u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u * *pTables*[4], int *nBitSize*)**

Four channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pTables Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.

nBitSize Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_PALETTE_BITSIZE_ERROR** if *nBitSize* is < 1 or > 16.

7.48.2.167 **NppStatus nppiLUTPalette_8u24u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pTable*, int *nBitSize*)

One channel 8-bit unsigned bit range restricted 24-bit palette look-up-table color conversion with 24-bit destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step (3 bytes per pixel).

oSizeROI Region-of-Interest (ROI).

pTable Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)

nBitSize Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_PALETTE_BITSIZE_ERROR** if *nBitSize* is < 1 or > 8.

7.48.2.168 **NppStatus nppiLUTPalette_8u32u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp32u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32u * *pTable*, int *nBitSize*)

One channel 8-bit unsigned bit range restricted 32-bit palette look-up-table color conversion with 32-bit destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step (4 bytes per pixel).
oSizeROI Region-of-Interest (ROI).
pTable Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
nBitSize Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_PALETTE_BITSIZE_ERROR** if *nBitSize* is < 1 or > 8.

7.48.2.169 NppStatus nppiLUTPalette_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pTables*[3], int *nBitSize*)

Four channel 8-bit unsigned bit range restricted palette look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. Alpha channel is the last channel and is not processed.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pTables Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.
nBitSize Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_PALETTE_BITSIZE_ERROR** if *nBitSize* is < 1 or > 8.

7.48.2.170 NppStatus nppiLUTPalette_8u_C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pTable*, int *nBitSize*)

One channel 8-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pTable* Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
- nBitSize* Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_PALETTE_BITSIZE_ERROR** if *nBitSize* is < 1 or > 8.

7.48.2.171 NppStatus nppiLUTPalette_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pTables*[3], int *nBitSize*)

Three channel 8-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pTables* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.
- nBitSize* Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_PALETTE_BITSIZE_ERROR** if *nBitSize* is < 1 or > 8.

7.48.2.172 NppStatus nppiLUTPalette_8u_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pTables*[4], int *nBitSize*)

Four channel 8-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pTables Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.

nBitSize Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_PALETTE_BITSIZE_ERROR](#) if *nBitSize* is < 1 or > 8.

7.48.2.173 `NppStatus nppiLUTPaletteSwap_16u_C3A0C4R (const Npp16u * pSrc, int nSrcStep, int nAlphaValue, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u * pTables[3], int nBitSize)`

Three channel 16-bit unsigned source bit range restricted palette look-up-table color conversion to four channel 16-bit unsigned destination output with alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. This function also reverses the source pixel channel order in the destination so the Alpha channel is the first channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#) (3 unsigned short integers per pixel).

nAlphaValue Signed alpha value that will be used to initialize the pixel alpha channel position in all modified destination pixels.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#) (4 unsigned short integers per pixel with alpha).

oSizeROI [Region-of-Interest \(ROI\)](#).

pTables Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values. Alpha values < 0 or > 65535 will cause destination pixel alpha channel values to be unmodified.

nBitSize Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_PALETTE_BITSIZE_ERROR](#) if *nBitSize* is < 1 or > 16.

7.48.2.174 NppStatus nppiLUTPaletteSwap_8u_C3A0C4R (**const Npp8u * pSrc**, **int nSrcStep**, **int nAlphaValue**, **Npp8u * pDst**, **int nDstStep**, **NppiSize oSizeROI**, **const Npp8u * pTables[3]**, **int nBitSize**)

Three channel 8-bit unsigned source bit range restricted palette look-up-table color conversion to four channel 8-bit unsigned destination output with alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. This function also reverses the source pixel channel order in the destination so the Alpha channel is the first channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#) (3 bytes per pixel).

nAlphaValue Signed alpha value that will be used to initialize the pixel alpha channel position in all modified destination pixels.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#) (4 bytes per pixel with alpha).

oSizeROI [Region-of-Interest \(ROI\)](#).

pTables Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values. Alpha values < 0 or > 255 will cause destination pixel alpha channel values to be unmodified.

nBitSize Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP_LUT_PALETTE_BITSIZESIZE_ERROR](#) if nBitSize is < 1 or > 8.

7.49 Compression

Image compression primitives.

Modules

- [Quantization Functions](#)

Typedefs

- typedef struct [NppiDecodeHuffmanSpec](#) [NppiDecodeHuffmanSpec](#)

Functions

- [NppStatus nppiDecodeHuffmanSpecGetBufSize_JPEG](#) (int *pSize)
Returns the length of the NppiDecodeHuffmanSpec structure.
- [NppStatus nppiDecodeHuffmanSpecInitHost_JPEG](#) (const [Npp8u](#) *pRawHuffmanTable, [NppiHuffmanTableType](#) eTableType, [NppiDecodeHuffmanSpec](#) *pHuffmanSpec)
Creates a Huffman table in a format that is suitable for the decoder on the host.
- [NppStatus nppiDecodeHuffmanSpecInitAllocHost_JPEG](#) (const [Npp8u](#) *pRawHuffmanTable, [NppiHuffmanTableType](#) eTableType, [NppiDecodeHuffmanSpec](#) **ppHuffmanSpec)
Allocates memory and creates a Huffman table in a format that is suitable for the decoder on the host.
- [NppStatus nppiDecodeHuffmanSpecFreeHost_JPEG](#) ([NppiDecodeHuffmanSpec](#) *pHuffmanSpec)
Frees the host memory allocated by nppiDecodeHuffmanSpecInitAllocHost_JPEG.
- [NppStatus nppiDecodeHuffmanScanHost_JPEG_8u16s_P1R](#) (const [Npp8u](#) *pSrc, [Npp32s](#) nLength, [Npp32s](#) nRestartInterval, [Npp32s](#) nSs, [Npp32s](#) nSe, [Npp32s](#) nAh, [Npp32s](#) nAl, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiDecodeHuffmanSpec](#) *pHuffmanTableDC, [NppiDecodeHuffmanSpec](#) *pHuffmanTableAC, [NppiSize](#) oSizeROI)
Huffman Decoding of the JPEG decoding on the host.
- [NppStatus nppiDecodeHuffmanScanHost_JPEG_8u16s_P3R](#) (const [Npp8u](#) *pSrc, [Npp32s](#) nLength, [Npp32s](#) nRestartInterval, [Npp32s](#) nSs, [Npp32s](#) nSe, [Npp32s](#) nAh, [Npp32s](#) nAl, [Npp16s](#) *apDst[3], [Npp32s](#) aDstStep[3], [NppiDecodeHuffmanSpec](#) *apHuffmanDCTable[3], [NppiDecodeHuffmanSpec](#) *apHuffmanACTable[3], [NppiSize](#) aSizeROI[3])
Huffman Decoding of the JPEG decoding on the host.

7.49.1 Detailed Description

Image compression primitives.

The JPEG standard defines a flow of level shift, DCT and quantization for forward JPEG transform and inverse level shift, IDCT and de-quantization for inverse JPEG transform. This group has the functions for both forward and inverse functions.

7.49.2 Typedef Documentation

7.49.2.1 typedef struct NppiDecodeHuffmanSpec NppiDecodeHuffmanSpec

7.49.3 Function Documentation

7.49.3.1 NppStatus nppiDecodeHuffmanScanHost_JPEG_8u16s_P1R (const Npp8u * *pSrc*, Npp32s *nLength*, Npp32s *restartInterval*, Npp32s *Ss*, Npp32s *Se*, Npp32s *Ah*, Npp32s *Al*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiDecodeHuffmanSpec * *pHuffmanTableDC*, NppiDecodeHuffmanSpec * *pHuffmanTableAC*, NppiSize *oSizeROI*)

Huffman Decoding of the JPEG decoding on the host.

Input is expected in byte stuffed huffman encoded JPEG scan and output is expected to be 64x1 macro blocks.

Parameters:

pSrc Byte-stuffed huffman encoded JPEG scan.

nLength Byte length of the input.

restartInterval Restart Interval, see JPEG standard.

Ss Start Coefficient, see JPEG standard.

Se End Coefficient, see JPEG standard.

Ah Bit Approximation High, see JPEG standard.

Al Bit Approximation Low, see JPEG standard.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pHuffmanTableDC DC Huffman table.

pHuffmanTableAC AC Huffman table.

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

Error codes:

- [NPP_SIZE_ERROR](#) For negative input height/width or not a multiple of 8 width/height.
- [NPP_STEP_ERROR](#) If input image width is not multiple of 8 or does not match ROI.
- [NPP_NULL_POINTER_ERROR](#) If the destination pointer is 0.

7.49.3.2 NppStatus nppiDecodeHuffmanScanHost_JPEG_8u16s_P3R (const Npp8u * *pSrc*, Npp32s *nLength*, Npp32s *nRestartInterval*, Npp32s *nSs*, Npp32s *nSe*, Npp32s *nAh*, Npp32s *nAl*, Npp16s * *apDst*[3], Npp32s *aDstStep*[3], NppiDecodeHuffmanSpec * *apHuffmanDCTable*[3], NppiDecodeHuffmanSpec * *apHuffmanACTable*[3], NppiSize *aSizeROI*[3])

Huffman Decoding of the JPEG decoding on the host.

Input is expected in byte stuffed huffman encoded JPEG scan and output is expected to be 64x1 macro blocks.

Parameters:

pSrc Byte-stuffed huffman encoded JPEG scan.
nLength Byte length of the input.
nRestartInterval Restart Interval, see JPEG standard.
nSs Start Coefficient, see JPEG standard.
nSe End Coefficient, see JPEG standard.
nAh Bit Approximation High, see JPEG standard.
nAl Bit Approximation Low, see JPEG standard.
apDst [Destination-Image Pointer](#).
aDstStep [Destination-Image Line Step](#).
apHuffmanDCTable DC Huffman tables.
apHuffmanACTable AC Huffman tables.
aSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

Error codes:

- [NPP_SIZE_ERROR](#) For negative input height/width or not a multiple of 8 width/height.
- [NPP_STEP_ERROR](#) If input image width is not multiple of 8 or does not match ROI.
- [NPP_NULL_POINTER_ERROR](#) If the destination pointer is 0.

7.49.3.3 NppStatus nppiDecodeHuffmanSpecFreeHost_JPEG (NppiDecodeHuffmanSpec * *pHuffmanSpec*)

Frees the host memory allocated by nppiDecodeHuffmanSpecInitAllocHost_JPEG.

Parameters:

pHuffmanSpec Pointer to the Huffman table for the decoder

7.49.3.4 NppStatus nppiDecodeHuffmanSpecGetBufSize_JPEG (int * *pSize*)

Returns the length of the NppiDecodeHuffmanSpec structure.

Parameters:

pSize Pointer to a variable that will receive the length of the NppiDecodeHuffmanSpec structure.

Returns:

Error codes:

- [NPP_NULL_POINTER_ERROR](#) If one of the pointers is 0.

7.49.3.5 `NppStatus nppiDecodeHuffmanSpecInitAllocHost_JPEG (const Npp8u *
pRawHuffmanTable, NppiHuffmanTableType eTableType, NppiDecodeHuffmanSpec **
ppHuffmanSpec)`

Allocates memory and creates a Huffman table in a format that is suitable for the decoder on the host.

Parameters:

pRawHuffmanTable Huffman table formatted as specified in the JPEG standard.

eTableType Enum specifying type of table (nppiDCTable or nppiACTable).

ppHuffmanSpec Pointer to returned pointer to the Huffman table for the decoder

Returns:

Error codes:

- [NPP_NULL_POINTER_ERROR](#) If one of the pointers is 0.

7.49.3.6 `NppStatus nppiDecodeHuffmanSpecInitHost_JPEG (const Npp8u *pRawHuffmanTable,
NppiHuffmanTableType eTableType, NppiDecodeHuffmanSpec *pHuffmanSpec)`

Creates a Huffman table in a format that is suitable for the decoder on the host.

Parameters:

pRawHuffmanTable Huffman table formatted as specified in the JPEG standard.

eTableType Enum specifying type of table (nppiDCTable or nppiACTable).

pHuffmanSpec Pointer to the Huffman table for the decoder

Returns:

Error codes:

- [NPP_NULL_POINTER_ERROR](#) If one of the pointers is 0.

7.50 Quantization Functions

Typedefs

- typedef struct [NppiDCTState](#) [NppiDCTState](#)

Functions

- [NppStatus](#) [nppiQuantFwdRawTableInit_JPEG_8u](#) ([Npp8u](#) *hpQuantRawTable, int nQualityFactor)

Apply quality factor to raw 8-bit quantization table.

- [NppStatus](#) [nppiQuantFwdTableInit_JPEG_8u16u](#) (const [Npp8u](#) *hpQuantRawTable, [Npp16u](#) *hpQuantFwdRawTable)

Initializes a quantization table for [nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R\(\)](#).

- [NppStatus](#) [nppiQuantInvTableInit_JPEG_8u16u](#) (const [Npp8u](#) *hpQuantRawTable, [Npp16u](#) *hpQuantFwdRawTable)

Initializes a quantization table for [nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R\(\)](#).

- [NppStatus](#) [nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp16s](#) *pDst, int nDstStep, const [Npp16u](#) *pQuantFwdTable, [NppiSize](#) oSizeROI)

Forward DCT, quantization and level shift part of the JPEG encoding.

- [NppStatus](#) [nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp8u](#) *pDst, int nDstStep, const [Npp16u](#) *pQuantInvTable, [NppiSize](#) oSizeROI)

Inverse DCT, de-quantization and level shift part of the JPEG decoding.

- [NppStatus](#) [nppiDCTInitAlloc](#) ([NppiDCTState](#) **ppState)

Initializes DCT state structure and allocates additional resources.

- [NppStatus](#) [nppiDCTFree](#) ([NppiDCTState](#) *pState)

Frees the additional resources of the DCT state structure.

- [NppStatus](#) [nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R_NEW](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp16s](#) *pDst, int nDstStep, const [Npp8u](#) *pQuantizationTable, [NppiSize](#) oSizeROI, [NppiDCTState](#) *pState)

Forward DCT, quantization and level shift part of the JPEG encoding.

- [NppStatus](#) [nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R_NEW](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp8u](#) *pDst, int nDstStep, const [Npp8u](#) *pQuantizationTable, [NppiSize](#) oSizeROI, [NppiDCTState](#) *pState)

Inverse DCT, de-quantization and level shift part of the JPEG decoding.

7.50.1 Typedef Documentation

7.50.1.1 typedef struct NppiDCTState NppiDCTState

7.50.2 Function Documentation

7.50.2.1 NppStatus nppiDCTFree (NppiDCTState * *pState*)

Frees the additional resources of the DCT state structure.

See also:

[nppiDCTInitAlloc](#)

Parameters:

pState Pointer to DCT state structure.

Returns:

NPP_SUCCESS Indicates no error. Any other value indicates an error or a warning
 NPP_SIZE_ERROR Indicates an error condition if any image dimension has zero or negative value
 NPP_NULL_POINTER_ERROR Indicates an error condition if pState pointer is NULL

7.50.2.2 NppStatus nppiDCTInitAlloc (NppiDCTState ** *ppState*)

Initializes DCT state structure and allocates additional resources.

See also:

[nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R_NEW\(\)](#), [nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R_NEW](#).

Parameters:

ppState Pointer to pointer to DCT state structure.

Returns:

NPP_SUCCESS Indicates no error. Any other value indicates an error or a warning
 NPP_SIZE_ERROR Indicates an error condition if any image dimension has zero or negative value
 NPP_NULL_POINTER_ERROR Indicates an error condition if pBufSize pointer is NULL

7.50.2.3 NppStatus nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, const Npp16u * *pQuantFwdTable*, NppiSize *oSizeROI*)

Forward DCT, quantization and level shift part of the JPEG encoding.

Input is expected in 8x8 macro blocks and output is expected to be in 64x1 macro blocks.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pQuantFwdTable Forward quantization tables for JPEG encoding created using `nppiQuantInvTableInit_JPEG_8u16u()`.

oSizeROI Region-of-Interest (ROI).

Returns:

Error codes:

- `NPP_SIZE_ERROR` For negative input height/width or not a multiple of 8 width/height.
- `NPP_STEP_ERROR` If input image width is not multiple of 8 or does not match ROI.
- `NPP_NULL_POINTER_ERROR` If the destination pointer is 0.

7.50.2.4 `NppStatus nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R_NEW (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, const Npp8u * pQuantizationTable, NppiSize oSizeROI, NppiDCTState * pState)`

Forward DCT, quantization and level shift part of the JPEG encoding.

Input is expected in 8x8 macro blocks and output is expected to be in 64x1 macro blocks. The new version of the primitive takes the ROI in image pixel size and works with DCT coefficients that are in zig-zag order.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Image width in pixels x 8 x sizeof(Npp16s).

pQuantizationTable Quantization Table in zig-zag order.

oSizeROI Region-of-Interest (ROI).

pState Pointer to DCT state structure. This structure must be initialized allocated and initialized using `nppiDCTInitAlloc()`.

Returns:

Error codes:

- `NPP_SIZE_ERROR` For negative input height/width or not a multiple of 8 width/height.
- `NPP_STEP_ERROR` If input image width is not multiple of 8 or does not match ROI.
- `NPP_NULL_POINTER_ERROR` If the destination pointer is 0.

7.50.2.5 `NppStatus nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, const Npp16u * pQuantInvTable, NppiSize oSizeROI)`

Inverse DCT, de-quantization and level shift part of the JPEG decoding.

Input is expected in 64x1 macro blocks and output is expected to be in 8x8 macro blocks.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Image width in pixels x 8 x sizeof(Npp16s).

pDst Destination-Image Pointer.

nDstStep Image width in pixels x 8 x sizeof(Npp16s).

pQuantInvTable Inverse quantization tables for JPEG decoding created using `nppiQuantInvTableInit_JPEG_8u16u()`.

oSizeROI Region-of-Interest (ROI).

Returns:

Error codes:

- `NPP_SIZE_ERROR` For negative input height/width or not a multiple of 8 width/height.
- `NPP_STEP_ERROR` If input image width is not multiple of 8 or does not match ROI.
- `NPP_NULL_POINTER_ERROR` If the destination pointer is 0.

7.50.2.6 `NppStatus nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R_NEW (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, const Npp8u * pQuantizationTable, NppiSize oSizeROI, NppiDCTState * pState)`

Inverse DCT, de-quantization and level shift part of the JPEG decoding.

Input is expected in 64x1 macro blocks and output is expected to be in 8x8 macro blocks. The new version of the primitive takes the ROI in image pixel size and works with DCT coefficients that are in zig-zag order.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Image width in pixels x 8 x sizeof(Npp16s).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pQuantizationTable Quantization Table in zig-zag order.

oSizeROI Region-of-Interest (ROI).

pState Pointer to DCT state structure. This structure must be initialized allocated and initialized using `nppiDCTInitAlloc()`.

Returns:

Error codes:

- `NPP_SIZE_ERROR` For negative input height/width or not a multiple of 8 width/height.
- `NPP_STEP_ERROR` If input image width is not multiple of 8 or does not match ROI.
- `NPP_NULL_POINTER_ERROR` If the destination pointer is 0.

7.50.2.7 NppStatus nppiQuantFwdRawTableInit_JPEG_8u (Npp8u * *hpQuantRawTable*, int *nQualityFactor*)

Apply quality factor to raw 8-bit quantization table.

This is effectively an in-place method that modifies a given raw quantization table based on a quality factor. Note that this method is a host method and that the pointer to the raw quantization table is a host pointer.

Parameters:

hpQuantRawTable Raw quantization table.

nQualityFactor Quality factor for the table. Range is [1:100].

Returns:

Error code: [NPP_NULL_POINTER_ERROR](#) is returned if *hpQuantRawTable* is 0.

7.50.2.8 NppStatus nppiQuantFwdTableInit_JPEG_8u16u (const Npp8u * *hpQuantRawTable*, Npp16u * *hpQuantFwdRawTable*)

Initializes a quantization table for [nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R\(\)](#).

The method creates a 16-bit version of the raw table and converts the data order from zigzag layout to original row-order layout since raw quantization tables are typically stored in zigzag format.

This method is a host method. It consumes and produces host data. I.e. the pointers passed to this function must be host pointers. The resulting table needs to be transferred to device memory in order to be used with [nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R\(\)](#) function.

Parameters:

hpQuantRawTable Host pointer to raw quantization table as returned by [nppiQuantFwdRawTableInit_JPEG_8u\(\)](#). The raw quantization table is assumed to be in zigzag order.

hpQuantFwdRawTable Forward quantization table for use with [nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R\(\)](#).

Returns:

Error code: [NPP_NULL_POINTER_ERROR](#) if *hpQuantRawTable* or *hpQuantFwdRawTable* is 0.

7.50.2.9 NppStatus nppiQuantInvTableInit_JPEG_8u16u (const Npp8u * *hpQuantRawTable*, Npp16u * *hpQuantFwdRawTable*)

Initializes a quantization table for [nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R\(\)](#).

The [nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R\(\)](#) method uses a quantization table in a 16-bit format allowing for faster processing. In addition it converts the data order from zigzag layout to original row-order layout. Typically raw quantization tables are stored in zigzag format.

This method is a host method and consumes and produces host data. I.e. the pointers passed to this function must be host pointers. The resulting table needs to be transferred to device memory in order to be used with [nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R\(\)](#) function.

Parameters:

hpQuantRawTable Raw quantization table.

hpQuantFwdRawTable Inverse quantization table.

Returns:

[NPP_NULL_POINTER_ERROR](#) pQuantRawTable or pQuantFwdRawTable is 0.

7.51 Labeling and Segmentation

Pixel labeling and image segmentation operations.

Modules

- [GraphCut](#)

Typedefs

- typedef struct [NppiGraphcutState](#) [NppiGraphcutState](#)

7.51.1 Detailed Description

Pixel labeling and image segmentation operations.

NOTE: All of these Graphcut functions will be deprecated in a future release.

7.51.2 Typedef Documentation

7.51.2.1 typedef struct [NppiGraphcutState](#) [NppiGraphcutState](#)

7.52 GraphCut

Graphcut

- **NppStatus nppiGraphcutGetSize** (**NppiSize** oSize, int *pBufSize)
Calculates the size of the temporary buffer for graph-cut with 4 neighborhood labeling.
- **NppStatus nppiGraphcut8GetSize** (**NppiSize** oSize, int *pBufSize)
Calculates the size of the temporary buffer for graph-cut with 8 neighborhood labeling.
- **NppStatus nppiGraphcutInitAlloc** (**NppiSize** oSize, **NppiGraphcutState** **ppState, **Npp8u** *pDeviceMem)
Initializes graph-cut state structure and allocates additional resources for graph-cut with 8 neighborhood labeling.
- **NppStatus nppiGraphcut8InitAlloc** (**NppiSize** oSize, **NppiGraphcutState** **ppState, **Npp8u** *pDeviceMem)
Allocates and initializes the graph-cut state structure and additional resources for graph-cut with 8 neighborhood labeling.
- **NppStatus nppiGraphcutFree** (**NppiGraphcutState** *pState)
Frees the additional resources of the graph-cut state structure.
- **NppStatus nppiGraphcut_32s8u** (**Npp32s** *pTerminals, **Npp32s** *pLeftTransposed, **Npp32s** *pRightTransposed, **Npp32s** *pTop, **Npp32s** *pBottom, int nStep, int nTransposedStep, **NppiSize** size, **Npp8u** *pLabel, int nLabelStep, **NppiGraphcutState** *pState)
Graphcut of a flow network (32bit signed integer edge capacities).
- **NppStatus nppiGraphcut8_32s8u** (**Npp32s** *pTerminals, **Npp32s** *pLeftTransposed, **Npp32s** *pRightTransposed, **Npp32s** *pTop, **Npp32s** *pTopLeft, **Npp32s** *pTopRight, **Npp32s** *pBottom, **Npp32s** *pBottomLeft, **Npp32s** *pBottomRight, int nStep, int nTransposedStep, **NppiSize** size, **Npp8u** *pLabel, int nLabelStep, **NppiGraphcutState** *pState)
Graphcut of a flow network (32bit signed integer edge capacities).
- **NppStatus nppiGraphcut_32f8u** (**Npp32f** *pTerminals, **Npp32f** *pLeftTransposed, **Npp32f** *pRightTransposed, **Npp32f** *pTop, **Npp32f** *pBottom, int nStep, int nTransposedStep, **NppiSize** size, **Npp8u** *pLabel, int nLabelStep, **NppiGraphcutState** *pState)
Graphcut of a flow network (32bit float edge capacities).
- **NppStatus nppiGraphcut8_32f8u** (**Npp32f** *pTerminals, **Npp32f** *pLeftTransposed, **Npp32f** *pRightTransposed, **Npp32f** *pTop, **Npp32f** *pTopLeft, **Npp32f** *pTopRight, **Npp32f** *pBottom, **Npp32f** *pBottomLeft, **Npp32f** *pBottomRight, int nStep, int nTransposedStep, **NppiSize** size, **Npp8u** *pLabel, int nLabelStep, **NppiGraphcutState** *pState)
Graphcut of a flow network (32bit float edge capacities).

7.52.1 Function Documentation

7.52.1.1 `NppStatus nppiGraphcut8_32f8u (Npp32f * pTerminals, Npp32f * pLeftTransposed, Npp32f * pRightTransposed, Npp32f * pTop, Npp32f * pTopLeft, Npp32f * pTopRight, Npp32f * pBottom, Npp32f * pBottomLeft, Npp32f * pBottomRight, int nStep, int nTransposedStep, NppiSize size, Npp8u * pLabel, int nLabelStep, NppiGraphcutState * pState)`

Graphcut of a flow network (32bit float edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 8-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array (`terminals(x) = source(x) - sink(x)`). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example `left(0,*) == 0`). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and >0).

NOTE: This Graphcut function will be deprecated in a future release.

See also:

[nppiGraphcut8InitAlloc\(\)](#), [nppiGraphcutFree\(\)](#), [nppiGraphcut8GetSize\(\)](#).

Parameters:

pTerminals Pointer to differences of terminal edge capacities (`terminal(x) = source(x) - sink(x)`)

pLeftTransposed Pointer to transposed left edge capacities (`left(0,*)` must be 0)

pRightTransposed Pointer to transposed right edge capacities (`right(width-1,*)` must be 0)

pTop Pointer to top edge capacities (`top(*,0)` must be 0)

pTopLeft Pointer to top left edge capacities (`topleft(*,0)` & `topleft(0,*)` must be 0)

pTopRight Pointer to top right edge capacities (`topright(*,0)` & `topright(width-1,*)` must be 0)

pBottom Pointer to bottom edge capacities (`bottom(*,height-1)` must be 0)

pBottomLeft Pointer to bottom left edge capacities (`bottomleft(*,height-1)` & `bottomleft(0,*)` must be 0)

pBottomRight Pointer to bottom right edge capacities (`bottomright(*,height-1)` & `bottomright(width-1,*)` must be 0)

nStep Step in bytes between any pair of sequential rows of edge capacities

nTransposedStep Step in bytes between any pair of sequential rows of transposed edge capacities

size Graph size

pLabel Pointer to destination label image

nLabelStep Step in bytes between any pair of sequential rows of label image

pState Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcut8InitAlloc\(\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.52.1.2 `NppStatus nppiGraphcut8_32s8u (Npp32s * pTerminals, Npp32s * pLeftTransposed, Npp32s * pRightTransposed, Npp32s * pTop, Npp32s * pTopLeft, Npp32s * pTopRight, Npp32s * pBottom, Npp32s * pBottomLeft, Npp32s * pBottomRight, int nStep, int nTransposedStep, NppiSize size, Npp8u * pLabel, int nLabelStep, NppiGraphcutState * pState)`

Graphcut of a flow network (32bit signed integer edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 8-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array (`terminals(x) = source(x) - sink(x)`). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example `left(0,*) == 0`). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and >0).

NOTE: This Graphcut function will be deprecated in a future release.

See also:

[nppiGraphcut8InitAlloc\(\)](#), [nppiGraphcutFree\(\)](#), [nppiGraphcut8GetSize\(\)](#).

Parameters:

pTerminals Pointer to differences of terminal edge capacities (`terminal(x) = source(x) - sink(x)`)

pLeftTransposed Pointer to transposed left edge capacities (`left(0,*)` must be 0)

pRightTransposed Pointer to transposed right edge capacities (`right(width-1,*)` must be 0)

pTop Pointer to top edge capacities (`top(*,0)` must be 0)

pTopLeft Pointer to top left edge capacities (`opleft(*,0)` & `opleft(0,*)` must be 0)

pTopRight Pointer to top right edge capacities (`topright(*,0)` & `topright(width-1,*)` must be 0)

pBottom Pointer to bottom edge capacities (`bottom(*,height-1)` must be 0)

pBottomLeft Pointer to bottom left edge capacities (`bottomleft(*,height-1)` & `bottomleft(0,*)` must be 0)

pBottomRight Pointer to bottom right edge capacities (`bottomright(*,height-1)` & `bottomright(width-1,*)` must be 0)

nStep Step in bytes between any pair of sequential rows of edge capacities

nTransposedStep Step in bytes between any pair of sequential rows of transposed edge capacities

size Graph size

pLabel Pointer to destination label image

nLabelStep Step in bytes between any pair of sequential rows of label image

pState Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcut8InitAlloc\(\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.52.1.3 NppStatus nppiGraphcut8GetSize (NppiSize *oSize*, int * *pBufSize*)

Calculates the size of the temporary buffer for graph-cut with 8 neighborhood labeling.

NOTE: This Graphcut function will be deprecated in a future release.

See also:

[nppiGraphcut8InitAlloc\(\)](#), [nppiGraphcut8_32s8u\(\)](#).

Parameters:

oSize Graph size.

pBufSize Pointer to variable that returns the size of the temporary buffer.

Returns:

NPP_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP_SIZE_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP_NULL_POINTER_ERROR Indicates an error condition if pBufSize pointer is NULL

7.52.1.4 NppStatus nppiGraphcut8InitAlloc (NppiSize *oSize*, NppiGraphcutState ** *ppState*, Npp8u * *pDeviceMem*)

Allocates and initializes the graph-cut state structure and additional resources for graph-cut with 8 neighborhood labeling.

NOTE: This Graphcut function will be deprecated in a future release.

See also:

[nppiGraphcut8_32s8u\(\)](#), [nppiGraphcut8GetSize\(\)](#).

Parameters:

oSize Graph size

ppState Pointer to pointer to graph-cut state structure.

pDeviceMem to the sufficient amount of device memory. The CUDA runtime or NPP memory allocators must be used to allocate this memory. The minimum amount of device memory required to run graph-cut on a for a specific image size is computed by [nppiGraphcut8GetSize\(\)](#).

Returns:

NPP_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP_SIZE_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP_NULL_POINTER_ERROR Indicates an error condition if pBufSize pointer is NULL

7.52.1.5 NppStatus nppiGraphcut_32f8u (Npp32f * *pTerminals*, Npp32f * *pLeftTransposed*, Npp32f * *pRightTransposed*, Npp32f * *pTop*, Npp32f * *pBottom*, int *nStep*, int *nTransposedStep*, NppiSize *size*, Npp8u * *pLabel*, int *nLabelStep*, NppiGraphcutState * *pState*)

Graphcut of a flow network (32bit float edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 4-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array (`terminals(x) = source(x) - sink(x)`). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example `left(0,*) == 0`). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and >0).

NOTE: This Graphcut function will be deprecated in a future release.

See also:

[nppiGraphcutInitAlloc\(\)](#), [nppiGraphcutFree\(\)](#), [nppiGraphcutGetSize\(\)](#).

Parameters:

pTerminals Pointer to differences of terminal edge capacities (`terminal(x) = source(x) - sink(x)`)
pLeftTransposed Pointer to transposed left edge capacities (`left(0,*)` must be 0)
pRightTransposed Pointer to transposed right edge capacities (`right(width-1,*)` must be 0)
pTop Pointer to top edge capacities (`top(*,0)` must be 0)
pBottom Pointer to bottom edge capacities (`bottom(*,height-1)` must be 0)
nStep Step in bytes between any pair of sequential rows of edge capacities
nTransposedStep Step in bytes between any pair of sequential rows of transposed edge capacities
size Graph size
pLabel Pointer to destination label image
nLabelStep Step in bytes between any pair of sequential rows of label image
pState Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcutInitAlloc\(\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.52.1.6 `NppStatus nppiGraphcut_32s8u (Npp32s * pTerminals, Npp32s * pLeftTransposed, Npp32s * pRightTransposed, Npp32s * pTop, Npp32s * pBottom, int nStep, int nTransposedStep, NppiSize size, Npp8u * pLabel, int nLabelStep, NppiGraphcutState * pState)`

Graphcut of a flow network (32bit signed integer edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 4-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array (`terminals(x) = source(x) - sink(x)`). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example `left(0,*) == 0`). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and >0).

NOTE: This Graphcut function will be deprecated in a future release.

See also:

[nppiGraphcutInitAlloc\(\)](#), [nppiGraphcutFree\(\)](#), [nppiGraphcutGetSize\(\)](#).

Parameters:

pTerminals Pointer to differences of terminal edge capacities ($\text{terminal}(x) = \text{source}(x) - \text{sink}(x)$)

pLeftTransposed Pointer to transposed left edge capacities ($\text{left}(0,*)$ must be 0)

pRightTransposed Pointer to transposed right edge capacities ($\text{right}(\text{width}-1,*)$ must be 0)

pTop Pointer to top edge capacities ($\text{top}(*,0)$ must be 0)

pBottom Pointer to bottom edge capacities ($\text{bottom}(*,\text{height}-1)$ must be 0)

nStep Step in bytes between any pair of sequential rows of edge capacities

nTransposedStep Step in bytes between any pair of sequential rows of transposed edge capacities

size Graph size

pLabel Pointer to destination label image

nLabelStep Step in bytes between any pair of sequential rows of label image

pState Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcutInitAlloc\(\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.52.1.7 NppStatus nppiGraphcutFree (NppiGraphcutState * pState)

Frees the additional resources of the graph-cut state structure.

NOTE: This Graphcut function will be deprecated in a future release.

See also:

[nppiGraphcutInitAlloc](#)
[nppiGraphcut8InitAlloc](#)

Parameters:

pState Pointer to graph-cut state structure.

Returns:

NPP_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP_SIZE_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP_NULL_POINTER_ERROR Indicates an error condition if pState pointer is NULL

7.52.1.8 NppStatus nppiGraphcutGetSize (NppiSize oSize, int * pBufSize)

Calculates the size of the temporary buffer for graph-cut with 4 neighborhood labeling.

NOTE: This Graphcut function will be deprecated in a future release.

See also:

[nppiGraphcutInitAlloc\(\)](#), [nppiGraphcut_32s8u\(\)](#).

Parameters:

oSize Graph size.

pBufSize Pointer to variable that returns the size of the temporary buffer.

Returns:

NPP_SUCCESS Indicates no error. Any other value indicates an error or a warning
NPP_SIZE_ERROR Indicates an error condition if any image dimension has zero or negative value
NPP_NULL_POINTER_ERROR Indicates an error condition if pBufSize pointer is NULL

7.52.1.9 NppStatus nppiGraphcutInitAlloc (NppiSize oSize, NppiGraphcutState ** ppState, Npp8u * pDeviceMem)

Initializes graph-cut state structure and allocates additional resources for graph-cut with 8 neighborhood labeling.

NOTE: This Graphcut function will be deprecated in a future release.

See also:

[nppiGraphcut_32s8u\(\)](#), [nppiGraphcutGetSize\(\)](#).

Parameters:

oSize Graph size

ppState Pointer to pointer to graph-cut state structure.

pDeviceMem pDeviceMem to the sufficient amount of device memory. The CUDA runtime or NPP memory allocators must be used to allocate this memory. The minimum amount of device memory required to run graph-cut on a for a specific image size is computed by [nppiGraphcutGetSize\(\)](#).

Returns:

NPP_SUCCESS Indicates no error. Any other value indicates an error or a warning
NPP_SIZE_ERROR Indicates an error condition if any image dimension has zero or negative value
NPP_NULL_POINTER_ERROR Indicates an error condition if pBufSize pointer is NULL

7.53 Data Exchange and Initialization

Primitives for initializing, copying and converting image data.

Modules

- [Set](#)

Primitives for setting pixels to a specific value.

- [Copy](#)
- [Convert](#)
- [Scale](#)
- [Copy Constant Border](#)
- [Copy Replicate Border](#)
- [Copy Wrap Border](#)
- [Copy Sub-Pixel](#)
- [Duplicate Channel](#)
- [Transpose](#)
- [Swap Channels](#)

7.53.1 Detailed Description

Primitives for initializing, copying and converting image data.

7.54 Set

Primitives for setting pixels to a specific value.

Set

Set all pixels within the ROI to a specific value.

- **NppStatus nppiSet_8s_C1R** (const **Npp8s** nValue, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
8-bit image set.
- **NppStatus nppiSet_8s_C2R** (const **Npp8s** aValue[2], **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
8-bit two-channel image set.
- **NppStatus nppiSet_8s_C3R** (const **Npp8s** aValue[3], **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
8-bit three-channel image set.
- **NppStatus nppiSet_8s_C4R** (const **Npp8s** aValue[4], **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
8-bit four-channel image set.
- **NppStatus nppiSet_8s_AC4R** (const **Npp8s** aValue[3], **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
8-bit four-channel image set ignoring alpha channel.
- **NppStatus nppiSet_8u_C1R** (const **Npp8u** nValue, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
8-bit unsigned image set.
- **NppStatus nppiSet_8u_C2R** (const **Npp8u** aValue[2], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
2 channel 8-bit unsigned image set.
- **NppStatus nppiSet_8u_C3R** (const **Npp8u** aValue[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 8-bit unsigned image set.
- **NppStatus nppiSet_8u_C4R** (const **Npp8u** aValue[4], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned image set.
- **NppStatus nppiSet_8u_AC4R** (const **Npp8u** aValue[3], **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned image set method, not affecting Alpha channel.
- **NppStatus nppiSet_16u_C1R** (const **Npp16u** nValue, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
16-bit unsigned image set.

- `NppStatus nppiSet_16u_C2R` (const `Npp16u` aValue[2], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
2 channel 16-bit unsigned image set.
- `NppStatus nppiSet_16u_C3R` (const `Npp16u` aValue[3], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 16-bit unsigned image set.
- `NppStatus nppiSet_16u_C4R` (const `Npp16u` aValue[4], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 16-bit unsigned image set.
- `NppStatus nppiSet_16u_AC4R` (const `Npp16u` aValue[3], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 16-bit unsigned image set method, not affecting Alpha channel.
- `NppStatus nppiSet_16s_C1R` (const `Npp16s` nValue, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
16-bit image set.
- `NppStatus nppiSet_16s_C2R` (const `Npp16s` aValue[2], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
2 channel 16-bit image set.
- `NppStatus nppiSet_16s_C3R` (const `Npp16s` aValue[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 16-bit image set.
- `NppStatus nppiSet_16s_C4R` (const `Npp16s` aValue[4], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 16-bit image set.
- `NppStatus nppiSet_16s_AC4R` (const `Npp16s` aValue[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 16-bit image set method, not affecting Alpha channel.
- `NppStatus nppiSet_16sc_C1R` (const `Npp16sc` oValue, `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
16-bit complex integer image set.
- `NppStatus nppiSet_16sc_C2R` (const `Npp16sc` aValue[2], `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
16-bit complex integer two-channel image set.
- `NppStatus nppiSet_16sc_C3R` (const `Npp16sc` aValue[3], `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
16-bit complex integer three-channel image set.
- `NppStatus nppiSet_16sc_C4R` (const `Npp16sc` aValue[4], `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI)

16-bit complex integer four-channel image set.

- **NppStatus nppiSet_16sc_AC4R** (const **Npp16sc** aValue[3], **Npp16sc** *pDst, int nDstStep, **NppiSize** oSizeROI)

16-bit complex integer four-channel image set ignoring alpha.

- **NppStatus nppiSet_32s_C1R** (const **Npp32s** nValue, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)

32-bit image set.

- **NppStatus nppiSet_32s_C2R** (const **Npp32s** aValue[2], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)

2 channel 32-bit image set.

- **NppStatus nppiSet_32s_C3R** (const **Npp32s** aValue[3], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 32-bit image set.

- **NppStatus nppiSet_32s_C4R** (const **Npp32s** aValue[4], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 32-bit image set.

- **NppStatus nppiSet_32s_AC4R** (const **Npp32s** aValue[3], **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 32-bit image set method, not affecting Alpha channel.

- **NppStatus nppiSet_32u_C1R** (const **Npp32u** nValue, **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI)

32-bit unsigned image set.

- **NppStatus nppiSet_32u_C2R** (const **Npp32u** aValue[2], **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI)

2 channel 32-bit unsigned image set.

- **NppStatus nppiSet_32u_C3R** (const **Npp32u** aValue[3], **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI)

3 channel 32-bit unsigned image set.

- **NppStatus nppiSet_32u_C4R** (const **Npp32u** aValue[4], **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 32-bit unsigned image set.

- **NppStatus nppiSet_32u_AC4R** (const **Npp32u** aValue[3], **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 32-bit unsigned image set method, not affecting Alpha channel.

- **NppStatus nppiSet_32sc_C1R** (const **Npp32sc** oValue, **Npp32sc** *pDst, int nDstStep, **NppiSize** oSizeROI)

Single channel 32-bit complex integer image set.

- [NppStatus nppiSet_32sc_C2R](#) (const [Npp32sc](#) aValue[2], [Npp32sc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Two channel 32-bit complex integer image set.
- [NppStatus nppiSet_32sc_C3R](#) (const [Npp32sc](#) aValue[3], [Npp32sc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Three channel 32-bit complex integer image set.
- [NppStatus nppiSet_32sc_C4R](#) (const [Npp32sc](#) aValue[4], [Npp32sc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Four channel 32-bit complex integer image set.
- [NppStatus nppiSet_32sc_AC4R](#) (const [Npp32sc](#) aValue[3], [Npp32sc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
32-bit complex integer four-channel image set ignoring alpha.
- [NppStatus nppiSet_32f_C1R](#) (const [Npp32f](#) nValue, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
32-bit floating point image set.
- [NppStatus nppiSet_32f_C2R](#) (const [Npp32f](#) aValue[2], [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
2 channel 32-bit floating point image set.
- [NppStatus nppiSet_32f_C3R](#) (const [Npp32f](#) aValue[3], [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
3 channel 32-bit floating point image set.
- [NppStatus nppiSet_32f_C4R](#) (const [Npp32f](#) aValue[4], [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
4 channel 32-bit floating point image set.
- [NppStatus nppiSet_32f_AC4R](#) (const [Npp32f](#) aValue[3], [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
4 channel 32-bit floating point image set method, not affecting Alpha channel.
- [NppStatus nppiSet_32fc_C1R](#) (const [Npp32fc](#) oValue, [Npp32fc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Single channel 32-bit complex image set.
- [NppStatus nppiSet_32fc_C2R](#) (const [Npp32fc](#) aValue[2], [Npp32fc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Two channel 32-bit complex image set.
- [NppStatus nppiSet_32fc_C3R](#) (const [Npp32fc](#) aValue[3], [Npp32fc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Three channel 32-bit complex image set.
- [NppStatus nppiSet_32fc_C4R](#) (const [Npp32fc](#) aValue[4], [Npp32fc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Four channel 32-bit complex image set.

- `NppStatus nppiSet_32fc_AC4R` (const `Npp32fc` aValue[3], `Npp32fc` *pDst, int nDstStep, `NppiSize` oSizeROI)

32-bit complex four-channel image set ignoring alpha.

Masked Set

The masked set primitives have an additional "mask image" input.

The mask controls which pixels within the ROI are set. For details see [Masked Operation](#).

- `NppStatus nppiSet_8u_C1MR` (`Npp8u` nValue, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked 8-bit unsigned image set.

- `NppStatus nppiSet_8u_C3MR` (const `Npp8u` aValue[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked 3 channel 8-bit unsigned image set.

- `NppStatus nppiSet_8u_C4MR` (const `Npp8u` aValue[4], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked 4 channel 8-bit unsigned image set.

- `NppStatus nppiSet_8u_AC4MR` (const `Npp8u` aValue[3], `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked 4 channel 8-bit unsigned image set method, not affecting Alpha channel.

- `NppStatus nppiSet_16u_C1MR` (`Npp16u` nValue, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked 16-bit unsigned image set.

- `NppStatus nppiSet_16u_C3MR` (const `Npp16u` aValue[3], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked 3 channel 16-bit unsigned image set.

- `NppStatus nppiSet_16u_C4MR` (const `Npp16u` aValue[4], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked 4 channel 16-bit unsigned image set.

- `NppStatus nppiSet_16u_AC4MR` (const `Npp16u` aValue[3], `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked 4 channel 16-bit unsigned image set method, not affecting Alpha channel.

- `NppStatus nppiSet_16s_C1MR` (`Npp16s` nValue, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked 16-bit image set.

- `NppStatus nppiSet_16s_C3MR` (const `Npp16s` aValue[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked 3 channel 16-bit image set.

- `NppStatus nppiSet_16s_C4MR` (const `Npp16s` aValue[4], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)
Masked 4 channel 16-bit image set.
- `NppStatus nppiSet_16s_AC4MR` (const `Npp16s` aValue[3], `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)
Masked 4 channel 16-bit image set method, not affecting Alpha channel.
- `NppStatus nppiSet_32s_C1MR` (`Npp32s` nValue, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)
Masked 32-bit image set.
- `NppStatus nppiSet_32s_C3MR` (const `Npp32s` aValue[3], `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)
Masked 3 channel 32-bit image set.
- `NppStatus nppiSet_32s_C4MR` (const `Npp32s` aValue[4], `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)
Masked 4 channel 32-bit image set.
- `NppStatus nppiSet_32s_AC4MR` (const `Npp32s` aValue[3], `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)
Masked 4 channel 16-bit image set method, not affecting Alpha channel.
- `NppStatus nppiSet_32f_C1MR` (`Npp32f` nValue, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)
Masked 32-bit floating point image set.
- `NppStatus nppiSet_32f_C3MR` (const `Npp32f` aValue[3], `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)
Masked 3 channel 32-bit floating point image set.
- `NppStatus nppiSet_32f_C4MR` (const `Npp32f` aValue[4], `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)
Masked 4 channel 32-bit floating point image set.
- `NppStatus nppiSet_32f_AC4MR` (const `Npp32f` aValue[3], `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)
Masked 4 channel 32-bit floating point image set method, not affecting Alpha channel.

Channel Set

The select-channel set primitives set a single color channel in multi-channel images to a given value.

The channel is selected by adjusting the pDst pointer to point to the desired color channel (see [Channel-of-Interest API](#)).

- `NppStatus nppiSet_8u_C3CR` (`Npp8u` nValue, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)
3 channel 8-bit unsigned image set affecting only single channel.

- **NppStatus nppiSet_8u_C4CR** (**Npp8u** nValue, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned image set affecting only single channel.
- **NppStatus nppiSet_16u_C3CR** (**Npp16u** nValue, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 16-bit unsigned image set affecting only single channel.
- **NppStatus nppiSet_16u_C4CR** (**Npp16u** nValue, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 16-bit unsigned image set affecting only single channel.
- **NppStatus nppiSet_16s_C3CR** (**Npp16s** nValue, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 16-bit signed image set affecting only single channel.
- **NppStatus nppiSet_16s_C4CR** (**Npp16s** nValue, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 16-bit signed image set affecting only single channel.
- **NppStatus nppiSet_32s_C3CR** (**Npp32s** nValue, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 32-bit unsigned image set affecting only single channel.
- **NppStatus nppiSet_32s_C4CR** (**Npp32s** nValue, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 32-bit unsigned image set affecting only single channel.
- **NppStatus nppiSet_32f_C3CR** (**Npp32f** nValue, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
3 channel 32-bit floating point image set affecting only single channel.
- **NppStatus nppiSet_32f_C4CR** (**Npp32f** nValue, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 32-bit floating point image set affecting only single channel.

7.54.1 Detailed Description

Primitives for setting pixels to a specific value.

7.54.2 Function Documentation

7.54.2.1 NppStatus nppiSet_16s_AC4MR (const Npp16s aValue[3], Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const Npp8u *pMask, int nMaskStep)

Masked 4 channel 16-bit image set method, not affecting Alpha channel.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep [Mask-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.2 NppStatus nppiSet_16s_AC4R (const Npp16s aValue[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 16-bit image set method, not affecting Alpha channel.

Parameters:

aValue The pixel-value to be set.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.3 NppStatus nppiSet_16s_C1MR (Npp16s nValue, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)

Masked 16-bit image set.

Parameters:

nValue The pixel-value to be set.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.4 NppStatus nppiSet_16s_C1R (const Npp16s nValue, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)

16-bit image set.

Parameters:

nValue The pixel-value to be set.

pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.5 NppStatus nppiSet_16s_C2R (const Npp16s aValue[2], Npp16s * pDst, int nDstStep, NppiSize oSizeROI)

2 channel 16-bit image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.6 NppStatus nppiSet_16s_C3CR (Npp16s nValue, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 16-bit signed image set affecting only single channel.

Parameters:

nValue The pixel-value to be set.

pDst Select-Channel Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.7 NppStatus nppiSet_16s_C3MR (const Npp16s aValue[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)

Masked 3 channel 16-bit image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.8 NppStatus nppiSet_16s_C3R (const Npp16s aValue[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 16-bit image set.

Parameters:

aValue The pixel-value to be set.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.9 NppStatus nppiSet_16s_C4CR (Npp16s nValue, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 16-bit signed image set affecting only single channel.

Parameters:

nValue The pixel-value to be set.

pDst [Select-Channel Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.10 NppStatus nppiSet_16s_C4MR (const Npp16s aValue[4], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)

Masked 4 channel 16-bit image set.

Parameters:

aValue The pixel-value to be set.

pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.11 NppStatus nppiSet_16s_C4R (const Npp16s aValue[4], Npp16s * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 16-bit image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.12 NppStatus nppiSet_16sc_AC4R (const Npp16sc aValue[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI)

16-bit complex integer four-channel image set ignoring alpha.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.13 NppStatus nppiSet_16sc_C1R (const Npp16sc oValue, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI)

16-bit complex integer image set.

Parameters:

oValue The pixel-value to be set.

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.14 NppStatus nppiSet_16sc_C2R (const Npp16sc aValue[2], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI)

16-bit complex integer two-channel image set.

Parameters:

aValue The pixel-value to be set.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.15 NppStatus nppiSet_16sc_C3R (const Npp16sc aValue[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI)

16-bit complex integer three-channel image set.

Parameters:

aValue The pixel-value to be set.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.16 NppStatus nppiSet_16sc_C4R (const Npp16sc aValue[4], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI)

16-bit complex integer four-channel image set.

Parameters:

aValue The pixel-value to be set.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.17 `NppStatus nppiSet_16u_AC4MR (const Npp16u aValue[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 16-bit unsigned image set method, not affecting Alpha channel.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.18 `NppStatus nppiSet_16u_AC4R (const Npp16u aValue[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image set method, not affecting Alpha channel.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.19 `NppStatus nppiSet_16u_C1MR (Npp16u nValue, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 16-bit unsigned image set.

Parameters:

nValue The pixel-value to be set.

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.20 NppStatus nppiSet_16u_C1R (const Npp16u *nValue*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

16-bit unsigned image set.

Parameters:

nValue The pixel-value to be set.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.21 NppStatus nppiSet_16u_C2R (const Npp16u *aValue*[2], Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 16-bit unsigned image set.

Parameters:

aValue The pixel-value to be set.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.22 NppStatus nppiSet_16u_C3CR (Npp16u *nValue*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 16-bit unsigned image set affecting only single channel.

Parameters:

nValue The pixel-value to be set.

pDst Select-Channel Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.23 `NppStatus nppiSet_16u_C3MR (const Npp16u aValue[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 3 channel 16-bit unsigned image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.24 `NppStatus nppiSet_16u_C3R (const Npp16u aValue[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 16-bit unsigned image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.25 `NppStatus nppiSet_16u_C4CR (Npp16u nValue, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image set affecting only single channel.

Parameters:

nValue The pixel-value to be set.

pDst Select-Channel Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.26 `NppStatus nppiSet_16u_C4MR (const Npp16u aValue[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 16-bit unsigned image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.27 `NppStatus nppiSet_16u_C4R (const Npp16u aValue[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.28 `NppStatus nppiSet_32f_AC4MR (const Npp32f aValue[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 32-bit floating point image set method, not affecting Alpha channel.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.29 NppStatus nppiSet_32f_AC4R (const Npp32f aValue[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 32-bit floating point image set method, not affecting Alpha channel.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.30 NppStatus nppiSet_32f_C1MR (Npp32f nValue, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)

Masked 32-bit floating point image set.

Parameters:

nValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.31 NppStatus nppiSet_32f_C1R (const Npp32f *nValue*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

32-bit floating point image set.

Parameters:

nValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.32 NppStatus nppiSet_32f_C2R (const Npp32f *aValue*[2], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 32-bit floating point image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.33 NppStatus nppiSet_32f_C3CR (Npp32f *nValue*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 32-bit floating point image set affecting only single channel.

Parameters:

nValue The pixel-value to be set.

pDst Select-Channel Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.34 NppStatus nppiSet_32f_C3MR (const Npp32f *aValue*[3], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

Masked 3 channel 32-bit floating point image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.35 NppStatus nppiSet_32f_C3R (const Npp32f *aValue*[3], Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 32-bit floating point image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.36 NppStatus nppiSet_32f_C4CR (Npp32f *nValue*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit floating point image set affecting only single channel.

Parameters:

nValue The pixel-value to be set.

pDst Select-Channel Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.37 `NppStatus nppiSet_32f_C4MR (const Npp32f aValue[4], Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 32-bit floating point image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.38 `NppStatus nppiSet_32f_C4R (const Npp32f aValue[4], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit floating point image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.39 `NppStatus nppiSet_32fc_AC4R (const Npp32fc aValue[3], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit complex four-channel image set ignoring alpha.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.40 NppStatus nppiSet_32fc_C1R (const Npp32fc oValue, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)

Single channel 32-bit complex image set.

Parameters:

oValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.41 NppStatus nppiSet_32fc_C2R (const Npp32fc aValue[2], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)

Two channel 32-bit complex image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.42 NppStatus nppiSet_32fc_C3R (const Npp32fc aValue[3], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)

Three channel 32-bit complex image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.43 NppStatus nppiSet_32fc_C4R (const Npp32fc *aValue*[4], Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit complex image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.44 NppStatus nppiSet_32s_AC4MR (const Npp32s *aValue*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

Masked 4 channel 16-bit image set method, not affecting Alpha channel.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.45 NppStatus nppiSet_32s_AC4R (const Npp32s *aValue*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit image set method, not affecting Alpha channel.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.46 `NppStatus nppiSet_32s_C1MR (Npp32s nValue, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 32-bit image set.

Parameters:

nValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.47 `NppStatus nppiSet_32s_C1R (const Npp32s nValue, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit image set.

Parameters:

nValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.48 `NppStatus nppiSet_32s_C2R (const Npp32s aValue[2], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 32-bit image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.49 NppStatus nppiSet_32s_C3CR (Npp32s *nValue*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 32-bit unsigned image set affecting only single channel.

Parameters:

nValue The pixel-value to be set.
pDst [Select-Channel Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.50 NppStatus nppiSet_32s_C3MR (const Npp32s *aValue*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

Masked 3 channel 32-bit image set.

Parameters:

aValue The pixel-value to be set.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pMask [Mask-Image Pointer](#).
nMaskStep [Mask-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.51 NppStatus nppiSet_32s_C3R (const Npp32s *aValue*[3], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 32-bit image set.

Parameters:

aValue The pixel-value to be set.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.52 NppStatus nppiSet_32s_C4CR (Npp32s *nValue*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit unsigned image set affecting only single channel.

Parameters:

nValue The pixel-value to be set.
pDst [Select-Channel Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.53 NppStatus nppiSet_32s_C4MR (const Npp32s *aValue*[4], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

Masked 4 channel 32-bit image set.

Parameters:

aValue The pixel-value to be set.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pMask [Mask-Image Pointer](#).
nMaskStep [Mask-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.54 NppStatus nppiSet_32s_C4R (const Npp32s *aValue*[4], Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit image set.

Parameters:

aValue The pixel-value to be set.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.55 NppStatus nppiSet_32sc_AC4R (const Npp32sc *aValue*[3], Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

32-bit complex integer four-channel image set ignoring alpha.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.56 NppStatus nppiSet_32sc_C1R (const Npp32sc *oValue*, Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit complex integer image set.

Parameters:

oValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.57 NppStatus nppiSet_32sc_C2R (const Npp32sc *aValue*[2], Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Two channel 32-bit complex integer image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.58 NppStatus nppiSet_32sc_C3R (const Npp32sc aValue[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)

Three channel 32-bit complex integer image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.59 NppStatus nppiSet_32sc_C4R (const Npp32sc aValue[4], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)

Four channel 32-bit complex integer image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.60 NppStatus nppiSet_32u_AC4R (const Npp32u aValue[3], Npp32u * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 32-bit unsigned image set method, not affecting Alpha channel.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.61 NppStatus nppiSet_32u_C1R (const Npp32u *nValue*, Npp32u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

32-bit unsigned image set.

Parameters:

nValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.62 NppStatus nppiSet_32u_C2R (const Npp32u *aValue*[2], Npp32u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 32-bit unsigned image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.63 NppStatus nppiSet_32u_C3R (const Npp32u *aValue*[3], Npp32u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 32-bit unsigned image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.64 NppStatus nppiSet_32u_C4R (const Npp32u aValue[4], Npp32u * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 32-bit unsigned image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.65 NppStatus nppiSet_8s_AC4R (const Npp8s aValue[3], Npp8s * pDst, int nDstStep, NppiSize oSizeROI)

8-bit four-channel image set ignoring alpha channel.

Parameters:

aValue The pixel value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.66 NppStatus nppiSet_8s_C1R (const Npp8s nValue, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)

8-bit image set.

Parameters:

nValue The pixel value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.54.2.67 NppStatus nppiSet_8s_C2R (const Npp8s *aValue*[2], Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

8-bit two-channel image set.

Parameters:

aValue The pixel value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.68 NppStatus nppiSet_8s_C3R (const Npp8s *aValue*[3], Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

8-bit three-channel image set.

Parameters:

aValue The pixel value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.69 NppStatus nppiSet_8s_C4R (const Npp8s *aValue*[4], Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

8-bit four-channel image set.

Parameters:

aValue The pixel value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.70 `NppStatus nppiSet_8u_AC4MR (const Npp8u aValue[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 8-bit unsigned image set method, not affecting Alpha channel.

Parameters:

aValue The pixel-value to be set.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.71 `NppStatus nppiSet_8u_AC4R (const Npp8u aValue[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image set method, not affecting Alpha channel.

Parameters:

aValue The pixel-value to be set.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.72 `NppStatus nppiSet_8u_C1MR (Npp8u nValue, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 8-bit unsigned image set.

Parameters:

nValue The pixel value to be set.
pDst Pointer Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.73 NppStatus nppiSet_8u_C1R (const Npp8u *nValue*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

8-bit unsigned image set.

Parameters:

nValue The pixel value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.74 NppStatus nppiSet_8u_C2R (const Npp8u *aValue*[2], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned image set.

Parameters:

aValue The pixel value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.75 NppStatus nppiSet_8u_C3CR (Npp8u *nValue*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned image set affecting only single channel.

Parameters:

nValue The pixel-value to be set.

pDst Select-Channel Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.76 NppStatus nppiSet_8u_C3MR (const Npp8u aValue[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)

Masked 3 channel 8-bit unsigned image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.77 NppStatus nppiSet_8u_C3R (const Npp8u aValue[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned image set.

Parameters:

aValue The pixel value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.78 NppStatus nppiSet_8u_C4CR (Npp8u nValue, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned image set affecting only single channel.

Parameters:

nValue The pixel-value to be set.

pDst Select-Channel Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.79 NppStatus nppiSet_8u_C4MR (const Npp8u *aValue*[4], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

Masked 4 channel 8-bit unsigned image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.54.2.80 NppStatus nppiSet_8u_C4R (const Npp8u *aValue*[4], Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned image set.

Parameters:

aValue The pixel-value to be set.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55 Copy

Copy

Copy pixels from one image to another.

- **NppStatus nppiCopy_8s_C1R** (const **Npp8s** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
8-bit image copy.
- **NppStatus nppiCopy_8s_C2R** (const **Npp8s** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Two-channel 8-bit image copy.
- **NppStatus nppiCopy_8s_C3R** (const **Npp8s** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three-channel 8-bit image copy.
- **NppStatus nppiCopy_8s_C4R** (const **Npp8s** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 8-bit image copy.
- **NppStatus nppiCopy_8s_AC4R** (const **Npp8s** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 8-bit image copy, ignoring alpha channel.
- **NppStatus nppiCopy_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
8-bit unsigned image copy.
- **NppStatus nppiCopy_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three channel 8-bit unsigned image copy.
- **NppStatus nppiCopy_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned image copy.
- **NppStatus nppiCopy_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
4 channel 8-bit unsigned image copy, not affecting Alpha channel.
- **NppStatus nppiCopy_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
16-bit unsigned image copy.
- **NppStatus nppiCopy_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three channel 16-bit unsigned image copy.

- `NppStatus nppiCopy_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 16-bit unsigned image copy.
- `NppStatus nppiCopy_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 16-bit unsigned image copy, not affecting Alpha channel.
- `NppStatus nppiCopy_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
16-bit image copy.
- `NppStatus nppiCopy_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three channel 16-bit image copy.
- `NppStatus nppiCopy_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 16-bit image copy.
- `NppStatus nppiCopy_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 16-bit image copy, not affecting Alpha.
- `NppStatus nppiCopy_16sc_C1R` (const `Npp16sc` *pSrc, int nSrcStep, `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
16-bit complex image copy.
- `NppStatus nppiCopy_16sc_C2R` (const `Npp16sc` *pSrc, int nSrcStep, `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
Two-channel 16-bit complex image copy.
- `NppStatus nppiCopy_16sc_C3R` (const `Npp16sc` *pSrc, int nSrcStep, `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three-channel 16-bit complex image copy.
- `NppStatus nppiCopy_16sc_C4R` (const `Npp16sc` *pSrc, int nSrcStep, `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four-channel 16-bit complex image copy.
- `NppStatus nppiCopy_16sc_AC4R` (const `Npp16sc` *pSrc, int nSrcStep, `Npp16sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four-channel 16-bit complex image copy, ignoring alpha.
- `NppStatus nppiCopy_32s_C1R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
32-bit image copy.
- `NppStatus nppiCopy_32s_C3R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three channel 32-bit image copy.

- `NppStatus nppiCopy_32s_C4R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 32-bit image copy.
- `NppStatus nppiCopy_32s_AC4R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 32-bit image copy, not affecting Alpha.
- `NppStatus nppiCopy_32sc_C1R` (const `Npp32sc` *pSrc, int nSrcStep, `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
32-bit complex image copy.
- `NppStatus nppiCopy_32sc_C2R` (const `Npp32sc` *pSrc, int nSrcStep, `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
Two-channel 32-bit complex image copy.
- `NppStatus nppiCopy_32sc_C3R` (const `Npp32sc` *pSrc, int nSrcStep, `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three-channel 32-bit complex image copy.
- `NppStatus nppiCopy_32sc_C4R` (const `Npp32sc` *pSrc, int nSrcStep, `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four-channel 32-bit complex image copy.
- `NppStatus nppiCopy_32sc_AC4R` (const `Npp32sc` *pSrc, int nSrcStep, `Npp32sc` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four-channel 32-bit complex image copy, ignoring alpha.
- `NppStatus nppiCopy_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)
32-bit floating point image copy.
- `NppStatus nppiCopy_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three channel 32-bit floating point image copy.
- `NppStatus nppiCopy_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 32-bit floating point image copy.
- `NppStatus nppiCopy_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)
4 channel 32-bit floating point image copy, not affecting Alpha.
- `NppStatus nppiCopy_32fc_C1R` (const `Npp32fc` *pSrc, int nSrcStep, `Npp32fc` *pDst, int nDstStep, `NppiSize` oSizeROI)
32-bit floating-point complex image copy.
- `NppStatus nppiCopy_32fc_C2R` (const `Npp32fc` *pSrc, int nSrcStep, `Npp32fc` *pDst, int nDstStep, `NppiSize` oSizeROI)

Two-channel 32-bit floating-point complex image copy.

- [NppStatus nppiCopy_32fc_C3R](#) (const [Npp32fc](#) *pSrc, int nSrcStep, [Npp32fc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)

Three-channel 32-bit floating-point complex image copy.

- [NppStatus nppiCopy_32fc_C4R](#) (const [Npp32fc](#) *pSrc, int nSrcStep, [Npp32fc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)

Four-channel 32-bit floating-point complex image copy.

- [NppStatus nppiCopy_32fc_AC4R](#) (const [Npp32fc](#) *pSrc, int nSrcStep, [Npp32fc](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)

Four-channel 32-bit floating-point complex image copy, ignoring alpha.

Masked Copy

The masked copy primitives have an additional "mask image" input.

The mask controls which pixels within the ROI are copied. For details see [Masked Operation](#).

- [NppStatus nppiCopy_8u_C1MR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) *pMask, int nMaskStep)

Masked Operation 8-bit unsigned image copy.

- [NppStatus nppiCopy_8u_C3MR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) *pMask, int nMaskStep)

Masked Operation three channel 8-bit unsigned image copy.

- [NppStatus nppiCopy_8u_C4MR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) *pMask, int nMaskStep)

Masked Operation four channel 8-bit unsigned image copy.

- [NppStatus nppiCopy_8u_AC4MR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) *pMask, int nMaskStep)

Masked Operation four channel 8-bit unsigned image copy, ignoring alpha.

- [NppStatus nppiCopy_16u_C1MR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) *pMask, int nMaskStep)

Masked Operation 16-bit unsigned image copy.

- [NppStatus nppiCopy_16u_C3MR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) *pMask, int nMaskStep)

Masked Operation three channel 16-bit unsigned image copy.

- [NppStatus nppiCopy_16u_C4MR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) *pMask, int nMaskStep)

Masked Operation four channel 16-bit unsigned image copy.

- [NppStatus nppiCopy_16u_AC4MR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) *pMask, int nMaskStep)

Masked Operation four channel 16-bit unsigned image copy, ignoring alpha.

- `NppStatus nppiCopy_16s_C1MR` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation 16-bit signed image copy.

- `NppStatus nppiCopy_16s_C3MR` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation three channel 16-bit signed image copy.

- `NppStatus nppiCopy_16s_C4MR` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation four channel 16-bit signed image copy.

- `NppStatus nppiCopy_16s_AC4MR` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation four channel 16-bit signed image copy, ignoring alpha.

- `NppStatus nppiCopy_32s_C1MR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation 32-bit signed image copy.

- `NppStatus nppiCopy_32s_C3MR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation three channel 32-bit signed image copy.

- `NppStatus nppiCopy_32s_C4MR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation four channel 32-bit signed image copy.

- `NppStatus nppiCopy_32s_AC4MR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation four channel 32-bit signed image copy, ignoring alpha.

- `NppStatus nppiCopy_32f_C1MR` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation 32-bit float image copy.

- `NppStatus nppiCopy_32f_C3MR` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation three channel 32-bit float image copy.

- `NppStatus nppiCopy_32f_C4MR` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation four channel 32-bit float image copy.

- `NppStatus nppiCopy_32f_AC4MR` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, int nMaskStep)

Masked Operation four channel 32-bit float image copy, ignoring alpha.

Channel Copy

The channel copy primitives copy a single color channel from a multi-channel source image to any other color channel in a multi-channel destination image.

The channel is selected by adjusting the respective image pointers to point to the desired color channel (see [Channel-of-Interest API](#)).

- `NppStatus nppiCopy_8u_C3CR` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Select-channel 8-bit unsigned image copy for three-channel images.

- `NppStatus nppiCopy_8u_C4CR` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Select-channel 8-bit unsigned image copy for four-channel images.

- `NppStatus nppiCopy_16s_C3CR` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Select-channel 16-bit signed image copy for three-channel images.

- `NppStatus nppiCopy_16s_C4CR` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Select-channel 16-bit signed image copy for four-channel images.

- `NppStatus nppiCopy_16u_C3CR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Select-channel 16-bit unsigned image copy for three-channel images.

- `NppStatus nppiCopy_16u_C4CR` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Select-channel 16-bit unsigned image copy for four-channel images.

- `NppStatus nppiCopy_32s_C3CR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Select-channel 32-bit signed image copy for three-channel images.

- `NppStatus nppiCopy_32s_C4CR` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Select-channel 32-bit signed image copy for four-channel images.

- `NppStatus nppiCopy_32f_C3CR` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Select-channel 32-bit float image copy for three-channel images.

- `NppStatus nppiCopy_32f_C4CR` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Select-channel 32-bit float image copy for four-channel images.

Extract Channel Copy

The channel extract primitives copy a single color channel from a multi-channel source image to single-channel destination image.

The channel is selected by adjusting the source image pointer to point to the desired color channel (see [Channel-of-Interest API](#)).

- `NppStatus nppiCopy_8u_C3C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three-channel to single-channel 8-bit unsigned image copy.

- `NppStatus nppiCopy_8u_C4C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four-channel to single-channel 8-bit unsigned image copy.

- `NppStatus nppiCopy_16s_C3C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three-channel to single-channel 16-bit signed image copy.

- `NppStatus nppiCopy_16s_C4C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four-channel to single-channel 16-bit signed image copy.

- `NppStatus nppiCopy_16u_C3C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three-channel to single-channel 16-bit unsigned image copy.

- `NppStatus nppiCopy_16u_C4C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four-channel to single-channel 16-bit unsigned image copy.

- `NppStatus nppiCopy_32s_C3C1R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three-channel to single-channel 32-bit signed image copy.

- `NppStatus nppiCopy_32s_C4C1R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four-channel to single-channel 32-bit signed image copy.

- `NppStatus nppiCopy_32f_C3C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three-channel to single-channel 32-bit float image copy.

- `NppStatus nppiCopy_32f_C4C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four-channel to single-channel 32-bit float image copy.

Insert Channel Copy

The channel insert primitives copy a single-channel source image into one of the color channels in a multi-channel destination image.

The channel is selected by adjusting the destination image pointer to point to the desired color channel (see [Channel-of-Interest API](#)).

- `NppStatus nppiCopy_8u_C1C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single-channel to three-channel 8-bit unsigned image copy.

- `NppStatus nppiCopy_8u_C1C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single-channel to four-channel 8-bit unsigned image copy.

- `NppStatus nppiCopy_16s_C1C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single-channel to three-channel 16-bit signed image copy.

- `NppStatus nppiCopy_16s_C1C4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single-channel to four-channel 16-bit signed image copy.

- `NppStatus nppiCopy_16u_C1C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single-channel to three-channel 16-bit unsigned image copy.

- `NppStatus nppiCopy_16u_C1C4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single-channel to four-channel 16-bit unsigned image copy.

- `NppStatus nppiCopy_32s_C1C3R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single-channel to three-channel 32-bit signed image copy.

- `NppStatus nppiCopy_32s_C1C4R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single-channel to four-channel 32-bit signed image copy.

- `NppStatus nppiCopy_32f_C1C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single-channel to three-channel 32-bit float image copy.

- `NppStatus nppiCopy_32f_C1C4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single-channel to four-channel 32-bit float image copy.

Packed-to-Planar Copy

Split a packed multi-channel image into a planar image.

E.g. copy the three channels of an RGB image into three separate single-channel images.

- `NppStatus nppiCopy_8u_C3P3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *const aDst[3], int nDstStep, `NppiSize` oSizeROI)

Three-channel 8-bit unsigned packed to planar image copy.

- `NppStatus nppiCopy_8u_C4P4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *const aDst[4], int nDstStep, `NppiSize` oSizeROI)

Four-channel 8-bit unsigned packed to planar image copy.

- `NppStatus nppiCopy_16s_C3P3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *const aDst[3], int nDstStep, `NppiSize` oSizeROI)

Three-channel 16-bit signed packed to planar image copy.

- `NppStatus nppiCopy_16s_C4P4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *const aDst[4], int nDstStep, `NppiSize` oSizeROI)

Four-channel 16-bit signed packed to planar image copy.

- `NppStatus nppiCopy_16u_C3P3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *const aDst[3], int nDstStep, `NppiSize` oSizeROI)

Three-channel 16-bit unsigned packed to planar image copy.

- `NppStatus nppiCopy_16u_C4P4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *const aDst[4], int nDstStep, `NppiSize` oSizeROI)

Four-channel 16-bit unsigned packed to planar image copy.

- `NppStatus nppiCopy_32s_C3P3R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *const aDst[3], int nDstStep, `NppiSize` oSizeROI)

Three-channel 32-bit signed packed to planar image copy.

- `NppStatus nppiCopy_32s_C4P4R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *const aDst[4], int nDstStep, `NppiSize` oSizeROI)

Four-channel 32-bit signed packed to planar image copy.

- `NppStatus nppiCopy_32f_C3P3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *const aDst[3], int nDstStep, `NppiSize` oSizeROI)

Three-channel 32-bit float packed to planar image copy.

- `NppStatus nppiCopy_32f_C4P4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *const aDst[4], int nDstStep, `NppiSize` oSizeROI)

Four-channel 32-bit float packed to planar image copy.

Planar-to-Packed Copy

Combine multiple image planes into a packed multi-channel image.

E.g. copy three single-channel images into a single 3-channel image.

- **NppStatus nppiCopy_8u_P3C3R** (const **Npp8u** *const aSrc[3], int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three-channel 8-bit unsigned planar to packed image copy.
- **NppStatus nppiCopy_8u_P4C4R** (const **Npp8u** *const aSrc[4], int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 8-bit unsigned planar to packed image copy.
- **NppStatus nppiCopy_16u_P3C3R** (const **Npp16u** *const aSrc[3], int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three-channel 16-bit unsigned planar to packed image copy.
- **NppStatus nppiCopy_16u_P4C4R** (const **Npp16u** *const aSrc[4], int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 16-bit unsigned planar to packed image copy.
- **NppStatus nppiCopy_16s_P3C3R** (const **Npp16s** *const aSrc[3], int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three-channel 16-bit signed planar to packed image copy.
- **NppStatus nppiCopy_16s_P4C4R** (const **Npp16s** *const aSrc[4], int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 16-bit signed planar to packed image copy.
- **NppStatus nppiCopy_32s_P3C3R** (const **Npp32s** *const aSrc[3], int nSrcStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three-channel 32-bit signed planar to packed image copy.
- **NppStatus nppiCopy_32s_P4C4R** (const **Npp32s** *const aSrc[4], int nSrcStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 32-bit signed planar to packed image copy.
- **NppStatus nppiCopy_32f_P3C3R** (const **Npp32f** *const aSrc[3], int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three-channel 32-bit float planar to packed image copy.
- **NppStatus nppiCopy_32f_P4C4R** (const **Npp32f** *const aSrc[4], int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 32-bit float planar to packed image copy.

7.55.1 Function Documentation

7.55.1.1 **NppStatus nppiCopy_16s_AC4MR** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, int nMaskStep)

Masked Operation four channel 16-bit signed image copy, ignoring alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.2 NppStatus nppiCopy_16s_AC4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 16-bit image copy, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.3 NppStatus nppiCopy_16s_C1C3R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single-channel to three-channel 16-bit signed image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.4 NppStatus nppiCopy_16s_C1C4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single-channel to four-channel 16-bit signed image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.5 NppStatus nppiCopy_16s_C1MR (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

[Masked Operation](#) 16-bit signed image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.6 NppStatus nppiCopy_16s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

16-bit image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.7 **NppStatus nppiCopy_16s_C3C1R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel to single-channel 16-bit signed image copy.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.8 **NppStatus nppiCopy_16s_C3CR** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 16-bit signed image copy for three-channel images.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.9 **NppStatus nppiCopy_16s_C3MR** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

Masked Operation three channel 16-bit signed image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.10 NppStatus nppiCopy_16s_C3P3R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s *const *aDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 16-bit signed packed to planar image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
aDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.11 NppStatus nppiCopy_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.12 NppStatus nppiCopy_16s_C4C1R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel to single-channel 16-bit signed image copy.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.13 `NppStatus nppiCopy_16s_C4CR` (const `Npp16s * pSrc`, int `nSrcStep`, `Npp16s * pDst`, int `nDstStep`, `NppiSize oSizeROI`)

Select-channel 16-bit signed image copy for four-channel images.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.14 `NppStatus nppiCopy_16s_C4MR` (const `Npp16s * pSrc`, int `nSrcStep`, `Npp16s * pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u * pMask`, int `nMaskStep`)

Masked Operation four channel 16-bit signed image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.15 `NppStatus nppiCopy_16s_C4P4R` (const `Npp16s * pSrc`, int `nSrcStep`, `Npp16s * const aDst[4]`, int `nDstStep`, `NppiSize oSizeROI`)

Four-channel 16-bit signed packed to planar image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
aDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.16 NppStatus nppiCopy_16s_C4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 16-bit image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.17 NppStatus nppiCopy_16s_P3C3R (const Npp16s *const *aSrc*[3], int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 16-bit signed planar to packed image copy.

Parameters:

aSrc Planar Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.18 NppStatus nppiCopy_16s_P4C4R (const Npp16s *const *aSrc*[4], int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit signed planar to packed image copy.

Parameters:

aSrc Planar Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.19 NppStatus nppiCopy_16sc_AC4R (const Npp16sc * *pSrc*, int *nSrcStep*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit complex image copy, ignoring alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.20 NppStatus nppiCopy_16sc_C1R (const Npp16sc * *pSrc*, int *nSrcStep*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

16-bit complex image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.21 NppStatus nppiCopy_16sc_C2R (const Npp16sc * *pSrc*, int *nSrcStep*, Npp16sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Two-channel 16-bit complex image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.22 `NppStatus nppiCopy_16sc_C3R (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel 16-bit complex image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.23 `NppStatus nppiCopy_16sc_C4R (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 16-bit complex image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.24 `NppStatus nppiCopy_16u_AC4MR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 16-bit unsigned image copy, ignoring alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.25 NppStatus nppiCopy_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

4 channel 16-bit unsigned image copy, not affecting Alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.26 NppStatus nppiCopy_16u_C1C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

Single-channel to three-channel 16-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.27 NppStatus nppiCopy_16u_C1C4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)

Single-channel to four-channel 16-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.28 `NppStatus nppiCopy_16u_C1MR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) 16-bit unsigned image copy.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
pMask [Mask-Image Pointer](#).
nMaskStep [Mask-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.29 `NppStatus nppiCopy_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

16-bit unsigned image copy.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.30 `NppStatus nppiCopy_16u_C3C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel to single-channel 16-bit unsigned image copy.

Parameters:

pSrc [Select-Channel Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.31 `NppStatus nppiCopy_16u_C3CR` (const `Npp16u * pSrc`, int `nSrcStep`, `Npp16u * pDst`, int `nDstStep`, `NppiSize oSizeROI`)

Select-channel 16-bit unsigned image copy for three-channel images.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.32 `NppStatus nppiCopy_16u_C3MR` (const `Npp16u * pSrc`, int `nSrcStep`, `Npp16u * pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u * pMask`, int `nMaskStep`)

Masked Operation three channel 16-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.33 `NppStatus nppiCopy_16u_C3P3R` (const `Npp16u * pSrc`, int `nSrcStep`, `Npp16u * const aDst[3]`, int `nDstStep`, `NppiSize oSizeROI`)

Three-channel 16-bit unsigned packed to planar image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
aDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.34 NppStatus nppiCopy_16u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.35 NppStatus nppiCopy_16u_C4C1R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel to single-channel 16-bit unsigned image copy.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.36 NppStatus nppiCopy_16u_C4CR (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 16-bit unsigned image copy for four-channel images.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.37 `NppStatus nppiCopy_16u_C4MR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 16-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.38 `NppStatus nppiCopy_16u_C4P4R (const Npp16u * pSrc, int nSrcStep, Npp16u * const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 16-bit unsigned packed to planar image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
aDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.39 `NppStatus nppiCopy_16u_C4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.40 `NppStatus nppiCopy_16u_P3C3R (const Npp16u *const aSrc[3], int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel 16-bit unsigned planar to packed image copy.

Parameters:

aSrc Planar [Source-Planar-Image Pointer Array](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.41 `NppStatus nppiCopy_16u_P4C4R (const Npp16u *const aSrc[4], int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 16-bit unsigned planar to packed image copy.

Parameters:

aSrc Planar [Source-Planar-Image Pointer Array](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.42 `NppStatus nppiCopy_32f_AC4MR (const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp8u *pMask, int nMaskStep)`

[Masked Operation](#) four channel 32-bit float image copy, ignoring alpha.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
pMask [Mask-Image Pointer](#).
nMaskStep [Mask-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.43 NppStatus nppiCopy_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit floating point image copy, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.44 NppStatus nppiCopy_32f_C1C3R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single-channel to three-channel 32-bit float image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.45 NppStatus nppiCopy_32f_C1C4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single-channel to four-channel 32-bit float image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.46 `NppStatus nppiCopy_32f_C1MR (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) 32-bit float image copy.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
pMask [Mask-Image Pointer](#).
nMaskStep [Mask-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.47 `NppStatus nppiCopy_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit floating point image copy.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.48 `NppStatus nppiCopy_32f_C3C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel to single-channel 32-bit float image copy.

Parameters:

pSrc [Select-Channel Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.49 **NppStatus nppiCopy_32f_C3CR** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 32-bit float image copy for three-channel images.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.50 **NppStatus nppiCopy_32f_C3MR** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

Masked Operation three channel 32-bit float image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.51 **NppStatus nppiCopy_32f_C3P3R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *aDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit float packed to planar image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
aDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.52 NppStatus nppiCopy_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 32-bit floating point image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.53 NppStatus nppiCopy_32f_C4C1R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel to single-channel 32-bit float image copy.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.54 NppStatus nppiCopy_32f_C4CR (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 32-bit float image copy for four-channel images.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.55 `NppStatus nppiCopy_32f_C4MR (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 32-bit float image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.56 `NppStatus nppiCopy_32f_C4P4R (const Npp32f * pSrc, int nSrcStep, Npp32f * const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 32-bit float packed to planar image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
aDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.57 `NppStatus nppiCopy_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit floating point image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.58 NppStatus nppiCopy_32f_P3C3R (const Npp32f *const *aSrc*[3], int *nSrcStep*, Npp32f **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit float planar to packed image copy.

Parameters:

aSrc Planar [Source-Planar-Image Pointer Array](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.59 NppStatus nppiCopy_32f_P4C4R (const Npp32f *const *aSrc*[4], int *nSrcStep*, Npp32f **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit float planar to packed image copy.

Parameters:

aSrc Planar [Source-Planar-Image Pointer Array](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.60 NppStatus nppiCopy_32fc_AC4R (const Npp32fc **pSrc*, int *nSrcStep*, Npp32fc **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating-point complex image copy, ignoring alpha.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.61 NppStatus nppiCopy_32fc_C1R (const Npp32fc * *pSrc*, int *nSrcStep*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

32-bit floating-point complex image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.62 NppStatus nppiCopy_32fc_C2R (const Npp32fc * *pSrc*, int *nSrcStep*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Two-channel 32-bit floating-point complex image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.63 NppStatus nppiCopy_32fc_C3R (const Npp32fc * *pSrc*, int *nSrcStep*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit floating-point complex image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.64 **NppStatus nppiCopy_32fc_C4R** (const Npp32fc * *pSrc*, int *nSrcStep*, Npp32fc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating-point complex image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.65 **NppStatus nppiCopy_32s_AC4MR** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

[Masked Operation](#) four channel 32-bit signed image copy, ignoring alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.66 **NppStatus nppiCopy_32s_AC4R** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit image copy, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.67 **NppStatus nppiCopy_32s_C1C3R** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single-channel to three-channel 32-bit signed image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.68 **NppStatus nppiCopy_32s_C1C4R** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single-channel to four-channel 32-bit signed image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.69 **NppStatus nppiCopy_32s_C1MR** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

Masked Operation 32-bit signed image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.70 NppStatus nppiCopy_32s_C1R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

32-bit image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.71 NppStatus nppiCopy_32s_C3C1R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel to single-channel 32-bit signed image copy.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.72 NppStatus nppiCopy_32s_C3CR (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 32-bit signed image copy for three-channel images.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.73 `NppStatus nppiCopy_32s_C3MR (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) three channel 32-bit signed image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.74 `NppStatus nppiCopy_32s_C3P3R (const Npp32s * pSrc, int nSrcStep, Npp32s * const aDst[3], int nDstStep, NppiSize oSizeROI)`

Three-channel 32-bit signed packed to planar image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
aDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.75 `NppStatus nppiCopy_32s_C3R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 32-bit image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.76 `NppStatus nppiCopy_32s_C4C1R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel to single-channel 32-bit signed image copy.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.77 `NppStatus nppiCopy_32s_C4CR (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Select-channel 32-bit signed image copy for four-channel images.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.78 `NppStatus nppiCopy_32s_C4MR (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation four channel 32-bit signed image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.79 NppStatus nppiCopy_32s_C4P4R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s *const *aDst*[4], int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit signed packed to planar image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
aDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.80 NppStatus nppiCopy_32s_C4R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.81 NppStatus nppiCopy_32s_P3C3R (const Npp32s *const *aSrc*[3], int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit signed planar to packed image copy.

Parameters:

aSrc Planar Source-Planar-Image Pointer Array.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.82 **NppStatus nppiCopy_32s_P4C4R** (const Npp32s *const *aSrc*[4], int *nSrcStep*, Npp32s **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit signed planar to packed image copy.

Parameters:

aSrc Planar [Source-Planar-Image Pointer Array](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.83 **NppStatus nppiCopy_32sc_AC4R** (const Npp32sc **pSrc*, int *nSrcStep*, Npp32sc **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit complex image copy, ignoring alpha.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.84 **NppStatus nppiCopy_32sc_C1R** (const Npp32sc **pSrc*, int *nSrcStep*, Npp32sc **pDst*, int *nDstStep*, NppiSize *oSizeROI*)

32-bit complex image copy.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.85 NppStatus nppiCopy_32sc_C2R (const Npp32sc * *pSrc*, int *nSrcStep*, Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Two-channel 32-bit complex image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.86 NppStatus nppiCopy_32sc_C3R (const Npp32sc * *pSrc*, int *nSrcStep*, Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit complex image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.87 NppStatus nppiCopy_32sc_C4R (const Npp32sc * *pSrc*, int *nSrcStep*, Npp32sc * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit complex image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.88 NppStatus nppiCopy_8s_AC4R (const Npp8s * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 8-bit image copy, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.89 NppStatus nppiCopy_8s_C1R (const Npp8s * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

8-bit image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.90 NppStatus nppiCopy_8s_C2R (const Npp8s * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Two-channel 8-bit image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.91 NppStatus nppiCopy_8s_C3R (const Npp8s * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 8-bit image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.92 NppStatus nppiCopy_8s_C4R (const Npp8s * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 8-bit image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.93 NppStatus nppiCopy_8u_AC4MR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

Masked Operation four channel 8-bit unsigned image copy, ignoring alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.94 NppStatus nppiCopy_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned image copy, not affecting Alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.95 NppStatus nppiCopy_8u_C1C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single-channel to three-channel 8-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.96 NppStatus nppiCopy_8u_C1C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single-channel to four-channel 8-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.97 `NppStatus nppiCopy_8u_C1MR (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) 8-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.98 `NppStatus nppiCopy_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

8-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.99 `NppStatus nppiCopy_8u_C3C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel to single-channel 8-bit unsigned image copy.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.100 **NppStatus nppiCopy_8u_C3CR** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 8-bit unsigned image copy for three-channel images.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.101 **NppStatus nppiCopy_8u_C3MR** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u * *pMask*, int *nMaskStep*)

[Masked Operation](#) three channel 8-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.102 **NppStatus nppiCopy_8u_C3P3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * const *aDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 8-bit unsigned packed to planar image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
aDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.103 NppStatus nppiCopy_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.104 NppStatus nppiCopy_8u_C4C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel to single-channel 8-bit unsigned image copy.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.105 NppStatus nppiCopy_8u_C4CR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 8-bit unsigned image copy for four-channel images.

Parameters:

pSrc Select-Channel Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Select-Channel Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.55.1.106 `NppStatus nppiCopy_8u_C4MR (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 8-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.107 `NppStatus nppiCopy_8u_C4P4R (const Npp8u * pSrc, int nSrcStep, Npp8u *const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 8-bit unsigned packed to planar image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
aDst Destination-Planar-Image Pointer Array.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.108 `NppStatus nppiCopy_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.109 NppStatus nppiCopy_8u_P3C3R (const Npp8u *const aSrc[3], int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

Three-channel 8-bit unsigned planar to packed image copy.

Parameters:

aSrc Planar [Source-Image Pointer](#).
nSrcStep [Source-Planar-Image Pointer Array](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.55.1.110 NppStatus nppiCopy_8u_P4C4R (const Npp8u *const aSrc[4], int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)

Four-channel 8-bit unsigned planar to packed image copy.

Parameters:

aSrc Planar [Source-Planar-Image Pointer Array](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56 Convert

Convert to Increase Bit-Depth

The integer conversion methods do not involve any scaling.

Also, even when increasing the bit-depth loss of information may occur:

- When converting integers (e.g. `Npp32u`) to float (e.g. `Npp32f`) integervalue not accurately representable by the float are rounded to the closest floating-point value.
- When converting signed integers to unsigned integers all negative values are lost (saturated to 0).
- `NppStatus nppiConvert_8u16u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Single channel 8-bit unsigned to 16-bit unsigned conversion.
- `NppStatus nppiConvert_8u16u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three channel 8-bit unsigned to 16-bit unsigned conversion.
- `NppStatus nppiConvert_8u16u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four channel 8-bit unsigned to 16-bit unsigned conversion.
- `NppStatus nppiConvert_8u16u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.
- `NppStatus nppiConvert_8u16s_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Single channel 8-bit unsigned to 16-bit signed conversion.
- `NppStatus nppiConvert_8u16s_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three channel 8-bit unsigned to 16-bit signed conversion.
- `NppStatus nppiConvert_8u16s_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four channel 8-bit unsigned to 16-bit signed conversion.
- `NppStatus nppiConvert_8u16s_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.
- `NppStatus nppiConvert_8u32s_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Single channel 8-bit unsigned to 32-bit signed conversion.
- `NppStatus nppiConvert_8u32s_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three channel 8-bit unsigned to 32-bit signed conversion.

- `NppStatus nppiConvert_8u32s_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned to 32-bit signed conversion.

- `NppStatus nppiConvert_8u32s_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.

- `NppStatus nppiConvert_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit unsigned to 32-bit floating-point conversion.

- `NppStatus nppiConvert_8u32f_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three channel 8-bit unsigned to 32-bit floating-point conversion.

- `NppStatus nppiConvert_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned to 32-bit floating-point conversion.

- `NppStatus nppiConvert_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

- `NppStatus nppiConvert_8s32s_C1R` (const `Npp8s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit signed to 32-bit signed conversion.

- `NppStatus nppiConvert_8s32s_C3R` (const `Npp8s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three channel 8-bit signed to 32-bit signed conversion.

- `NppStatus nppiConvert_8s32s_C4R` (const `Npp8s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit signed to 32-bit signed conversion.

- `NppStatus nppiConvert_8s32s_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit signed to 32-bit signed conversion, not affecting Alpha.

- `NppStatus nppiConvert_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit signed to 32-bit floating-point conversion.

- `NppStatus nppiConvert_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three channel 8-bit signed to 32-bit floating-point conversion.

- [NppStatus nppiConvert_8s32f_C4R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Four channel 8-bit signed to 32-bit floating-point conversion.
- [NppStatus nppiConvert_8s32f_AC4R](#) (const [Npp8s](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Four channel 8-bit signed to 32-bit floating-point conversion, not affecting Alpha.
- [NppStatus nppiConvert_16u32s_C1R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Single channel 16-bit unsigned to 32-bit signed conversion.
- [NppStatus nppiConvert_16u32s_C3R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Three channel 16-bit unsigned to 32-bit signed conversion.
- [NppStatus nppiConvert_16u32s_C4R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Four channel 16-bit unsigned to 32-bit signed conversion.
- [NppStatus nppiConvert_16u32s_AC4R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Four channel 16-bit unsigned to 32-bit signed conversion, not affecting Alpha.
- [NppStatus nppiConvert_16u32f_C1R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Single channel 16-bit unsigned to 32-bit floating-point conversion.
- [NppStatus nppiConvert_16u32f_C3R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Three channel 16-bit unsigned to 32-bit floating-point conversion.
- [NppStatus nppiConvert_16u32f_C4R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Four channel 16-bit unsigned to 32-bit floating-point conversion.
- [NppStatus nppiConvert_16u32f_AC4R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Four channel 16-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.
- [NppStatus nppiConvert_16s32s_C1R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Single channel 16-bit signed to 32-bit signed conversion.
- [NppStatus nppiConvert_16s32s_C3R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Three channel 16-bit signed to 32-bit signed conversion.
- [NppStatus nppiConvert_16s32s_C4R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI)
Four channel 16-bit signed to 32-bit signed conversion.

- **NppStatus nppiConvert_16s32s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four channel 16-bit signed to 32-bit signed conversion, not affecting Alpha.
- **NppStatus nppiConvert_16s32f_C1R** (const **Npp16s** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Single channel 16-bit signed to 32-bit floating-point conversion.
- **NppStatus nppiConvert_16s32f_C3R** (const **Npp16s** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three channel 16-bit signed to 32-bit floating-point conversion.
- **NppStatus nppiConvert_16s32f_C4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four channel 16-bit signed to 32-bit floating-point conversion.
- **NppStatus nppiConvert_16s32f_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four channel 16-bit signed to 32-bit floating-point conversion, not affecting Alpha.
- **NppStatus nppiConvert_8s8u_C1Rs** (const **Npp8s** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Single channel 8-bit signed to 8-bit unsigned conversion with saturation.
- **NppStatus nppiConvert_8s16u_C1Rs** (const **Npp8s** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Single channel 8-bit signed to 16-bit unsigned conversion with saturation.
- **NppStatus nppiConvert_8s16s_C1R** (const **Npp8s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Single channel 8-bit signed to 16-bit signed conversion.
- **NppStatus nppiConvert_8s32u_C1Rs** (const **Npp8s** *pSrc, int nSrcStep, **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Single channel 8-bit signed to 32-bit unsigned conversion with saturation.
- **NppStatus nppiConvert_16s16u_C1Rs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Single channel 16-bit signed to 16-bit unsigned conversion with saturation.
- **NppStatus nppiConvert_16s32u_C1Rs** (const **Npp16s** *pSrc, int nSrcStep, **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Single channel 16-bit signed to 32-bit unsigned conversion with saturation.
- **NppStatus nppiConvert_16u32u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Single channel 16-bit unsigned to 32-bit unsigned conversion.
- **NppStatus nppiConvert_32s32u_C1Rs** (const **Npp32s** *pSrc, int nSrcStep, **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Single channel 32-bit signed to 32-bit unsigned conversion with saturation.

- `NppStatus nppiConvert_32s32f_C1R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single channel 32-bit signed to 32-bit floating-point conversion.

- `NppStatus nppiConvert_32u32f_C1R` (const `Npp32u` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single channel 32-bit unsigned to 32-bit floating-point conversion.

Convert to Decrease Bit-Depth

The integer conversion methods do not involve any scaling.

When converting floating-point values to integers the user may choose the most appropriate rounding-mode. Typically information is lost when converting to lower bit depth:

- All converted values are saturated to the destination type's range. E.g. any values larger than the largest value of the destination type are clamped to the destination's maximum.
- Converting floating-point values to integer also involves rounding, effectively losing all fractional value information in the process.

- `NppStatus nppiConvert_16u8u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single channel 16-bit unsigned to 8-bit unsigned conversion.

- `NppStatus nppiConvert_16u8u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three channel 16-bit unsigned to 8-bit unsigned conversion.

- `NppStatus nppiConvert_16u8u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit unsigned to 8-bit unsigned conversion.

- `NppStatus nppiConvert_16u8u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.

- `NppStatus nppiConvert_16s8u_C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single channel 16-bit signed to 8-bit unsigned conversion.

- `NppStatus nppiConvert_16s8u_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three channel 16-bit signed to 8-bit unsigned conversion.

- `NppStatus nppiConvert_16s8u_C4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed to 8-bit unsigned conversion.

- **NppStatus nppiConvert_16s8u_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.
- **NppStatus nppiConvert_32s8u_C1R** (const **Npp32s** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Single channel 32-bit signed to 8-bit unsigned conversion.
- **NppStatus nppiConvert_32s8u_C3R** (const **Npp32s** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three channel 32-bit signed to 8-bit unsigned conversion.
- **NppStatus nppiConvert_32s8u_C4R** (const **Npp32s** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four channel 32-bit signed to 8-bit unsigned conversion.
- **NppStatus nppiConvert_32s8u_AC4R** (const **Npp32s** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.
- **NppStatus nppiConvert_32s8s_C1R** (const **Npp32s** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Single channel 32-bit signed to 8-bit signed conversion.
- **NppStatus nppiConvert_32s8s_C3R** (const **Npp32s** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Three channel 32-bit signed to 8-bit signed conversion.
- **NppStatus nppiConvert_32s8s_C4R** (const **Npp32s** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four channel 32-bit signed to 8-bit signed conversion.
- **NppStatus nppiConvert_32s8s_AC4R** (const **Npp32s** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four channel 32-bit signed to 8-bit signed conversion, not affecting Alpha.
- **NppStatus nppiConvert_8u8s_C1RSfs** (const **Npp8u** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)
Single channel 8-bit unsigned to 8-bit signed conversion.
- **NppStatus nppiConvert_16u8s_C1RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)
Single channel 16-bit unsigned to 8-bit signed conversion.
- **NppStatus nppiConvert_16s8s_C1RSfs** (const **Npp16s** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)
Single channel 16-bit signed to 8-bit signed conversion.
- **NppStatus nppiConvert_16u16s_C1RSfs** (const **Npp16u** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)

Single channel 16-bit unsigned to 16-bit signed conversion.

- `NppStatus nppiConvert_32u8u_C1RSfs` (const `Npp32u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

Single channel 32-bit unsigned to 8-bit unsigned conversion.

- `NppStatus nppiConvert_32u8s_C1RSfs` (const `Npp32u` *pSrc, int nSrcStep, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

Single channel 32-bit unsigned to 8-bit signed conversion.

- `NppStatus nppiConvert_32u16u_C1RSfs` (const `Npp32u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

Single channel 32-bit unsigned to 16-bit unsigned conversion.

- `NppStatus nppiConvert_32u16s_C1RSfs` (const `Npp32u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

Single channel 32-bit unsigned to 16-bit signed conversion.

- `NppStatus nppiConvert_32u32s_C1RSfs` (const `Npp32u` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

Single channel 32-bit unsigned to 32-bit signed conversion.

- `NppStatus nppiConvert_32s16u_C1RSfs` (const `Npp32s` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

Single channel 32-bit signed to 16-bit unsigned conversion.

- `NppStatus nppiConvert_32s16s_C1RSfs` (const `Npp32s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

Single channel 32-bit signed to 16-bit signed conversion.

- `NppStatus nppiConvert_32f8u_C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)

Single channel 32-bit floating point to 8-bit unsigned conversion.

- `NppStatus nppiConvert_32f8u_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)

Three channel 32-bit floating point to 8-bit unsigned conversion.

- `NppStatus nppiConvert_32f8u_C4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)

Four channel 32-bit floating point to 8-bit unsigned conversion.

- `NppStatus nppiConvert_32f8u_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)

Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.

- `NppStatus nppiConvert_32f8s_C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)

Single channel 32-bit floating point to 8-bit signed conversion.

- **NppStatus** **nppiConvert_32f8s_C3R** (const **Npp32f** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)
Three channel 32-bit floating point to 8-bit signed conversion.
- **NppStatus** **nppiConvert_32f8s_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)
Four channel 32-bit floating point to 8-bit signed conversion.
- **NppStatus** **nppiConvert_32f8s_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)
Four channel 32-bit floating point to 8-bit signed conversion, not affecting Alpha.
- **NppStatus** **nppiConvert_32f16u_C1R** (const **Npp32f** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)
Single channel 32-bit floating point to 16-bit unsigned conversion.
- **NppStatus** **nppiConvert_32f16u_C3R** (const **Npp32f** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)
Three channel 32-bit floating point to 16-bit unsigned conversion.
- **NppStatus** **nppiConvert_32f16u_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)
Four channel 32-bit floating point to 16-bit unsigned conversion.
- **NppStatus** **nppiConvert_32f16u_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)
Four channel 32-bit floating point to 16-bit unsigned conversion, not affecting Alpha.
- **NppStatus** **nppiConvert_32f16s_C1R** (const **Npp32f** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)
Single channel 32-bit floating point to 16-bit signed conversion.
- **NppStatus** **nppiConvert_32f16s_C3R** (const **Npp32f** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)
Three channel 32-bit floating point to 16-bit signed conversion.
- **NppStatus** **nppiConvert_32f16s_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)
Four channel 32-bit floating point to 16-bit signed conversion.
- **NppStatus** **nppiConvert_32f16s_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)
Four channel 32-bit floating point to 16-bit signed conversion.
- **NppStatus** **nppiConvert_32f8u_C1RSfs** (const **Npp32f** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)
Single channel 32-bit floating point to 8-bit unsigned conversion.
- **NppStatus** **nppiConvert_32f8s_C1RSfs** (const **Npp32f** *pSrc, int nSrcStep, **Npp8s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)
Single channel 32-bit floating point to 8-bit signed conversion.

- **NppStatus nppiConvert_32f16u_C1RSfs** (const **Npp32f** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)
Single channel 32-bit floating point to 16-bit unsigned conversion.
- **NppStatus nppiConvert_32f16s_C1RSfs** (const **Npp32f** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)
Single channel 32-bit floating point to 16-bit signed conversion.
- **NppStatus nppiConvert_32f32u_C1RSfs** (const **Npp32f** *pSrc, int nSrcStep, **Npp32u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)
Single channel 32-bit floating point to 32-bit unsigned conversion.
- **NppStatus nppiConvert_32f32s_C1RSfs** (const **Npp32f** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)
Single channel 32-bit floating point to 32-bit signed conversion.

7.56.1 Function Documentation

7.56.1.1 **NppStatus nppiConvert_16s16u_C1Rs** (const **Npp16s** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Single channel 16-bit signed to 16-bit unsigned conversion with saturation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.2 **NppStatus nppiConvert_16s32f_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four channel 16-bit signed to 32-bit floating-point conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.3 NppStatus nppiConvert_16s32f_C1R (const Npp16s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)

Single channel 16-bit signed to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.4 NppStatus nppiConvert_16s32f_C3R (const Npp16s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)

Three channel 16-bit signed to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.5 NppStatus nppiConvert_16s32f_C4R (const Npp16s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)

Four channel 16-bit signed to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.6 NppStatus nppiConvert_16s32s_AC4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed to 32-bit signed conversion, not affecting Alpha.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.7 NppStatus nppiConvert_16s32s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 16-bit signed to 32-bit signed conversion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.8 NppStatus nppiConvert_16s32s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit signed to 32-bit signed conversion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.9 NppStatus nppiConvert_16s32s_C4R (const Npp16s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)

Four channel 16-bit signed to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.10 NppStatus nppiConvert_16s32u_C1Rs (const Npp16s * pSrc, int nSrcStep, Npp32u * pDst, int nDstStep, NppiSize oSizeROI)

Single channel 16-bit signed to 32-bit unsigned conversion with saturation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.11 NppStatus nppiConvert_16s8s_C1RSfs (const Npp16s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)

Single channel 16-bit signed to 8-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eRoundMode Rounding Mode Parameter.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.12 NppStatus nppiConvert_16s8u_AC4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.13 NppStatus nppiConvert_16s8u_C1R (const Npp16s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 16-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.14 NppStatus nppiConvert_16s8u_C3R (const Npp16s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.15 NppStatus nppiConvert_16s8u_C4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.16 NppStatus nppiConvert_16u16s_C1RSfs (const Npp16u * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*, int *nScaleFactor*)

Single channel 16-bit unsigned to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eRoundMode Rounding Mode Parameter.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.17 NppStatus nppiConvert_16u32f_AC4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.18 NppStatus nppiConvert_16u32f_C1R (const Npp16u * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 16-bit unsigned to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.19 NppStatus nppiConvert_16u32f_C3R (const Npp16u * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit unsigned to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.20 `NppStatus nppiConvert_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 16-bit unsigned to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.21 `NppStatus nppiConvert_16u32s_AC4R (const Npp16u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 16-bit unsigned to 32-bit signed conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.22 `NppStatus nppiConvert_16u32s_C1R (const Npp16u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 16-bit unsigned to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.23 NppStatus nppiConvert_16u32s_C3R (const Npp16u * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit unsigned to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.24 NppStatus nppiConvert_16u32s_C4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit unsigned to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.25 NppStatus nppiConvert_16u32u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, Npp32u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 16-bit unsigned to 32-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.26 `NppStatus nppiConvert_16u8s_C1RSfs (const Npp16u * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 16-bit unsigned to 8-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eRoundMode Rounding Mode Parameter.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.27 `NppStatus nppiConvert_16u8u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.28 `NppStatus nppiConvert_16u8u_C1R (const Npp16u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 16-bit unsigned to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.29 NppStatus nppiConvert_16u8u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit unsigned to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.30 NppStatus nppiConvert_16u8u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit unsigned to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.31 NppStatus nppiConvert_32f16s_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*)

Four channel 32-bit floating point to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.32 `NppStatus nppiConvert_32f16s_C1R (const Npp32f * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Single channel 32-bit floating point to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.33 `NppStatus nppiConvert_32f16s_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.34 `NppStatus nppiConvert_32f16s_C3R (const Npp32f * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Three channel 32-bit floating point to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.35 NppStatus nppiConvert_32f16s_C4R (const Npp32f * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)

Four channel 32-bit floating point to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.36 NppStatus nppiConvert_32f16u_AC4R (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)

Four channel 32-bit floating point to 16-bit unsigned conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.37 `NppStatus nppiConvert_32f16u_C1R (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Single channel 32-bit floating point to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.38 `NppStatus nppiConvert_32f16u_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.39 `NppStatus nppiConvert_32f16u_C3R (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Three channel 32-bit floating point to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.40 NppStatus nppiConvert_32f16u_C4R (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)

Four channel 32-bit floating point to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.41 NppStatus nppiConvert_32f32s_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)

Single channel 32-bit floating point to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.42 `NppStatus nppiConvert_32f32u_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp32u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 32-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.43 `NppStatus nppiConvert_32f8s_AC4R (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Four channel 32-bit floating point to 8-bit signed conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.44 `NppStatus nppiConvert_32f8s_C1R (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Single channel 32-bit floating point to 8-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.45 `NppStatus nppiConvert_32f8s_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 8-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.46 `NppStatus nppiConvert_32f8s_C3R (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Three channel 32-bit floating point to 8-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.47 NppStatus nppiConvert_32f8s_C4R (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)

Four channel 32-bit floating point to 8-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.48 NppStatus nppiConvert_32f8u_AC4R (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)

Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.49 NppStatus nppiConvert_32f8u_C1R (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)

Single channel 32-bit floating point to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.50 `NppStatus nppiConvert_32f8u_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.51 `NppStatus nppiConvert_32f8u_C3R (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Three channel 32-bit floating point to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.52 `NppStatus nppiConvert_32f8u_C4R (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Four channel 32-bit floating point to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Flag specifying how fractional float values are rounded to integer values.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.53 `NppStatus nppiConvert_32s16s_C1RSfs (const Npp32s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit unsigned to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Rounding Mode Parameter.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.54 `NppStatus nppiConvert_32s16u_C1RSfs (const Npp32s * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit unsigned to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Rounding Mode Parameter.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.55 NppStatus nppiConvert_32s32f_C1R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit signed to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.56 NppStatus nppiConvert_32s32u_C1Rs (const Npp32s * *pSrc*, int *nSrcStep*, Npp32u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit signed to 32-bit unsigned conversion with saturation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.57 NppStatus nppiConvert_32s8s_AC4R (const Npp32s * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit signed to 8-bit signed conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.58 NppStatus nppiConvert_32s8s_C1R (const Npp32s * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit signed to 8-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.59 NppStatus nppiConvert_32s8s_C3R (const Npp32s * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 32-bit signed to 8-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.60 NppStatus nppiConvert_32s8s_C4R (const Npp32s * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit signed to 8-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.61 NppStatus nppiConvert_32s8u_AC4R (const Npp32s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.62 NppStatus nppiConvert_32s8u_C1R (const Npp32s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.63 NppStatus nppiConvert_32s8u_C3R (const Npp32s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 32-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.64 `NppStatus nppiConvert_32s8u_C4R (const Npp32s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 32-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.65 `NppStatus nppiConvert_32u16s_C1RSfs (const Npp32u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit unsigned to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eRoundMode Rounding Mode Parameter.
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.66 `NppStatus nppiConvert_32u16u_C1RSfs (const Npp32u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit unsigned to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Rounding Mode Parameter.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.67 NppStatus nppiConvert_32u32f_C1R (const Npp32u * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit unsigned to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.68 NppStatus nppiConvert_32u32s_C1RSfs (const Npp32u * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*, int *nScaleFactor*)

Single channel 32-bit unsigned to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eRoundMode Rounding Mode Parameter.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.69 `NppStatus nppiConvert_32u8s_C1RSfs (const Npp32u * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit unsigned to 8-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eRoundMode Rounding Mode Parameter.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.70 `NppStatus nppiConvert_32u8u_C1RSfs (const Npp32u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit unsigned to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eRoundMode Rounding Mode Parameter.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.71 `NppStatus nppiConvert_8s16s_C1R (const Npp8s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit signed to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.72 **NppStatus nppiConvert_8s16u_C1Rs** (const Npp8s * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit signed to 16-bit unsigned conversion with saturation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.73 **NppStatus nppiConvert_8s32f_AC4R** (const Npp8s * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit signed to 32-bit floating-point conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.74 **NppStatus nppiConvert_8s32f_C1R** (const Npp8s * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit signed to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.75 `NppStatus nppiConvert_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit signed to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.76 `NppStatus nppiConvert_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit signed to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.77 NppStatus nppiConvert_8s32s_AC4R (const Npp8s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit signed to 32-bit signed conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.78 NppStatus nppiConvert_8s32s_C1R (const Npp8s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit signed to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.79 NppStatus nppiConvert_8s32s_C3R (const Npp8s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit signed to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.80 NppStatus nppiConvert_8s32s_C4R (const Npp8s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit signed to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.81 NppStatus nppiConvert_8s32u_C1Rs (const Npp8s * *pSrc*, int *nSrcStep*, Npp32u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit signed to 32-bit unsigned conversion with saturation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.82 NppStatus nppiConvert_8s8u_C1Rs (const Npp8s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit signed to 8-bit unsigned conversion with saturation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.83 `NppStatus nppiConvert_8u16s_AC4R (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.84 `NppStatus nppiConvert_8u16s_C1R (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit unsigned to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.85 `NppStatus nppiConvert_8u16s_C3R (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit unsigned to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.86 NppStatus nppiConvert_8u16s_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.87 NppStatus nppiConvert_8u16u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.88 NppStatus nppiConvert_8u16u_C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit unsigned to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.89 NppStatus nppiConvert_8u16u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit unsigned to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.90 NppStatus nppiConvert_8u16u_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.91 NppStatus nppiConvert_8u32f_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.56.1.92 NppStatus nppiConvert_8u32f_C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit unsigned to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.93 NppStatus nppiConvert_8u32f_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit unsigned to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.94 NppStatus nppiConvert_8u32f_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.95 `NppStatus nppiConvert_8u32s_AC4R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.96 `NppStatus nppiConvert_8u32s_C1R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit unsigned to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.97 `NppStatus nppiConvert_8u32s_C3R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit unsigned to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.98 NppStatus nppiConvert_8u32s_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.56.1.99 NppStatus nppiConvert_8u8s_C1RSfs (const Npp8u * *pSrc*, int *nSrcStep*, Npp8s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*, int *nScaleFactor*)

Single channel 8-bit unsigned to 8-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eRoundMode Rounding Mode Parameter.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57 Scale

Scaled Bit-Depth Conversion

Scale bit-depth up and down.

To map source pixel `srcPixelValue` to destination pixel `dstPixelValue` the following equation is used:

$$\text{dstPixelValue} = \text{dstMinRangeValue} + \text{scaleFactor} * (\text{srcPixelValue} - \text{srcMinRangeValue})$$

where $\text{scaleFactor} = (\text{dstMaxRangeValue} - \text{dstMinRangeValue}) / (\text{srcMaxRangeValue} - \text{srcMinRangeValue})$.

For conversions between integer data types, the entire integer numeric range of the input data type is mapped onto the entire integer numeric range of the output data type.

For conversions to floating point data types the floating point data range is defined by the user supplied floating point values of `nMax` and `nMin` which are used as the `dstMaxRangeValue` and `dstMinRangeValue` respectively in the `scaleFactor` and `dstPixelValue` calculations and also as the saturation values to which output data is clamped.

When converting from floating-point values to integer values, `nMax` and `nMin` are used as the `srcMaxRangeValue` and `srcMinRangeValue` respectively in the `scaleFactor` and `dstPixelValue` calculations. Output values are saturated and clamped to the full output integer pixel value range.

- `NppStatus nppiScale_8u16u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Single channel 8-bit unsigned to 16-bit unsigned conversion.
- `NppStatus nppiScale_8u16u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three channel 8-bit unsigned to 16-bit unsigned conversion.
- `NppStatus nppiScale_8u16u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four channel 8-bit unsigned to 16-bit unsigned conversion.
- `NppStatus nppiScale_8u16u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.
- `NppStatus nppiScale_8u16s_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Single channel 8-bit unsigned to 16-bit signed conversion.
- `NppStatus nppiScale_8u16s_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Three channel 8-bit unsigned to 16-bit signed conversion.
- `NppStatus nppiScale_8u16s_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)
Four channel 8-bit unsigned to 16-bit signed conversion.
- `NppStatus nppiScale_8u16s_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.

- `NppStatus nppiScale_8u32s_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit unsigned to 32-bit signed conversion.

- `NppStatus nppiScale_8u32s_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Three channel 8-bit unsigned to 32-bit signed conversion.

- `NppStatus nppiScale_8u32s_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned to 32-bit signed conversion.

- `NppStatus nppiScale_8u32s_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.

- `NppStatus nppiScale_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

Single channel 8-bit unsigned to 32-bit floating-point conversion.

- `NppStatus nppiScale_8u32f_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

Three channel 8-bit unsigned to 32-bit floating-point conversion.

- `NppStatus nppiScale_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

Four channel 8-bit unsigned to 32-bit floating-point conversion.

- `NppStatus nppiScale_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

- `NppStatus nppiScale_16u8u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

Single channel 16-bit unsigned to 8-bit unsigned conversion.

- `NppStatus nppiScale_16u8u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

Three channel 16-bit unsigned to 8-bit unsigned conversion.

- `NppStatus nppiScale_16u8u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

Four channel 16-bit unsigned to 8-bit unsigned conversion.

- `NppStatus nppiScale_16u8u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.

- `NppStatus nppiScale_16s8u_C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)
Single channel 16-bit signed to 8-bit unsigned conversion.
- `NppStatus nppiScale_16s8u_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)
Three channel 16-bit signed to 8-bit unsigned conversion.
- `NppStatus nppiScale_16s8u_C4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)
Four channel 16-bit signed to 8-bit unsigned conversion.
- `NppStatus nppiScale_16s8u_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)
Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.
- `NppStatus nppiScale_32s8u_C1R` (const `Npp32s` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)
Single channel 32-bit signed to 8-bit unsigned conversion.
- `NppStatus nppiScale_32s8u_C3R` (const `Npp32s` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)
Three channel 32-bit signed to 8-bit unsigned conversion.
- `NppStatus nppiScale_32s8u_C4R` (const `Npp32s` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)
Four channel 32-bit signed to 8-bit unsigned conversion.
- `NppStatus nppiScale_32s8u_AC4R` (const `Npp32s` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)
Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.
- `NppStatus nppiScale_32f8u_C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)
Single channel 32-bit floating point to 8-bit unsigned conversion.
- `NppStatus nppiScale_32f8u_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)
Three channel 32-bit floating point to 8-bit unsigned conversion.
- `NppStatus nppiScale_32f8u_C4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)
Four channel 32-bit floating point to 8-bit unsigned conversion.
- `NppStatus nppiScale_32f8u_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)
Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.

7.57.1 Function Documentation

7.57.1.1 **NppStatus nppiScale_16s8u_AC4R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)

Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

hint algorithm performance or accuracy selector, currently ignored

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.57.1.2 **NppStatus nppiScale_16s8u_C1R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)

Single channel 16-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

hint algorithm performance or accuracy selector, currently ignored

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.57.1.3 **NppStatus nppiScale_16s8u_C3R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)

Three channel 16-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

hint algorithm performance or accuracy selector, currently ignored

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.57.1.4 NppStatus nppiScale_16s8u_C4R (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppHintAlgorithm hint)

Four channel 16-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

hint algorithm performance or accuracy selector, currently ignored

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.57.1.5 NppStatus nppiScale_16u8u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppHintAlgorithm hint)

Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

hint algorithm performance or accuracy selector, currently ignored

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.57.1.6 NppStatus nppiScale_16u8u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)

Single channel 16-bit unsigned to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

hint algorithm performance or accuracy selector, currently ignored

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.57.1.7 NppStatus nppiScale_16u8u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)

Three channel 16-bit unsigned to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

hint algorithm performance or accuracy selector, currently ignored

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.57.1.8 NppStatus nppiScale_16u8u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)

Four channel 16-bit unsigned to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

hint algorithm performance or accuracy selector, currently ignored

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.57.1.9 NppStatus nppiScale_32f8u_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32f *nMin*, Npp32f *nMax*)

Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nMin specifies the minimum saturation value to which every output value will be clamped.

nMax specifies the maximum saturation value to which every output value will be clamped.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, NPP_SCALE_RANGE_ERROR indicates an error condition if $nMax \leq nMin$.

7.57.1.10 NppStatus nppiScale_32f8u_C1R (const Npp32f * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32f *nMin*, Npp32f *nMax*)

Single channel 32-bit floating point to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nMin specifies the minimum saturation value to which every output value will be clamped.

nMax specifies the maximum saturation value to which every output value will be clamped.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, NPP_SCALE_RANGE_ERROR indicates an error condition if $nMax \leq nMin$.

7.57.1.11 NppStatus nppiScale_32f8u_C3R (const Npp32f * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32f *nMin*, Npp32f *nMax*)

Three channel 32-bit floating point to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nMin specifies the minimum saturation value to which every output value will be clamped.

nMax specifies the maximum saturation value to which every output value will be clamped.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if $nMax \leq nMin$.

7.57.1.12 `NppStatus nppiScale_32f8u_C4R (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nMin, Npp32f nMax)`

Four channel 32-bit floating point to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nMin specifies the minimum saturation value to which every output value will be clamped.

nMax specifies the maximum saturation value to which every output value will be clamped.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if $nMax \leq nMin$.

7.57.1.13 `NppStatus nppiScale_32s8u_AC4R (const Npp32s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppHintAlgorithm hint)`

Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

hint algorithm performance or accuracy selector, currently ignored

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.14 NppStatus nppiScale_32s8u_C1R (const Npp32s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)

Single channel 32-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

hint algorithm performance or accuracy selector, currently ignored

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.57.1.15 NppStatus nppiScale_32s8u_C3R (const Npp32s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)

Three channel 32-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

hint algorithm performance or accuracy selector, currently ignored

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.57.1.16 NppStatus nppiScale_32s8u_C4R (const Npp32s * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)

Four channel 32-bit signed to 8-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

hint algorithm performance or accuracy selector, currently ignored

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.57.1.17 NppStatus nppiScale_8u16s_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.18 NppStatus nppiScale_8u16s_C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit unsigned to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.19 NppStatus nppiScale_8u16s_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit unsigned to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.20 NppStatus nppiScale_8u16s_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned to 16-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.21 NppStatus nppiScale_8u16u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.22 NppStatus nppiScale_8u16u_C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit unsigned to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.23 NppStatus nppiScale_8u16u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit unsigned to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.24 NppStatus nppiScale_8u16u_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned to 16-bit unsigned conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.25 NppStatus nppiScale_8u32f_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32f *nMin*, Npp32f *nMax*)

Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nMin specifies the minimum saturation value to which every output value will be clamped.
nMax specifies the maximum saturation value to which every output value will be clamped.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if $nMax \leq nMin$.

7.57.1.26 `NppStatus nppiScale_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nMin, Npp32f nMax)`

Single channel 8-bit unsigned to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nMin specifies the minimum saturation value to which every output value will be clamped.

nMax specifies the maximum saturation value to which every output value will be clamped.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if $nMax \leq nMin$.

7.57.1.27 `NppStatus nppiScale_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nMin, Npp32f nMax)`

Three channel 8-bit unsigned to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nMin specifies the minimum saturation value to which every output value will be clamped.

nMax specifies the maximum saturation value to which every output value will be clamped.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if $nMax \leq nMin$.

7.57.1.28 `NppStatus nppiScale_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nMin, Npp32f nMax)`

Four channel 8-bit unsigned to 32-bit floating-point conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nMin specifies the minimum saturation value to which every output value will be clamped.

nMax specifies the maximum saturation value to which every output value will be clamped.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if $nMax \leq nMin$.

7.57.1.29 NppStatus nppiScale_8u32s_AC4R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)

Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.30 NppStatus nppiScale_8u32s_C1R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)

Single channel 8-bit unsigned to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.31 `NppStatus nppiScale_8u32s_C3R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit unsigned to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.57.1.32 `NppStatus nppiScale_8u32s_C4R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned to 32-bit signed conversion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.58 Copy Constant Border

CopyConstBorder

Methods for copying images and padding borders with a constant, user-specifiable color.

- **NppStatus nppiCopyConstBorder_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, **Npp8u** nValue)
1 channel 8-bit unsigned integer image copy with constant border color.
- **NppStatus nppiCopyConstBorder_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp8u** aValue[3])
3 channel 8-bit unsigned integer image copy with constant border color.
- **NppStatus nppiCopyConstBorder_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp8u** aValue[4])
4 channel 8-bit unsigned integer image copy with constant border color.
- **NppStatus nppiCopyConstBorder_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp8u** aValue[3])
4 channel 8-bit unsigned integer image copy with constant border color with alpha channel unaffected.
- **NppStatus nppiCopyConstBorder_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, **Npp16u** nValue)
1 channel 16-bit unsigned integer image copy with constant border color.
- **NppStatus nppiCopyConstBorder_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp16u** aValue[3])
3 channel 16-bit unsigned integer image copy with constant border color.
- **NppStatus nppiCopyConstBorder_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp16u** aValue[4])
4 channel 16-bit unsigned integer image copy with constant border color.
- **NppStatus nppiCopyConstBorder_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp16u** aValue[3])
4 channel 16-bit unsigned integer image copy with constant border color with alpha channel unaffected.
- **NppStatus nppiCopyConstBorder_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, **Npp16s** nValue)
1 channel 16-bit signed integer image copy with constant border color.

- **NppStatus nppiCopyConstBorder_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp16s** aValue[3])
3 channel 16-bit signed integer image copy with constant border color.
- **NppStatus nppiCopyConstBorder_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp16s** aValue[4])
4 channel 16-bit signed integer image copy with constant border color.
- **NppStatus nppiCopyConstBorder_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp16s** aValue[3])
4 channel 16-bit signed integer image copy with constant border color with alpha channel unaffected.
- **NppStatus nppiCopyConstBorder_32s_C1R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, **Npp32s** nValue)
1 channel 32-bit signed integer image copy with constant border color.
- **NppStatus nppiCopyConstBorder_32s_C3R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp32s** aValue[3])
3 channel 32-bit signed integer image copy with constant border color.
- **NppStatus nppiCopyConstBorder_32s_C4R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp32s** aValue[4])
4 channel 32-bit signed integer image copy with constant border color.
- **NppStatus nppiCopyConstBorder_32s_AC4R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp32s** aValue[3])
4 channel 32-bit signed integer image copy with constant border color with alpha channel unaffected.
- **NppStatus nppiCopyConstBorder_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, **Npp32f** nValue)
1 channel 32-bit floating point image copy with constant border color.
- **NppStatus nppiCopyConstBorder_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp32f** aValue[3])
3 channel 32-bit floating point image copy with constant border color.
- **NppStatus nppiCopyConstBorder_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp32f** aValue[4])
4 channel 32-bit floating point image copy with constant border color.

- **NppStatus nppiCopyConstBorder_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp32f** aValue[3])

4 channel 32-bit floating point image copy with constant border color with alpha channel unaffected.

7.58.1 Function Documentation

- 7.58.1.1 NppStatus nppiCopyConstBorder_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp16s** aValue[3])

4 channel 16-bit signed integer image copy with constant border color with alpha channel unaffected.

See **nppiCopyConstBorder_16s_C1R()** for detailed documentation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSizeROI Size of the source region-of-interest.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

aValue Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.58.1.2 NppStatus nppiCopyConstBorder_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, **Npp16s** nValue)

1 channel 16-bit signed integer image copy with constant border color.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSizeROI Size of the source region of pixels.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the constant border color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

nValue The pixel value to be set for border pixels.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.3 `NppStatus nppiCopyConstBorder_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp16s aValue[3])`

3 channel 16-bit signed integer image copy with constant border color.

See [nppiCopyConstBorder_16s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

aValue Vector of the RGBA values of the border pixels to be set.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.4 `NppStatus nppiCopyConstBorder_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp16s aValue[4])`

4 channel 16-bit signed integer image copy with constant border color.

See [nppiCopyConstBorder_16s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

aValue Vector of the RGBA values of the border pixels to be set.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.5 `NppStatus nppiCopyConstBorder_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp16u aValue[3])`

4 channel 16-bit unsigned integer image copy with constant border color with alpha channel unaffected.

See [nppiCopyConstBorder_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSizeROI Size of the source region-of-interest.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

aValue Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.6 `NppStatus nppiCopyConstBorder_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, Npp16u nValue)`

1 channel 16-bit unsigned integer image copy with constant border color.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSizeROI Size of the source region of pixels.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the constant border color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

nValue The pixel value to be set for border pixels.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.7 `NppStatus nppiCopyConstBorder_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp16u aValue[3])`

3 channel 16-bit unsigned integer image copy with constant border color.

See [nppiCopyConstBorder_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

aValue Vector of the RGBA values of the border pixels to be set.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.8 `NppStatus nppiCopyConstBorder_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp16u aValue[4])`

4 channel 16-bit unsigned integer image copy with constant border color.

See [nppiCopyConstBorder_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.
aValue Vector of the RGBA values of the border pixels to be set.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.9 NppStatus nppiCopyConstBorder_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp32f aValue[3])

4 channel 32-bit floating point image copy with constant border color with alpha channel unaffected.

See [nppiCopyConstBorder_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region-of-interest.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.
aValue Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.10 NppStatus nppiCopyConstBorder_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, Npp32f nValue)

1 channel 32-bit floating point image copy with constant border color.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region of pixels.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the constant border color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

nValue The pixel value to be set for border pixels.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.11 `NppStatus nppiCopyConstBorder_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp32f aValue[3])`

3 channel 32-bit floating point image copy with constant border color.

See [nppiCopyConstBorder_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

aValue Vector of the RGBA values of the border pixels to be set.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.12 `NppStatus nppiCopyConstBorder_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp32f aValue[4])`

4 channel 32-bit floating point image copy with constant border color.

See [nppiCopyConstBorder_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.
aValue Vector of the RGBA values of the border pixels to be set.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.13 `NppStatus nppiCopyConstBorder_32s_AC4R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp32s aValue[3])`

4 channel 32-bit signed integer image copy with constant border color with alpha channel unaffected.

See [nppiCopyConstBorder_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.
aValue Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.14 `NppStatus nppiCopyConstBorder_32s_C1R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, Npp32s nValue)`

1 channel 32-bit signed integer image copy with constant border color.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

oSrcSizeROI Size of the source region of pixels.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the constant border color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

nValue The pixel value to be set for border pixels.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.15 `NppStatus nppiCopyConstBorder_32s_C3R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp32s aValue[3])`

3 channel 32-bit signed integer image copy with constant border color.

See [nppiCopyConstBorder_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSizeROI Size of the source region-of-interest.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

aValue Vector of the RGBA values of the border pixels to be set.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.16 `NppStatus nppiCopyConstBorder_32s_C4R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp32s aValue[4])`

4 channel 32-bit signed integer image copy with constant border color.

See [nppiCopyConstBorder_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region-of-interest.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.
aValue Vector of the RGBA values of the border pixels to be set.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.17 `NppStatus nppiCopyConstBorder_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp8u aValue[3])`

4 channel 8-bit unsigned integer image copy with constant border color with alpha channel unaffected.

See [nppiCopyConstBorder_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region-of-interest.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.
aValue Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.18 `NppStatus nppiCopyConstBorder_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, Npp8u nValue)`

1 channel 8-bit unsigned integer image copy with constant border color.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region of pixels.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the constant border color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

nValue The pixel value to be set for border pixels.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.19 `NppStatus nppiCopyConstBorder_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp8u aValue[3])`

3 channel 8-bit unsigned integer image copy with constant border color.

See [nppiCopyConstBorder_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

aValue Vector of the RGBA values of the border pixels to be set.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.58.1.20 `NppStatus nppiCopyConstBorder_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp8u aValue[4])`

4 channel 8-bit unsigned integer image copy with constant border color.

See [nppiCopyConstBorder_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

aValue Vector of the RGBA values of the border pixels to be set.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59 Copy Replicate Border

CopyReplicateBorder

Methods for copying images and padding borders with a replicates of the nearest source image pixel color.

- `NppStatus nppiCopyReplicateBorder_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp8u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 8-bit unsigned integer image copy with nearest source image pixel color.

- `NppStatus nppiCopyReplicateBorder_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp8u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

3 channel 8-bit unsigned integer image copy with nearest source image pixel color.

- `NppStatus nppiCopyReplicateBorder_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp8u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 8-bit unsigned integer image copy with nearest source image pixel color.

- `NppStatus nppiCopyReplicateBorder_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp8u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 8-bit unsigned integer image copy with nearest source image pixel color with alpha channel unaffected.

- `NppStatus nppiCopyReplicateBorder_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp16u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 16-bit unsigned integer image copy with nearest source image pixel color.

- `NppStatus nppiCopyReplicateBorder_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp16u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

3 channel 16-bit unsigned integer image copy with nearest source image pixel color.

- `NppStatus nppiCopyReplicateBorder_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp16u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit unsigned integer image copy with nearest source image pixel color.

- `NppStatus nppiCopyReplicateBorder_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp16u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit unsigned image copy with nearest source image pixel color with alpha channel unaffected.

- `NppStatus nppiCopyReplicateBorder_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp16s` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 16-bit signed integer image copy with nearest source image pixel color.

- **NppStatus nppiCopyReplicateBorder_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

3 channel 16-bit signed integer image copy with nearest source image pixel color.
- **NppStatus nppiCopyReplicateBorder_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit signed integer image copy with nearest source image pixel color.
- **NppStatus nppiCopyReplicateBorder_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit signed integer image copy with nearest source image pixel color with alpha channel unaffected.
- **NppStatus nppiCopyReplicateBorder_32s_C1R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 32-bit signed integer image copy with nearest source image pixel color.
- **NppStatus nppiCopyReplicateBorder_32s_C3R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

3 channel 32-bit signed image copy with nearest source image pixel color.
- **NppStatus nppiCopyReplicateBorder_32s_C4R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 32-bit signed integer image copy with nearest source image pixel color.
- **NppStatus nppiCopyReplicateBorder_32s_AC4R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 32-bit signed integer image copy with nearest source image pixel color with alpha channel unaffected.
- **NppStatus nppiCopyReplicateBorder_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 32-bit floating point image copy with nearest source image pixel color.
- **NppStatus nppiCopyReplicateBorder_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

3 channel 32-bit floating point image copy with nearest source image pixel color.
- **NppStatus nppiCopyReplicateBorder_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 32-bit floating point image copy with nearest source image pixel color.

- [NppStatus nppiCopyReplicateBorder_32f_AC4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcSizeROI, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 32-bit floating point image copy with nearest source image pixel color with alpha channel unaffected.

7.59.1 Function Documentation

7.59.1.1 [NppStatus nppiCopyReplicateBorder_16s_AC4R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcSizeROI, [Npp16s](#) *pDst, int nDstStep, [NppiSize](#) oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit signed integer image copy with nearest source image pixel color with alpha channel unaffected.

See [nppiCopyReplicateBorder_16s_C1R\(\)](#) for detailed documentation.

Parameters:

[pSrc](#) [Source-Image Pointer](#).
[nSrcStep](#) [Source-Image Line Step](#).
[oSrcSizeROI](#) Size of the source region-of-interest.
[pDst](#) [Destination-Image Pointer](#).
[nDstStep](#) [Destination-Image Line Step](#).
[oDstSizeROI](#) Size of the destination region-of-interest.
[nTopBorderHeight](#) Height of top border.
[nLeftBorderWidth](#) Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.2 [NppStatus nppiCopyReplicateBorder_16s_C1R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcSizeROI, [Npp16s](#) *pDst, int nDstStep, [NppiSize](#) oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 16-bit signed integer image copy with nearest source image pixel color.

Parameters:

[pSrc](#) [Source-Image Pointer](#).
[nSrcStep](#) [Source-Image Line Step](#).
[oSrcSizeROI](#) Size of the source region of pixels.
[pDst](#) [Destination-Image Pointer](#).
[nDstStep](#) [Destination-Image Line Step](#).
[oDstSizeROI](#) Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and nearest source image pixel color (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the nearest source image pixel color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.3 NppStatus nppiCopyReplicateBorder_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

3 channel 16-bit signed integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder_16s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.4 NppStatus nppiCopyReplicateBorder_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit signed integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder_16s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.5 NppStatus nppiCopyReplicateBorder_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit unsigned image copy with nearest source image pixel color with alpha channel unaffected.

See [nppiCopyReplicateBorder_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.6 NppStatus nppiCopyReplicateBorder_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 16-bit unsigned integer image copy with nearest source image pixel color.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region of pixels.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and nearest source image pixel color (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the nearest source image pixel color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.7 `NppStatus nppiCopyReplicateBorder_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

3 channel 16-bit unsigned integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region-of-interest.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.8 `NppStatus nppiCopyReplicateBorder_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

4 channel 16-bit unsigned integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region-of-interest.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.9 NppStatus nppiCopyReplicateBorder_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 32-bit floating point image copy with nearest source image pixel color with alpha channel unaffected.

See [nppiCopyReplicateBorder_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.10 NppStatus nppiCopyReplicateBorder_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

1 channel 32-bit floating point image copy with nearest source image pixel color.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region of pixels.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and nearest source image pixel color (outer part).
nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the nearest source image pixel color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.
nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.11 NppStatus nppiCopyReplicateBorder_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

3 channel 32-bit floating point image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSizeROI Size of the source region-of-interest.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.12 NppStatus nppiCopyReplicateBorder_32f_C4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 32-bit floating point image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSizeROI Size of the source region-of-interest.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.13 NppStatus nppiCopyReplicateBorder_32s_AC4R (const Npp32s * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 32-bit signed integer image copy with nearest source image pixel color with alpha channel unaffected.

See [nppiCopyReplicateBorder_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region-of-interest.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.14 NppStatus nppiCopyReplicateBorder_32s_C1R (const Npp32s * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

1 channel 32-bit signed integer image copy with nearest source image pixel color.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region of pixels.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and nearest source image pixel color (outer part).
nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the nearest source image pixel color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.
nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.15 NppStatus nppiCopyReplicateBorder_32s_C3R (const Npp32s * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

3 channel 32-bit signed image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.16 NppStatus nppiCopyReplicateBorder_32s_C4R (const Npp32s * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 32-bit signed integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.17 NppStatus nppiCopyReplicateBorder_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 8-bit unsigned integer image copy with nearest source image pixel color with alpha channel unaffected.

See [nppiCopyReplicateBorder_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region-of-interest.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.18 NppStatus nppiCopyReplicateBorder_8u_C1R (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

1 channel 8-bit unsigned integer image copy with nearest source image pixel color.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region of pixels.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and nearest source image pixel color (outer part).
nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the nearest source image pixel color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.
nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.19 `NppStatus nppiCopyReplicateBorder_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

3 channel 8-bit unsigned integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.59.1.20 `NppStatus nppiCopyReplicateBorder_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

4 channel 8-bit unsigned integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60 Copy Wrap Border

CopyWrapBorder

Methods for copying images and padding borders with wrapped replications of the source image pixel colors.

- **NppStatus nppiCopyWrapBorder_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
1 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.
- **NppStatus nppiCopyWrapBorder_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
3 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.
- **NppStatus nppiCopyWrapBorder_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
4 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.
- **NppStatus nppiCopyWrapBorder_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
4 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.
- **NppStatus nppiCopyWrapBorder_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
1 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.
- **NppStatus nppiCopyWrapBorder_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
3 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.
- **NppStatus nppiCopyWrapBorder_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.
- **NppStatus nppiCopyWrapBorder_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

- **NppStatus nppiCopyWrapBorder_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

- **NppStatus nppiCopyWrapBorder_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

3 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

- **NppStatus nppiCopyWrapBorder_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

- **NppStatus nppiCopyWrapBorder_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

- **NppStatus nppiCopyWrapBorder_32s_C1R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

- **NppStatus nppiCopyWrapBorder_32s_C3R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

3 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

- **NppStatus nppiCopyWrapBorder_32s_C4R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

- **NppStatus nppiCopyWrapBorder_32s_AC4R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

- **NppStatus nppiCopyWrapBorder_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

- **NppStatus nppiCopyWrapBorder_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

3 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

- **NppStatus nppiCopyWrapBorder_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

- **NppStatus nppiCopyWrapBorder_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

7.60.1 Function Documentation

7.60.1.1 **NppStatus nppiCopyWrapBorder_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

See [nppiCopyWrapBorder_16s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.2 NppStatus nppiCopyWrapBorder_16s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

1 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region of pixels.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and a border consisting of wrapped replication of the source image pixel colors (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the wrapped replication of the corresponding column of source image pixels colors. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.3 NppStatus nppiCopyWrapBorder_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

3 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder_16s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.4 NppStatus nppiCopyWrapBorder_16s_C4R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder_16s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.5 NppStatus nppiCopyWrapBorder_16u_AC4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

See [nppiCopyWrapBorder_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.6 NppStatus nppiCopyWrapBorder_16u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

1 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region of pixels.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and a border consisting of wrapped replication of the source image pixel colors (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the wrapped replication of the corresponding column of source image pixels colors. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.7 NppStatus nppiCopyWrapBorder_16u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

3 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.8 NppStatus nppiCopyWrapBorder_16u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region-of-interest.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.9 NppStatus nppiCopyWrapBorder_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

1 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

See [nppiCopyWrapBorder_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region-of-interest.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.10 NppStatus nppiCopyWrapBorder_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

1 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region of pixels.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and a border consisting of wrapped replication of the source image pixel colors (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the wrapped replication of the corresponding column of source image pixels colors. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.11 NppStatus nppiCopyWrapBorder_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

3 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.12 NppStatus nppiCopyWrapBorder_32f_C4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.13 NppStatus nppiCopyWrapBorder_32s_AC4R (const Npp32s * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

See [nppiCopyWrapBorder_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.14 `NppStatus nppiCopyWrapBorder_32s_C1R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

1 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region of pixels.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and a border consisting of wrapped replication of the source image pixel colors (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the wrapped replication of the corresponding column of source image pixels colors. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.15 `NppStatus nppiCopyWrapBorder_32s_C3R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

3 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.16 NppStatus nppiCopyWrapBorder_32s_C4R (const Npp32s * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.17 NppStatus nppiCopyWrapBorder_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

See [nppiCopyWrapBorder_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSizeROI Size of the source region-of-interest.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.18 NppStatus nppiCopyWrapBorder_8u_C1R (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

1 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region of pixels.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and a border consisting of wrapped replication of the source image pixel colors (outer part).

nTopBorderHeight Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the wrapped replication of the corresponding column of source image pixels colors. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

nLeftBorderWidth Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.19 NppStatus nppiCopyWrapBorder_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

3 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSizeROI Size of the source region-of-interest.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nTopBorderHeight Height of top border.

nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.60.1.20 NppStatus nppiCopyWrapBorder_8u_C4R (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSizeROI Size of the source region-of-interest.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Size of the destination region-of-interest.
nTopBorderHeight Height of top border.
nLeftBorderWidth Width of left border.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61 Copy Sub-Pixel

CopySubpix

Functions for copying linearly interpolated images using source image subpixel coordinates

- **NppStatus nppiCopySubpix_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
1 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
3 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
4 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
4 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.
- **NppStatus nppiCopySubpix_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
1 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
3 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
4 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
4 channel 16-bit unsigned linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.
- **NppStatus nppiCopySubpix_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
1 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
3 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
4 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.

- **NppStatus nppiCopySubpix_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
4 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.
- **NppStatus nppiCopySubpix_32s_C1R** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
1 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_32s_C3R** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
3 channel 32-bit signed linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_32s_C4R** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
4 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_32s_AC4R** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
4 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.
- **NppStatus nppiCopySubpix_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
1 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
3 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
4 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.
- **NppStatus nppiCopySubpix_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)
4 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

7.61.1 Function Documentation

7.61.1.1 **NppStatus nppiCopySubpix_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)

4 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

See **nppiCopySubpix_16s_C1R()** for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nDx Fractional part of source image X coordinate.
nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.2 **NppStatus nppiCopySubpix_16s_C1R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

1 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.
nDx Fractional part of source image X coordinate.
nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.3 **NppStatus nppiCopySubpix_16s_C3R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

3 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix_16s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.

nDx Fractional part of source image X coordinate.

nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.4 NppStatus nppiCopySubpix_16s_C4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oDstSizeROI, Npp32f nDx, Npp32f nDy)

4 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix_16s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nDx Fractional part of source image X coordinate.

nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.5 NppStatus nppiCopySubpix_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, Npp32f nDx, Npp32f nDy)

4 channel 16-bit unsigned linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

See [nppiCopySubpix_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nDx Fractional part of source image X coordinate.

nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.6 **NppStatus nppiCopySubpix_16u_C1R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

1 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

nDx Fractional part of source image X coordinate.

nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.7 **NppStatus nppiCopySubpix_16u_C3R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

3 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

nDx Fractional part of source image X coordinate.

nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.8 **NppStatus nppiCopySubpix_16u_C4R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

4 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix_16u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nDx Fractional part of source image X coordinate.
nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.9 **NppStatus nppiCopySubpix_32f_AC4R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

4 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

See [nppiCopySubpix_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nDx Fractional part of source image X coordinate.
nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.10 **NppStatus nppiCopySubpix_32f_C1R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

1 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.
nDx Fractional part of source image X coordinate.
nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.11 NppStatus nppiCopySubpix_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

3 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nDx Fractional part of source image X coordinate.
nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.12 NppStatus nppiCopySubpix_32f_C4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

4 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix_32f_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nDx Fractional part of source image X coordinate.
nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.13 NppStatus nppiCopySubpix_32s_AC4R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

4 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

See [nppiCopySubpix_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.
nDx Fractional part of source image X coordinate.
nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.14 NppStatus nppiCopySubpix_32s_C1R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

1 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.
nDx Fractional part of source image X coordinate.
nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.15 NppStatus nppiCopySubpix_32s_C3R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

3 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.

nDx Fractional part of source image X coordinate.

nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.16 NppStatus nppiCopySubpix_32s_C4R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, Npp32f nDx, Npp32f nDy)

4 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix_32s_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nDx Fractional part of source image X coordinate.

nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.17 NppStatus nppiCopySubpix_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, Npp32f nDx, Npp32f nDy)

4 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

See [nppiCopySubpix_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nDx Fractional part of source image X coordinate.

nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.18 **NppStatus nppiCopySubpix_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

1 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

nDx Fractional part of source image X coordinate.

nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.19 **NppStatus nppiCopySubpix_8u_C3R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

3 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nDx Fractional part of source image X coordinate.

nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.61.1.20 **NppStatus nppiCopySubpix_8u_C4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

4 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix_8u_C1R\(\)](#) for detailed documentation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Size of the destination region-of-interest.

nDx Fractional part of source image X coordinate.

nDy Fractional part of source image Y coordinate.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.62 Duplicate Channel

Dup

Functions for duplicating a single channel image in a multiple channel image.

- **NppStatus nppiDup_8u_C1C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI)
1 channel 8-bit unsigned integer source image duplicated in all 3 channels of destination image.
- **NppStatus nppiDup_8u_C1C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI)
1 channel 8-bit unsigned integer source image duplicated in all 4 channels of destination image.
- **NppStatus nppiDup_8u_C1AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oDstSizeROI)
1 channel 8-bit unsigned integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.
- **NppStatus nppiDup_16u_C1C3R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI)
1 channel 16-bit unsigned integer source image duplicated in all 3 channels of destination image.
- **NppStatus nppiDup_16u_C1C4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI)
1 channel 16-bit unsigned integer source image duplicated in all 4 channels of destination image.
- **NppStatus nppiDup_16u_C1AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oDstSizeROI)
1 channel 16-bit unsigned integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.
- **NppStatus nppiDup_16s_C1C3R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI)
1 channel 16-bit signed integer source image duplicated in all 3 channels of destination image.
- **NppStatus nppiDup_16s_C1C4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI)
1 channel 16-bit signed integer source image duplicated in all 4 channels of destination image.
- **NppStatus nppiDup_16s_C1AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI)
1 channel 16-bit signed integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.
- **NppStatus nppiDup_32s_C1C3R** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI)
1 channel 32-bit signed integer source image duplicated in all 3 channels of destination image.
- **NppStatus nppiDup_32s_C1C4R** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI)

1 channel 32-bit signed integer source image duplicated in all 4 channels of destination image.

- **NppStatus nppiDup_32s_C1AC4R** (const **Npp32s** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oDstSizeROI)

1 channel 32-bit signed integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

- **NppStatus nppiDup_32f_C1C3R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI)

1 channel 32-bit floating point source image duplicated in all 3 channels of destination image.

- **NppStatus nppiDup_32f_C1C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI)

1 channel 32-bit floating point source image duplicated in all 4 channels of destination image.

- **NppStatus nppiDup_32f_C1AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oDstSizeROI)

1 channel 32-bit floating point source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

7.62.1 Function Documentation

7.62.1.1 **NppStatus nppiDup_16s_C1AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI)

1 channel 16-bit signed integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstSizeROI Size of the destination region-of-interest.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.62.1.2 **NppStatus nppiDup_16s_C1C3R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI)

1 channel 16-bit signed integer source image duplicated in all 3 channels of destination image.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.62.1.3 NppStatus nppiDup_16s_C1C4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oDstSizeROI)

1 channel 16-bit signed integer source image duplicated in all 4 channels of destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.62.1.4 NppStatus nppiDup_16u_C1AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI)

1 channel 16-bit unsigned integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.62.1.5 NppStatus nppiDup_16u_C1C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI)

1 channel 16-bit unsigned integer source image duplicated in all 3 channels of destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.62.1.6 NppStatus nppiDup_16u_C1C4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI)

1 channel 16-bit unsigned integer source image duplicated in all 4 channels of destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.62.1.7 NppStatus nppiDup_32f_C1AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI)

1 channel 32-bit floating point source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.62.1.8 **NppStatus nppiDup_32f_C1C3R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)

1 channel 32-bit floating point source image duplicated in all 3 channels of destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.62.1.9 **NppStatus nppiDup_32f_C1C4R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)

1 channel 32-bit floating point source image duplicated in all 4 channels of destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.62.1.10 **NppStatus nppiDup_32s_C1AC4R** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)

1 channel 32-bit signed integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.62.1.11 NppStatus nppiDup_32s_C1C3R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI)

1 channel 32-bit signed integer source image duplicated in all 3 channels of destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.62.1.12 NppStatus nppiDup_32s_C1C4R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI)

1 channel 32-bit signed integer source image duplicated in all 4 channels of destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.62.1.13 NppStatus nppiDup_8u_C1AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI)

1 channel 8-bit unsigned integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.62.1.14 NppStatus nppiDup_8u_C1C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)

1 channel 8-bit unsigned integer source image duplicated in all 3 channels of destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.62.1.15 NppStatus nppiDup_8u_C1C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)

1 channel 8-bit unsigned integer source image duplicated in all 4 channels of destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Size of the destination region-of-interest.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.63 Transpose

Transpose

Methods for transposing images of various types.

Like matrix transpose, image transpose is a mirror along the image's diagonal (upper-left to lower-right corner).

- `NppStatus nppiTranspose_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSrcROI)
1 channel 8-bit unsigned int image transpose.
- `NppStatus nppiTranspose_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSrcROI)
3 channel 8-bit unsigned int image transpose.
- `NppStatus nppiTranspose_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSrcROI)
4 channel 8-bit unsigned int image transpose.
- `NppStatus nppiTranspose_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSrcROI)
1 channel 16-bit unsigned int image transpose.
- `NppStatus nppiTranspose_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSrcROI)
3 channel 16-bit unsigned int image transpose.
- `NppStatus nppiTranspose_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSrcROI)
4 channel 16-bit unsigned int image transpose.
- `NppStatus nppiTranspose_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSrcROI)
1 channel 16-bit signed int image transpose.
- `NppStatus nppiTranspose_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSrcROI)
3 channel 16-bit signed int image transpose.
- `NppStatus nppiTranspose_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSrcROI)
4 channel 16-bit signed int image transpose.
- `NppStatus nppiTranspose_32s_C1R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSrcROI)
1 channel 32-bit signed int image transpose.
- `NppStatus nppiTranspose_32s_C3R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSrcROI)

3 channel 32-bit signed int image transpose.

- **NppStatus nppiTranspose_32s_C4R** (const [Npp32s](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSrcROI)

4 channel 32-bit signed int image transpose.

- **NppStatus nppiTranspose_32f_C1R** (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSrcROI)

1 channel 32-bit floating point image transpose.

- **NppStatus nppiTranspose_32f_C3R** (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSrcROI)

3 channel 32-bit floating point image transpose.

- **NppStatus nppiTranspose_32f_C4R** (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSrcROI)

4 channel 32-bit floating point image transpose.

7.63.1 Function Documentation

7.63.1.1 **NppStatus nppiTranspose_16s_C1R** (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pDst, int nDstStep, [NppiSize](#) oSrcROI)

1 channel 16-bit signed int image transpose.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst Pointer to the destination ROI.

nDstStep [Destination-Image Line Step](#).

oSrcROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.63.1.2 **NppStatus nppiTranspose_16s_C3R** (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pDst, int nDstStep, [NppiSize](#) oSrcROI)

3 channel 16-bit signed int image transpose.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst Pointer to the destination ROI.

nDstStep [Destination-Image Line Step](#).

oSrcROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.63.1.3 NppStatus nppiTranspose_16s_C4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSrcROI)

4 channel 16-bit signed int image transpose.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Pointer to the destination ROI](#).

nDstStep [Destination-Image Line Step](#).

oSrcROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.63.1.4 NppStatus nppiTranspose_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSrcROI)

1 channel 16-bit unsigned int image transpose.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Pointer to the destination ROI](#).

nDstStep [Destination-Image Line Step](#).

oSrcROI [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.63.1.5 NppStatus nppiTranspose_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSrcROI)

3 channel 16-bit unsigned int image transpose.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst Pointer to the destination ROI.
nDstStep Destination-Image Line Step.
oSrcROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.63.1.6 NppStatus nppiTranspose_16u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSrcROI*)

4 channel 16-bit unsigned int image transpose.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Pointer to the destination ROI.
nDstStep Destination-Image Line Step.
oSrcROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.63.1.7 NppStatus nppiTranspose_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSrcROI*)

1 channel 32-bit floating point image transpose.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Pointer to the destination ROI.
nDstStep Destination-Image Line Step.
oSrcROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.63.1.8 NppStatus nppiTranspose_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSrcROI*)

3 channel 32-bit floating point image transpose.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Pointer to the destination ROI.

nDstStep Destination-Image Line Step.

oSrcROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.63.1.9 NppStatus nppiTranspose_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSrcROI)

4 channel 32-bit floating point image transpose.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Pointer to the destination ROI.

nDstStep Destination-Image Line Step.

oSrcROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.63.1.10 NppStatus nppiTranspose_32s_C1R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSrcROI)

1 channel 32-bit signed int image transpose.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Pointer to the destination ROI.

nDstStep Destination-Image Line Step.

oSrcROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.63.1.11 NppStatus nppiTranspose_32s_C3R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSrcROI*)

3 channel 32-bit signed int image transpose.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Pointer to the destination ROI.
nDstStep Destination-Image Line Step.
oSrcROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.63.1.12 NppStatus nppiTranspose_32s_C4R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSrcROI*)

4 channel 32-bit signed int image transpose.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Pointer to the destination ROI.
nDstStep Destination-Image Line Step.
oSrcROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.63.1.13 NppStatus nppiTranspose_8u_C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSrcROI*)

1 channel 8-bit unsigned int image transpose.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Pointer to the destination ROI.
nDstStep Destination-Image Line Step.
oSrcROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.63.1.14 NppStatus nppiTranspose_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSrcROI*)

3 channel 8-bit unsigned int image transpose.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Pointer to the destination ROI.

nDstStep Destination-Image Line Step.

oSrcROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.63.1.15 NppStatus nppiTranspose_8u_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSrcROI*)

4 channel 8-bit unsigned int image transpose.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Pointer to the destination ROI.

nDstStep Destination-Image Line Step.

oSrcROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.64 Swap Channels

SwapChannels

Functions for swapping and duplicating channels in multiple channel images.

The methods support arbitrary permutations of the original channels, including replication and setting one or more channels to a constant value.

- **NppStatus** **nppiSwapChannels_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])
3 channel 8-bit unsigned integer source image to 3 channel destination image.
- **NppStatus** **nppiSwapChannels_8u_C3IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])
3 channel 8-bit unsigned integer in place image.
- **NppStatus** **nppiSwapChannels_8u_C4C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])
4 channel 8-bit unsigned integer source image to 3 channel destination image.
- **NppStatus** **nppiSwapChannels_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[4])
4 channel 8-bit unsigned integer source image to 4 channel destination image.
- **NppStatus** **nppiSwapChannels_8u_C4IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const int aDstOrder[4])
4 channel 8-bit unsigned integer in place image.
- **NppStatus** **nppiSwapChannels_8u_C3C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[4], const **Npp8u** nValue)
3 channel 8-bit unsigned integer source image to 4 channel destination image.
- **NppStatus** **nppiSwapChannels_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])
4 channel 8-bit unsigned integer source image to 4 channel destination image with destination alpha channel unaffected.
- **NppStatus** **nppiSwapChannels_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])
3 channel 16-bit unsigned integer source image to 3 channel destination image.
- **NppStatus** **nppiSwapChannels_16u_C3IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])
3 channel 16-bit unsigned integer in place image.
- **NppStatus** **nppiSwapChannels_16u_C4C3R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])
4 channel 16-bit unsigned integer source image to 3 channel destination image.
- **NppStatus** **nppiSwapChannels_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[4])

4 channel 16-bit unsigned integer source image to 4 channel destination image.

- `NppStatus nppiSwapChannels_16u_C4IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])`

4 channel 16-bit unsigned integer in place image.

- `NppStatus nppiSwapChannels_16u_C3C4R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4], const Npp16u nValue)`

3 channel 16-bit unsigned integer source image to 4 channel destination image.

- `NppStatus nppiSwapChannels_16u_AC4R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

4 channel 16-bit unsigned integer source image to 4 channel destination image with destination alpha channel unaffected.

- `NppStatus nppiSwapChannels_16s_C3R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

3 channel 16-bit signed integer source image to 3 channel destination image.

- `NppStatus nppiSwapChannels_16s_C3IR (Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

3 channel 16-bit signed integer in place image.

- `NppStatus nppiSwapChannels_16s_C4C3R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

4 channel 16-bit signed integer source image to 3 channel destination image.

- `NppStatus nppiSwapChannels_16s_C4R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4])`

4 channel 16-bit signed integer source image to 4 channel destination image.

- `NppStatus nppiSwapChannels_16s_C4IR (Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])`

4 channel 16-bit signed integer in place image.

- `NppStatus nppiSwapChannels_16s_C3C4R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4], const Npp16s nValue)`

3 channel 16-bit signed integer source image to 4 channel destination image.

- `NppStatus nppiSwapChannels_16s_AC4R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

4 channel 16-bit signed integer source image to 4 channel destination image with destination alpha channel unaffected.

- `NppStatus nppiSwapChannels_32s_C3R (const Npp32s *pSrc, int nSrcStep, Npp32s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

3 channel 32-bit signed integer source image to 3 channel destination image.

- `NppStatus nppiSwapChannels_32s_C3IR (Npp32s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

3 channel 32-bit signed integer in place image.

- [NppStatus nppiSwapChannels_32s_C4C3R](#) (const [Npp32s](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[3])
4 channel 32-bit signed integer source image to 3 channel destination image.
- [NppStatus nppiSwapChannels_32s_C4R](#) (const [Npp32s](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[4])
4 channel 32-bit signed integer source image to 4 channel destination image.
- [NppStatus nppiSwapChannels_32s_C4IR](#) ([Npp32s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[4])
4 channel 32-bit signed integer in place image.
- [NppStatus nppiSwapChannels_32s_C3C4R](#) (const [Npp32s](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[4], const [Npp32s](#) nValue)
3 channel 32-bit signed integer source image to 4 channel destination image.
- [NppStatus nppiSwapChannels_32s_AC4R](#) (const [Npp32s](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[3])
4 channel 32-bit signed integer source image to 4 channel destination image with destination alpha channel unaffected.
- [NppStatus nppiSwapChannels_32f_C3R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[3])
3 channel 32-bit floating point source image to 3 channel destination image.
- [NppStatus nppiSwapChannels_32f_C3IR](#) ([Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[3])
3 channel 32-bit floating point in place image.
- [NppStatus nppiSwapChannels_32f_C4C3R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[3])
4 channel 32-bit floating point source image to 3 channel destination image.
- [NppStatus nppiSwapChannels_32f_C4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[4])
4 channel 32-bit floating point source image to 4 channel destination image.
- [NppStatus nppiSwapChannels_32f_C4IR](#) ([Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[4])
4 channel 32-bit floating point in place image.
- [NppStatus nppiSwapChannels_32f_C3C4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[4], const [Npp32f](#) nValue)
3 channel 32-bit floating point source image to 4 channel destination image.
- [NppStatus nppiSwapChannels_32f_AC4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, const int aDstOrder[3])
4 channel 32-bit floating point source image to 4 channel destination image with destination alpha channel unaffected.

7.64.1 Function Documentation

7.64.1.1 NppStatus nppiSwapChannels_16s_AC4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 16-bit signed integer source image to 4 channel destination image with destination alpha channel unaffected.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDstOrder Host memory integer array describing how channel values are permuted. The *n*-th entry of the array contains the number of the channel that is stored in the *n*-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.2 NppStatus nppiSwapChannels_16s_C3C4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4], const Npp16s *nValue*)

3 channel 16-bit signed integer source image to 4 channel destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDstOrder Host memory integer array describing how channel values are permuted. The *n*-th entry of the array contains the number of the channel that is stored in the *n*-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [3,2,1,0] converts this to VBGR channel order.

nValue (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. *nValue* is either written or not written to a particular channel depending on the *aDstOrder* entry for that destination channel. An *aDstOrder* value of 3 will output *nValue* to that channel, an *aDstOrder* value greater than 3 will leave that particular destination channel value unmodified.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.3 **NppStatus nppiSwapChannels_16s_C3IR** (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 16-bit signed integer in place image.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.4 **NppStatus nppiSwapChannels_16s_C3R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 16-bit signed integer source image to 3 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.5 **NppStatus nppiSwapChannels_16s_C4C3R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 16-bit signed integer source image to 3 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to a 3 channel BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.6 NppStatus nppiSwapChannels_16s_C4IR (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4])

4 channel 16-bit signed integer in place image.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.7 NppStatus nppiSwapChannels_16s_C4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4])

4 channel 16-bit signed integer source image to 4 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.8 NppStatus nppiSwapChannels_16u_AC4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 16-bit unsigned integer source image to 4 channel destination image with destination alpha channel unaffected.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.9 NppStatus nppiSwapChannels_16u_C3C4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4], const Npp16u *nValue*)

3 channel 16-bit unsigned integer source image to 4 channel destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [3,2,1,0] converts this to VBGR channel order.

nValue (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. *nValue* is either written or not written to a particular channel depending on the *aDstOrder* entry for that destination channel. An *aDstOrder* value of 3 will output *nValue* to that channel, an *aDstOrder* value greater than 3 will leave that particular destination channel value unmodified.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.10 `NppStatus nppiSwapChannels_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

3 channel 16-bit unsigned integer in place image.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.11 `NppStatus nppiSwapChannels_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

3 channel 16-bit unsigned integer source image to 3 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.12 `NppStatus nppiSwapChannels_16u_C4C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

4 channel 16-bit unsigned integer source image to 3 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to a 3 channel BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.13 NppStatus nppiSwapChannels_16u_C4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])

4 channel 16-bit unsigned integer in place image.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.14 NppStatus nppiSwapChannels_16u_C4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4])

4 channel 16-bit unsigned integer source image to 4 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.15 `NppStatus nppiSwapChannels_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

4 channel 32-bit floating point source image to 4 channel destination image with destination alpha channel unaffected.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.16 `NppStatus nppiSwapChannels_32f_C3C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4], const Npp32f nValue)`

3 channel 32-bit floating point source image to 4 channel destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [3,2,1,0] converts this to VBGR channel order.

nValue (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. *nValue* is either written or not written to a particular channel depending on the *aDstOrder* entry for that destination channel. An *aDstOrder* value of 3 will output *nValue* to that channel, an *aDstOrder* value greater than 3 will leave that particular destination channel value unmodified.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.17 **NppStatus nppiSwapChannels_32f_C3IR** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 32-bit floating point in place image.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [oSizeROI Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.18 **NppStatus nppiSwapChannels_32f_C3R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 32-bit floating point source image to 3 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.19 **NppStatus nppiSwapChannels_32f_C4C3R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 32-bit floating point source image to 3 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to a 3 channel BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.20 NppStatus nppiSwapChannels_32f_C4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])

4 channel 32-bit floating point in place image.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.21 NppStatus nppiSwapChannels_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4])

4 channel 32-bit floating point source image to 4 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.22 NppStatus nppiSwapChannels_32s_AC4R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 32-bit signed integer source image to 4 channel destination image with destination alpha channel unaffected.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.23 NppStatus nppiSwapChannels_32s_C3C4R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4], const Npp32s *nValue*)

3 channel 32-bit signed integer source image to 4 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [3,2,1,0] converts this to VBGR channel order.

nValue (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. *nValue* is either written or not written to a particular channel depending on the *aDstOrder* entry for that destination channel. An *aDstOrder* value of 3 will output *nValue* to that channel, an *aDstOrder* value greater than 3 will leave that particular destination channel value unmodified.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.24 NppStatus nppiSwapChannels_32s_C3IR (Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[3])

3 channel 32-bit signed integer in place image.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.25 NppStatus nppiSwapChannels_32s_C3R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])

3 channel 32-bit signed integer source image to 3 channel destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.26 NppStatus nppiSwapChannels_32s_C4C3R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])

4 channel 32-bit signed integer source image to 3 channel destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to a 3 channel BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.27 NppStatus nppiSwapChannels_32s_C4IR (Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])

4 channel 32-bit signed integer in place image.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.28 NppStatus nppiSwapChannels_32s_C4R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4])

4 channel 32-bit signed integer source image to 4 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.29 NppStatus nppiSwapChannels_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 8-bit unsigned integer source image to 4 channel destination image with destination alpha channel unaffected.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDstOrder Host memory integer array describing how channel values are permuted. The *n*-th entry of the array contains the number of the channel that is stored in the *n*-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [3,2,1,0] converts this to VBGR channel order. of the array contains the number of the channel that is stored in the *n*-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.30 NppStatus nppiSwapChannels_8u_C3C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4], const Npp8u *nValue*)

3 channel 8-bit unsigned integer source image to 4 channel destination image.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDstOrder Host memory integer array describing how channel values are permuted. The *n*-th entry of the array contains the number of the channel that is stored in the *n*-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [3,2,1,0] converts this to VBGR channel order.

nValue (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. *nValue* is either written or not written to a particular channel depending on the *aDstOrder* entry for that destination channel. An *aDstOrder* value of 3 will output *nValue* to that channel, an *aDstOrder* value greater than 3 will leave that particular destination channel value unmodified.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.31 `NppStatus nppiSwapChannels_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

3 channel 8-bit unsigned integer in place image.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.32 `NppStatus nppiSwapChannels_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

3 channel 8-bit unsigned integer source image to 3 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.33 `NppStatus nppiSwapChannels_8u_C4C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

4 channel 8-bit unsigned integer source image to 3 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to a 3 channel BGR channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.34 NppStatus nppiSwapChannels_8u_C4IR (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4])

4 channel 8-bit unsigned integer in place image.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.64.1.35 NppStatus nppiSwapChannels_8u_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4])

4 channel 8-bit unsigned integer source image to 4 channel destination image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDstOrder Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65 Filtering Functions

Linear and non-linear image filtering functions.

Modules

- [1D Linear Filter](#)

FilterSobelVertSecondBorder

Filters the image using a second derivative, vertical Sobel filter kernel with border control:

$$\begin{pmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -2 & 0 & 1 \\ 4 & 0 & -8 & 0 & 4 \\ 6 & 0 & -12 & 0 & 6 \\ 4 & 0 & -8 & 0 & 4 \\ 1 & 0 & -2 & 0 & 1 \end{pmatrix}$$

- [NppStatus nppiFilterSobelVertSecondBorder_8u16s_C1R](#) (const [Npp8u](#) *pSrc, [Npp32s](#) nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiMaskSize](#) eMaskSize, [NppiBorderType](#) eBorderType)

Single channel 8-bit unsigned to 16-bit signed second derivative, vertical Sobel filter with border control.

- [NppStatus nppiFilterSobelVertSecondBorder_8s16s_C1R](#) (const [Npp8s](#) *pSrc, [Npp32s](#) nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiMaskSize](#) eMaskSize, [NppiBorderType](#) eBorderType)

Single channel 8-bit signed to 16-bit signed second derivative, vertical Sobel filter with border control.

- [NppStatus nppiFilterSobelVertSecondBorder_32f_C1R](#) (const [Npp32f](#) *pSrc, [Npp32s](#) nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp32f](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiMaskSize](#) eMaskSize, [NppiBorderType](#) eBorderType)

Single channel 32-bit floating-point second derivative, vertical Sobel filter with border control.

FilterSobelCrossBorder

Filters the image using a second cross derivative Sobel filter kernel with border control:

$$\begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -4 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 0 & -4 & -2 \\ 1 & 2 & 0 & -2 & -1 \end{pmatrix}$$

- [NppStatus nppiFilterSobelCrossBorder_8u16s_C1R](#) (const [Npp8u](#) *pSrc, [Npp32s](#) nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiMaskSize](#) eMaskSize, [NppiBorderType](#) eBorderType)

Single channel 8-bit unsigned to 16-bit signed second cross derivative Sobel filter with border control.

- `NppStatus nppiFilterSobelCrossBorder_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 8-bit signed to 16-bit signed second cross derivative Sobel filter with border control.

- `NppStatus nppiFilterSobelCrossBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point second cross derivative Sobel filter with border control.

FilterRobertsDown

Filters the image using a horizontal Roberts filter kernel:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterRobertsDown_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit unsigned horizontal Roberts filter.

- `NppStatus nppiFilterRobertsDown_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 8-bit unsigned horizontal Roberts filter.

- `NppStatus nppiFilterRobertsDown_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned horizontal Roberts filter.

- `NppStatus nppiFilterRobertsDown_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned horizontal Roberts filter, ignoring alpha-channel.

- `NppStatus nppiFilterRobertsDown_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 16-bit signed horizontal Roberts filter.

- `NppStatus nppiFilterRobertsDown_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 16-bit signed horizontal Roberts filter.

- `NppStatus nppiFilterRobertsDown_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed horizontal Roberts filter.

- `NppStatus nppiFilterRobertsDown_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed horizontal Roberts filter, ignoring alpha-channel.

- `NppStatus nppiFilterRobertsDown_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 32-bit floating-point horizontal Roberts filter.

- `NppStatus nppiFilterRobertsDown_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 32-bit floating-point horizontal Roberts filter.

- `NppStatus nppiFilterRobertsDown_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 32-bit floating-point horizontal Roberts filter.

- `NppStatus nppiFilterRobertsDown_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 32-bit floating-point horizontal Roberts filter, ignoring alpha-channel.

FilterRobertsDownBorder

Filters the image using a horizontal Roberts filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the `NPP_BORDER_REPLICATE` border type operation is supported.

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterRobertsDownBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned horizontal Roberts filter with border control.

- `NppStatus nppiFilterRobertsDownBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned horizontal Roberts filter with border control.

- `NppStatus nppiFilterRobertsDownBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned horizontal Roberts filter with border control.

- `NppStatus nppiFilterRobertsDownBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned horizontal Roberts filter with border control, ignoring alpha-channel.

- `NppStatus nppiFilterRobertsDownBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 16-bit signed horizontal Roberts filter with border control.

- `NppStatus nppiFilterRobertsDownBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 16-bit signed horizontal Roberts filter with border control.

- `NppStatus nppiFilterRobertsDownBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed horizontal Roberts filter with border control.

- `NppStatus nppiFilterRobertsDownBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed horizontal Roberts filter with border control, ignoring alpha-channel.

- `NppStatus nppiFilterRobertsDownBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point horizontal Roberts filter with border control.

- `NppStatus nppiFilterRobertsDownBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point horizontal Roberts filter with border control.

- `NppStatus nppiFilterRobertsDownBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point horizontal Roberts filter with border control.

- `NppStatus nppiFilterRobertsDownBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point horizontal Roberts filter with border control, ignoring alpha-channel.

FilterRobertsUp

Filters the image using a vertical Roberts filter kernel:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

- `NppStatus nppiFilterRobertsUp_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit unsigned vertical Roberts filter.

- `NppStatus nppiFilterRobertsUp_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 8-bit unsigned vertical Roberts filter.

- `NppStatus nppiFilterRobertsUp_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned vertical Roberts filter.

- `NppStatus nppiFilterRobertsUp_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned vertical Roberts filter, ignoring alpha-channel.

- `NppStatus nppiFilterRobertsUp_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 16-bit signed vertical Roberts filter.

- `NppStatus nppiFilterRobertsUp_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 16-bit signed vertical Roberts filter.

- `NppStatus nppiFilterRobertsUp_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed vertical Roberts filter.

- `NppStatus nppiFilterRobertsUp_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed vertical Roberts filter, ignoring alpha-channel.

- `NppStatus nppiFilterRobertsUp_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 32-bit floating-point vertical Roberts filter.

- `NppStatus nppiFilterRobertsUp_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 32-bit floating-point vertical Roberts filter.

- `NppStatus nppiFilterRobertsUp_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 32-bit floating-point vertical Roberts filter.

- `NppStatus nppiFilterRobertsUp_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 32-bit floating-point vertical Roberts filter, ignoring alpha-channel.

FilterRobertsUpBorder

Filters the image using a vertical Roberts filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

- `NppStatus nppiFilterRobertsUpBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned vertical Roberts filter with border control.

- `NppStatus nppiFilterRobertsUpBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned vertical Roberts filter with border control.

- `NppStatus nppiFilterRobertsUpBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned vertical Roberts filter with border control.

- `NppStatus nppiFilterRobertsUpBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned vertical Roberts filter with border control, ignoring alpha-channel.

- `NppStatus nppiFilterRobertsUpBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 16-bit signed vertical Roberts filter with border control.

- `NppStatus nppiFilterRobertsUpBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 16-bit signed vertical Roberts filter with border control.

- `NppStatus nppiFilterRobertsUpBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed vertical Roberts filter with border control.

- `NppStatus nppiFilterRobertsUpBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed vertical Roberts filter with border control, ignoring alpha-channel.

- `NppStatus nppiFilterRobertsUpBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point vertical Roberts filter with border control.

- `NppStatus nppiFilterRobertsUpBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point vertical Roberts filter with border control.

- `NppStatus nppiFilterRobertsUpBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point vertical Roberts filter with border control.

- `NppStatus nppiFilterRobertsUpBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point vertical Roberts filter with border control, ignoring alpha-channel.

FilterLaplace

Filters the image using a Laplacian filter kernel:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & -3 & -4 & -3 & -1 \\ -3 & 0 & 6 & 0 & -3 \\ -4 & 6 & 20 & 6 & -4 \\ -3 & 0 & 6 & 0 & -3 \\ -1 & -3 & -4 & -3 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterLaplace_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 8-bit unsigned Laplace filter.

- `NppStatus nppiFilterLaplace_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Three channel 8-bit unsigned Laplace filter.

- `NppStatus nppiFilterLaplace_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 8-bit unsigned Laplace filter.

- `NppStatus nppiFilterLaplace_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 8-bit unsigned Laplace filter, ignoring alpha channel.

- `NppStatus nppiFilterLaplace_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 16-bit signed Laplace filter.

- `NppStatus nppiFilterLaplace_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Three channel 16-bit signed Laplace filter.

- `NppStatus nppiFilterLaplace_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Four channel 16-bit signed Laplace filter.
- `NppStatus nppiFilterLaplace_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Four channel 16-bit signed Laplace filter, ignoring alpha channel.
- `NppStatus nppiFilterLaplace_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 32-bit floating-point Laplace filter.
- `NppStatus nppiFilterLaplace_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Three channel 32-bit floating-point Laplace filter.
- `NppStatus nppiFilterLaplace_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Four channel 32-bit floating-point Laplace filter.
- `NppStatus nppiFilterLaplace_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Four channel 32-bit floating-point Laplace filter, ignoring alpha channel.
- `NppStatus nppiFilterLaplace_8u16s_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 8-bit unsigned to 16-bit signed Laplace filter.
- `NppStatus nppiFilterLaplace_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 8-bit signed to 16-bit signed Laplace filter.

FilterLaplaceBorder

Filters the image using a Laplacian filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the `NPP_BORDER_REPLICATE` border type operation is supported.

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & -3 & -4 & -3 & -1 \\ -3 & 0 & 6 & 0 & -3 \\ -4 & 6 & 20 & 6 & -4 \\ -3 & 0 & 6 & 0 & -3 \\ -1 & -3 & -4 & -3 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterLaplaceBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned Laplace filter with border control.

- `NppStatus nppiFilterLaplaceBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned Laplace filter with border control.

- `NppStatus nppiFilterLaplaceBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned Laplace filter with border control.

- `NppStatus nppiFilterLaplaceBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned Laplace filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterLaplaceBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 16-bit signed Laplace filter with border control.

- `NppStatus nppiFilterLaplaceBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Three channel 16-bit signed Laplace filter with border control.

- `NppStatus nppiFilterLaplaceBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 16-bit signed Laplace filter with border control.

- `NppStatus nppiFilterLaplaceBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 16-bit signed Laplace filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterLaplaceBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point Laplace filter with border control.

- `NppStatus nppiFilterLaplaceBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point Laplace filter with border control.

- `NppStatus nppiFilterLaplaceBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point Laplace filter with border control.

- **NppStatus** **nppiFilterLaplaceBorder_32f_AC4R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Four channel 32-bit floating-point Laplace filter with border control, ignoring alpha channel.

- **NppStatus** **nppiFilterLaplaceBorder_8u16s_C1R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Single channel 8-bit unsigned to 16-bit signed Laplace filter with border control.

- **NppStatus** **nppiFilterLaplaceBorder_8s16s_C1R** (const **Npp8s** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Single channel 8-bit signed to 16-bit signed Laplace filter with border control.

FilterGauss

Filters the image using a Gaussian filter kernel:

Note that all FilterGauss functions currently support mask sizes up to 15x15.

$$\begin{pmatrix} 1/16 & 2/16 & 1/16 \\ 2/16 & 4/16 & 2/16 \\ 1/16 & 2/16 & 1/16 \end{pmatrix} \begin{pmatrix} 2/571 & 7/571 & 12/571 & 7/571 & 2/571 \\ 7/571 & 31/571 & 52/571 & 31/571 & 7/571 \\ 12/571 & 52/571 & 127/571 & 52/571 & 12/571 \\ 7/571 & 31/571 & 52/571 & 31/571 & 7/571 \\ 2/571 & 7/571 & 12/571 & 7/571 & 2/571 \end{pmatrix}$$

- **NppStatus** **nppiFilterGauss_8u_C1R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize)

Single channel 8-bit unsigned Gauss filter.

- **NppStatus** **nppiFilterGauss_8u_C3R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize)

Three channel 8-bit unsigned Gauss filter.

- **NppStatus** **nppiFilterGauss_8u_C4R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize)

Four channel 8-bit unsigned Gauss filter.

- **NppStatus** **nppiFilterGauss_8u_AC4R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize)

Four channel 8-bit unsigned Gauss filter, ignoring alpha channel.

- **NppStatus** **nppiFilterGauss_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize)

Single channel 16-bit unsigned Gauss filter.

- **NppStatus** **nppiFilterGauss_16u_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize)

Three channel 16-bit unsigned Gauss filter.

- `NppStatus nppiFilterGauss_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 16-bit unsigned Gauss filter.

- `NppStatus nppiFilterGauss_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 16-bit unsigned Gauss filter, ignoring alpha channel.

- `NppStatus nppiFilterGauss_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 16-bit signed Gauss filter.

- `NppStatus nppiFilterGauss_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Three channel 16-bit signed Gauss filter.

- `NppStatus nppiFilterGauss_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 16-bit signed Gauss filter.

- `NppStatus nppiFilterGauss_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 16-bit signed Gauss filter, ignoring alpha channel.

- `NppStatus nppiFilterGauss_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 32-bit floating-point Gauss filter.

- `NppStatus nppiFilterGauss_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Three channel 32-bit floating-point Gauss filter.

- `NppStatus nppiFilterGauss_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 32-bit floating-point Gauss filter.

- `NppStatus nppiFilterGauss_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 32-bit floating-point Gauss filter, ignoring alpha channel.

FilterGaussAdvanced

Filters the image using a separable Gaussian filter kernel with user supplied floating point coefficients:

- `NppStatus nppiFilterGaussAdvanced_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)

Single channel 8-bit unsigned Gauss filter.

- `NppStatus nppiFilterGaussAdvanced_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Three channel 8-bit unsigned Gauss filter.
- `NppStatus nppiFilterGaussAdvanced_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Four channel 8-bit unsigned Gauss filter.
- `NppStatus nppiFilterGaussAdvanced_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Four channel 8-bit unsigned Gauss filter, ignoring alpha channel.
- `NppStatus nppiFilterGaussAdvanced_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Single channel 16-bit unsigned Gauss filter.
- `NppStatus nppiFilterGaussAdvanced_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Three channel 16-bit unsigned Gauss filter.
- `NppStatus nppiFilterGaussAdvanced_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Four channel 16-bit unsigned Gauss filter.
- `NppStatus nppiFilterGaussAdvanced_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Four channel 16-bit unsigned Gauss filter, ignoring alpha channel.
- `NppStatus nppiFilterGaussAdvanced_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Single channel 16-bit signed Gauss filter.
- `NppStatus nppiFilterGaussAdvanced_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Three channel 16-bit signed Gauss filter.
- `NppStatus nppiFilterGaussAdvanced_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Four channel 16-bit signed Gauss filter.
- `NppStatus nppiFilterGaussAdvanced_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Four channel 16-bit signed Gauss filter, ignoring alpha channel.
- `NppStatus nppiFilterGaussAdvanced_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)
Single channel 32-bit floating-point Gauss filter.
- `NppStatus nppiFilterGaussAdvanced_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel)

Three channel 32-bit floating-point Gauss filter.

- `NppStatus nppiFilterGaussAdvanced_32f_C4R` (const `Npp32f *pSrc`, `Npp32s nSrcStep`, `Npp32f *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, const int `nFilterTaps`, const `Npp32f *pKernel`)

Four channel 32-bit floating-point Gauss filter.

- `NppStatus nppiFilterGaussAdvanced_32f_AC4R` (const `Npp32f *pSrc`, `Npp32s nSrcStep`, `Npp32f *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, const int `nFilterTaps`, const `Npp32f *pKernel`)

Four channel 32-bit floating-point Gauss filter, ignoring alpha channel.

FilterGaussBorder

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the `NPP_BORDER_REPLICATE` border type operation is supported.

Note that all `FilterGaussBorder` functions currently support mask sizes up to 15x15.

Filters the image using a Gaussian filter kernel:

$$\begin{pmatrix} 1/16 & 2/16 & 1/16 \\ 2/16 & 4/16 & 2/16 \\ 1/16 & 2/16 & 1/16 \end{pmatrix} \begin{pmatrix} 2/571 & 7/571 & 12/571 & 7/571 & 2/571 \\ 7/571 & 31/571 & 52/571 & 31/571 & 7/571 \\ 12/571 & 52/571 & 127/571 & 52/571 & 12/571 \\ 7/571 & 31/571 & 52/571 & 31/571 & 7/571 \\ 2/571 & 7/571 & 12/571 & 7/571 & 2/571 \end{pmatrix}$$

- `NppStatus nppiFilterGaussBorder_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `NppiSize oSrcSize`, `NppiPoint oSrcOffset`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiMaskSize eMaskSize`, `NppiBorderType eBorderType`)

Single channel 8-bit unsigned Gauss filter with border control.

- `NppStatus nppiFilterGaussBorder_8u_C3R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `NppiSize oSrcSize`, `NppiPoint oSrcOffset`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiMaskSize eMaskSize`, `NppiBorderType eBorderType`)

Three channel 8-bit unsigned Gauss filter with border control.

- `NppStatus nppiFilterGaussBorder_8u_C4R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `NppiSize oSrcSize`, `NppiPoint oSrcOffset`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiMaskSize eMaskSize`, `NppiBorderType eBorderType`)

Four channel 8-bit unsigned Gauss filter with border control.

- `NppStatus nppiFilterGaussBorder_8u_AC4R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `NppiSize oSrcSize`, `NppiPoint oSrcOffset`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiMaskSize eMaskSize`, `NppiBorderType eBorderType`)

Four channel 8-bit unsigned Gauss filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterGaussBorder_16u_C1R` (const `Npp16u *pSrc`, `Npp32s nSrcStep`, `NppiSize oSrcSize`, `NppiPoint oSrcOffset`, `Npp16u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiMaskSize eMaskSize`, `NppiBorderType eBorderType`)

Single channel 16-bit unsigned Gauss filter with border control.

- **NppStatus nppiFilterGaussBorder_16u_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Three channel 16-bit unsigned Gauss filter with border control.

- **NppStatus nppiFilterGaussBorder_16u_C4R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Four channel 16-bit unsigned Gauss filter with border control.

- **NppStatus nppiFilterGaussBorder_16u_AC4R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Four channel 16-bit unsigned Gauss filter with border control, ignoring alpha channel.

- **NppStatus nppiFilterGaussBorder_16s_C1R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Single channel 16-bit signed Gauss filter with border control.

- **NppStatus nppiFilterGaussBorder_16s_C3R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Three channel 16-bit signed Gauss filter with border control.

- **NppStatus nppiFilterGaussBorder_16s_C4R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Four channel 16-bit signed Gauss filter with border control.

- **NppStatus nppiFilterGaussBorder_16s_AC4R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Four channel 16-bit signed Gauss filter with border control, ignoring alpha channel.

- **NppStatus nppiFilterGaussBorder_32f_C1R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Single channel 32-bit floating-point Gauss filter with border control.

- **NppStatus nppiFilterGaussBorder_32f_C3R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Three channel 32-bit floating-point Gauss filter with border control.

- **NppStatus nppiFilterGaussBorder_32f_C4R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Four channel 32-bit floating-point Gauss filter with border control.

- **NppStatus nppiFilterGaussBorder_32f_AC4R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Four channel 32-bit floating-point Gauss filter with border control, ignoring alpha channel.

FilterGaussAdvancedBorder

Filters the image using a separable Gaussian filter kernel with user supplied floating point coefficients with border control: If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

- **NppStatus nppiFilterGaussAdvancedBorder_8u_C1R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const int nFilterTaps, const **Npp32f** *pKernel, **NppiBorderType** eBorderType)

Single channel 8-bit unsigned Gauss filter with border control.

- **NppStatus nppiFilterGaussAdvancedBorder_8u_C3R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const int nFilterTaps, const **Npp32f** *pKernel, **NppiBorderType** eBorderType)

Three channel 8-bit unsigned Gauss filter with border control.

- **NppStatus nppiFilterGaussAdvancedBorder_8u_C4R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const int nFilterTaps, const **Npp32f** *pKernel, **NppiBorderType** eBorderType)

Four channel 8-bit unsigned Gauss filter with border control.

- **NppStatus nppiFilterGaussAdvancedBorder_8u_AC4R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const int nFilterTaps, const **Npp32f** *pKernel, **NppiBorderType** eBorderType)

Four channel 8-bit unsigned Gauss filter with border control, ignoring alpha channel.

- **NppStatus nppiFilterGaussAdvancedBorder_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const int nFilterTaps, const **Npp32f** *pKernel, **NppiBorderType** eBorderType)

Single channel 16-bit unsigned Gauss filter with border control.

- **NppStatus nppiFilterGaussAdvancedBorder_16u_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const int nFilterTaps, const **Npp32f** *pKernel, **NppiBorderType** eBorderType)

Three channel 16-bit unsigned Gauss filter with border control.

- **NppStatus nppiFilterGaussAdvancedBorder_16u_C4R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const int nFilterTaps, const **Npp32f** *pKernel, **NppiBorderType** eBorderType)

Four channel 16-bit unsigned Gauss filter with border control.

- **NppStatus nppiFilterGaussAdvancedBorder_16u_AC4R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const int nFilterTaps, const **Npp32f** *pKernel, **NppiBorderType** eBorderType)

Four channel 16-bit unsigned Gauss filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterGaussAdvancedBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel, `NppiBorderType` eBorderType)

Single channel 16-bit signed Gauss filter with border control.

- `NppStatus nppiFilterGaussAdvancedBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel, `NppiBorderType` eBorderType)

Three channel 16-bit signed Gauss filter with border control.

- `NppStatus nppiFilterGaussAdvancedBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel, `NppiBorderType` eBorderType)

Four channel 16-bit signed Gauss filter with border control.

- `NppStatus nppiFilterGaussAdvancedBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel, `NppiBorderType` eBorderType)

Four channel 16-bit signed Gauss filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterGaussAdvancedBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point Gauss filter with border control.

- `NppStatus nppiFilterGaussAdvancedBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point Gauss filter with border control.

- `NppStatus nppiFilterGaussAdvancedBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point Gauss filter with border control.

- `NppStatus nppiFilterGaussAdvancedBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const int nFilterTaps, const `Npp32f` *pKernel, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point Gauss filter with border control, ignoring alpha channel.

FilterHighPass

Filters the image using a high-pass filter kernel:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterHighPass_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 8-bit unsigned high-pass filter.
- `NppStatus nppiFilterHighPass_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Three channel 8-bit unsigned high-pass filter.
- `NppStatus nppiFilterHighPass_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Four channel 8-bit unsigned high-pass filter.
- `NppStatus nppiFilterHighPass_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Four channel 8-bit unsigned high-pass filter, ignoring alpha channel.
- `NppStatus nppiFilterHighPass_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 16-bit unsigned high-pass filter.
- `NppStatus nppiFilterHighPass_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Three channel 16-bit unsigned high-pass filter.
- `NppStatus nppiFilterHighPass_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Four channel 16-bit unsigned high-pass filter.
- `NppStatus nppiFilterHighPass_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Four channel 16-bit unsigned high-pass filter, ignoring alpha channel.
- `NppStatus nppiFilterHighPass_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 16-bit signed high-pass filter.
- `NppStatus nppiFilterHighPass_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Three channel 16-bit signed high-pass filter.
- `NppStatus nppiFilterHighPass_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Four channel 16-bit signed high-pass filter.
- `NppStatus nppiFilterHighPass_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Four channel 16-bit signed high-pass filter, ignoring alpha channel.
- `NppStatus nppiFilterHighPass_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 32-bit floating-point high-pass filter.

- `NppStatus nppiFilterHighPass_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Three channel 32-bit floating-point high-pass filter.

- `NppStatus nppiFilterHighPass_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 32-bit floating-point high-pass filter.

- `NppStatus nppiFilterHighPass_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

FilterHighPassBorder

Filters the image using a high-pass filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the `NPP_BORDER_REPLICATE` border type operation is supported.

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterHighPassBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned high-pass filter.

- `NppStatus nppiFilterHighPassBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned high-pass filter.

- `NppStatus nppiFilterHighPassBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned high-pass filter.

- `NppStatus nppiFilterHighPassBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned high-pass filter, ignoring alpha channel.

- `NppStatus nppiFilterHighPassBorder_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 16-bit unsigned high-pass filter.

- `NppStatus nppiFilterHighPassBorder_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Three channel 16-bit unsigned high-pass filter.

- `NppStatus nppiFilterHighPassBorder_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned high-pass filter.

- `NppStatus nppiFilterHighPassBorder_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned high-pass filter, ignoring alpha channel.

- `NppStatus nppiFilterHighPassBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 16-bit signed high-pass filter.

- `NppStatus nppiFilterHighPassBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Three channel 16-bit signed high-pass filter.

- `NppStatus nppiFilterHighPassBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 16-bit signed high-pass filter.

- `NppStatus nppiFilterHighPassBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 16-bit signed high-pass filter, ignoring alpha channel.

- `NppStatus nppiFilterHighPassBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point high-pass filter.

- `NppStatus nppiFilterHighPassBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point high-pass filter.

- `NppStatus nppiFilterHighPassBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point high-pass filter.

- `NppStatus nppiFilterHighPassBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

FilterLowPass

Filters the image using a low-pass filter kernel:

$$\begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} \begin{pmatrix} 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \end{pmatrix}$$

- `NppStatus nppiFilterLowPass_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 8-bit unsigned low-pass filter.

- `NppStatus nppiFilterLowPass_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Three channel 8-bit unsigned low-pass filter.

- `NppStatus nppiFilterLowPass_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 8-bit unsigned low-pass filter.

- `NppStatus nppiFilterLowPass_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 8-bit unsigned low-pass filter, ignoring alpha channel.

- `NppStatus nppiFilterLowPass_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 16-bit unsigned low-pass filter.

- `NppStatus nppiFilterLowPass_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Three channel 16-bit unsigned low-pass filter.

- `NppStatus nppiFilterLowPass_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 16-bit unsigned low-pass filter.

- `NppStatus nppiFilterLowPass_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 16-bit unsigned low-pass filter, ignoring alpha channel.

- `NppStatus nppiFilterLowPass_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 16-bit signed low-pass filter.

- `NppStatus nppiFilterLowPass_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Three channel 16-bit signed low-pass filter.

- `NppStatus nppiFilterLowPass_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 16-bit signed low-pass filter.

- `NppStatus nppiFilterLowPass_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 16-bit signed low-pass filter, ignoring alpha channel.

- `NppStatus nppiFilterLowPass_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 32-bit floating-point low-pass filter.

- `NppStatus nppiFilterLowPass_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Three channel 32-bit floating-point low-pass filter.

- `NppStatus nppiFilterLowPass_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 32-bit floating-point low-pass filter.

- `NppStatus nppiFilterLowPass_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

FilterLowPassBorder

Filters the image using a low-pass filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the `NPP_BORDER_REPLICATE` border type operation is supported.

$$\begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} \begin{pmatrix} 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \end{pmatrix}$$

- `NppStatus nppiFilterLowPassBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned high-pass filter.

- `NppStatus nppiFilterLowPassBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned high-pass filter.

- `NppStatus nppiFilterLowPassBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned high-pass filter.

- `NppStatus nppiFilterLowPassBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned high-pass filter, ignoring alpha channel.

- `NppStatus nppiFilterLowPassBorder_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 16-bit unsigned high-pass filter.

- `NppStatus nppiFilterLowPassBorder_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Three channel 16-bit unsigned high-pass filter.

- `NppStatus nppiFilterLowPassBorder_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned high-pass filter.

- `NppStatus nppiFilterLowPassBorder_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned high-pass filter, ignoring alpha channel.

- `NppStatus nppiFilterLowPassBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 16-bit signed high-pass filter.

- `NppStatus nppiFilterLowPassBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Three channel 16-bit signed high-pass filter.

- `NppStatus nppiFilterLowPassBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 16-bit signed high-pass filter.

- `NppStatus nppiFilterLowPassBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 16-bit signed high-pass filter, ignoring alpha channel.

- `NppStatus nppiFilterLowPassBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point high-pass filter.

- `NppStatus nppiFilterLowPassBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point high-pass filter.

- `NppStatus nppiFilterLowPassBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point high-pass filter.

- `NppStatus nppiFilterLowPassBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

FilterSharpen

Filters the image using a sharpening filter kernel:

$$\begin{pmatrix} -1/8 & -1/8 & -1/8 \\ -1/8 & 16/8 & -1/8 \\ -1/8 & -1/8 & -1/8 \end{pmatrix}$$

- `NppStatus nppiFilterSharpen_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit unsigned sharpening filter.

- `NppStatus nppiFilterSharpen_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 8-bit unsigned sharpening filter.

- `NppStatus nppiFilterSharpen_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned sharpening filter.

- `NppStatus nppiFilterSharpen_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned sharpening filter, ignoring alpha channel.

- `NppStatus nppiFilterSharpen_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 16-bit unsigned sharpening filter.

- `NppStatus nppiFilterSharpen_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 16-bit unsigned sharpening filter.

- `NppStatus nppiFilterSharpen_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit unsigned sharpening filter.

- `NppStatus nppiFilterSharpen_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit unsigned sharpening filter, ignoring alpha channel.

- `NppStatus nppiFilterSharpen_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 16-bit signed sharpening filter.

- `NppStatus nppiFilterSharpen_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 16-bit signed sharpening filter.

- `NppStatus nppiFilterSharpen_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed sharpening filter.

- `NppStatus nppiFilterSharpen_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed sharpening filter, ignoring alpha channel.

- `NppStatus nppiFilterSharpen_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 32-bit floating-point sharpening filter.

- `NppStatus nppiFilterSharpen_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 32-bit floating-point sharpening filter.

- `NppStatus nppiFilterSharpen_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 32-bit floating-point sharpening filter.

- `NppStatus nppiFilterSharpen_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 32-bit floating-point sharpening filter, ignoring alpha channel.

FilterSharpenBorder

Filters the image using a sharpening filter kernel with border control.

If any portion of the 3x3 mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

$$\begin{pmatrix} -1/8 & -1/8 & -1/8 \\ -1/8 & 16/8 & -1/8 \\ -1/8 & -1/8 & -1/8 \end{pmatrix}$$

- **NppStatus nppiFilterSharpenBorder_8u_C1R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)

Single channel 8-bit unsigned sharpening filter with border control.

- **NppStatus nppiFilterSharpenBorder_8u_C3R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)

Three channel 8-bit unsigned sharpening filter with border control.

- **NppStatus nppiFilterSharpenBorder_8u_C4R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)

Four channel 8-bit unsigned sharpening filter with border control.

- **NppStatus nppiFilterSharpenBorder_8u_AC4R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)

Four channel 8-bit unsigned sharpening filter with border control, ignoring alpha channel.

- **NppStatus nppiFilterSharpenBorder_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)

Single channel 16-bit unsigned sharpening filter with border control.

- **NppStatus nppiFilterSharpenBorder_16u_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)

Three channel 16-bit unsigned sharpening filter with border control.

- **NppStatus nppiFilterSharpenBorder_16u_C4R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)

Four channel 16-bit unsigned sharpening filter with border control.

- **NppStatus nppiFilterSharpenBorder_16u_AC4R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)

Four channel 16-bit unsigned sharpening filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterSharpenBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 16-bit signed sharpening filter with border control.

- `NppStatus nppiFilterSharpenBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 16-bit signed sharpening filter with border control.

- `NppStatus nppiFilterSharpenBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed sharpening filter with border control.

- `NppStatus nppiFilterSharpenBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed sharpening filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterSharpenBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point sharpening filter with border control.

- `NppStatus nppiFilterSharpenBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point sharpening filter with border control.

- `NppStatus nppiFilterSharpenBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point sharpening filter with border control.

- `NppStatus nppiFilterSharpenBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point sharpening filter with border control, ignoring alpha channel.

FilterUnsharpBorder

Filters the image using a unsharp-mask sharpening filter kernel with border control.

The algorithm involves the following steps: Smooth the original image with a Gaussian filter, with the width controlled by the `nRadius`. Subtract the smoothed image from the original to create a high-pass filtered image. Apply any clipping needed on the high-pass image, as controlled by the `nThreshold`. Add a certain percentage of the high-pass filtered image to the original image, with the percentage controlled by the `nWeight`. In pseudocode this algorithm can be written as: $\text{HighPass} = \text{Image} - \text{Gaussian}(\text{Image})$ $\text{Result} = \text{Image} + \text{nWeight} * \text{HighPass} * (|\text{HighPass}| \geq \text{nThreshold})$ where `nWeight` is the amount, `nThreshold` is the threshold, and `>=` indicates a Boolean operation, 1 if true, or 0 otherwise.

If any portion of the mask overlaps the source image boundary, the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

- `NppStatus nppiFilterUnsharpBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Single channel 8-bit unsigned unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Three channel 8-bit unsigned unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Four channel 8-bit unsigned unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Four channel 8-bit unsigned unsharp filter (alpha channel is not processed).

- `NppStatus nppiFilterUnsharpBorder_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Single channel 16-bit unsigned unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Three channel 16-bit unsigned unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Four channel 16-bit unsigned unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Four channel 16-bit unsigned unsharp filter (alpha channel is not processed).

- `NppStatus nppiFilterUnsharpBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Single channel 16-bit signed unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Three channel 16-bit signed unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Four channel 16-bit signed unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Four channel 16-bit signed unsharp filter (alpha channel is not processed).

- `NppStatus nppiFilterUnsharpBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Single channel 32-bit floating point unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Three channel 32-bit floating point unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Four channel 32-bit floating point unsharp filter.

- `NppStatus nppiFilterUnsharpBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `Npp32f` nRadius, `Npp32f` nSigma, `Npp32f` nWeight, `Npp32f` nThreshold, `NppiBorderType` eBorderType, `Npp8u` *pDeviceBuffer)

Four channel 32-bit floating point unsharp filter (alpha channel is not processed).

- `NppStatus nppiFilterUnsharpGetBufferSize_8u_C1R` (const `Npp32f` nRadius, const `Npp32f` nSigma, int *hpBufferSize)

Single channel 8-bit unsigned unsharp filter scratch memory size.

- `NppStatus nppiFilterUnsharpGetBufferSize_8u_C3R` (const `Npp32f` nRadius, const `Npp32f` nSigma, int *hpBufferSize)

Three channel 8-bit unsigned unsharp filter scratch memory size.

- **NppStatus** **nppiFilterUnsharpGetBufferSize_8u_C4R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Four channel 8-bit unsigned unsharp filter scratch memory size.
- **NppStatus** **nppiFilterUnsharpGetBufferSize_8u_AC4R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Four channel 8-bit unsigned unsharp filter scratch memory size (alpha channel is not processed).
- **NppStatus** **nppiFilterUnsharpGetBufferSize_16u_C1R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Single channel 16-bit unsigned unsharp filter scratch memory size.
- **NppStatus** **nppiFilterUnsharpGetBufferSize_16u_C3R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Three channel 16-bit unsigned unsharp filter scratch memory size.
- **NppStatus** **nppiFilterUnsharpGetBufferSize_16u_C4R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Four channel 16-bit unsigned unsharp filter scratch memory size.
- **NppStatus** **nppiFilterUnsharpGetBufferSize_16u_AC4R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Four channel 16-bit unsigned unsharp filter scratch memory size (alpha channel is not processed).
- **NppStatus** **nppiFilterUnsharpGetBufferSize_16s_C1R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Single channel 16-bit signed unsharp filter scratch memory size.
- **NppStatus** **nppiFilterUnsharpGetBufferSize_16s_C3R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Three channel 16-bit signed unsharp filter scratch memory size.
- **NppStatus** **nppiFilterUnsharpGetBufferSize_16s_C4R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Four channel 16-bit signed unsharp filter scratch memory size.
- **NppStatus** **nppiFilterUnsharpGetBufferSize_16s_AC4R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Four channel 16-bit signed unsharp filter scratch memory size (alpha channel is not processed).
- **NppStatus** **nppiFilterUnsharpGetBufferSize_32f_C1R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Single channel 32-bit floating point unsharp filter scratch memory size.
- **NppStatus** **nppiFilterUnsharpGetBufferSize_32f_C3R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Three channel 32-bit floating point unsharp filter scratch memory size.
- **NppStatus** **nppiFilterUnsharpGetBufferSize_32f_C4R** (const **Npp32f** nRadius, const **Npp32f** nSigma, int *hpBufferSize)
Four channel 32-bit floating point unsharp filter scratch memory size.

- **NppStatus** `nppiFilterUnsharpGetBufferSize_32f_AC4R` (const **Npp32f** `nRadius`, const **Npp32f** `nSigma`, int `*hpBufferSize`)

Four channel 32-bit floating point unsharp filter scratch memory size (alpha channel is not processed).

7.65.1 Detailed Description

Linear and non-linear image filtering functions.

Filtering functions are classified as [Neighborhood Operations](#). It is the user's responsibility to avoid [Sampling Beyond Image Boundaries](#).

7.65.2 Function Documentation

7.65.2.1 **NppStatus** `nppiFilterGauss_16s_AC4R` (const **Npp16s** * `pSrc`, **Npp32s** `nSrcStep`, **Npp16s** * `pDst`, **Npp32s** `nDstStep`, **NppiSize** `oSizeROI`, **NppiMaskSize** `eMaskSize`)

Four channel 16-bit signed Gauss filter, ignoring alpha channel.

Parameters:

`pSrc` [Source-Image Pointer](#).
`nSrcStep` [Source-Image Line Step](#).
`pDst` [Destination-Image Pointer](#).
`nDstStep` [Destination-Image Line Step](#).
`oSizeROI` [Region-of-Interest \(ROI\)](#).
`eMaskSize` Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.2 **NppStatus** `nppiFilterGauss_16s_C1R` (const **Npp16s** * `pSrc`, **Npp32s** `nSrcStep`, **Npp16s** * `pDst`, **Npp32s** `nDstStep`, **NppiSize** `oSizeROI`, **NppiMaskSize** `eMaskSize`)

Single channel 16-bit signed Gauss filter.

Parameters:

`pSrc` [Source-Image Pointer](#).
`nSrcStep` [Source-Image Line Step](#).
`pDst` [Destination-Image Pointer](#).
`nDstStep` [Destination-Image Line Step](#).
`oSizeROI` [Region-of-Interest \(ROI\)](#).
`eMaskSize` Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.3 **NppStatus nppiFilterGauss_16s_C3R** (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 16-bit signed Gauss filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.4 **NppStatus nppiFilterGauss_16s_C4R** (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit signed Gauss filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.5 **NppStatus nppiFilterGauss_16u_AC4R** (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit unsigned Gauss filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.6 NppStatus nppiFilterGauss_16u_C1R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 16-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.7 NppStatus nppiFilterGauss_16u_C3R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 16-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.8 NppStatus nppiFilterGauss_16u_C4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.9 **NppStatus nppiFilterGauss_32f_AC4R** (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 32-bit floating-point Gauss filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.10 **NppStatus nppiFilterGauss_32f_C1R** (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 32-bit floating-point Gauss filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.11 **NppStatus nppiFilterGauss_32f_C3R** (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 32-bit floating-point Gauss filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.12 NppStatus nppiFilterGauss_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 32-bit floating-point Gauss filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.13 NppStatus nppiFilterGauss_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 8-bit unsigned Gauss filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.14 NppStatus nppiFilterGauss_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.15 **NppStatus nppiFilterGauss_8u_C3R** (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 8-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.16 **NppStatus nppiFilterGauss_8u_C4R** (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 8-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.17 **NppStatus nppiFilterGaussAdvanced_16s_AC4R** (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const int *nFilterTaps*, const Npp32f * *pKernel*)

Four channel 16-bit signed Gauss filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.18 `NppStatus nppiFilterGaussAdvanced_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Single channel 16-bit signed Gauss filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.19 `NppStatus nppiFilterGaussAdvanced_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Three channel 16-bit signed Gauss filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.20 `NppStatus nppiFilterGaussAdvanced_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Four channel 16-bit signed Gauss filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.21 `NppStatus nppiFilterGaussAdvanced_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Four channel 16-bit unsigned Gauss filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.22 `NppStatus nppiFilterGaussAdvanced_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Single channel 16-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.23 `NppStatus nppiFilterGaussAdvanced_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Three channel 16-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.24 `NppStatus nppiFilterGaussAdvanced_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Four channel 16-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.25 `NppStatus nppiFilterGaussAdvanced_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Four channel 32-bit floating-point Gauss filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.26 `NppStatus nppiFilterGaussAdvanced_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Single channel 32-bit floating-point Gauss filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.27 `NppStatus nppiFilterGaussAdvanced_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Three channel 32-bit floating-point Gauss filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.28 `NppStatus nppiFilterGaussAdvanced_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Four channel 32-bit floating-point Gauss filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.29 `NppStatus nppiFilterGaussAdvanced_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Four channel 8-bit unsigned Gauss filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.30 `NppStatus nppiFilterGaussAdvanced_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Single channel 8-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.31 `NppStatus nppiFilterGaussAdvanced_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Three channel 8-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.32 `NppStatus nppiFilterGaussAdvanced_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel)`

Four channel 8-bit unsigned Gauss filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.33 `NppStatus nppiFilterGaussAdvancedBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Four channel 16-bit signed Gauss filter with border control, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of *nFilterTaps* kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.34 `NppStatus nppiFilterGaussAdvancedBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Single channel 16-bit signed Gauss filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.
pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.35 `NppStatus nppiFilterGaussAdvancedBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Three channel 16-bit signed Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.
pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.36 `NppStatus nppiFilterGaussAdvancedBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Four channel 16-bit signed Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.37 `NppStatus nppiFilterGaussAdvancedBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Four channel 16-bit unsigned Gauss filter with border control, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.38 `NppStatus nppiFilterGaussAdvancedBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Single channel 16-bit unsigned Gauss filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.39 `NppStatus nppiFilterGaussAdvancedBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Three channel 16-bit unsigned Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.40 `NppStatus nppiFilterGaussAdvancedBorder_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Four channel 16-bit unsigned Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.41 `NppStatus nppiFilterGaussAdvancedBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Four channel 32-bit floating-point Gauss filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.42 `NppStatus nppiFilterGaussAdvancedBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Single channel 32-bit floating-point Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.43 `NppStatus nppiFilterGaussAdvancedBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Three channel 32-bit floating-point Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.44 `NppStatus nppiFilterGaussAdvancedBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Four channel 32-bit floating-point Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.45 `NppStatus nppiFilterGaussAdvancedBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Four channel 8-bit unsigned Gauss filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.46 `NppStatus nppiFilterGaussAdvancedBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Single channel 8-bit unsigned Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.47 `NppStatus nppiFilterGaussAdvancedBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Three channel 8-bit unsigned Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.48 `NppStatus nppiFilterGaussAdvancedBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nFilterTaps, const Npp32f * pKernel, NppiBorderType eBorderType)`

Four channel 8-bit unsigned Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nFilterTaps The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

pKernel Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.49 `NppStatus nppiFilterGaussBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 16-bit signed Gauss filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.50 `NppStatus nppiFilterGaussBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 16-bit signed Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.51 `NppStatus nppiFilterGaussBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 16-bit signed Gauss filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.52 `NppStatus nppiFilterGaussBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 16-bit signed Gauss filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.53 `NppStatus nppiFilterGaussBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 16-bit unsigned Gauss filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.54 `NppStatus nppiFilterGaussBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 16-bit unsigned Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.55 NppStatus nppiFilterGaussBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)

Three channel 16-bit unsigned Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.56 NppStatus nppiFilterGaussBorder_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)

Four channel 16-bit unsigned Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.57 `NppStatus nppiFilterGaussBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 32-bit floating-point Gauss filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.58 `NppStatus nppiFilterGaussBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.59 NppStatus nppiFilterGaussBorder_32f_C3R (**const Npp32f * pSrc**, **Npp32s nSrcStep**, **NppiSize oSrcSize**, **NppiPoint oSrcOffset**, **Npp32f * pDst**, **Npp32s nDstStep**, **NppiSize oSizeROI**, **NppiMaskSize eMaskSize**, **NppiBorderType eBorderType**)

Three channel 32-bit floating-point Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.60 NppStatus nppiFilterGaussBorder_32f_C4R (**const Npp32f * pSrc**, **Npp32s nSrcStep**, **NppiSize oSrcSize**, **NppiPoint oSrcOffset**, **Npp32f * pDst**, **Npp32s nDstStep**, **NppiSize oSizeROI**, **NppiMaskSize eMaskSize**, **NppiBorderType eBorderType**)

Four channel 32-bit floating-point Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.61 `NppStatus nppiFilterGaussBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 8-bit unsigned Gauss filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.62 `NppStatus nppiFilterGaussBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.63 `NppStatus nppiFilterGaussBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 8-bit unsigned Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.64 `NppStatus nppiFilterGaussBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 8-bit unsigned Gauss filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.65 NppStatus nppiFilterHighPass_16s_AC4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit signed high-pass filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.66 NppStatus nppiFilterHighPass_16s_C1R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 16-bit signed high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.67 NppStatus nppiFilterHighPass_16s_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 16-bit signed high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.68 NppStatus nppiFilterHighPass_16s_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit signed high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.69 NppStatus nppiFilterHighPass_16u_AC4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit unsigned high-pass filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.70 NppStatus nppiFilterHighPass_16u_C1R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 16-bit unsigned high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.71 NppStatus nppiFilterHighPass_16u_C3R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 16-bit unsigned high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.72 NppStatus nppiFilterHighPass_16u_C4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit unsigned high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.73 NppStatus nppiFilterHighPass_32f_AC4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.74 **NppStatus nppiFilterHighPass_32f_C1R** (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 32-bit floating-point high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.75 **NppStatus nppiFilterHighPass_32f_C3R** (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 32-bit floating-point high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.76 **NppStatus nppiFilterHighPass_32f_C4R** (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 32-bit floating-point high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.77 NppStatus nppiFilterHighPass_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 8-bit unsigned high-pass filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.78 NppStatus nppiFilterHighPass_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.79 NppStatus nppiFilterHighPass_8u_C3R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 8-bit unsigned high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.80 `NppStatus nppiFilterHighPass_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Four channel 8-bit unsigned high-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.81 `NppStatus nppiFilterHighPassBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 16-bit signed high-pass filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.82 `NppStatus nppiFilterHighPassBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 16-bit signed high-pass filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.83 `NppStatus nppiFilterHighPassBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 16-bit signed high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.84 `NppStatus nppiFilterHighPassBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 16-bit signed high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.85 `NppStatus nppiFilterHighPassBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 16-bit unsigned high-pass filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.86 `NppStatus nppiFilterHighPassBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 16-bit unsigned high-pass filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.87 `NppStatus nppiFilterHighPassBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 16-bit unsigned high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.88 `NppStatus nppiFilterHighPassBorder_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 16-bit unsigned high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.89 `NppStatus nppiFilterHighPassBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.90 `NppStatus nppiFilterHighPassBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.91 NppStatus nppiFilterHighPassBorder_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*, NppiBorderType *eBorderType*)

Three channel 32-bit floating-point high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.92 NppStatus nppiFilterHighPassBorder_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*, NppiBorderType *eBorderType*)

Four channel 32-bit floating-point high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.93 `NppStatus nppiFilterHighPassBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 8-bit unsigned high-pass filter, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.94 `NppStatus nppiFilterHighPassBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.95 `NppStatus nppiFilterHighPassBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 8-bit unsigned high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.96 `NppStatus nppiFilterHighPassBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 8-bit unsigned high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.97 NppStatus nppiFilterLaplace_16s_AC4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit signed Laplace filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.98 NppStatus nppiFilterLaplace_16s_C1R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 16-bit signed Laplace filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.99 NppStatus nppiFilterLaplace_16s_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 16-bit signed Laplace filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.100 NppStatus nppiFilterLaplace_16s_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit signed Laplace filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.101 NppStatus nppiFilterLaplace_32f_AC4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 32-bit floating-point Laplace filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.102 NppStatus nppiFilterLaplace_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 32-bit floating-point Laplace filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.103 NppStatus nppiFilterLaplace_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 32-bit floating-point Laplace filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.104 NppStatus nppiFilterLaplace_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 32-bit floating-point Laplace filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.105 NppStatus nppiFilterLaplace_8s16s_C1R (const Npp8s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit signed to 16-bit signed Laplace filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.106 NppStatus nppiFilterLaplace_8u16s_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned to 16-bit signed Laplace filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.107 NppStatus nppiFilterLaplace_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 8-bit unsigned Laplace filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.108 NppStatus nppiFilterLaplace_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned Laplace filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.109 **NppStatus nppiFilterLaplace_8u_C3R** (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 8-bit unsigned Laplace filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.110 **NppStatus nppiFilterLaplace_8u_C4R** (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 8-bit unsigned Laplace filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.111 **NppStatus nppiFilterLaplaceBorder_16s_AC4R** (const Npp16s * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*, NppiBorderType *eBorderType*)

Four channel 16-bit signed Laplace filter with border control, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to *pSrc*.
oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.112 `NppStatus nppiFilterLaplaceBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 16-bit signed Laplace filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.113 `NppStatus nppiFilterLaplaceBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 16-bit signed Laplace filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.114 NppStatus nppiFilterLaplaceBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)

Four channel 16-bit signed Laplace filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.115 NppStatus nppiFilterLaplaceBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)

Four channel 32-bit floating-point Laplace filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.116 `NppStatus nppiFilterLaplaceBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point Laplace filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.117 `NppStatus nppiFilterLaplaceBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 32-bit floating-point Laplace filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.118 `NppStatus nppiFilterLaplaceBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 32-bit floating-point Laplace filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.119 `NppStatus nppiFilterLaplaceBorder_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit signed to 16-bit signed Laplace filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.120 `NppStatus nppiFilterLaplaceBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed Laplace filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.121 `NppStatus nppiFilterLaplaceBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 8-bit unsigned Laplace filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.122 `NppStatus nppiFilterLaplaceBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned Laplace filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.123 `NppStatus nppiFilterLaplaceBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 8-bit unsigned Laplace filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.124 NppStatus nppiFilterLaplaceBorder_8u_C4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*, NppiBorderType *eBorderType*)

Four channel 8-bit unsigned Laplace filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.125 NppStatus nppiFilterLowPass_16s_AC4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit signed low-pass filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.126 NppStatus nppiFilterLowPass_16s_C1R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 16-bit signed low-pass filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.127 NppStatus nppiFilterLowPass_16s_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 16-bit signed low-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.128 NppStatus nppiFilterLowPass_16s_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit signed low-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.129 NppStatus nppiFilterLowPass_16u_AC4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit unsigned low-pass filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.130 NppStatus nppiFilterLowPass_16u_C1R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 16-bit unsigned low-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.131 NppStatus nppiFilterLowPass_16u_C3R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 16-bit unsigned low-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.132 NppStatus nppiFilterLowPass_16u_C4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit unsigned low-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.133 NppStatus nppiFilterLowPass_32f_AC4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.134 NppStatus nppiFilterLowPass_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 32-bit floating-point low-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.135 NppStatus nppiFilterLowPass_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 32-bit floating-point low-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.136 NppStatus nppiFilterLowPass_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 32-bit floating-point low-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.137 NppStatus nppiFilterLowPass_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 8-bit unsigned low-pass filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.138 NppStatus nppiFilterLowPass_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned low-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.139 NppStatus nppiFilterLowPass_8u_C3R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 8-bit unsigned low-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.140 NppStatus nppiFilterLowPass_8u_C4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 8-bit unsigned low-pass filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.141 `NppStatus nppiFilterLowPassBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 16-bit signed high-pass filter, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.142 `NppStatus nppiFilterLowPassBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 16-bit signed high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.143 `NppStatus nppiFilterLowPassBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 16-bit signed high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.144 `NppStatus nppiFilterLowPassBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 16-bit signed high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.145 `NppStatus nppiFilterLowPassBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 16-bit unsigned high-pass filter, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.146 `NppStatus nppiFilterLowPassBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 16-bit unsigned high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.147 `NppStatus nppiFilterLowPassBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 16-bit unsigned high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.148 `NppStatus nppiFilterLowPassBorder_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 16-bit unsigned high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.149 `NppStatus nppiFilterLowPassBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.150 `NppStatus nppiFilterLowPassBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.151 `NppStatus nppiFilterLowPassBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 32-bit floating-point high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.152 `NppStatus nppiFilterLowPassBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 32-bit floating-point high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.153 `NppStatus nppiFilterLowPassBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 8-bit unsigned high-pass filter, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.154 `NppStatus nppiFilterLowPassBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.155 `NppStatus nppiFilterLowPassBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Three channel 8-bit unsigned high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.156 `NppStatus nppiFilterLowPassBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Four channel 8-bit unsigned high-pass filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.157 NppStatus nppiFilterRobertsDown_16s_AC4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed horizontal Roberts filter, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.158 NppStatus nppiFilterRobertsDown_16s_C1R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 16-bit signed horizontal Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.159 NppStatus nppiFilterRobertsDown_16s_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit signed horizontal Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.160 NppStatus nppiFilterRobertsDown_16s_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed horizontal Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.161 NppStatus nppiFilterRobertsDown_32f_AC4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point horizontal Roberts filter, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.162 NppStatus nppiFilterRobertsDown_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit floating-point horizontal Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.163 NppStatus nppiFilterRobertsDown_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI)

Three channel 32-bit floating-point horizontal Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.164 NppStatus nppiFilterRobertsDown_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI)

Four channel 32-bit floating-point horizontal Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.165 NppStatus nppiFilterRobertsDown_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)

Four channel 8-bit unsigned horizontal Roberts filter, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.166 NppStatus nppiFilterRobertsDown_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit unsigned horizontal Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.167 NppStatus nppiFilterRobertsDown_8u_C3R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit unsigned horizontal Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.168 NppStatus nppiFilterRobertsDown_8u_C4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned horizontal Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.169 `NppStatus nppiFilterRobertsDownBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed horizontal Roberts filter with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.170 `NppStatus nppiFilterRobertsDownBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 16-bit signed horizontal Roberts filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.171 `NppStatus nppiFilterRobertsDownBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 16-bit signed horizontal Roberts filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.172 `NppStatus nppiFilterRobertsDownBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed horizontal Roberts filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.173 `NppStatus nppiFilterRobertsDownBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point horizontal Roberts filter with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.174 `NppStatus nppiFilterRobertsDownBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 32-bit floating-point horizontal Roberts filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.175 `NppStatus nppiFilterRobertsDownBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 32-bit floating-point horizontal Roberts filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.176 `NppStatus nppiFilterRobertsDownBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point horizontal Roberts filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.177 `NppStatus nppiFilterRobertsDownBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned horizontal Roberts filter with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.178 `NppStatus nppiFilterRobertsDownBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned horizontal Roberts filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.179 `NppStatus nppiFilterRobertsDownBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 8-bit unsigned horizontal Roberts filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.180 `NppStatus nppiFilterRobertsDownBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned horizontal Roberts filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.181 `NppStatus nppiFilterRobertsUp_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 16-bit signed vertical Roberts filter, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.182 `NppStatus nppiFilterRobertsUp_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Single channel 16-bit signed vertical Roberts filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.183 NppStatus nppiFilterRobertsUp_16s_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit signed vertical Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.184 NppStatus nppiFilterRobertsUp_16s_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed vertical Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.185 NppStatus nppiFilterRobertsUp_32f_AC4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point vertical Roberts filter, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.186 NppStatus nppiFilterRobertsUp_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit floating-point vertical Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.187 NppStatus nppiFilterRobertsUp_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 32-bit floating-point vertical Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.188 NppStatus nppiFilterRobertsUp_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point vertical Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.189 NppStatus nppiFilterRobertsUp_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned vertical Roberts filter, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.190 NppStatus nppiFilterRobertsUp_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit unsigned vertical Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.191 NppStatus nppiFilterRobertsUp_8u_C3R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit unsigned vertical Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.192 `NppStatus nppiFilterRobertsUp_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned vertical Roberts filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.193 `NppStatus nppiFilterRobertsUpBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed vertical Roberts filter with border control, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.194 `NppStatus nppiFilterRobertsUpBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 16-bit signed vertical Roberts filter with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.195 `NppStatus nppiFilterRobertsUpBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 16-bit signed vertical Roberts filter with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.196 `NppStatus nppiFilterRobertsUpBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed vertical Roberts filter with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.197 `NppStatus nppiFilterRobertsUpBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point vertical Roberts filter with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.198 `NppStatus nppiFilterRobertsUpBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 32-bit floating-point vertical Roberts filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.199 `NppStatus nppiFilterRobertsUpBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 32-bit floating-point vertical Roberts filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.200 `NppStatus nppiFilterRobertsUpBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point vertical Roberts filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.201 `NppStatus nppiFilterRobertsUpBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned vertical Roberts filter with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.202 `NppStatus nppiFilterRobertsUpBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned vertical Roberts filter with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.203 `NppStatus nppiFilterRobertsUpBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 8-bit unsigned vertical Roberts filter with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.204 `NppStatus nppiFilterRobertsUpBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned vertical Roberts filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.205 `NppStatus nppiFilterSharpen_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 16-bit signed sharpening filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.206 `NppStatus nppiFilterSharpen_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Single channel 16-bit signed sharpening filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.207 **NppStatus nppiFilterSharpen_16s_C3R** (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit signed sharpening filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.208 **NppStatus nppiFilterSharpen_16s_C4R** (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed sharpening filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.209 **NppStatus nppiFilterSharpen_16u_AC4R** (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit unsigned sharpening filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.210 `NppStatus nppiFilterSharpen_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Single channel 16-bit unsigned sharpening filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.211 `NppStatus nppiFilterSharpen_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Three channel 16-bit unsigned sharpening filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.212 NppStatus nppiFilterSharpen_16u_C4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit unsigned sharpening filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.213 NppStatus nppiFilterSharpen_32f_AC4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point sharpening filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.214 NppStatus nppiFilterSharpen_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit floating-point sharpening filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.215 NppStatus nppiFilterSharpen_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 32-bit floating-point sharpening filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.216 NppStatus nppiFilterSharpen_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point sharpening filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.217 NppStatus nppiFilterSharpen_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned sharpening filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.218 `NppStatus nppiFilterSharpen_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Single channel 8-bit unsigned sharpening filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.219 `NppStatus nppiFilterSharpen_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Three channel 8-bit unsigned sharpening filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.220 `NppStatus nppiFilterSharpen_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned sharpening filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.221 `NppStatus nppiFilterSharpenBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed sharpening filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.222 `NppStatus nppiFilterSharpenBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 16-bit signed sharpening filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.223 `NppStatus nppiFilterSharpenBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 16-bit signed sharpening filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.224 `NppStatus nppiFilterSharpenBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed sharpening filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.225 `NppStatus nppiFilterSharpenBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit unsigned sharpening filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.226 `NppStatus nppiFilterSharpenBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 16-bit unsigned sharpening filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.227 `NppStatus nppiFilterSharpenBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 16-bit unsigned sharpening filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.228 `NppStatus nppiFilterSharpenBorder_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit unsigned sharpening filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.229 `NppStatus nppiFilterSharpenBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point sharpening filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.230 `NppStatus nppiFilterSharpenBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 32-bit floating-point sharpening filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.231 `NppStatus nppiFilterSharpenBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 32-bit floating-point sharpening filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.232 `NppStatus nppiFilterSharpenBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point sharpening filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.233 `NppStatus nppiFilterSharpenBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned sharpening filter with border control, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.234 `NppStatus nppiFilterSharpenBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned sharpening filter with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.65.2.235 `NppStatus nppiFilterSharpenBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 8-bit unsigned sharpening filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.236 `NppStatus nppiFilterSharpenBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned sharpening filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.237 `NppStatus nppiFilterSobelCrossBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point second cross derivative Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eMaskSize Enumeration value specifying the mask size.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.238 `NppStatus nppiFilterSobelCrossBorder_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit signed to 16-bit signed second cross derivative Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eMaskSize Enumeration value specifying the mask size.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.239 `NppStatus nppiFilterSobelCrossBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed second cross derivative Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.240 `NppStatus nppiFilterSobelVertSecondBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point second derivative, vertical Sobel filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.241 `NppStatus nppiFilterSobelVertSecondBorder_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit signed to 16-bit signed second derivative, vertical Sobel filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.242 `NppStatus nppiFilterSobelVertSecondBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed second derivative, vertical Sobel filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.243 `NppStatus nppiFilterUnsharpBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Four channel 16-bit signed unsharp filter (alpha channel is not processed).

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

- nRadius*** The radius of the Gaussian filter, in pixels, not counting the center pixel.
- nSigma*** The standard deviation of the Gaussian filter, in pixel.
- nWeight*** The percentage of the difference between the original and the high pass image that is added back into the original.
- nThreshold*** The threshold needed to apply the difference amount.
- eBorderType*** The border type operation to be applied at source image border boundaries.
- pDeviceBuffer*** Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.244 **NppStatus** **nppiFilterUnsharpBorder_16s_C1R** (const Npp16s * *pSrc*, Npp32s *nSrcStep*, NppiPoint *oSrcOffset*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, Npp32f *nRadius*, Npp32f *nSigma*, Npp32f *nWeight*, Npp32f *nThreshold*, NppiBorderType *eBorderType*, Npp8u * *pDeviceBuffer*)

Single channel 16-bit signed unsharp filter.

Parameters:

- pSrc*** [Source-Image Pointer](#).
- nSrcStep*** [Source-Image Line Step](#).
- oSrcOffset*** The pixel offset that *pSrc* points to relative to the origin of the source image.
- pDst*** [Destination-Image Pointer](#).
- nDstStep*** [Destination-Image Line Step](#).
- oSizeROI*** [Region-of-Interest \(ROI\)](#).
- nRadius*** The radius of the Gaussian filter, in pixels, not counting the center pixel.
- nSigma*** The standard deviation of the Gaussian filter, in pixel.
- nWeight*** The percentage of the difference between the original and the high pass image that is added back into the original.
- nThreshold*** The threshold needed to apply the difference amount.
- eBorderType*** The border type operation to be applied at source image border boundaries.
- pDeviceBuffer*** Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.245 **NppStatus** **nppiFilterUnsharpBorder_16s_C3R** (const Npp16s * *pSrc*, Npp32s *nSrcStep*, NppiPoint *oSrcOffset*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, Npp32f *nRadius*, Npp32f *nSigma*, Npp32f *nWeight*, Npp32f *nThreshold*, NppiBorderType *eBorderType*, Npp8u * *pDeviceBuffer*)

Three channel 16-bit signed unsharp filter.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
nWeight The percentage of the difference between the original and the high pass image that is added back into the original.
nThreshold The threshold needed to apply the difference amount.
eBorderType The border type operation to be applied at source image border boundaries.
pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.246 `NppStatus nppiFilterUnsharpBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Four channel 16-bit signed unsharp filter.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
nWeight The percentage of the difference between the original and the high pass image that is added back into the original.
nThreshold The threshold needed to apply the difference amount.
eBorderType The border type operation to be applied at source image border boundaries.
pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.247 `NppStatus nppiFilterUnsharpBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Four channel 16-bit unsigned unsharp filter (alpha channel is not processed).

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

nWeight The percentage of the difference between the original and the high pass image that is added back into the original.

nThreshold The threshold needed to apply the difference amount.

eBorderType The border type operation to be applied at source image border boundaries.

pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.248 `NppStatus nppiFilterUnsharpBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Single channel 16-bit unsigned unsharp filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

nWeight The percentage of the difference between the original and the high pass image that is added back into the original.

nThreshold The threshold needed to apply the difference amount.

eBorderType The border type operation to be applied at source image border boundaries.

pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.249 `NppStatus nppiFilterUnsharpBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Three channel 16-bit unsigned unsharp filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

nWeight The percentage of the difference between the original and the high pass image that is added back into the original.

nThreshold The threshold needed to apply the difference amount.

eBorderType The border type operation to be applied at source image border boundaries.

pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.250 `NppStatus nppiFilterUnsharpBorder_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Four channel 16-bit unsigned unsharp filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

nWeight The percentage of the difference between the original and the high pass image that is added back into the original.

nThreshold The threshold needed to apply the difference amount.

eBorderType The border type operation to be applied at source image border boundaries.

pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.251 `NppStatus nppiFilterUnsharpBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Four channel 32-bit floating point unsharp filter (alpha channel is not processed).

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

nWeight The percentage of the difference between the original and the high pass image that is added back into the original.

nThreshold The threshold needed to apply the difference amount.

eBorderType The border type operation to be applied at source image border boundaries.

pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.252 `NppStatus nppiFilterUnsharpBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Single channel 32-bit floating point unsharp filter.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
nWeight The percentage of the difference between the original and the high pass image that is added back into the original.
nThreshold The threshold needed to apply the difference amount.
eBorderType The border type operation to be applied at source image border boundaries.
pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.253 `NppStatus npippiFilterUnsharpBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Three channel 32-bit floating point unsharp filter.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
nWeight The percentage of the difference between the original and the high pass image that is added back into the original.
nThreshold The threshold needed to apply the difference amount.
eBorderType The border type operation to be applied at source image border boundaries.
pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.254 `NppStatus nppiFilterUnsharpBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Four channel 32-bit floating point unsharp filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

nWeight The percentage of the difference between the original and the high pass image that is added back into the original.

nThreshold The threshold needed to apply the difference amount.

eBorderType The border type operation to be applied at source image border boundaries.

pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.255 `NppStatus nppiFilterUnsharpBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Four channel 8-bit unsigned unsharp filter (alpha channel is not processed).

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

nWeight The percentage of the difference between the original and the high pass image that is added back into the original.

nThreshold The threshold needed to apply the difference amount.

eBorderType The border type operation to be applied at source image border boundaries.

pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.256 NppStatus nppiFilterUnsharpBorder_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, NppiPoint *oSrcOffset*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, Npp32f *nRadius*, Npp32f *nSigma*, Npp32f *nWeight*, Npp32f *nThreshold*, NppiBorderType *eBorderType*, Npp8u * *pDeviceBuffer*)

Single channel 8-bit unsigned unsharp filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

nWeight The percentage of the difference between the original and the high pass image that is added back into the original.

nThreshold The threshold needed to apply the difference amount.

eBorderType The border type operation to be applied at source image border boundaries.

pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.257 NppStatus nppiFilterUnsharpBorder_8u_C3R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, NppiPoint *oSrcOffset*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, Npp32f *nRadius*, Npp32f *nSigma*, Npp32f *nWeight*, Npp32f *nThreshold*, NppiBorderType *eBorderType*, Npp8u * *pDeviceBuffer*)

Three channel 8-bit unsigned unsharp filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

nWeight The percentage of the difference between the original and the high pass image that is added back into the original.

nThreshold The threshold needed to apply the difference amount.

eBorderType The border type operation to be applied at source image border boundaries.

pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.258 `NppStatus nppiFilterUnsharpBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRadius, Npp32f nSigma, Npp32f nWeight, Npp32f nThreshold, NppiBorderType eBorderType, Npp8u * pDeviceBuffer)`

Four channel 8-bit unsigned unsharp filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

nWeight The percentage of the difference between the original and the high pass image that is added back into the original.

nThreshold The threshold needed to apply the difference amount.

eBorderType The border type operation to be applied at source image border boundaries.

pDeviceBuffer Pointer to the user-allocated device scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.65.2.259 `NppStatus nppiFilterUnsharpGetBufferSize_16s_AC4R (const Npp32f nRadius, const Npp32f nSigma, int * hpBufferSize)`

Four channel 16-bit signed unsharp filter scratch memory size (alpha channel is not processed).

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.260 NppStatus nppiFilterUnsharpGetBufferSize_16s_C1R (const Npp32f nRadius, const Npp32f nSigma, int * hpBufferSize)

Single channel 16-bit signed unsharp filter scratch memory size.

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.261 NppStatus nppiFilterUnsharpGetBufferSize_16s_C3R (const Npp32f nRadius, const Npp32f nSigma, int * hpBufferSize)

Three channel 16-bit signed unsharp filter scratch memory size.

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.262 NppStatus nppiFilterUnsharpGetBufferSize_16s_C4R (const Npp32f nRadius, const Npp32f nSigma, int * hpBufferSize)

Four channel 16-bit signed unsharp filter scratch memory size.

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.263 NppStatus nppiFilterUnsharpGetBufferSize_16u_AC4R (const Npp32f *nRadius*, const Npp32f *nSigma*, int * *hpBufferSize*)

Four channel 16-bit unsigned unsharp filter scratch memory size (alpha channel is not processed).

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.264 NppStatus nppiFilterUnsharpGetBufferSize_16u_C1R (const Npp32f *nRadius*, const Npp32f *nSigma*, int * *hpBufferSize*)

Single channel 16-bit unsigned unsharp filter scratch memory size.

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.265 NppStatus nppiFilterUnsharpGetBufferSize_16u_C3R (const Npp32f *nRadius*, const Npp32f *nSigma*, int * *hpBufferSize*)

Three channel 16-bit unsigned unsharp filter scratch memory size.

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.266 NppStatus nppiFilterUnsharpGetBufferSize_16u_C4R (const Npp32f *nRadius*, const Npp32f *nSigma*, int * *hpBufferSize*)

Four channel 16-bit unsigned unsharp filter scratch memory size.

Parameters:

- nRadius* The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.267 NppStatus nppiFilterUnsharpGetBufferSize_32f_AC4R (const Npp32f nRadius, const Npp32f nSigma, int * hpBufferSize)

Four channel 32-bit floating point unsharp filter scratch memory size (alpha channel is not processed).

Parameters:

- nRadius* The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.268 NppStatus nppiFilterUnsharpGetBufferSize_32f_C1R (const Npp32f nRadius, const Npp32f nSigma, int * hpBufferSize)

Single channel 32-bit floating point unsharp filter scratch memory size.

Parameters:

- nRadius* The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.269 NppStatus nppiFilterUnsharpGetBufferSize_32f_C3R (const Npp32f nRadius, const Npp32f nSigma, int * hpBufferSize)

Three channel 32-bit floating point unsharp filter scratch memory size.

Parameters:

- nRadius* The radius of the Gaussian filter, in pixels, not counting the center pixel.
nSigma The standard deviation of the Gaussian filter, in pixel.
hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.270 NppStatus nppiFilterUnsharpGetBufferSize_32f_C4R (const Npp32f *nRadius*, const Npp32f *nSigma*, int * *hpBufferSize*)

Four channel 32-bit floating point unsharp filter scratch memory size.

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.271 NppStatus nppiFilterUnsharpGetBufferSize_8u_AC4R (const Npp32f *nRadius*, const Npp32f *nSigma*, int * *hpBufferSize*)

Four channel 8-bit unsigned unsharp filter scratch memory size (alpha channel is not processed).

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.272 NppStatus nppiFilterUnsharpGetBufferSize_8u_C1R (const Npp32f *nRadius*, const Npp32f *nSigma*, int * *hpBufferSize*)

Single channel 8-bit unsigned unsharp filter scratch memory size.

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.273 NppStatus nppiFilterUnsharpGetBufferSize_8u_C3R (const Npp32f *nRadius*, const Npp32f *nSigma*, int * *hpBufferSize*)

Three channel 8-bit unsigned unsharp filter scratch memory size.

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.65.2.274 NppStatus nppiFilterUnsharpGetBufferSize_8u_C4R (const Npp32f *nRadius*, const Npp32f *nSigma*, int * *hpBufferSize*)

Four channel 8-bit unsigned unsharp filter scratch memory size.

Parameters:

nRadius The radius of the Gaussian filter, in pixels, not counting the center pixel.

nSigma The standard deviation of the Gaussian filter, in pixel.

hpBufferSize Pointer to the size of the scratch buffer required for the unsharp operation.

Returns:

[Image Data Related Error Codes](#)

7.66 1D Linear Filter

Modules

- [1D Window Sum](#)
- [1D Window Sum with Border Control](#)
- [Convolution](#)
- [2D Fixed Linear Filters](#)
- [Rank Filters](#)
- [Fixed Filters](#)

Fixed filters perform linear filtering operations (i.e.

FilterColumn

Apply convolution filter with user specified 1D column of weights.

Result pixel is equal to the sum of the products between the kernel coefficients (pKernel array) and corresponding neighboring column pixel values in the source image defined by nKernelDim and nAnchorY, divided by nDivisor.

- [NppStatus nppiFilterColumn_8u_C1R](#) (const [Npp8u](#) *pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) *pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

8-bit unsigned single-channel 1D column convolution.

- [NppStatus nppiFilterColumn_8u_C3R](#) (const [Npp8u](#) *pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) *pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

8-bit unsigned three-channel 1D column convolution.

- [NppStatus nppiFilterColumn_8u_C4R](#) (const [Npp8u](#) *pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) *pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

8-bit unsigned four-channel 1D column convolution.

- [NppStatus nppiFilterColumn_8u_AC4R](#) (const [Npp8u](#) *pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) *pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

8-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

- [NppStatus nppiFilterColumn_16u_C1R](#) (const [Npp16u](#) *pSrc, [Npp32s](#) nSrcStep, [Npp16u](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) *pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

16-bit unsigned single-channel 1D column convolution.

- [NppStatus nppiFilterColumn_16u_C3R](#) (const [Npp16u](#) *pSrc, [Npp32s](#) nSrcStep, [Npp16u](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) *pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

16-bit unsigned three-channel 1D column convolution.

- `NppStatus nppiFilterColumn_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit unsigned four-channel 1D column convolution.

- `NppStatus nppiFilterColumn_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

- `NppStatus nppiFilterColumn_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit single-channel 1D column convolution.

- `NppStatus nppiFilterColumn_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit three-channel 1D column convolution.

- `NppStatus nppiFilterColumn_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit four-channel 1D column convolution.

- `NppStatus nppiFilterColumn_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit four-channel 1D column convolution ignoring alpha-channel.

- `NppStatus nppiFilterColumn_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

32-bit float single-channel 1D column convolution.

- `NppStatus nppiFilterColumn_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

32-bit float three-channel 1D column convolution.

- `NppStatus nppiFilterColumn_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

32-bit float four-channel 1D column convolution.

- `NppStatus nppiFilterColumn_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

32-bit float four-channel 1D column convolution ignoring alpha-channel.

- `NppStatus nppiFilterColumn_64f_C1R` (const `Npp64f` *pSrc, `Npp32s` nSrcStep, `Npp64f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp64f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

64-bit float single-channel 1D column convolution.

FilterColumnBorder

General purpose 1D convolution column filter with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

- **NppStatus nppiFilterColumnBorder_8u_C1R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor, **Npp32s** nDivisor, **NppiBorderType** eBorderType)

Single channel 8-bit unsigned 1D column convolution filter with border control.

- **NppStatus nppiFilterColumnBorder_8u_C3R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor, **Npp32s** nDivisor, **NppiBorderType** eBorderType)

Three channel 8-bit unsigned 1D column convolution filter with border control.

- **NppStatus nppiFilterColumnBorder_8u_C4R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor, **Npp32s** nDivisor, **NppiBorderType** eBorderType)

Four channel channel 8-bit unsigned 1D column convolution filter with border control.

- **NppStatus nppiFilterColumnBorder_8u_AC4R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor, **Npp32s** nDivisor, **NppiBorderType** eBorderType)

Four channel 8-bit unsigned convolution 1D column filter with border control, ignoring alpha channel.

- **NppStatus nppiFilterColumnBorder_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor, **Npp32s** nDivisor, **NppiBorderType** eBorderType)

Single channel 16-bit unsigned convolution 1D column filter with border control.

- **NppStatus nppiFilterColumnBorder_16u_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor, **Npp32s** nDivisor, **NppiBorderType** eBorderType)

Three channel 16-bit unsigned 1D column convolution filter with border control.

- **NppStatus nppiFilterColumnBorder_16u_C4R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp32s** *pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor, **Npp32s** nDivisor, **NppiBorderType** eBorderType)

Four channel channel 16-bit 1D column unsigned convolution filter with border control.

- `NppStatus nppiFilterColumnBorder_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned 1D column convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterColumnBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Single channel 16-bit 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Three channel 16-bit 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel channel 16-bit 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel 16-bit 1D column convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterColumnBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Single channel 32-bit float 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 32-bit float 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit float 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit float 1D column convolution filter with border control, ignoring alpha channel.

FilterColumn32f

FilterColumn using floating-point weights.

- `NppStatus nppiFilterColumn32f_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
8-bit unsigned single-channel 1D column convolution.
- `NppStatus nppiFilterColumn32f_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
8-bit unsigned three-channel 1D column convolution.
- `NppStatus nppiFilterColumn32f_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
8-bit unsigned four-channel 1D column convolution.
- `NppStatus nppiFilterColumn32f_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
8-bit unsigned four-channel 1D column convolution ignoring alpha-channel.
- `NppStatus nppiFilterColumn32f_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
16-bit unsigned single-channel 1D column convolution.
- `NppStatus nppiFilterColumn32f_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
16-bit unsigned three-channel 1D column convolution.
- `NppStatus nppiFilterColumn32f_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
16-bit unsigned four-channel 1D column convolution.
- `NppStatus nppiFilterColumn32f_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
16-bit unsigned four-channel 1D column convolution ignoring alpha-channel.
- `NppStatus nppiFilterColumn32f_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
16-bit single-channel 1D column convolution.
- `NppStatus nppiFilterColumn32f_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
16-bit three-channel 1D column convolution.
- `NppStatus nppiFilterColumn32f_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
16-bit four-channel 1D column convolution.
- `NppStatus nppiFilterColumn32f_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
16-bit four-channel 1D column convolution ignoring alpha-channel.

16-bit four-channel 1D column convolution ignoring alpha-channel.

FilterColumnBorder32f

General purpose 1D column convolution filter using floating-point weights with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

- `NppStatus nppiFilterColumnBorder32f_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder32f_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder32f_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder32f_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned 1D column convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterColumnBorder32f_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Single channel 16-bit unsigned 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder32f_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit unsigned 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder32f_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder32f_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned 1D column convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterColumnBorder32f_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Single channel 16-bit 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder32f_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder32f_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit 1D column convolution filter with border control.

- `NppStatus nppiFilterColumnBorder32f_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit 1D column convolution filter with border control, ignoring alpha channel.

FilterRow

Apply convolution filter with user specified 1D row of weights.

Result pixel is equal to the sum of the products between the kernel coefficients (pKernel array) and corresponding neighboring row pixel values in the source image defined by nKernelDim and nAnchorX, divided by nDivisor.

- `NppStatus nppiFilterRow_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

8-bit unsigned single-channel 1D row convolution.

- `NppStatus nppiFilterRow_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

8-bit unsigned three-channel 1D row convolution.

- `NppStatus nppiFilterRow_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

8-bit unsigned four-channel 1D row convolution.

- `NppStatus nppiFilterRow_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

8-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

- `NppStatus nppiFilterRow_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit unsigned single-channel 1D row convolution.

- `NppStatus nppiFilterRow_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit unsigned three-channel 1D row convolution.

- `NppStatus nppiFilterRow_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit unsigned four-channel 1D row convolution.

- `NppStatus nppiFilterRow_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

- `NppStatus nppiFilterRow_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit single-channel 1D row convolution.

- `NppStatus nppiFilterRow_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit three-channel 1D row convolution.

- `NppStatus nppiFilterRow_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit four-channel 1D row convolution.

- `NppStatus nppiFilterRow_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

16-bit four-channel 1D row convolution ignoring alpha-channel.

- `NppStatus nppiFilterRow_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

32-bit float single-channel 1D row convolution.

- `NppStatus nppiFilterRow_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

32-bit float three-channel 1D row convolution.

- `NppStatus nppiFilterRow_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

32-bit float four-channel 1D row convolution.

- `NppStatus nppiFilterRow_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
32-bit float four-channel 1D row convolution ignoring alpha-channel.
- `NppStatus nppiFilterRow_64f_C1R` (const `Npp64f` *pSrc, `Npp32s` nSrcStep, `Npp64f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp64f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
64-bit float single-channel 1D row convolution.

FilterRowBorder

General purpose 1D convolution row filter with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

- `NppStatus nppiFilterRowBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)
Single channel 8-bit unsigned 1D row convolution filter with border control.
- `NppStatus nppiFilterRowBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)
Three channel 8-bit unsigned 1D row convolution filter with border control.
- `NppStatus nppiFilterRowBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)
Four channel 8-bit unsigned 1D row convolution filter with border control.
- `NppStatus nppiFilterRowBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)
Four channel 8-bit unsigned convolution 1D row filter with border control, ignoring alpha channel.
- `NppStatus nppiFilterRowBorder_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)
Single channel 16-bit unsigned convolution 1D row filter with border control.

- `NppStatus nppiFilterRowBorder_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Three channel 16-bit unsigned 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel channel 16-bit 1D row unsigned convolution filter with border control.

- `NppStatus nppiFilterRowBorder_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned 1D row convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterRowBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Single channel 16-bit 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Three channel 16-bit 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel channel 16-bit 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel 16-bit 1D row convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterRowBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Single channel 32-bit float 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 32-bit float 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit float 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit float 1D row convolution filter with border control, ignoring alpha channel.

FilterRow32f

FilterRow using floating-point weights.

- `NppStatus nppiFilterRow32f_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

8-bit unsigned single-channel 1D row convolution.

- `NppStatus nppiFilterRow32f_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

8-bit unsigned three-channel 1D row convolution.

- `NppStatus nppiFilterRow32f_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

8-bit unsigned four-channel 1D row convolution.

- `NppStatus nppiFilterRow32f_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

8-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

- `NppStatus nppiFilterRow32f_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

16-bit unsigned single-channel 1D row convolution.

- `NppStatus nppiFilterRow32f_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

16-bit unsigned three-channel 1D row convolution.

- `NppStatus nppiFilterRow32f_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

16-bit unsigned four-channel 1D row convolution.

- `NppStatus nppiFilterRow32f_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

16-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

- `NppStatus nppiFilterRow32f_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

16-bit single-channel 1D row convolution.

- `NppStatus nppiFilterRow32f_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
16-bit three-channel 1D row convolution.
- `NppStatus nppiFilterRow32f_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
16-bit four-channel 1D row convolution.
- `NppStatus nppiFilterRow32f_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)
16-bit four-channel 1D row convolution ignoring alpha-channel.

FilterRowBorder32f

General purpose 1D row convolution filter using floating-point weights with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the `NPP_BORDER_REPLICATE` border type operation is supported.

- `NppStatus nppiFilterRowBorder32f_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)
Single channel 8-bit unsigned 1D row convolution filter with border control.
- `NppStatus nppiFilterRowBorder32f_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)
Three channel 8-bit unsigned 1D row convolution filter with border control.
- `NppStatus nppiFilterRowBorder32f_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)
Four channel 8-bit unsigned 1D row convolution filter with border control.
- `NppStatus nppiFilterRowBorder32f_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)
Four channel 8-bit unsigned 1D row convolution filter with border control, ignoring alpha channel.
- `NppStatus nppiFilterRowBorder32f_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)
Single channel 16-bit unsigned 1D row convolution filter with border control.
- `NppStatus nppiFilterRowBorder32f_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit unsigned 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder32f_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder32f_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned 1D row convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterRowBorder32f_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Single channel 16-bit 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder32f_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder32f_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit 1D row convolution filter with border control.

- `NppStatus nppiFilterRowBorder32f_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit 1D row convolution filter with border control, ignoring alpha channel.

FilterSobelVertSecond

Filters the image using a second derivative, vertical Sobel filter kernel:

$$\begin{pmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -2 & 0 & 1 \\ 4 & 0 & -8 & 0 & 4 \\ 6 & 0 & -12 & 0 & 6 \\ 4 & 0 & -8 & 0 & 4 \\ 1 & 0 & -2 & 0 & 1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelVertSecond_8u16s_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 8-bit unsigned to 16-bit signed second derivative, vertical Sobel filter.

- `NppStatus nppiFilterSobelVertSecond_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 8-bit signed to 16-bit signed second derivative, vertical Sobel filter.

- `NppStatus nppiFilterSobelVertSecond_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 32-bit floating-point second derivative, vertical Sobel filter.

FilterSobelCross

Filters the image using a second cross derivative Sobel filter kernel:

$$\begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -4 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 0 & -4 & -2 \\ 1 & 2 & 0 & -2 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelCross_8u16s_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 8-bit unsigned to 16-bit signed second cross derivative Sobel filter.

- `NppStatus nppiFilterSobelCross_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 8-bit signed to 16-bit signed second cross derivative Sobel filter.

- `NppStatus nppiFilterSobelCross_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Single channel 32-bit floating-point second cross derivative Sobel filter.

FilterSobelHorizBorder

Filters the image using a horizontal Sobel filter kernel with border control:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelHorizBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned horizontal Sobel filter with border control.

- `NppStatus nppiFilterSobelHorizBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned horizontal Sobel filter with border control.

- `NppStatus nppiFilterSobelHorizBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned horizontal Sobel filter with border control.

- `NppStatus nppiFilterSobelHorizBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed horizontal Sobel filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterSobelHorizBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 16-bit signed horizontal Sobel filter with border control.

- `NppStatus nppiFilterSobelHorizBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 16-bit signed horizontal Sobel filter with border control.

- `NppStatus nppiFilterSobelHorizBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed horizontal Sobel filter with border control.

- `NppStatus nppiFilterSobelHorizBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned horizontal Sobel filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterSobelHorizBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point horizontal Sobel filter with border control.

- `NppStatus nppiFilterSobelHorizBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point horizontal Sobel filter with border control.

- `NppStatus nppiFilterSobelHorizBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point horizontal Sobel filter with border control.

- `NppStatus nppiFilterSobelHorizBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point horizontal Sobel filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterSobelHorizBorder_8u16s_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned to 16-bit signed horizontal Sobel filter with border control.

- `NppStatus nppiFilterSobelHorizBorder_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 8-bit signed to 16-bit signed horizontal Sobel filter with border control.

- `NppStatus nppiFilterSobelHorizMaskBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point horizontal Sobel filter with border control.

FilterSobelVertBorder

Filters the image using a vertical Sobel filter kernel with border control:

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelVertBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned vertical Sobel filter with border control.

- `NppStatus nppiFilterSobelVertBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned vertical Sobel filter with border control.

- `NppStatus nppiFilterSobelVertBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned vertical Sobel filter with border control.

- `NppStatus nppiFilterSobelVertBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed vertical Sobel filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterSobelVertBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 16-bit signed vertical Sobel filter with border control.

- `NppStatus nppiFilterSobelVertBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 16-bit signed vertical Sobel filter with border control.

- `NppStatus nppiFilterSobelVertBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed vertical Sobel filter with border control.

- `NppStatus nppiFilterSobelVertBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned vertical Sobel filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterSobelVertBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point vertical Sobel filter with border control.

- `NppStatus nppiFilterSobelVertBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point vertical Sobel filter with border control.

- `NppStatus nppiFilterSobelVertBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point vertical Sobel filter with border control.

- `NppStatus nppiFilterSobelVertBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point vertical Sobel filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterSobelVertBorder_8u16s_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned to 16-bit signed vertical Sobel filter with border control.

- `NppStatus nppiFilterSobelVertBorder_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 8-bit signed to 16-bit signed vertical Sobel filter with border control.

- `NppStatus nppiFilterSobelVertMaskBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point vertical Sobel filter with border control.

FilterSobelHorizSecondBorder

Filters the image using a second derivative, horizontal Sobel filter kernel with border control:

$$\begin{pmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

- **NppStatus nppiFilterSobelHorizSecondBorder_8u16s_C1R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Single channel 8-bit unsigned to 16-bit signed second derivative, horizontal Sobel filter with border control.

- **NppStatus nppiFilterSobelHorizSecondBorder_8s16s_C1R** (const **Npp8s** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Single channel 8-bit signed to 16-bit signed second derivative, horizontal Sobel filter with border control.

- **NppStatus nppiFilterSobelHorizSecondBorder_32f_C1R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

Single channel 32-bit floating-point second derivative, horizontal Sobel filter with border control.

7.66.1 Function Documentation

7.66.1.1 **NppStatus nppiFilterColumn32f_16s_AC4R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32f** *pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor)

16-bit four-channel 1D column convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.2 NppStatus nppiFilterColumn32f_16s_C1R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

16-bit single-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.3 NppStatus nppiFilterColumn32f_16s_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

16-bit three-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.4 NppStatus nppiFilterColumn32f_16s_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

16-bit four-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.5 NppStatus nppiFilterColumn32f_16u_AC4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

16-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.6 NppStatus nppiFilterColumn32f_16u_C1R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

16-bit unsigned single-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.7 NppStatus nppiFilterColumn32f_16u_C3R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

16-bit unsigned three-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.8 NppStatus nppiFilterColumn32f_16u_C4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

16-bit unsigned four-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.9 NppStatus nppiFilterColumn32f_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

8-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.10 `NppStatus nppiFilterColumn32f_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned single-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.11 `NppStatus nppiFilterColumn32f_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned three-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.12 `NppStatus nppiFilterColumn32f_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned four-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.13 `NppStatus nppiFilterColumn_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit four-channel 1D column convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.14 `NppStatus nppiFilterColumn_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit single-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.15 `NppStatus nppiFilterColumn_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit three-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.16 `NppStatus nppiFilterColumn_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit four-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.17 `NppStatus nppiFilterColumn_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.18 `NppStatus nppiFilterColumn_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned single-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.19 `NppStatus nppiFilterColumn_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned three-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.20 `NppStatus nppiFilterColumn_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned four-channel 1D column convolution.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.21 `NppStatus nppiFilterColumn_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

32-bit float four-channel 1D column convolution ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.22 NppStatus nppiFilterColumn_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

32-bit float single-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.23 NppStatus nppiFilterColumn_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

32-bit float three-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.24 NppStatus nppiFilterColumn_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

32-bit float four-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.25 NppStatus nppiFilterColumn_64f_C1R (const Npp64f * *pSrc*, Npp32s *nSrcStep*, Npp64f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp64f * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

64-bit float single-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.26 `NppStatus nppiFilterColumn_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.27 `NppStatus nppiFilterColumn_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned single-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.28 `NppStatus nppiFilterColumn_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned three-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.29 `NppStatus nppiFilterColumn_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned four-channel 1D column convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.30 `NppStatus nppiFilterColumnBorder32f_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 16-bit 1D column convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.31 `NppStatus nppiFilterColumnBorder32f_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Single channel 16-bit 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.32 `NppStatus nppiFilterColumnBorder32f_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 16-bit 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
nMaskSize Width of the kernel.
nAnchor X offset of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.33 `NppStatus nppiFilterColumnBorder32f_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 16-bit 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
nMaskSize Width of the kernel.
nAnchor X offset of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.34 `NppStatus nppiFilterColumnBorder32f_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned 1D column convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.35 `NppStatus nppiFilterColumnBorder32f_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Single channel 16-bit unsigned 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.36 `NppStatus nppiFilterColumnBorder32f_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 16-bit unsigned 1D column convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- nMaskSize* Width of the kernel.
- nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.37 `NppStatus nppiFilterColumnBorder32f_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned 1D column convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- nMaskSize* Width of the kernel.
- nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.38 `NppStatus nppiFilterColumnBorder32f_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned 1D column convolution filter with border control, ignore alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.39 `NppStatus nppiFilterColumnBorder32f_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.40 `NppStatus nppiFilterColumnBorder32f_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned 1D column convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- nMaskSize* Width of the kernel.
- nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.41 `NppStatus nppiFilterColumnBorder32f_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned 1D column convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- nMaskSize* Width of the kernel.
- nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.42 `NppStatus nppiFilterColumnBorder_16s_AC4R` (`const Npp16s * pSrc`, `Npp32s nSrcStep`, `NppiSize oSrcSize`, `NppiPoint oSrcOffset`, `Npp16s * pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `const Npp32s * pKernel`, `Npp32s nMaskSize`, `Npp32s nAnchor`, `Npp32s nDivisor`, `NppiBorderType eBorderType`)

Four channel 16-bit 1D column convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.43 `NppStatus nppiFilterColumnBorder_16s_C1R` (`const Npp16s * pSrc`, `Npp32s nSrcStep`, `NppiSize oSrcSize`, `NppiPoint oSrcOffset`, `Npp16s * pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `const Npp32s * pKernel`, `Npp32s nMaskSize`, `Npp32s nAnchor`, `Npp32s nDivisor`, `NppiBorderType eBorderType`)

Single channel 16-bit 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.44 `NppStatus nppiFilterColumnBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Three channel 16-bit 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.45 `NppStatus nppiFilterColumnBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel channel 16-bit 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.46 `NppStatus nppiFilterColumnBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned 1D column convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.47 `NppStatus nppiFilterColumnBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Single channel 16-bit unsigned convolution 1D column filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.48 `NppStatus nppiFilterColumnBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Three channel 16-bit unsigned 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.49 `NppStatus nppiFilterColumnBorder_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel channel 16-bit 1D column unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.50 `NppStatus nppiFilterColumnBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 32-bit float 1D column convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.51 `NppStatus nppiFilterColumnBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Single channel 32-bit float 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.52 `NppStatus nppiFilterColumnBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 32-bit float 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.53 `NppStatus nppiFilterColumnBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 32-bit float 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.54 `NppStatus nppiFilterColumnBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned convolution 1D column filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.55 `NppStatus nppiFilterColumnBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.56 `NppStatus nppiFilterColumnBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.57 `NppStatus nppiFilterColumnBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel channel 8-bit unsigned 1D column convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.58 `NppStatus nppiFilterRow32f_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit four-channel 1D row convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.59 `NppStatus nppiFilterRow32f_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit single-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.60 `NppStatus nppiFilterRow32f_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit three-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.61 `NppStatus nppiFilterRow32f_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit four-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.62 `NppStatus nppiFilterRow32f_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.63 `NppStatus nppiFilterRow32f_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit unsigned single-channel 1D row convolution.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.64 `NppStatus nppiFilterRow32f_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit unsigned three-channel 1D row convolution.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.65 `NppStatus nppiFilterRow32f_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit unsigned four-channel 1D row convolution.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.66 `NppStatus nppiFilterRow32f_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.67 `NppStatus nppiFilterRow32f_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned single-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.68 `NppStatus nppiFilterRow32f_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned three-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.69 `NppStatus nppiFilterRow32f_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned four-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.70 `NppStatus nppiFilterRow_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit four-channel 1D row convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.71 `NppStatus nppiFilterRow_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit single-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.72 `NppStatus nppiFilterRow_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit three-channel 1D row convolution.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.73 `NppStatus nppiFilterRow_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit four-channel 1D row convolution.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.74 `NppStatus nppiFilterRow_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.75 `NppStatus nppiFilterRow_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned single-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.76 `NppStatus nppiFilterRow_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned three-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.77 `NppStatus nppiFilterRow_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned four-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.78 `NppStatus nppiFilterRow_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

32-bit float four-channel 1D row convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.79 `NppStatus nppiFilterRow_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

32-bit float single-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.80 `NppStatus nppiFilterRow_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

32-bit float three-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.81 `NppStatus nppiFilterRow_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

32-bit float four-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.82 `NppStatus nppiFilterRow_64f_C1R (const Npp64f * pSrc, Npp32s nSrcStep, Npp64f * pDst, Npp32s nDstStep, NppiSize oROI, const Npp64f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

64-bit float single-channel 1D row convolution.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.83 `NppStatus nppiFilterRow_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.84 `NppStatus nppiFilterRow_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned single-channel 1D row convolution.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.85 `NppStatus nppiFilterRow_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned three-channel 1D row convolution.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.86 `NppStatus nppiFilterRow_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned four-channel 1D row convolution.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.87 `NppStatus nppiFilterRowBorder32f_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 16-bit 1D row convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.88 `NppStatus nppiFilterRowBorder32f_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Single channel 16-bit 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.89 `NppStatus nppiFilterRowBorder32f_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 16-bit 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.90 `NppStatus nppiFilterRowBorder32f_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 16-bit 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
nMaskSize Width of the kernel.
nAnchor X offset of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.91 `NppStatus nppiFilterRowBorder32f_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned 1D row convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
nMaskSize Width of the kernel.
nAnchor X offset of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.92 `NppStatus nppiFilterRowBorder32f_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Single channel 16-bit unsigned 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.93 `NppStatus nppiFilterRowBorder32f_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 16-bit unsigned 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.94 `NppStatus nppiFilterRowBorder32f_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
nMaskSize Width of the kernel.
nAnchor X offset of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.95 `NppStatus nppiFilterRowBorder32f_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned 1D row convolution filter with border control, ignorint alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
nMaskSize Width of the kernel.
nAnchor X offset of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.96 `NppStatus nppiFilterRowBorder32f_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.97 `NppStatus nppiFilterRowBorder32f_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.98 `NppStatus nppiFilterRowBorder32f_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.99 `NppStatus nppiFilterRowBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel 16-bit 1D row convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.100 `NppStatus nppiFilterRowBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Single channel 16-bit 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.101 `NppStatus nppiFilterRowBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Three channel 16-bit 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.102 `NppStatus nppiFilterRowBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel channel 16-bit 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.103 `NppStatus nppiFilterRowBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned 1D row convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.104 `NppStatus nppiFilterRowBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Single channel 16-bit unsigned convolution 1D row filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.105 **NppStatus nppiFilterRowBorder_16u_C3R** (const Npp16u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const Npp32s * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*, Npp32s *nDivisor*, NppiBorderType *eBorderType*)

Three channel 16-bit unsigned 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.106 **NppStatus nppiFilterRowBorder_16u_C4R** (const Npp16u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const Npp32s * *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*, Npp32s *nDivisor*, NppiBorderType *eBorderType*)

Four channel channel 16-bit 1D row unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.107 `NppStatus nppiFilterRowBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 32-bit float 1D row convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.108 `NppStatus nppiFilterRowBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Single channel 32-bit float 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.109 `NppStatus nppiFilterRowBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 32-bit float 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.110 `NppStatus nppiFilterRowBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 32-bit float 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.111 `NppStatus nppiFilterRowBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned convolution 1D row filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.112 `NppStatus nppiFilterRowBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.113 `NppStatus nppiFilterRowBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.114 `NppStatus nppiFilterRowBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel channel 8-bit unsigned 1D row convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

nMaskSize Width of the kernel.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.115 NppStatus nppiFilterSobelCross_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 32-bit floating-point second cross derivative Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.116 NppStatus nppiFilterSobelCross_8s16s_C1R (const Npp8s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit signed to 16-bit signed second cross derivative Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.117 NppStatus nppiFilterSobelCross_8u16s_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned to 16-bit signed second cross derivative Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.118 `NppStatus nppiFilterSobelHorizBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned horizontal Sobel filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.119 `NppStatus nppiFilterSobelHorizBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 16-bit signed horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.120 `NppStatus nppiFilterSobelHorizBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 16-bit signed horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.121 `NppStatus nppiFilterSobelHorizBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.122 `NppStatus nppiFilterSobelHorizBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point horizontal Sobel filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.123 `NppStatus nppiFilterSobelHorizBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 32-bit floating-point horizontal Sobel filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.124 `NppStatus nppiFilterSobelHorizBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 32-bit floating-point horizontal Sobel filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.125 **NppStatus nppiFilterSobelHorizBorder_32f_C4R** (**const Npp32f * pSrc**, **Npp32s nSrcStep**, **NppiSize oSrcSize**, **NppiPoint oSrcOffset**, **Npp32f * pDst**, **Npp32s nDstStep**, **NppiSize oSizeROI**, **NppiBorderType eBorderType**)

Four channel 32-bit floating-point horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.126 **NppStatus nppiFilterSobelHorizBorder_8s16s_C1R** (**const Npp8s * pSrc**, **Npp32s nSrcStep**, **NppiSize oSrcSize**, **NppiPoint oSrcOffset**, **Npp16s * pDst**, **Npp32s nDstStep**, **NppiSize oSizeROI**, **NppiMaskSize eMaskSize**, **NppiBorderType eBorderType**)

Single channel 8-bit signed to 16-bit signed horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.127 `NppStatus nppiFilterSobelHorizBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed horizontal Sobel filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.128 `NppStatus nppiFilterSobelHorizBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed horizontal Sobel filter with border control, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.129 `NppStatus nppiFilterSobelHorizBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.130 `NppStatus nppiFilterSobelHorizBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 8-bit unsigned horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.131 `NppStatus nppiFilterSobelHorizBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.132 `NppStatus nppiFilterSobelHorizMaskBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eMaskSize Enumeration value specifying the mask size.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.133 `NppStatus nppiFilterSobelHorizSecondBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point second derivative, horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eMaskSize Enumeration value specifying the mask size.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.134 `NppStatus nppiFilterSobelHorizSecondBorder_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit signed to 16-bit signed second derivative, horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eMaskSize Enumeration value specifying the mask size.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.135 `NppStatus nppiFilterSobelHorizSecondBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed second derivative, horizontal Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.136 `NppStatus nppiFilterSobelVertBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned vertical Sobel filter with border control, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.137 `NppStatus nppiFilterSobelVertBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 16-bit signed vertical Sobel filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.138 `NppStatus nppiFilterSobelVertBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 16-bit signed vertical Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.139 `NppStatus nppiFilterSobelVertBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed vertical Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.140 `NppStatus nppiFilterSobelVertBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point vertical Sobel filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.141 `NppStatus nppiFilterSobelVertBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 32-bit floating-point vertical Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.142 `NppStatus nppiFilterSobelVertBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 32-bit floating-point vertical Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.143 **NppStatus nppiFilterSobelVertBorder_32f_C4R** (const Npp32f * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiBorderType *eBorderType*)

Four channel 32-bit floating-point vertical Sobel filter with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to *pSrc*.
oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.144 **NppStatus nppiFilterSobelVertBorder_8s16s_C1R** (const Npp8s * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*, NppiBorderType *eBorderType*)

Single channel 8-bit signed to 16-bit signed vertical Sobel filter with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to *pSrc*.
oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.66.1.145 `NppStatus nppiFilterSobelVertBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed vertical Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.146 `NppStatus nppiFilterSobelVertBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed vertical Sobel filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.147 `NppStatus nppiFilterSobelVertBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned vertical Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.148 `NppStatus nppiFilterSobelVertBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 8-bit unsigned vertical Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.149 `NppStatus nppiFilterSobelVertBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned vertical Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.150 `NppStatus nppiFilterSobelVertMaskBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point vertical Sobel filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eMaskSize Enumeration value specifying the mask size.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.151 `NppStatus nppiFilterSobelVertSecond_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 32-bit floating-point second derivative, vertical Sobel filter.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.152 `NppStatus nppiFilterSobelVertSecond_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit signed to 16-bit signed second derivative, vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.66.1.153 `NppStatus nppiFilterSobelVertSecond_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit unsigned to 16-bit signed second derivative, vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67 1D Window Sum

1D Window Sum

1D mask Window Sum for 8 and 16 bit images.

- `NppStatus nppiSumWindowColumn_8u32f_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)
One channel 8-bit unsigned 1D (column) sum to 32f.
- `NppStatus nppiSumWindowColumn_8u32f_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)
Three channel 8-bit unsigned 1D (column) sum to 32f.
- `NppStatus nppiSumWindowColumn_8u32f_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)
Four channel 8-bit unsigned 1D (column) sum to 32f.
- `NppStatus nppiSumWindowColumn_16u32f_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)
One channel 16-bit unsigned 1D (column) sum to 32f.
- `NppStatus nppiSumWindowColumn_16u32f_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)
Three channel 16-bit unsigned 1D (column) sum to 32f.
- `NppStatus nppiSumWindowColumn_16u32f_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)
Four channel 16-bit unsigned 1D (column) sum to 32f.
- `NppStatus nppiSumWindowColumn_16s32f_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)
One channel 16-bit signed 1D (column) sum to 32f.
- `NppStatus nppiSumWindowColumn_16s32f_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)
Three channel 16-bit signed 1D (column) sum to 32f.
- `NppStatus nppiSumWindowColumn_16s32f_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)
Four channel 16-bit signed 1D (column) sum to 32f.
- `NppStatus nppiSumWindowRow_8u32f_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)
One channel 8-bit unsigned 1D (row) sum to 32f.
- `NppStatus nppiSumWindowRow_8u32f_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)
Three channel 8-bit unsigned 1D (row) sum to 32f.

- **NppStatus nppiSumWindowRow_8u32f_C4R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)
Four channel 8-bit unsigned 1D (row) sum to 32f.
- **NppStatus nppiSumWindowRow_16u32f_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)
One channel 16-bit unsigned 1D (row) sum to 32f.
- **NppStatus nppiSumWindowRow_16u32f_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)
Three channel 16-bit unsigned 1D (row) sum to 32f.
- **NppStatus nppiSumWindowRow_16u32f_C4R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)
Four channel 16-bit unsigned 1D (row) sum to 32f.
- **NppStatus nppiSumWindowRow_16s32f_C1R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)
One channel 16-bit signed 1D (row) sum to 32f.
- **NppStatus nppiSumWindowRow_16s32f_C3R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)
Three channel 16-bit signed 1D (row) sum to 32f.
- **NppStatus nppiSumWindowRow_16s32f_C4R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)
Four channel 16-bit signed 1D (row) sum to 32f.

7.67.1 Function Documentation

7.67.1.1 NppStatus nppiSumWindowColumn_16s32f_C1R (const Npp16s *pSrc, Npp32s nSrcStep, Npp32f *pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)

One channel 16-bit signed 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

- pSrc** Source-Image Pointer.
- nSrcStep** Source-Image Line Step.
- pDst** Destination-Image Pointer.
- nDstStep** Destination-Image Line Step.
- oROI** Region-of-Interest (ROI).
- nMaskSize** Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.2 NppStatus nppiSumWindowColumn_16s32f_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)

Three channel 16-bit signed 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.3 NppStatus nppiSumWindowColumn_16s32f_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)

Four channel 16-bit signed 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.4 NppStatus nppiSumWindowColumn_16u32f_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)

One channel 16-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.5 NppStatus nppiSumWindowColumn_16u32f_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)

Three channel 16-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 3-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.6 `NppStatus nppiSumWindowColumn_16u32f_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)`

Four channel 16-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.7 `NppStatus nppiSumWindowColumn_8u32f_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)`

One channel 8-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 8 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.8 NppStatus nppiSumWindowColumn_8u32f_C3R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

Three channel 8-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 3-channel 8 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by *nMaskSize* and *nAnchor*.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.9 NppStatus nppiSumWindowColumn_8u32f_C4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

Four channel 8-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 8 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by *nMaskSize* and *nAnchor*.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.10 NppStatus nppiSumWindowRow_16s32f_C1R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

One channel 16-bit signed 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.11 NppStatus nppiSumWindowRow_16s32f_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

Three channel 16-bit signed 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.12 NppStatus nppiSumWindowRow_16s32f_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

Four channel 16-bit signed 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.13 NppStatus nppiSumWindowRow_16u32f_C1R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

One channel 16-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.14 NppStatus nppiSumWindowRow_16u32f_C3R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

Three channel 16-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.15 NppStatus nppiSumWindowRow_16u32f_C4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

Four channel 16-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.16 NppStatus nppiSumWindowRow_8u32f_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

One channel 8-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 8-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.17 NppStatus nppiSumWindowRow_8u32f_C3R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

Three channel 8-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 8-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.67.1.18 `NppStatus nppiSumWindowRow_8u32f_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)`

Four channel 8-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 8-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by `iKernelDim` and `iAnchorX`.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68 1D Window Sum with Border Control

1D Window Sum Border

1D mask Window Sum for 8 and 16 bit images with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

- `NppStatus nppiSumWindowColumnBorder_8u32f_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

One channel 8-bit unsigned 1D (column) sum to 32f with border control.

- `NppStatus nppiSumWindowColumnBorder_8u32f_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned 1D (column) sum to 32f with border control.

- `NppStatus nppiSumWindowColumnBorder_8u32f_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned 1D (column) sum to 32f with border control.

- `NppStatus nppiSumWindowColumnBorder_16u32f_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

One channel 16-bit unsigned 1D (column) sum to 32f with border control.

- `NppStatus nppiSumWindowColumnBorder_16u32f_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit unsigned 1D (column) sum to 32f with border control.

- `NppStatus nppiSumWindowColumnBorder_16u32f_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned 1D (column) sum to 32f with border control.

- `NppStatus nppiSumWindowColumnBorder_16s32f_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

One channel 16-bit signed 1D (column) sum to 32f with border control.

- `NppStatus nppiSumWindowColumnBorder_16s32f_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit signed 1D (column) sum to 32f with border control.

- `NppStatus nppiSumWindowColumnBorder_16s32f_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit signed 1D (column) sum to 32f with border control.

- `NppStatus nppiSumWindowRowBorder_8u32f_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

One channel 8-bit unsigned 1D (row) sum to 32f with border control.

- `NppStatus nppiSumWindowRowBorder_8u32f_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned 1D (row) sum to 32f with border control.

- `NppStatus nppiSumWindowRowBorder_8u32f_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned 1D (row) sum to 32f with border control.

- `NppStatus nppiSumWindowRowBorder_16u32f_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

One channel 16-bit unsigned 1D (row) sum to 32f with border control.

- `NppStatus nppiSumWindowRowBorder_16u32f_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit unsigned 1D (row) sum to 32f with border control.

- `NppStatus nppiSumWindowRowBorder_16u32f_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned 1D (row) sum to 32f with border control.

- `NppStatus nppiSumWindowRowBorder_16s32f_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

One channel 16-bit signed 1D (row) sum to 32f with border control.

- `NppStatus nppiSumWindowRowBorder_16s32f_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit signed 1D (row) sum to 32f with border control.

- `NppStatus nppiSumWindowRowBorder_16s32f_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit signed 1D (row) sum to 32f with border control.

7.68.1 Function Documentation

7.68.1.1 `NppStatus nppiSumWindowColumnBorder_16s32f_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

One channel 16-bit signed 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oROI* [Region-of-Interest \(ROI\)](#).
- nMaskSize* Length of the linear kernel array.
- nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.2 `NppStatus nppiSumWindowColumnBorder_16s32f_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 16-bit signed 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oROI* [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.3 NppStatus nppiSumWindowColumnBorder_16s32f_C4R (const Npp16s *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)

Four channel 16-bit signed 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.4 NppStatus nppiSumWindowColumnBorder_16u32f_C1R (const Npp16u *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)

One channel 16-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.5 NppStatus npapiSumWindowColumnBorder_16u32f_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)

Three channel 16-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 3-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.6 NppStatus nppiSumWindowColumnBorder_16u32f_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)

Four channel 16-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.7 NppStatus nppiSumWindowColumnBorder_8u32f_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)

One channel 8-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 8 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.8 NppStatus nppiSumWindowColumnBorder_8u32f_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)

Three channel 8-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 3-channel 8 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.9 NppStatus nppiSumWindowColumnBorder_8u32f_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)

Four channel 8-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 8 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor Y offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.10 `NppStatus nppiSumWindowRowBorder_16s32f_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

One channel 16-bit signed 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.11 `NppStatus nppiSumWindowRowBorder_16s32f_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 16-bit signed 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oROI [Region-of-Interest \(ROI\)](#).
nMaskSize Length of the linear kernel array.
nAnchor X offset of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.12 `NppStatus nppiSumWindowRowBorder_16s32f_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 16-bit signed 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oROI [Region-of-Interest \(ROI\)](#).
nMaskSize Length of the linear kernel array.
nAnchor X offset of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.13 `NppStatus nppiSumWindowRowBorder_16u32f_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

One channel 16-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.14 `NppStatus nppiSumWindowRowBorder_16u32f_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 16-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.15 `NppStatus nppiSumWindowRowBorder_16u32f_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.16 `NppStatus nppiSumWindowRowBorder_8u32f_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

One channel 8-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 8-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oROI [Region-of-Interest \(ROI\)](#).
nMaskSize Length of the linear kernel array.
nAnchor X offset of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.17 `NppStatus nppiSumWindowRowBorder_8u32f_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 8-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oROI [Region-of-Interest \(ROI\)](#).
nMaskSize Length of the linear kernel array.
nAnchor X offset of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.68.1.18 `NppStatus nppiSumWindowRowBorder_8u32f_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 8-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nMaskSize Length of the linear kernel array.

nAnchor X offset of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69 Convolution

Filter

General purpose 2D convolution filter.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor.

- `NppStatus nppiFilter_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Single channel 8-bit unsigned convolution filter.

- `NppStatus nppiFilter_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Three channel 8-bit unsigned convolution filter.

- `NppStatus nppiFilter_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Four channel channel 8-bit unsigned convolution filter.

- `NppStatus nppiFilter_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Four channel 8-bit unsigned convolution filter, ignoring alpha channel.

- `NppStatus nppiFilter_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Single channel 16-bit unsigned convolution filter.

- `NppStatus nppiFilter_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Three channel 16-bit unsigned convolution filter.

- `NppStatus nppiFilter_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Four channel channel 16-bit unsigned convolution filter.

- `NppStatus nppiFilter_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Four channel 16-bit unsigned convolution filter, ignoring alpha channel.

- `NppStatus nppiFilter_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Single channel 16-bit convolution filter.

- `NppStatus nppiFilter_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Three channel 16-bit convolution filter.

- `NppStatus nppiFilter_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Four channel 16-bit convolution filter.

- `NppStatus nppiFilter_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

Four channel 16-bit convolution filter, ignoring alpha channel.

- `NppStatus nppiFilter_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Single channel 32-bit float convolution filter.

- `NppStatus nppiFilter_32f_C2R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Two channel 32-bit float convolution filter.

- `NppStatus nppiFilter_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Three channel 32-bit float convolution filter.

- `NppStatus nppiFilter_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Four channel 32-bit float convolution filter.

- `NppStatus nppiFilter_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Four channel 32-bit float convolution filter, ignoring alpha channel.

- `NppStatus nppiFilter_64f_C1R` (const `Npp64f` *pSrc, `Npp32s` nSrcStep, `Npp64f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp64f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Single channel 64-bit float convolution filter.

Filter32f

General purpose 2D convolution filter using floating-point weights.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed.

- `NppStatus nppiFilter32f_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Single channel 8-bit unsigned convolution filter.

- `NppStatus nppiFilter32f_8u_C2R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Two channel 8-bit unsigned convolution filter.

- `NppStatus nppiFilter32f_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Three channel 8-bit unsigned convolution filter.

- `NppStatus nppiFilter32f_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Four channel 8-bit unsigned convolution filter.

- `NppStatus nppiFilter32f_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Four channel 8-bit unsigned convolution filter, ignore alpha channel.

- `NppStatus nppiFilter32f_8s_C1R` (const `Npp8s` *pSrc, int nSrcStep, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Single channel 8-bit signed convolution filter.

- `NppStatus nppiFilter32f_8s_C2R` (const `Npp8s` *pSrc, int nSrcStep, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Two channel 8-bit signed convolution filter.

- `NppStatus nppiFilter32f_8s_C3R` (const `Npp8s` *pSrc, int nSrcStep, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Three channel 8-bit signed convolution filter.

- `NppStatus nppiFilter32f_8s_C4R` (const `Npp8s` *pSrc, int nSrcStep, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Four channel 8-bit signed convolution filter.

- `NppStatus nppiFilter32f_8s_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Four channel 8-bit signed convolution filter, ignoring alpha channel.

- `NppStatus nppiFilter32f_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Single channel 16-bit unsigned convolution filter.

- `NppStatus nppiFilter32f_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Three channel 16-bit unsigned convolution filter.

- `NppStatus nppiFilter32f_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Four channel 16-bit unsigned convolution filter.

- `NppStatus nppiFilter32f_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Four channel 16-bit unsigned convolution filter, ignoring alpha channel.
- `NppStatus nppiFilter32f_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Single channel 16-bit convolution filter.
- `NppStatus nppiFilter32f_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Three channel 16-bit convolution filter.
- `NppStatus nppiFilter32f_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Four channel 16-bit convolution filter.
- `NppStatus nppiFilter32f_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Four channel 16-bit convolution filter, ignoring alpha channel.
- `NppStatus nppiFilter32f_32s_C1R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Single channel 32-bit convolution filter.
- `NppStatus nppiFilter32f_32s_C3R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Three channel 32-bit convolution filter.
- `NppStatus nppiFilter32f_32s_C4R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Four channel 32-bit convolution filter.
- `NppStatus nppiFilter32f_32s_AC4R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Four channel 32-bit convolution filter, ignoring alpha channel.
- `NppStatus nppiFilter32f_8u16s_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Single channel 8-bit unsigned to 16-bit signed convolution filter.
- `NppStatus nppiFilter32f_8u16s_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Three channel 8-bit unsigned to 16-bit signed convolution filter.
- `NppStatus nppiFilter32f_8u16s_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Four channel 8-bit unsigned to 16-bit signed convolution filter.
- `NppStatus nppiFilter32f_8u16s_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)
Four channel 8-bit unsigned to 16-bit signed convolution filter, ignoring alpha channel.

- `NppStatus nppiFilter32f_8s16s_C1R` (const `Npp8s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Single channel 8-bit to 16-bit signed convolution filter.

- `NppStatus nppiFilter32f_8s16s_C3R` (const `Npp8s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Three channel 8-bit to 16-bit signed convolution filter.

- `NppStatus nppiFilter32f_8s16s_C4R` (const `Npp8s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Four channel 8-bit to 16-bit signed convolution filter.

- `NppStatus nppiFilter32f_8s16s_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

Four channel 8-bit to 16-bit signed convolution filter, ignoring alpha channel.

FilterBorder

General purpose 2D convolution filter with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

- `NppStatus nppiFilterBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel channel 8-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterBorder_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Single channel 16-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Three channel 16-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel channel 16-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Single channel 16-bit convolution filter with border control.

- `NppStatus nppiFilterBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Three channel 16-bit convolution filter with border control.

- `NppStatus nppiFilterBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel channel 16-bit convolution filter with border control.

- `NppStatus nppiFilterBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

Four channel 16-bit convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 32-bit float convolution filter with border control.

- `NppStatus nppiFilterBorder_32f_C2R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Two channel 32-bit float convolution filter with border control.

- `NppStatus nppiFilterBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 32-bit float convolution filter with border control.

- `NppStatus nppiFilterBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit float convolution filter with border control.

- `NppStatus nppiFilterBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit float convolution filter with border control, ignoring alpha channel.

FilterBorder32f

General purpose 2D convolution filter using floating-point weights with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

- `NppStatus nppiFilterBorder32f_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder32f_8u_C2R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Two channel 8-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder32f_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder32f_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder32f_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned convolution filter with border control, ignorint alpha channel.

- `NppStatus nppiFilterBorder32f_8s_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 8-bit signed convolution filter with border control.

- `NppStatus nppiFilterBorder32f_8s_C2R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Two channel 8-bit signed convolution filter with border control.

- `NppStatus nppiFilterBorder32f_8s_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 8-bit signed convolution filter with border control.

- `NppStatus nppiFilterBorder32f_8s_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit signed convolution filter with border control.

- `NppStatus nppiFilterBorder32f_8s_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit signed convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterBorder32f_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 16-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder32f_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder32f_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned convolution filter with border control.

- `NppStatus nppiFilterBorder32f_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterBorder32f_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 16-bit convolution filter with border control.

- `NppStatus nppiFilterBorder32f_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit convolution filter with border control.

- `NppStatus nppiFilterBorder32f_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit convolution filter with border control.

- `NppStatus nppiFilterBorder32f_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterBorder32f_32s_C1R` (const `Npp32s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 32-bit convolution filter with border control.

- `NppStatus nppiFilterBorder32f_32s_C3R` (const `Npp32s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 32-bit convolution filter with border control.

- `NppStatus nppiFilterBorder32f_32s_C4R` (const `Npp32s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit convolution filter with border control.

- `NppStatus nppiFilterBorder32f_32s_AC4R` (const `Npp32s` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit convolution filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterBorder32f_8u16s_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned to 16-bit signed convolution filter with border control.

- `NppStatus nppiFilterBorder32f_8u16s_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` *pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned to 16-bit signed convolution filter with border control.

- **NppStatus nppiFilterBorder32f_8u16s_C4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** *pKernel, **NppiSize** oKernelSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four channel 8-bit unsigned to 16-bit signed convolution filter with border control.

- **NppStatus nppiFilterBorder32f_8u16s_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** *pKernel, **NppiSize** oKernelSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four channel 8-bit unsigned to 16-bit signed convolution filter with border control, ignoring alpha channel.

- **NppStatus nppiFilterBorder32f_8s16s_C1R** (const **Npp8s** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** *pKernel, **NppiSize** oKernelSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Single channel 8-bit to 16-bit signed convolution filter with border control.

- **NppStatus nppiFilterBorder32f_8s16s_C3R** (const **Npp8s** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** *pKernel, **NppiSize** oKernelSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Three channel 8-bit to 16-bit signed convolution filter with border control.

- **NppStatus nppiFilterBorder32f_8s16s_C4R** (const **Npp8s** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** *pKernel, **NppiSize** oKernelSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four channel 8-bit to 16-bit signed convolution filter with border control.

- **NppStatus nppiFilterBorder32f_8s16s_AC4R** (const **Npp8s** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** *pKernel, **NppiSize** oKernelSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four channel 8-bit to 16-bit signed convolution filter with border control, ignoring alpha channel.

7.69.1 Function Documentation

7.69.1.1 **NppStatus nppiFilter32f_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** *pKernel, **NppiSize** oKernelSize, **NppiPoint** oAnchor)

Four channel 16-bit convolution filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.2 `NppStatus nppiFilter32f_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 16-bit convolution filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.3 `NppStatus nppiFilter32f_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Three channel 16-bit convolution filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.4 NppStatus nppiFilter32f_16s_C4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Four channel 16-bit convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.5 NppStatus nppiFilter32f_16u_AC4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Four channel 16-bit unsigned convolution filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.6 NppStatus nppiFilter32f_16u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Single channel 16-bit unsigned convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.69.1.7 NppStatus nppiFilter32f_16u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Three channel 16-bit unsigned convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.69.1.8 NppStatus nppiFilter32f_16u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Four channel 16-bit unsigned convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.9 NppStatus nppiFilter32f_32s_AC4R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Four channel 32-bit convolution filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.10 `NppStatus nppiFilter32f_32s_C1R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 32-bit convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.69.1.11 `NppStatus nppiFilter32f_32s_C3R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Three channel 32-bit convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.69.1.12 `NppStatus nppiFilter32f_32s_C4R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 32-bit convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.13 `NppStatus nppiFilter32f_8s16s_AC4R (const Npp8s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit to 16-bit signed convolution filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.14 NppStatus nppiFilter32f_8s16s_C1R (const Npp8s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Single channel 8-bit to 16-bit signed convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.69.1.15 NppStatus nppiFilter32f_8s16s_C3R (const Npp8s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Three channel 8-bit to 16-bit signed convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.69.1.16 `NppStatus nppiFilter32f_8s16s_C4R (const Npp8s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit to 16-bit signed convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.17 `NppStatus nppiFilter32f_8s_AC4R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit signed convolution filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.18 `NppStatus nppiFilter32f_8s_C1R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 8-bit signed convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.19 `NppStatus nppiFilter32f_8s_C2R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Two channel 8-bit signed convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.20 `NppStatus nppiFilter32f_8s_C3R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Three channel 8-bit signed convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.21 `NppStatus nppiFilter32f_8s_C4R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit signed convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.22 NppStatus nppiFilter32f_8u16s_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Four channel 8-bit unsigned to 16-bit signed convolution filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.23 NppStatus nppiFilter32f_8u16s_C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Single channel 8-bit unsigned to 16-bit signed convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.24 NppStatus nppiFilter32f_8u16s_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Three channel 8-bit unsigned to 16-bit signed convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.25 NppStatus nppiFilter32f_8u16s_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Four channel 8-bit unsigned to 16-bit signed convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.26 `NppStatus nppiFilter32f_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned convolution filter, ignorint alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.27 `NppStatus nppiFilter32f_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 8-bit unsigned convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.28 `NppStatus nppiFilter32f_8u_C2R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Two channel 8-bit unsigned convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.29 `NppStatus nppiFilter32f_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Three channel 8-bit unsigned convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.30 `NppStatus nppiFilter32f_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.31 `NppStatus nppiFilter_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel 16-bit convolution filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.32 `NppStatus nppiFilter_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Single channel 16-bit convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.33 `NppStatus nppiFilter_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Three channel 16-bit convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.34 `NppStatus nppiFilter_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel channel 16-bit convolution filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.35 `NppStatus nppiFilter_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel 16-bit unsigned convolution filter, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.36 `NppStatus nppiFilter_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Single channel 16-bit unsigned convolution filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.37 `NppStatus nppiFilter_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Three channel 16-bit unsigned convolution filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.38 `NppStatus nppiFilter_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel channel 16-bit unsigned convolution filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.39 `NppStatus nppiFilter_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 32-bit float convolution filter, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.40 `NppStatus nppiFilter_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 32-bit float convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.41 `NppStatus nppiFilter_32f_C2R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Two channel 32-bit float convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.42 NppStatus nppiFilter_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Three channel 32-bit float convolution filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.43 NppStatus nppiFilter_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const Npp32f * *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Four channel 32-bit float convolution filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.44 `NppStatus nppiFilter_64f_C1R (const Npp64f * pSrc, Npp32s nSrcStep, Npp64f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp64f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 64-bit float convolution filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.45 `NppStatus nppiFilter_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel 8-bit unsigned convolution filter, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.46 `NppStatus nppiFilter_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Single channel 8-bit unsigned convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.47 `NppStatus nppiFilter_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Three channel 8-bit unsigned convolution filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.48 `NppStatus nppiFilter_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel channel 8-bit unsigned convolution filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.49 `NppStatus nppiFilterBorder32f_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.50 `NppStatus nppiFilterBorder32f_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 16-bit convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.51 `NppStatus nppiFilterBorder32f_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 16-bit convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.52 `NppStatus nppiFilterBorder32f_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
oKernelSize Width and Height of the rectangular kernel.
oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.53 `NppStatus nppiFilterBorder32f_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
oKernelSize Width and Height of the rectangular kernel.
oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.54 `NppStatus nppiFilterBorder32f_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 16-bit unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.55 `NppStatus nppiFilterBorder32f_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 16-bit unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.56 `NppStatus nppiFilterBorder32f_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.57 `NppStatus nppiFilterBorder32f_32s_AC4R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.58 `NppStatus nppiFilterBorder32f_32s_C1R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 32-bit convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.59 `NppStatus nppiFilterBorder32f_32s_C3R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 32-bit convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.60 `NppStatus nppiFilterBorder32f_32s_C4R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
oKernelSize Width and Height of the rectangular kernel.
oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.61 `NppStatus nppiFilterBorder32f_8s16s_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit to 16-bit signed convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
oKernelSize Width and Height of the rectangular kernel.
oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.62 `NppStatus nppiFilterBorder32f_8s16s_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 8-bit to 16-bit signed convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.63 `NppStatus nppiFilterBorder32f_8s16s_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 8-bit to 16-bit signed convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.64 `NppStatus nppiFilterBorder32f_8s16s_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit to 16-bit signed convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.65 `NppStatus nppiFilterBorder32f_8s_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit signed convolution filter with border control, ignoring alpha channel.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.66 `NppStatus nppiFilterBorder32f_8s_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 8-bit signed convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.67 `NppStatus nppiFilterBorder32f_8s_C2R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Two channel 8-bit signed convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.68 `NppStatus nppiFilterBorder32f_8s_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 8-bit signed convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.69 `NppStatus nppiFilterBorder32f_8s_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit signed convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.70 `NppStatus nppiFilterBorder32f_8u16s_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned to 16-bit signed convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.71 `NppStatus nppiFilterBorder32f_8u16s_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.72 `NppStatus nppiFilterBorder32f_8u16s_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned to 16-bit signed convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.73 `NppStatus nppiFilterBorder32f_8u16s_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned to 16-bit signed convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.74 `NppStatus nppiFilterBorder32f_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned convolution filter with border control, ignorint alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.75 `NppStatus nppiFilterBorder32f_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.76 `NppStatus nppiFilterBorder32f_8u_C2R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Two channel 8-bit unsigned convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.77 `NppStatus nppiFilterBorder32f_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.78 `NppStatus nppiFilterBorder32f_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.79 `NppStatus nppiFilterBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel 16-bit convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.80 `NppStatus nppiFilterBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Single channel 16-bit convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.81 `NppStatus nppiFilterBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Three channel 16-bit convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.82 `NppStatus nppiFilterBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel channel 16-bit convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.83 `NppStatus nppiFilterBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.84 `NppStatus nppiFilterBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Single channel 16-bit unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.85 `NppStatus nppiFilterBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Three channel 16-bit unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.86 `NppStatus nppiFilterBorder_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel channel 16-bit unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.87 `NppStatus nppiFilterBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit float convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.88 `NppStatus nppiFilterBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 32-bit float convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.89 `NppStatus nppiFilterBorder_32f_C2R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Two channel 32-bit float convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.90 `NppStatus nppiFilterBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 32-bit float convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.91 `NppStatus nppiFilterBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit float convolution filter with border control.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.92 `NppStatus nppiFilterBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned convolution filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.93 `NppStatus nppiFilterBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.94 `NppStatus nppiFilterBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided.
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.69.1.95 `NppStatus nppiFilterBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel channel 8-bit unsigned convolution filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pKernel Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

oKernelSize Width and Height of the rectangular kernel.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

nDivisor The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70 2D Fixed Linear Filters

FilterBox

Computes the average pixel values of the pixels under a rectangular mask.

- `NppStatus nppiFilterBox_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Single channel 8-bit unsigned box filter.

- `NppStatus nppiFilterBox_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Three channel 8-bit unsigned box filter.

- `NppStatus nppiFilterBox_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 8-bit unsigned box filter.

- `NppStatus nppiFilterBox_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 8-bit unsigned box filter, ignoring alpha channel.

- `NppStatus nppiFilterBox_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Single channel 16-bit unsigned box filter.

- `NppStatus nppiFilterBox_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Three channel 16-bit unsigned box filter.

- `NppStatus nppiFilterBox_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 16-bit unsigned box filter.

- `NppStatus nppiFilterBox_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 16-bit unsigned box filter, ignoring alpha channel.

- `NppStatus nppiFilterBox_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Single channel 16-bit box filter.

- `NppStatus nppiFilterBox_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Three channel 16-bit box filter.

- `NppStatus nppiFilterBox_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 16-bit box filter.

- `NppStatus nppiFilterBox_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 16-bit box filter, ignoring alpha channel.
- `NppStatus nppiFilterBox_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Single channel 32-bit floating-point box filter.
- `NppStatus nppiFilterBox_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Three channel 32-bit floating-point box filter.
- `NppStatus nppiFilterBox_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 32-bit floating-point box filter.
- `NppStatus nppiFilterBox_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 32-bit floating-point box filter, ignoring alpha channel.
- `NppStatus nppiFilterBox_64f_C1R` (const `Npp64f` *pSrc, `Npp32s` nSrcStep, `Npp64f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Single channel 64-bit floating-point box filter.

FilterBoxBorder

Computes the average pixel values of the pixels under a rectangular mask with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the `NPP_BORDER_REPLICATE` border type operation is supported. *

- `NppStatus nppiFilterBoxBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)
Single channel 8-bit unsigned box filter with border control.
- `NppStatus nppiFilterBoxBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)
Three channel 8-bit unsigned box filter with border control.
- `NppStatus nppiFilterBoxBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)
Four channel 8-bit unsigned box filter with border control.
- `NppStatus nppiFilterBoxBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned box filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterBoxBorder_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 16-bit unsigned box filter with border control.

- `NppStatus nppiFilterBoxBorder_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit unsigned box filter with border control.

- `NppStatus nppiFilterBoxBorder_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned box filter with border control.

- `NppStatus nppiFilterBoxBorder_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned box filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterBoxBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 16-bit box filter with border control.

- `NppStatus nppiFilterBoxBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit box filter with border control.

- `NppStatus nppiFilterBoxBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit box filter with border control.

- `NppStatus nppiFilterBoxBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit box filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterBoxBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point box filter with border control.

- `NppStatus nppiFilterBoxBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point box filter with border control.

- **NppStatus nppiFilterBoxBorder_32f_C4R** (const [Npp32f](#) *pSrc, [Npp32s](#) nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp32f](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [NppiPoint](#) oAnchor, [NppiBorderType](#) eBorderType)

Four channel 32-bit floating-point box filter with border control.

- **NppStatus nppiFilterBoxBorder_32f_AC4R** (const [Npp32f](#) *pSrc, [Npp32s](#) nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp32f](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [NppiPoint](#) oAnchor, [NppiBorderType](#) eBorderType)

Four channel 32-bit floating-point box filter with border control, ignoring alpha channel.

7.70.1 Function Documentation

7.70.1.1 **NppStatus nppiFilterBox_16s_AC4R** (const [Npp16s](#) *pSrc, [Npp32s](#) nSrcStep, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [NppiPoint](#) oAnchor)

Four channel 16-bit box filter, ignoring alpha channel.

Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[pDst](#) [Destination-Image Pointer](#).

[nDstStep](#) [Destination-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[oMaskSize](#) Width and Height of the neighborhood region for the local Avg operation.

[oAnchor](#) X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.2 **NppStatus nppiFilterBox_16s_C1R** (const [Npp16s](#) *pSrc, [Npp32s](#) nSrcStep, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [NppiPoint](#) oAnchor)

Single channel 16-bit box filter.

Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[pDst](#) [Destination-Image Pointer](#).

[nDstStep](#) [Destination-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[oMaskSize](#) Width and Height of the neighborhood region for the local Avg operation.

[oAnchor](#) X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.3 NppStatus nppiFilterBox_16s_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Three channel 16-bit box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.4 NppStatus nppiFilterBox_16s_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 16-bit box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.5 NppStatus nppiFilterBox_16u_AC4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 16-bit unsigned box filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.70.1.6 `NppStatus nppiFilterBox_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 16-bit unsigned box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.70.1.7 `NppStatus nppiFilterBox_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 16-bit unsigned box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.70.1.8 NppStatus nppiFilterBox_16u_C4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 16-bit unsigned box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.9 NppStatus nppiFilterBox_32f_AC4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 32-bit floating-point box filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.10 NppStatus nppiFilterBox_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Single channel 32-bit floating-point box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.11 `NppStatus nppiFilterBox_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 32-bit floating-point box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.12 `NppStatus nppiFilterBox_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 32-bit floating-point box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.13 NppStatus nppiFilterBox_64f_C1R (const Npp64f * *pSrc*, Npp32s *nSrcStep*, Npp64f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Single channel 64-bit floating-point box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.14 NppStatus nppiFilterBox_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 8-bit unsigned box filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.15 NppStatus nppiFilterBox_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Single channel 8-bit unsigned box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.70.1.16 `NppStatus nppiFilterBox_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 8-bit unsigned box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.70.1.17 `NppStatus nppiFilterBox_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned box filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.70.1.18 NppStatus nppiFilterBoxBorder_16s_AC4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Four channel 16-bit box filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.19 NppStatus nppiFilterBoxBorder_16s_C1R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Single channel 16-bit box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.20 `NppStatus nppiFilterBoxBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 16-bit box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.21 `NppStatus nppiFilterBoxBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.22 NppStatus nppiFilterBoxBorder_16u_AC4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Four channel 16-bit unsigned box filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.23 NppStatus nppiFilterBoxBorder_16u_C1R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Single channel 16-bit unsigned box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.24 NppStatus nppiFilterBoxBorder_16u_C3R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Three channel 16-bit unsigned box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.25 NppStatus nppiFilterBoxBorder_16u_C4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Four channel 16-bit unsigned box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.26 `NppStatus nppiFilterBoxBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit floating-point box filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.27 `NppStatus nppiFilterBoxBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 32-bit floating-point box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.28 `NppStatus nppiFilterBoxBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 32-bit floating-point box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.29 `NppStatus nppiFilterBoxBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit floating-point box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.30 `NppStatus nppiFilterBoxBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned box filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.31 `NppStatus nppiFilterBoxBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.32 `NppStatus nppiFilterBoxBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.70.1.33 `NppStatus nppiFilterBoxBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned box filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Avg operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71 Rank Filters

ImageMax Filter

Result pixel value is the maximum of pixel values under the rectangular mask region.

- `NppStatus nppiFilterMax_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Single channel 8-bit unsigned maximum filter.
- `NppStatus nppiFilterMax_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Three channel 8-bit unsigned maximum filter.
- `NppStatus nppiFilterMax_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 8-bit unsigned maximum filter.
- `NppStatus nppiFilterMax_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 8-bit unsigned maximum filter, ignoring alpha channel.
- `NppStatus nppiFilterMax_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Single channel 16-bit unsigned maximum filter.
- `NppStatus nppiFilterMax_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Three channel 16-bit unsigned maximum filter.
- `NppStatus nppiFilterMax_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 16-bit unsigned maximum filter.
- `NppStatus nppiFilterMax_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 16-bit unsigned maximum filter, ignoring alpha channel.
- `NppStatus nppiFilterMax_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Single channel 16-bit signed maximum filter.
- `NppStatus nppiFilterMax_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Three channel 16-bit signed maximum filter.
- `NppStatus nppiFilterMax_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 16-bit signed maximum filter.

- `NppStatus nppiFilterMax_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 16-bit signed maximum filter, ignoring alpha channel.

- `NppStatus nppiFilterMax_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Single channel 32-bit floating-point maximum filter.

- `NppStatus nppiFilterMax_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Three channel 32-bit floating-point maximum filter.

- `NppStatus nppiFilterMax_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 32-bit floating-point maximum filter.

- `NppStatus nppiFilterMax_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 32-bit floating-point maximum filter, ignoring alpha channel.

ImageMaxBorder Filter

Result pixel value is the maximum of pixel values under the rectangular mask region.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the `NPP_BORDER_REPLICATE` border type operation is supported.

- `NppStatus nppiFilterMaxBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned maximum filter with border control.

- `NppStatus nppiFilterMaxBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned maximum filter with border control.

- `NppStatus nppiFilterMaxBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned maximum filter with border control.

- `NppStatus nppiFilterMaxBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned maximum filter with border control, ignoring alpha channel.

- **NppStatus** **nppiFilterMaxBorder_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Single channel 16-bit unsigned maximum filter with border control.

- **NppStatus** **nppiFilterMaxBorder_16u_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Three channel 16-bit unsigned maximum filter with border control.

- **NppStatus** **nppiFilterMaxBorder_16u_C4R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four channel 16-bit unsigned maximum filter with border control.

- **NppStatus** **nppiFilterMaxBorder_16u_AC4R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four channel 16-bit unsigned maximum filter with border control, ignoring alpha channel.

- **NppStatus** **nppiFilterMaxBorder_16s_C1R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Single channel 16-bit signed maximum filter with border control.

- **NppStatus** **nppiFilterMaxBorder_16s_C3R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Three channel 16-bit signed maximum filter with border control.

- **NppStatus** **nppiFilterMaxBorder_16s_C4R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four channel 16-bit signed maximum filter with border control.

- **NppStatus** **nppiFilterMaxBorder_16s_AC4R** (const **Npp16s** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four channel 16-bit signed maximum filter with border control, ignoring alpha channel.

- **NppStatus** **nppiFilterMaxBorder_32f_C1R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Single channel 32-bit floating-point maximum filter with border control.

- **NppStatus** **nppiFilterMaxBorder_32f_C3R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Three channel 32-bit floating-point maximum filter with border control.

- `NppStatus nppiFilterMaxBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point maximum filter with border control.

- `NppStatus nppiFilterMaxBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point maximum filter with border control, ignoring alpha channel.

ImageMin Filter

Result pixel value is the minimum of pixel values under the rectangular mask region.

- `NppStatus nppiFilterMin_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Single channel 8-bit unsigned minimum filter.

- `NppStatus nppiFilterMin_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Three channel 8-bit unsigned minimum filter.

- `NppStatus nppiFilterMin_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 8-bit unsigned minimum filter.

- `NppStatus nppiFilterMin_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 8-bit unsigned minimum filter, ignoring alpha channel.

- `NppStatus nppiFilterMin_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Single channel 16-bit unsigned minimum filter.

- `NppStatus nppiFilterMin_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Three channel 16-bit unsigned minimum filter.

- `NppStatus nppiFilterMin_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 16-bit unsigned minimum filter.

- `NppStatus nppiFilterMin_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 16-bit unsigned minimum filter, ignoring alpha channel.

- `NppStatus nppiFilterMin_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Single channel 16-bit signed minimum filter.

- `NppStatus nppiFilterMin_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Three channel 16-bit signed minimum filter.
- `NppStatus nppiFilterMin_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 16-bit signed minimum filter.
- `NppStatus nppiFilterMin_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 16-bit signed minimum filter, ignoring alpha channel.
- `NppStatus nppiFilterMin_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Single channel 32-bit floating-point minimum filter.
- `NppStatus nppiFilterMin_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Three channel 32-bit floating-point minimum filter.
- `NppStatus nppiFilterMin_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 32-bit floating-point minimum filter.
- `NppStatus nppiFilterMin_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)
Four channel 32-bit floating-point minimum filter, ignoring alpha channel.

ImageMinBorder Filter

Result pixel value is the minimum of pixel values under the rectangular mask region.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the `NPP_BORDER_REPLICATE` border type operation is supported.

- `NppStatus nppiFilterMinBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)
Single channel 8-bit unsigned minimum filter with border control.
- `NppStatus nppiFilterMinBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)
Three channel 8-bit unsigned minimum filter with border control.
- `NppStatus nppiFilterMinBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned minimum filter with border control.

- `NppStatus nppiFilterMinBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned minimum filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterMinBorder_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 16-bit unsigned minimum filter with border control.

- `NppStatus nppiFilterMinBorder_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit unsigned minimum filter with border control.

- `NppStatus nppiFilterMinBorder_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned minimum filter with border control.

- `NppStatus nppiFilterMinBorder_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit unsigned minimum filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterMinBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 16-bit signed minimum filter with border control.

- `NppStatus nppiFilterMinBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 16-bit signed minimum filter with border control.

- `NppStatus nppiFilterMinBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit signed minimum filter with border control.

- `NppStatus nppiFilterMinBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 16-bit signed minimum filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterMinBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point minimum filter with border control.

- `NppStatus nppiFilterMinBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point minimum filter with border control.

- `NppStatus nppiFilterMinBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point minimum filter with border control.

- `NppStatus nppiFilterMinBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point minimum filter with border control, ignoring alpha channel.

ImageMedian Filter

Result pixel value is the median of pixel values under the rectangular mask region.

- `NppStatus nppiFilterMedian_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Single channel 8-bit unsigned median filter.

- `NppStatus nppiFilterMedian_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Three channel 8-bit unsigned median filter.

- `NppStatus nppiFilterMedian_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Four channel 8-bit unsigned median filter.

- `NppStatus nppiFilterMedian_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Four channel 8-bit unsigned median filter, ignoring alpha channel.

- `NppStatus nppiFilterMedian_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Single channel 16-bit unsigned median filter.

- `NppStatus nppiFilterMedian_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Three channel 16-bit unsigned median filter.

- `NppStatus nppiFilterMedian_16u_C4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Four channel 16-bit unsigned median filter.

- `NppStatus nppiFilterMedian_16u_AC4R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Four channel 16-bit unsigned median filter, ignoring alpha channel.

- `NppStatus nppiFilterMedian_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Single channel 16-bit signed median filter.

- `NppStatus nppiFilterMedian_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Three channel 16-bit signed median filter.

- `NppStatus nppiFilterMedian_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Four channel 16-bit signed median filter.

- `NppStatus nppiFilterMedian_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Four channel 16-bit signed median filter, ignoring alpha channel.

- `NppStatus nppiFilterMedian_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Single channel 32-bit floating-point median filter.

- `NppStatus nppiFilterMedian_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Three channel 32-bit floating-point median filter.

- `NppStatus nppiFilterMedian_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Four channel 32-bit floating-point median filter.

- `NppStatus nppiFilterMedian_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` *pBuffer)

Four channel 32-bit floating-point median filter, ignoring alpha channel.

- `NppStatus nppiFilterMedianGetBufferSize_8u_C1R` (`NppiSize` oSizeROI, `NppiSize` oMaskSize, `Npp32u` *nBufferSize)

Single channel 8-bit unsigned median filter scratch memory size.

- [NppStatus nppiFilterMedianGetBufferSize_8u_C3R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Three channel 8-bit unsigned median filter scratch memory size.
- [NppStatus nppiFilterMedianGetBufferSize_8u_C4R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Four channel 8-bit unsigned median filter scratch memory size.
- [NppStatus nppiFilterMedianGetBufferSize_8u_AC4R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Four channel 8-bit unsigned median filter, ignoring alpha channel.
- [NppStatus nppiFilterMedianGetBufferSize_16u_C1R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Single channel 16-bit unsigned median filter scratch memory size.
- [NppStatus nppiFilterMedianGetBufferSize_16u_C3R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Three channel 16-bit unsigned median filter scratch memory size.
- [NppStatus nppiFilterMedianGetBufferSize_16u_C4R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Four channel 16-bit unsigned median filter scratch memory size.
- [NppStatus nppiFilterMedianGetBufferSize_16u_AC4R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Four channel 16-bit unsigned median filter, ignoring alpha channel.
- [NppStatus nppiFilterMedianGetBufferSize_16s_C1R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Single channel 16-bit signed median filter scratch memory size.
- [NppStatus nppiFilterMedianGetBufferSize_16s_C3R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Three channel 16-bit signed median filter scratch memory size.
- [NppStatus nppiFilterMedianGetBufferSize_16s_C4R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Four channel 16-bit signed median filter scratch memory size.
- [NppStatus nppiFilterMedianGetBufferSize_16s_AC4R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Four channel 16-bit signed median filter, ignoring alpha channel.
- [NppStatus nppiFilterMedianGetBufferSize_32f_C1R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Single channel 32-bit floating-point median filter scratch memory size.
- [NppStatus nppiFilterMedianGetBufferSize_32f_C3R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)
Three channel 32-bit floating-point median filter scratch memory size.

- [NppStatus nppiFilterMedianGetBufferSize_32f_C4R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)

Four channel 32-bit floating-point median filter scratch memory size.

- [NppStatus nppiFilterMedianGetBufferSize_32f_AC4R](#) ([NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [Npp32u](#) *nBufferSize)

Four channel 32-bit floating-point median filter, ignoring alpha channel.

7.71.1 Function Documentation

7.71.1.1 [NppStatus nppiFilterMax_16s_AC4R](#) (const [Npp16s](#) *pSrc, [Npp32s](#) nSrcStep, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [NppiPoint](#) oAnchor)

Four channel 16-bit signed maximum filter, ignoring alpha channel.

Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[pDst](#) [Destination-Image Pointer](#).

[nDstStep](#) [Destination-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[oMaskSize](#) Width and Height of the neighborhood region for the local Max operation.

[oAnchor](#) X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.2 [NppStatus nppiFilterMax_16s_C1R](#) (const [Npp16s](#) *pSrc, [Npp32s](#) nSrcStep, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiSize](#) oMaskSize, [NppiPoint](#) oAnchor)

Single channel 16-bit signed maximum filter.

Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[pDst](#) [Destination-Image Pointer](#).

[nDstStep](#) [Destination-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[oMaskSize](#) Width and Height of the neighborhood region for the local Max operation.

[oAnchor](#) X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.3 NppStatus nppiFilterMax_16s_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Three channel 16-bit signed maximum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.4 NppStatus nppiFilterMax_16s_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 16-bit signed maximum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.5 NppStatus nppiFilterMax_16u_AC4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 16-bit unsigned maximum filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.71.1.6 NppStatus nppiFilterMax_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)

Single channel 16-bit unsigned maximum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.71.1.7 NppStatus nppiFilterMax_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)

Three channel 16-bit unsigned maximum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.71.1.8 **NppStatus nppiFilterMax_16u_C4R** (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 16-bit unsigned maximum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.9 **NppStatus nppiFilterMax_32f_AC4R** (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 32-bit floating-point maximum filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.10 **NppStatus nppiFilterMax_32f_C1R** (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Single channel 32-bit floating-point maximum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.71.1.11 `NppStatus nppiFilterMax_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 32-bit floating-point maximum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.71.1.12 `NppStatus nppiFilterMax_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 32-bit floating-point maximum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.71.1.13 `NppStatus nppiFilterMax_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned maximum filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.14 `NppStatus nppiFilterMax_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 8-bit unsigned maximum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.15 `NppStatus nppiFilterMax_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 8-bit unsigned maximum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.16 `NppStatus nppiFilterMax_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned maximum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.17 `NppStatus nppiFilterMaxBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit signed maximum filter with border control, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.18 `NppStatus nppiFilterMaxBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 16-bit signed maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.19 `NppStatus nppiFilterMaxBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 16-bit signed maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.20 NppStatus nppiFilterMaxBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Four channel 16-bit signed maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.21 NppStatus nppiFilterMaxBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Four channel 16-bit unsigned maximum filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.22 NppStatus nppiFilterMaxBorder_16u_C1R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Single channel 16-bit unsigned maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.23 NppStatus nppiFilterMaxBorder_16u_C3R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Three channel 16-bit unsigned maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.24 NppStatus nppiFilterMaxBorder_16u_C4R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Four channel 16-bit unsigned maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.25 NppStatus nppiFilterMaxBorder_32f_AC4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Four channel 32-bit floating-point maximum filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.26 NppStatus nppiFilterMaxBorder_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Single channel 32-bit floating-point maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.27 NppStatus nppiFilterMaxBorder_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Three channel 32-bit floating-point maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.28 NppStatus nppiFilterMaxBorder_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Four channel 32-bit floating-point maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.29 NppStatus nppiFilterMaxBorder_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*, NppiBorderType *eBorderType*)

Four channel 8-bit unsigned maximum filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.30 `NppStatus nppiFilterMaxBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.31 `NppStatus nppiFilterMaxBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.32 `NppStatus nppiFilterMaxBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned maximum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.33 `NppStatus nppiFilterMedian_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 16-bit signed median filter, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.34 `NppStatus nppiFilterMedian_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Single channel 16-bit signed median filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.35 `NppStatus nppiFilterMedian_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Three channel 16-bit signed median filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.36 `NppStatus nppiFilterMedian_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 16-bit signed median filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
oMaskSize Width and Height of the neighborhood region for the local Median operation.
oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.
pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.37 `NppStatus nppiFilterMedian_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 16-bit unsigned median filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
oMaskSize Width and Height of the neighborhood region for the local Median operation.
oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.
pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.38 `NppStatus nppiFilterMedian_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Single channel 16-bit unsigned median filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.39 `NppStatus nppiFilterMedian_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Three channel 16-bit unsigned median filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.40 `NppStatus nppiFilterMedian_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 16-bit unsigned median filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.41 `NppStatus nppiFilterMedian_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 32-bit floating-point median filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.42 `NppStatus nppiFilterMedian_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Single channel 32-bit floating-point median filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.43 `NppStatus nppiFilterMedian_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Three channel 32-bit floating-point median filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.44 `NppStatus nppiFilterMedian_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 32-bit floating-point median filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.45 `NppStatus nppiFilterMedian_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 8-bit unsigned median filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.46 `NppStatus nppiFilterMedian_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Single channel 8-bit unsigned median filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.47 `NppStatus nppiFilterMedian_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Three channel 8-bit unsigned median filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.48 `NppStatus nppiFilterMedian_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 8-bit unsigned median filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

pBuffer Pointer to the user-allocated scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.49 `NppStatus nppiFilterMedianGetBufferSize_16s_AC4R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u * nBufferSize)`

Four channel 16-bit signed median filter, ignoring alpha channel.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.50 `NppStatus nppiFilterMedianGetBufferSize_16s_C1R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u * nBufferSize)`

Single channel 16-bit signed median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.51 NppStatus nppiFilterMedianGetBufferSize_16s_C3R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u * *nBufferSize*)

Three channel 16-bit signed median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.52 NppStatus nppiFilterMedianGetBufferSize_16s_C4R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u * *nBufferSize*)

Four channel 16-bit signed median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.53 NppStatus nppiFilterMedianGetBufferSize_16u_AC4R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u * *nBufferSize*)

Four channel 16-bit unsigned median filter, ignoring alpha channel.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.54 NppStatus nppiFilterMedianGetBufferSize_16u_C1R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u * *nBufferSize*)

Single channel 16-bit unsigned median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).
oMaskSize Width and Height of the neighborhood region for the local Median operation.
nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.55 NppStatus nppiFilterMedianGetBufferSize_16u_C3R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u * nBufferSize)

Three channel 16-bit unsigned median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).
oMaskSize Width and Height of the neighborhood region for the local Median operation.
nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.56 NppStatus nppiFilterMedianGetBufferSize_16u_C4R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u * nBufferSize)

Four channel 16-bit unsigned median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).
oMaskSize Width and Height of the neighborhood region for the local Median operation.
nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.57 NppStatus nppiFilterMedianGetBufferSize_32f_AC4R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u * nBufferSize)

Four channel 32-bit floating-point median filter, ignoring alpha channel.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).
oMaskSize Width and Height of the neighborhood region for the local Median operation.
nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.58 NppStatus nppiFilterMedianGetBufferSize_32f_C1R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u * *nBufferSize*)

Single channel 32-bit floating-point median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.59 NppStatus nppiFilterMedianGetBufferSize_32f_C3R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u * *nBufferSize*)

Three channel 32-bit floating-point median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.60 NppStatus nppiFilterMedianGetBufferSize_32f_C4R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u * *nBufferSize*)

Four channel 32-bit floating-point median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.61 NppStatus nppiFilterMedianGetBufferSize_8u_AC4R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u * *nBufferSize*)

Four channel 8-bit unsigned median filter, ignoring alpha channel.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.62 NppStatus nppiFilterMedianGetBufferSize_8u_C1R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u * nBufferSize)

Single channel 8-bit unsigned median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.63 NppStatus nppiFilterMedianGetBufferSize_8u_C3R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u * nBufferSize)

Three channel 8-bit unsigned median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.64 NppStatus nppiFilterMedianGetBufferSize_8u_C4R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u * nBufferSize)

Four channel 8-bit unsigned median filter scratch memory size.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Median operation.

nBufferSize Pointer to the size of the scratch buffer required for the Median operation.

Returns:

[Image Data Related Error Codes](#)

7.71.1.65 NppStatus nppiFilterMin_16s_AC4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 16-bit signed minimum filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.66 NppStatus nppiFilterMin_16s_C1R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Single channel 16-bit signed minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.67 NppStatus nppiFilterMin_16s_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Three channel 16-bit signed minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.71.1.68 `NppStatus nppiFilterMin_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 16-bit signed minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.71.1.69 `NppStatus nppiFilterMin_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 16-bit unsigned minimum filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.71.1.70 `NppStatus nppiFilterMin_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 16-bit unsigned minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.71 `NppStatus nppiFilterMin_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 16-bit unsigned minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.72 `NppStatus nppiFilterMin_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 16-bit unsigned minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.73 `NppStatus nppiFilterMin_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 32-bit floating-point minimum filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.74 `NppStatus nppiFilterMin_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 32-bit floating-point minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.75 NppStatus nppiFilterMin_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Three channel 32-bit floating-point minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.76 NppStatus nppiFilterMin_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 32-bit floating-point minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.77 NppStatus nppiFilterMin_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 8-bit unsigned minimum filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.78 `NppStatus nppiFilterMin_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 8-bit unsigned minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.79 `NppStatus nppiFilterMin_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 8-bit unsigned minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.80 `NppStatus nppiFilterMin_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned minimum filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Max operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.81 `NppStatus nppiFilterMinBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit signed minimum filter with border control, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.82 `NppStatus nppiFilterMinBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 16-bit signed minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.83 `NppStatus nppiFilterMinBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 16-bit signed minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.84 `NppStatus nppiFilterMinBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit signed minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.85 `NppStatus nppiFilterMinBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned minimum filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.86 `NppStatus nppiFilterMinBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 16-bit unsigned minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.87 `NppStatus nppiFilterMinBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 16-bit unsigned minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.88 `NppStatus nppiFilterMinBorder_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.89 `NppStatus nppiFilterMinBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit floating-point minimum filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.90 `NppStatus nppiFilterMinBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 32-bit floating-point minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.91 `NppStatus nppiFilterMinBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 32-bit floating-point minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.92 `NppStatus nppiFilterMinBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit floating-point minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.93 `NppStatus nppiFilterMinBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned minimum filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.94 `NppStatus nppiFilterMinBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.95 `NppStatus nppiFilterMinBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.71.1.96 `NppStatus nppiFilterMinBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned minimum filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

oMaskSize Width and Height of the neighborhood region for the local Min operation.

oAnchor X and Y offsets of the kernel origin frame of reference relative to the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72 Fixed Filters

Fixed filters perform linear filtering operations (i.e.

FilterPrewittHoriz

Filters the image using a horizontal Prewitt filter kernel:

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterPrewittHoriz_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Single channel 8-bit unsigned horizontal Prewitt filter.
- `NppStatus nppiFilterPrewittHoriz_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Three channel 8-bit unsigned horizontal Prewitt filter.
- `NppStatus nppiFilterPrewittHoriz_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 8-bit unsigned horizontal Prewitt filter.
- `NppStatus nppiFilterPrewittHoriz_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 8-bit unsigned horizontal Prewitt filter, ignoring alpha channel.
- `NppStatus nppiFilterPrewittHoriz_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Single channel 16-bit signed horizontal Prewitt filter.
- `NppStatus nppiFilterPrewittHoriz_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Three channel 16-bit signed horizontal Prewitt filter.
- `NppStatus nppiFilterPrewittHoriz_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 16-bit signed horizontal Prewitt filter.
- `NppStatus nppiFilterPrewittHoriz_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 16-bit signed horizontal Prewitt filter, ignoring alpha channel.
- `NppStatus nppiFilterPrewittHoriz_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Single channel 32-bit floating-point horizontal Prewitt filter.
- `NppStatus nppiFilterPrewittHoriz_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 32-bit floating-point horizontal Prewitt filter.

- `NppStatus nppiFilterPrewittHoriz_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 32-bit floating-point horizontal Prewitt filter.

- `NppStatus nppiFilterPrewittHoriz_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 32-bit floating-point horizontal Prewitt filter, ignoring alpha channel.

FilterPrewittHorizBorder

Filters the image using a horizontal Prewitt filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterPrewittHorizBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned horizontal Prewitt filter with border control.

- `NppStatus nppiFilterPrewittHorizBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned horizontal Prewitt filter with border control.

- `NppStatus nppiFilterPrewittHorizBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned horizontal Prewitt filter with border control.

- `NppStatus nppiFilterPrewittHorizBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned horizontal Prewitt filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterPrewittHorizBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 16-bit signed horizontal Prewitt filter with border control.

- `NppStatus nppiFilterPrewittHorizBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 16-bit signed horizontal Prewitt filter with border control.

- `NppStatus nppiFilterPrewittHorizBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed horizontal Prewitt filter with border control.

- `NppStatus nppiFilterPrewittHorizBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed horizontal Prewitt filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterPrewittHorizBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point horizontal Prewitt filter with border control.

- `NppStatus nppiFilterPrewittHorizBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point horizontal Prewitt filter with border control.

- `NppStatus nppiFilterPrewittHorizBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point horizontal Prewitt filter with border control.

- `NppStatus nppiFilterPrewittHorizBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point horizontal Prewitt filter with border control, ignoring alpha channel.

FilterPrewittVert

Filters the image using a vertical Prewitt filter kernel:

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

- `NppStatus nppiFilterPrewittVert_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit unsigned vertical Prewitt filter.

- `NppStatus nppiFilterPrewittVert_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 8-bit unsigned vertical Prewitt filter.

- `NppStatus nppiFilterPrewittVert_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned vertical Prewitt filter.

- `NppStatus nppiFilterPrewittVert_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned vertical Prewitt filter, ignoring alpha channel.

- `NppStatus nppiFilterPrewittVert_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 16-bit signed vertical Prewitt filter.

- `NppStatus nppiFilterPrewittVert_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 16-bit signed vertical Prewitt filter.

- `NppStatus nppiFilterPrewittVert_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed vertical Prewitt filter.

- `NppStatus nppiFilterPrewittVert_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed vertical Prewitt filter, ignoring alpha channel.

- `NppStatus nppiFilterPrewittVert_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 32-bit floating-point vertical Prewitt filter.

- `NppStatus nppiFilterPrewittVert_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 32-bit floating-point vertical Prewitt filter.

- `NppStatus nppiFilterPrewittVert_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 32-bit floating-point vertical Prewitt filter.

- `NppStatus nppiFilterPrewittVert_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 32-bit floating-point vertical Prewitt filter, ignoring alpha channel.

FilterPrewittVertBorder

Filters the image using a vertical Prewitt filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the `NPP_BORDER_REPLICATE` border type operation is supported.

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix};$$

- `NppStatus nppiFilterPrewittVertBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned vertical Prewitt filter with border control.

- `NppStatus nppiFilterPrewittVertBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 8-bit unsigned vertical Prewitt filter with border control.

- `NppStatus nppiFilterPrewittVertBorder_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned vertical Prewitt filter with border control.

- `NppStatus nppiFilterPrewittVertBorder_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 8-bit unsigned vertical Prewitt filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterPrewittVertBorder_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 16-bit signed vertical Prewitt filter with border control.

- `NppStatus nppiFilterPrewittVertBorder_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 16-bit signed vertical Prewitt filter with border control.

- `NppStatus nppiFilterPrewittVertBorder_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed vertical Prewitt filter with border control.

- `NppStatus nppiFilterPrewittVertBorder_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 16-bit signed vertical Prewitt filter with border control, ignoring alpha channel.

- `NppStatus nppiFilterPrewittVertBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point vertical Prewitt filter with border control.

- `NppStatus nppiFilterPrewittVertBorder_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Three channel 32-bit floating-point vertical Prewitt filter with border control.

- `NppStatus nppiFilterPrewittVertBorder_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point vertical Prewitt filter with border control.

- `NppStatus nppiFilterPrewittVertBorder_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Four channel 32-bit floating-point vertical Prewitt filter with border control, ignoring alpha channel.

FilterScharrHoriz

Filters the image using a horizontal Scharr filter kernel:

$$\begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$$

- `NppStatus nppiFilterScharrHoriz_8u16s_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit unsigned to 16-bit signed horizontal Scharr filter.

- `NppStatus nppiFilterScharrHoriz_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit signed to 16-bit signed horizontal Scharr filter.

- `NppStatus nppiFilterScharrHoriz_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 32-bit floating-point horizontal Scharr filter.

FilterScharrVert

Filters the image using a vertical Scharr filter kernel:

$$\begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}$$

- `NppStatus nppiFilterScharrVert_8u16s_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit unsigned to 16-bit signed vertical Scharr filter.

- `NppStatus nppiFilterScharrVert_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit signed to 16-bit signed vertical Scharr filter.

- `NppStatus nppiFilterScharrVert_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 32-bit floating-point vertical Scharr filter.

FilterScharrHorizBorder

Filters the image using a horizontal Scharr filter kernel with border control:

$$\begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$$

- `NppStatus nppiFilterScharrHorizBorder_8u16s_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned to 16-bit signed horizontal Scharr filter kernel with border control.

- `NppStatus nppiFilterScharrHorizBorder_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 8-bit signed to 16-bit signed horizontal Scharr filter kernel with border control.

- `NppStatus nppiFilterScharrHorizBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point horizontal Scharr filter kernel with border control.

FilterScharrVertBorder

Filters the image using a vertical Scharr filter kernel kernel with border control:

$$\begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}$$

- `NppStatus nppiFilterScharrVertBorder_8u16s_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 8-bit unsigned to 16-bit signed vertical Scharr filter kernel with border control.

- `NppStatus nppiFilterScharrVertBorder_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 8-bit signed to 16-bit signed vertical Scharr filter kernel with border control.

- `NppStatus nppiFilterScharrVertBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

Single channel 32-bit floating-point vertical Scharr filter kernel with border control.

FilterSobelHoriz

Filters the image using a horizontal Sobel filter kernel:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelHoriz_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 8-bit unsigned horizontal Sobel filter.

- `NppStatus nppiFilterSobelHoriz_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 8-bit unsigned horizontal Sobel filter.

- `NppStatus nppiFilterSobelHoriz_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned horizontal Sobel filter.

- `NppStatus nppiFilterSobelHoriz_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed horizontal Sobel filter, ignoring alpha channel.

- `NppStatus nppiFilterSobelHoriz_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 16-bit signed horizontal Sobel filter.

- `NppStatus nppiFilterSobelHoriz_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 16-bit signed horizontal Sobel filter.

- `NppStatus nppiFilterSobelHoriz_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed horizontal Sobel filter.

- `NppStatus nppiFilterSobelHoriz_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 8-bit unsigned horizontal Sobel filter, ignoring alpha channel.

- `NppStatus nppiFilterSobelHoriz_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Single channel 32-bit floating-point horizontal Sobel filter.

- `NppStatus nppiFilterSobelHoriz_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Three channel 32-bit floating-point horizontal Sobel filter.

- `NppStatus nppiFilterSobelHoriz_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 32-bit floating-point horizontal Sobel filter.
- `NppStatus nppiFilterSobelHoriz_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 32-bit floating-point horizontal Sobel filter, ignoring alpha channel.
- `NppStatus nppiFilterSobelHoriz_8u16s_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 8-bit unsigned to 16-bit signed horizontal Sobel filter.
- `NppStatus nppiFilterSobelHoriz_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 8-bit signed to 16-bit signed horizontal Sobel filter.
- `NppStatus nppiFilterSobelHorizMask_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 32-bit floating-point horizontal Sobel filter.

FilterSobelVert

Filters the image using a vertical Sobel filter kernel:

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelVert_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Single channel 8-bit unsigned vertical Sobel filter.
- `NppStatus nppiFilterSobelVert_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Three channel 8-bit unsigned vertical Sobel filter.
- `NppStatus nppiFilterSobelVert_8u_C4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 8-bit unsigned vertical Sobel filter.
- `NppStatus nppiFilterSobelVert_8u_AC4R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 16-bit signed vertical Sobel filter, ignoring alpha channel.
- `NppStatus nppiFilterSobelVert_16s_C1R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Single channel 16-bit signed vertical Sobel filter.

- `NppStatus nppiFilterSobelVert_16s_C3R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Three channel 16-bit signed vertical Sobel filter.
- `NppStatus nppiFilterSobelVert_16s_C4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 16-bit signed vertical Sobel filter.
- `NppStatus nppiFilterSobelVert_16s_AC4R` (const `Npp16s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 8-bit unsigned vertical Sobel filter, ignoring alpha channel.
- `NppStatus nppiFilterSobelVert_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Single channel 32-bit floating-point vertical Sobel filter.
- `NppStatus nppiFilterSobelVert_32f_C3R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Three channel 32-bit floating-point vertical Sobel filter.
- `NppStatus nppiFilterSobelVert_32f_C4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 32-bit floating-point vertical Sobel filter.
- `NppStatus nppiFilterSobelVert_32f_AC4R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)
Four channel 32-bit floating-point vertical Sobel filter, ignoring alpha channel.
- `NppStatus nppiFilterSobelVert_8u16s_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 8-bit unsigned to 16-bit signed vertical Sobel filter.
- `NppStatus nppiFilterSobelVert_8s16s_C1R` (const `Npp8s` *pSrc, `Npp32s` nSrcStep, `Npp16s` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 8-bit signed to 16-bit signed vertical Sobel filter.
- `NppStatus nppiFilterSobelVertMask_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)
Single channel 32-bit floating-point vertical Sobel filter.

FilterSobelHorizSecond

Filters the image using a second derivative, horizontal Sobel filter kernel:

$$\begin{pmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

- [NppStatus nppiFilterSobelHorizSecond_8u16s_C1R](#) (const [Npp8u](#) *pSrc, [Npp32s](#) nSrcStep, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiMaskSize](#) eMaskSize)
Single channel 8-bit unsigned to 16-bit signed second derivative, horizontal Sobel filter.
- [NppStatus nppiFilterSobelHorizSecond_8s16s_C1R](#) (const [Npp8s](#) *pSrc, [Npp32s](#) nSrcStep, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiMaskSize](#) eMaskSize)
Single channel 8-bit signed to 16-bit signed second derivative, horizontal Sobel filter.
- [NppStatus nppiFilterSobelHorizSecond_32f_C1R](#) (const [Npp32f](#) *pSrc, [Npp32s](#) nSrcStep, [Npp32f](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiMaskSize](#) eMaskSize)
Single channel 32-bit floating-point second derivative, horizontal Sobel filter.

7.72.1 Detailed Description

Fixed filters perform linear filtering operations (i.e. convolutions) with predefined kernels of fixed sizes.

Some of the fixed filters have versions with border control. For these functions, if any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported for these functions.

7.72.2 Function Documentation

7.72.2.1 [NppStatus nppiFilterPrewittHoriz_16s_AC4R](#) (const [Npp16s](#) *pSrc, [Npp32s](#) nSrcStep, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI)

Four channel 16-bit signed horizontal Prewitt filter, ignoring alpha channel.

Parameters:

[pSrc](#) [Source-Image Pointer](#).
[nSrcStep](#) [Source-Image Line Step](#).
[pDst](#) [Destination-Image Pointer](#).
[nDstStep](#) [Destination-Image Line Step](#).
[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.2 [NppStatus nppiFilterPrewittHoriz_16s_C1R](#) (const [Npp16s](#) *pSrc, [Npp32s](#) nSrcStep, [Npp16s](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI)

Single channel 16-bit signed horizontal Prewitt filter.

Parameters:

[pSrc](#) [Source-Image Pointer](#).

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.3 NppStatus nppiFilterPrewittHoriz_16s_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit signed horizontal Prewitt filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.4 NppStatus nppiFilterPrewittHoriz_16s_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed horizontal Prewitt filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.5 NppStatus nppiFilterPrewittHoriz_32f_AC4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point horizontal Prewitt filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.6 NppStatus nppiFilterPrewittHoriz_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit floating-point horizontal Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.7 NppStatus nppiFilterPrewittHoriz_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 32-bit floating-point horizontal Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.8 NppStatus nppiFilterPrewittHoriz_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point horizontal Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.9 NppStatus nppiFilterPrewittHoriz_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned horizontal Prewitt filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.10 NppStatus nppiFilterPrewittHoriz_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit unsigned horizontal Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.11 **NppStatus nppiFilterPrewittHoriz_8u_C3R** (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit unsigned horizontal Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.12 **NppStatus nppiFilterPrewittHoriz_8u_C4R** (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned horizontal Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.13 **NppStatus nppiFilterPrewittHorizBorder_16s_AC4R** (const Npp16s * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiBorderType *eBorderType*)

Four channel 16-bit signed horizontal Prewitt filter with border control, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to *pSrc*.
oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.14 `NppStatus nppiFilterPrewittHorizBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 16-bit signed horizontal Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.15 `NppStatus nppiFilterPrewittHorizBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 16-bit signed horizontal Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.16 `NppStatus nppiFilterPrewittHorizBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed horizontal Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.17 `NppStatus nppiFilterPrewittHorizBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point horizontal Prewitt filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.18 `NppStatus nppiFilterPrewittHorizBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 32-bit floating-point horizontal Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.19 `NppStatus nppiFilterPrewittHorizBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 32-bit floating-point horizontal Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.20 `NppStatus nppiFilterPrewittHorizBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point horizontal Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.21 `NppStatus nppiFilterPrewittHorizBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned horizontal Prewitt filter with border control, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.22 `NppStatus nppiFilterPrewittHorizBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned horizontal Prewitt filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.23 `NppStatus nppiFilterPrewittHorizBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 8-bit unsigned horizontal Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.24 `NppStatus nppiFilterPrewittHorizBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned horizontal Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.25 `NppStatus nppiFilterPrewittVert_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 16-bit signed vertical Prewitt filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.26 `NppStatus nppiFilterPrewittVert_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Single channel 16-bit signed vertical Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.27 `NppStatus nppiFilterPrewittVert_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Three channel 16-bit signed vertical Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.28 NppStatus nppiFilterPrewittVert_16s_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed vertical Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.29 NppStatus nppiFilterPrewittVert_32f_AC4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point vertical Prewitt filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.30 NppStatus nppiFilterPrewittVert_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit floating-point vertical Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.31 NppStatus nppiFilterPrewittVert_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 32-bit floating-point vertical Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.32 NppStatus nppiFilterPrewittVert_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point vertical Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.33 NppStatus nppiFilterPrewittVert_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned vertical Prewitt filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.34 NppStatus nppiFilterPrewittVert_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)

Single channel 8-bit unsigned vertical Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.35 NppStatus nppiFilterPrewittVert_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)

Three channel 8-bit unsigned vertical Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.36 NppStatus nppiFilterPrewittVert_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)

Four channel 8-bit unsigned vertical Prewitt filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.37 `NppStatus nppiFilterPrewittVertBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed vertical Prewitt filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.38 `NppStatus nppiFilterPrewittVertBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 16-bit signed vertical Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.39 `NppStatus nppiFilterPrewittVertBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 16-bit signed vertical Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.40 `NppStatus nppiFilterPrewittVertBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed vertical Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.41 `NppStatus nppiFilterPrewittVertBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point vertical Prewitt filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.42 `NppStatus nppiFilterPrewittVertBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 32-bit floating-point vertical Prewitt filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.43 `NppStatus nppiFilterPrewittVertBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 32-bit floating-point vertical Prewitt filter with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.44 `NppStatus nppiFilterPrewittVertBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point vertical Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.45 `NppStatus nppiFilterPrewittVertBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned vertical Prewitt filter with border control, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.46 `NppStatus nppiFilterPrewittVertBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned vertical Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.47 `NppStatus nppiFilterPrewittVertBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 8-bit unsigned vertical Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.48 `NppStatus nppiFilterPrewittVertBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned vertical Prewitt filter with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.49 NppStatus nppiFilterScharrHoriz_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit floating-point horizontal Scharr filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.50 NppStatus nppiFilterScharrHoriz_8s16s_C1R (const Npp8s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit signed to 16-bit signed horizontal Scharr filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.51 NppStatus nppiFilterScharrHoriz_8u16s_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit unsigned to 16-bit signed horizontal Scharr filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.52 NppStatus nppiFilterScharrHorizBorder_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiBorderType *eBorderType*)

Single channel 32-bit floating-point horizontal Scharr filter kernel with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to *pSrc*.
oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.53 NppStatus nppiFilterScharrHorizBorder_8s16s_C1R (const Npp8s * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiBorderType *eBorderType*)

Single channel 8-bit signed to 16-bit signed horizontal Scharr filter kernel with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.54 `NppStatus nppiFilterScharrHorizBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed horizontal Scharr filter kernel with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.55 `NppStatus nppiFilterScharrVert_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Single channel 32-bit floating-point vertical Scharr filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.56 NppStatus nppiFilterScharrVert_8s16s_C1R (const Npp8s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit signed to 16-bit signed vertical Scharr filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.57 NppStatus nppiFilterScharrVert_8u16s_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit unsigned to 16-bit signed vertical Scharr filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.58 NppStatus nppiFilterScharrVertBorder_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiBorderType *eBorderType*)

Single channel 32-bit floating-point vertical Scharr filter kernel with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to *pSrc*.
oSrcOffset The pixel offset that *pSrc* points to relative to the origin of the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.59 `NppStatus nppiFilterScharrVertBorder_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit signed to 16-bit signed vertical Scharr filter kernel with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.60 `NppStatus nppiFilterScharrVertBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed vertical Scharr filter kernel with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset The pixel offset that pSrc points to relative to the origin of the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.61 NppStatus nppiFilterSobelHoriz_16s_AC4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned horizontal Sobel filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.62 NppStatus nppiFilterSobelHoriz_16s_C1R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 16-bit signed horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.63 NppStatus nppiFilterSobelHoriz_16s_C3R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit signed horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.64 NppStatus nppiFilterSobelHoriz_16s_C4R (const Npp16s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.65 NppStatus nppiFilterSobelHoriz_32f_AC4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point horizontal Sobel filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.66 NppStatus nppiFilterSobelHoriz_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit floating-point horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.67 `NppStatus nppiFilterSobelHoriz_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Three channel 32-bit floating-point horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.68 `NppStatus nppiFilterSobelHoriz_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 32-bit floating-point horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.69 `NppStatus nppiFilterSobelHoriz_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit signed to 16-bit signed horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.70 NppStatus nppiFilterSobelHoriz_8u16s_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned to 16-bit signed horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.71 NppStatus nppiFilterSobelHoriz_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed horizontal Sobel filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.72 NppStatus nppiFilterSobelHoriz_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit unsigned horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.73 **NppStatus nppiFilterSobelHoriz_8u_C3R** (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit unsigned horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.74 **NppStatus nppiFilterSobelHoriz_8u_C4R** (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.75 **NppStatus nppiFilterSobelHorizMask_32f_C1R** (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 32-bit floating-point horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.76 `NppStatus nppiFilterSobelHorizSecond_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 32-bit floating-point second derivative, horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.77 `NppStatus nppiFilterSobelHorizSecond_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit signed to 16-bit signed second derivative, horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.78 `NppStatus nppiFilterSobelHorizSecond_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit unsigned to 16-bit signed second derivative, horizontal Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.79 `NppStatus nppiFilterSobelVert_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned vertical Sobel filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.80 `NppStatus nppiFilterSobelVert_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Single channel 16-bit signed vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.81 `NppStatus nppiFilterSobelVert_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Three channel 16-bit signed vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.82 NppStatus nppiFilterSobelVert_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)

Four channel 16-bit signed vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.83 NppStatus nppiFilterSobelVert_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI)

Four channel 32-bit floating-point vertical Sobel filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.84 NppStatus nppiFilterSobelVert_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit floating-point vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.85 NppStatus nppiFilterSobelVert_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 32-bit floating-point vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.86 NppStatus nppiFilterSobelVert_32f_C4R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.87 NppStatus nppiFilterSobelVert_8s16s_C1R (const Npp8s * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit signed to 16-bit signed vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.88 NppStatus nppiFilterSobelVert_8u16s_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp16s * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned to 16-bit signed vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.89 NppStatus nppiFilterSobelVert_8u_AC4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed vertical Sobel filter, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.72.2.90 NppStatus nppiFilterSobelVert_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit unsigned vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.91 NppStatus nppiFilterSobelVert_8u_C3R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit unsigned vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.92 NppStatus nppiFilterSobelVert_8u_C4R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned vertical Sobel filter.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.72.2.93 `NppStatus nppiFilterSobelVertMask_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 32-bit floating-point vertical Sobel filter.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eMaskSize Enumeration value specifying the mask size.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.73 Geometry Transforms

Routines manipulating an image's geometry.

Modules

- [ResizeSqrPixel](#)

ResizeSqrPixel supports the following interpolation modes:.

- [Resize](#)

This function has been deprecated.

- [Remap](#)

Remap supports the following interpolation modes:.

- [Rotate](#)

Rotates an image around the origin (0,0) and then shifts it.

- [Mirror](#)

- [Affine Transforms](#)

- [Perspective Transform](#)

7.73.1 Detailed Description

Routines manipulating an image's geometry.

7.73.2 Geometric Transform API Specifics

This section covers some of the unique API features common to the geometric transform primitives.

7.73.2.1 Geometric Transforms and ROIs

Geometric transforms operate on source and destination ROIs. The way these ROIs affect the processing of pixels differs from other (non geometric) image-processing primitives: Only pixels in the intersection of the destination ROI and the transformed source ROI are being processed.

The typical processing proceeds as follows:

1. Transform the rectangular source ROI (given in source image coordinates) into the destination image space. This yields a quadrilateral.
2. Write only pixels in the intersection of the transformed source ROI and the destination ROI.

7.73.2.2 Pixel Interpolation

The majority of image geometry transform operation need to perform a resampling of the source image as source and destination pixels are not coincident.

NPP supports the following pixel interpolation modes (in order from fastest to slowest and lowest to highest quality):

- nearest neighbor
- linear interpolation
- cubic convolution
- supersampling
- interpolation using Lanczos window function

7.74 ResizeSqrPixel

ResizeSqrPixel supports the following interpolation modes:.

GetResizeRect

Returns [NppiRect](#) which represents the offset and size of the destination rectangle that would be generated by resizing the source [NppiRect](#) by the requested scale factors and shifts.

- [NppStatus](#) [nppiGetResizeRect](#) ([NppiRect](#) oSrcROI, [NppiRect](#) *pDstRect, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)

ResizeSqrPixel

Resizes images.

- [NppStatus](#) [nppiResizeSqrPixel_8u_C1R](#) (const [Npp8u](#) *pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) *pDst, int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
1 channel 8-bit unsigned image resize.
- [NppStatus](#) [nppiResizeSqrPixel_8u_C3R](#) (const [Npp8u](#) *pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) *pDst, int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
3 channel 8-bit unsigned image resize.
- [NppStatus](#) [nppiResizeSqrPixel_8u_C4R](#) (const [Npp8u](#) *pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) *pDst, int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 8-bit unsigned image resize.
- [NppStatus](#) [nppiResizeSqrPixel_8u_AC4R](#) (const [Npp8u](#) *pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) *pDst, int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 8-bit unsigned image resize not affecting alpha.
- [NppStatus](#) [nppiResizeSqrPixel_8u_P3R](#) (const [Npp8u](#) *const pSrc[3], [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) *pDst[3], int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
3 channel 8-bit unsigned planar image resize.
- [NppStatus](#) [nppiResizeSqrPixel_8u_P4R](#) (const [Npp8u](#) *const pSrc[4], [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) *pDst[4], int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 8-bit unsigned planar image resize.
- [NppStatus](#) [nppiResizeSqrPixel_16u_C1R](#) (const [Npp16u](#) *pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp16u](#) *pDst, int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
1 channel 16-bit unsigned image resize.

- `NppStatus nppiResizeSqrPixel_16u_C3R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
3 channel 16-bit unsigned image resize.
- `NppStatus nppiResizeSqrPixel_16u_C4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 16-bit unsigned image resize.
- `NppStatus nppiResizeSqrPixel_16u_AC4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 16-bit unsigned image resize not affecting alpha.
- `NppStatus nppiResizeSqrPixel_16u_P3R` (const `Npp16u` *const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst[3], int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
3 channel 16-bit unsigned planar image resize.
- `NppStatus nppiResizeSqrPixel_16u_P4R` (const `Npp16u` *const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst[4], int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 16-bit unsigned planar image resize.
- `NppStatus nppiResizeSqrPixel_16s_C1R` (const `Npp16s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16s` *pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
1 channel 16-bit signed image resize.
- `NppStatus nppiResizeSqrPixel_16s_C3R` (const `Npp16s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16s` *pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
3 channel 16-bit signed image resize.
- `NppStatus nppiResizeSqrPixel_16s_C4R` (const `Npp16s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16s` *pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 16-bit signed image resize.
- `NppStatus nppiResizeSqrPixel_16s_AC4R` (const `Npp16s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16s` *pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 16-bit signed image resize not affecting alpha.
- `NppStatus nppiResizeSqrPixel_16s_P3R` (const `Npp16s` *const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16s` *pDst[3], int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
3 channel 16-bit signed planar image resize.

- **NppStatus** **nppiResizeSqrPixel_16s_P4R** (const **Npp16s** *const pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16s** *pDst[4], int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 16-bit signed planar image resize.
- **NppStatus** **nppiResizeSqrPixel_32f_C1R** (const **Npp32f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** *pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
1 channel 32-bit floating point image resize.
- **NppStatus** **nppiResizeSqrPixel_32f_C3R** (const **Npp32f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** *pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
3 channel 32-bit floating point image resize.
- **NppStatus** **nppiResizeSqrPixel_32f_C4R** (const **Npp32f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** *pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 32-bit floating point image resize.
- **NppStatus** **nppiResizeSqrPixel_32f_AC4R** (const **Npp32f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** *pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 32-bit floating point image resize not affecting alpha.
- **NppStatus** **nppiResizeSqrPixel_32f_P3R** (const **Npp32f** *const pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** *pDst[3], int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
3 channel 32-bit floating point planar image resize.
- **NppStatus** **nppiResizeSqrPixel_32f_P4R** (const **Npp32f** *const pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** *pDst[4], int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 32-bit floating point planar image resize.
- **NppStatus** **nppiResizeSqrPixel_64f_C1R** (const **Npp64f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp64f** *pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
1 channel 64-bit floating point image resize.
- **NppStatus** **nppiResizeSqrPixel_64f_C3R** (const **Npp64f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp64f** *pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
3 channel 64-bit floating point image resize.
- **NppStatus** **nppiResizeSqrPixel_64f_C4R** (const **Npp64f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp64f** *pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 64-bit floating point image resize.

- **NppStatus** **nppiResizeSqrPixel_64f_AC4R** (const **Npp64f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp64f** *pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 64-bit floating point image resize not affecting alpha.
- **NppStatus** **nppiResizeSqrPixel_64f_P3R** (const **Npp64f** *const pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp64f** *pDst[3], int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
3 channel 64-bit floating point planar image resize.
- **NppStatus** **nppiResizeSqrPixel_64f_P4R** (const **Npp64f** *const pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp64f** *pDst[4], int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)
4 channel 64-bit floating point planar image resize.
- **NppStatus** **nppiResizeAdvancedGetBufferHostSize_8u_C1R** (**NppiSize** oSrcROI, **NppiSize** oDstROI, int *hpBufferSize, int eInterpolationMode)
*Buffer size for **nppiResizeSqrPixel_8u_C1R_Advanced**.*
- **NppStatus** **nppiResizeSqrPixel_8u_C1R_Advanced** (const **Npp8u** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** *pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, **Npp8u** *pBuffer, int eInterpolationMode)
1 channel 8-bit unsigned image resize.

7.74.1 Detailed Description

ResizeSqrPixel supports the following interpolation modes:.

```
NPPI_INTER_NN
NPPI_INTER_LINEAR
NPPI_INTER_CUBIC
NPPI_INTER_CUBIC2P_BSPLINE
NPPI_INTER_CUBIC2P_CATMULLROM
NPPI_INTER_CUBIC2P_BO5C03
NPPI_INTER_SUPER
NPPI_INTER_LANCZOS
```

ResizeSqrPixel attempts to choose source pixels that would approximately represent the center of the destination pixels. It does so by using the following scaling formula to select source pixels for interpolation:

```
nAdjustedXFactor = 1.0 / nXFactor;
nAdjustedYFactor = 1.0 / nYFactor;
nAdjustedXShift = nXShift * nAdjustedXFactor + ((1.0 - nAdjustedXFactor) * 0.5);
nAdjustedYShift = nYShift * nAdjustedYFactor + ((1.0 - nAdjustedYFactor) * 0.5);
nSrcX = nAdjustedXFactor * nDstX - nAdjustedXShift;
nSrcY = nAdjustedYFactor * nDstY - nAdjustedYShift;
```

In the ResizeSqrPixel functions below source image clip checking is handled as follows:

If the source pixel fractional x and y coordinates are greater than or equal to oSizeROI.x and less than oSizeROI.x + oSizeROI.width and greater than or equal to oSizeROI.y and less than oSizeROI.y + oSizeROI.height then the source pixel is considered to be within the source image clip rectangle and the source image is sampled. Otherwise the source image is not sampled and a destination pixel is not written to the destination image.

7.74.2 Error Codes

The resize primitives return the following error codes:

- [NPP_WRONG_INTERSECTION_ROI_ERROR](#) indicates an error condition if srcROIrect has no intersection with the source image.
- [NPP_RESIZE_NO_OPERATION_ERROR](#) if either destination ROI width or height is less than 1 pixel.
- [NPP_RESIZE_FACTOR_ERROR](#) Indicates an error condition if either nXFactor or nYFactor is less than or equal to zero.
- [NPP_INTERPOLATION_ERROR](#) if eInterpolation has an illegal value.
- [NPP_SIZE_ERROR](#) if source size width or height is less than 2 pixels.

7.74.3 Function Documentation

7.74.3.1 NppStatus nppiGetResizeRect (NppiRect oSrcROI, NppiRect * pDstRect, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)

Parameters:

oSrcROI Region of interest in the source image.

pDstRect User supplied host memory pointer to an [NppiRect](#) structure that will be filled in by this function with the region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.2 NppStatus nppiResizeAdvancedGetBufferHostSize_8u_C1R (NppiSize oSrcROI, NppiSize oDstROI, int * hpBufferSize, int eInterpolationMode)

Buffer size for [nppiResizeSqrPixel_8u_C1R_Advanced](#).

Parameters:

oSrcROI [Region-of-Interest \(ROI\)](#).

oDstROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

eInterpolationMode The type of eInterpolation to perform resampling. Currently only supports NPPI_INTER_LANCZOS3_Advanced.

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.74.3.3 NppStatus nppiResizeSqrPixel_16s_AC4R (const Npp16s * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp16s * *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)

4 channel 16-bit signed image resize not affecting alpha.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of interpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.4 NppStatus nppiResizeSqrPixel_16s_C1R (const Npp16s * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp16s * *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)

1 channel 16-bit signed image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.5 `NppStatus nppiResizeSqrPixel_16s_C3R (const Npp16s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16s * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 16-bit signed image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.6 `NppStatus nppiResizeSqrPixel_16s_C4R (const Npp16s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16s * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 16-bit signed image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.7 `NppStatus nppiResizeSqrPixel_16s_P3R (const Npp16s *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16s *pDst[3], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 16-bit signed planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.8 `NppStatus nppiResizeSqrPixel_16s_P4R (const Npp16s *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16s *pDst[4], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 16-bit signed planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.9 **NppStatus nppiResizeSqrPixel_16u_AC4R** (const Npp16u * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp16u * *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)

4 channel 16-bit unsigned image resize not affecting alpha.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of interpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.10 **NppStatus nppiResizeSqrPixel_16u_C1R** (const Npp16u * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp16u * *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)

1 channel 16-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.11 `NppStatus nppiResizeSqrPixel_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 16-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.12 `NppStatus nppiResizeSqrPixel_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 16-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.13 `NppStatus nppiResizeSqrPixel_16u_P3R (const Npp16u *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u *pDst[3], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 16-bit unsigned planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.14 `NppStatus nppiResizeSqrPixel_16u_P4R (const Npp16u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u *pDst[4], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 16-bit unsigned planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.15 `NppStatus nppiResizeSqrPixel_32f_AC4R (const Npp32f *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f *pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 32-bit floating point image resize not affecting alpha.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of interpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.16 `NppStatus nppiResizeSqrPixel_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

1 channel 32-bit floating point image resize.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.17 `NppStatus nppiResizeSqrPixel_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 32-bit floating point image resize.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.18 `NppStatus nppiResizeSqrPixel_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 32-bit floating point image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.19 `NppStatus nppiResizeSqrPixel_32f_P3R (const Npp32f *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 32-bit floating point planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.20 `NppStatus npqiResizeSqrPixel_32f_P4R (const Npp32f *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f *pDst[4], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 32-bit floating point planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.21 NppStatus nppiResizeSqrPixel_64f_AC4R (const Npp64f * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp64f * *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)

4 channel 64-bit floating point image resize not affecting alpha.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of interpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.22 NppStatus nppiResizeSqrPixel_64f_C1R (const Npp64f * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp64f * *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)

1 channel 64-bit floating point image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.23 `NppStatus nppiResizeSqrPixel_64f_C3R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 64-bit floating point image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.24 `NppStatus nppiResizeSqrPixel_64f_C4R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 64-bit floating point image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.25 `NppStatus nppiResizeSqrPixel_64f_P3R (const Npp64f *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f *pDst[3], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 64-bit floating point planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.26 `NppStatus nppiResizeSqrPixel_64f_P4R (const Npp64f *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f *pDst[4], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 64-bit floating point planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.27 **NppStatus nppiResizeSqrPixel_8u_AC4R** (const Npp8u * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp8u * *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)

4 channel 8-bit unsigned image resize not affecting alpha.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of interpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.28 **NppStatus nppiResizeSqrPixel_8u_C1R** (const Npp8u * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp8u * *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)

1 channel 8-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.29 `NppStatus nppiResizeSqrPixel_8u_C1R_Advanced (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, Npp8u * pBuffer, int eInterpolationMode)`

1 channel 8-bit unsigned image resize.

This primitive matches the behavior of GraphicsMagick++.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

pBuffer Device buffer that is used during calculations.

eInterpolationMode The type of eInterpolation to perform resampling. Currently only supports NPPI_INTER_LANCZOS3_Advanced.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.30 `NppStatus nppiResizeSqrPixel_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 8-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.31 `NppStatus nppiResizeSqrPixel_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 8-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
nXShift Source pixel shift in x-direction.
nYShift Source pixel shift in y-direction.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.32 `NppStatus nppiResizeSqrPixel_8u_P3R (const Npp8u *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u *pDst[3], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 8-bit unsigned planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.74.3.33 `NppStatus nppiResizeSqrPixel_8u_P4R (const Npp8u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u *pDst[4], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 8-bit unsigned planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

nXShift Source pixel shift in x-direction.

nYShift Source pixel shift in y-direction.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75 Resize

This function has been deprecated.

Resize

Resizes images.

- `NppStatus nppiResize_8u_C1R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)
1 channel 8-bit unsigned image resize.
- `NppStatus nppiResize_8u_C3R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)
3 channel 8-bit unsigned image resize.
- `NppStatus nppiResize_8u_C4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)
4 channel 8-bit unsigned image resize.
- `NppStatus nppiResize_8u_AC4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)
4 channel 8-bit unsigned image resize not affecting alpha.
- `NppStatus nppiResize_8u_P3R` (const `Npp8u` *const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst[3], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)
3 channel 8-bit unsigned planar image resize.
- `NppStatus nppiResize_8u_P4R` (const `Npp8u` *const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst[4], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)
4 channel 8-bit unsigned planar image resize.
- `NppStatus nppiResize_16u_C1R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)
1 channel 16-bit unsigned image resize.
- `NppStatus nppiResize_16u_C3R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)
3 channel 16-bit unsigned image resize.
- `NppStatus nppiResize_16u_C4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

4 channel 16-bit unsigned image resize.

- `NppStatus nppiResize_16u_AC4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

4 channel 16-bit unsigned image resize not affecting alpha.

- `NppStatus nppiResize_16u_P3R` (const `Npp16u` *const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst[3], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

3 channel 16-bit unsigned planar image resize.

- `NppStatus nppiResize_16u_P4R` (const `Npp16u` *const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst[4], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

4 channel 16-bit unsigned planar image resize.

- `NppStatus nppiResize_32f_C1R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

1 channel 32-bit floating point image resize.

- `NppStatus nppiResize_32f_C3R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

3 channel 32-bit floating point image resize.

- `NppStatus nppiResize_32f_C4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

4 channel 32-bit floating point image resize.

- `NppStatus nppiResize_32f_AC4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

4 channel 32-bit floating point image resize not affecting alpha.

- `NppStatus nppiResize_32f_P3R` (const `Npp32f` *const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst[3], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

3 channel 32-bit floating point planar image resize.

- `NppStatus nppiResize_32f_P4R` (const `Npp32f` *const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst[4], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

4 channel 32-bit floating point planar image resize.

7.75.1 Detailed Description

This function has been deprecated.

ResizeSqrPixel provides the same functionality and more.

Resize supports the following interpolation modes:

```
NPPI_INTER_NN
NPPI_INTER_LINEAR
NPPI_INTER_CUBIC
NPPI_INTER_SUPER
NPPI_INTER_LANCZOS
```

Resize uses the following scaling formula to select source pixels for interpolation:

```
scaledSrcSize.width = nXFactor * srcRectROI.width;
scaledSrcSize.height = nYFactor * srcRectROI.height;
nAdjustedXFactor = (srcRectROI.width - 1) / (scaledSrcSize.width - 1);
nAdjustedYFactor = (srcRectROI.height - 1) / (scaledSrcSize.height - 1);
nSrcX = nAdjustedXFactor * nDstX;
nSrcY = nAdjustedYFactor * nDstY;
```

In the Resize functions below source image clip checking is handled as follows:

If the source pixel fractional x and y coordinates are greater than or equal to oSizeROI.x and less than oSizeROI.x + oSizeROI.width and greater than or equal to oSizeROI.y and less than oSizeROI.y + oSizeROI.height then the source pixel is considered to be within the source image clip rectangle and the source image is sampled. Otherwise the source image is not sampled and a destination pixel is not written to the destination image.

7.75.2 Error Codes

The resize primitives return the following error codes:

- [NPP_WRONG_INTERSECTION_ROI_ERROR](#) indicates an error condition if srcROIRect has no intersection with the source image.
- [NPP_RESIZE_NO_OPERATION_ERROR](#) if either destination ROI width or height is less than 1 pixel.
- [NPP_RESIZE_FACTOR_ERROR](#) Indicates an error condition if either nXFactor or nYFactor is less than or equal to zero.
- [NPP_INTERPOLATION_ERROR](#) if eInterpolation has an illegal value.
- [NPP_SIZE_ERROR](#) if source size width or height is less than 2 pixels.

7.75.3 Function Documentation

7.75.3.1 `NppStatus nppiResize_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 16-bit unsigned image resize not affecting alpha.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of interpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.2 `NppStatus nppiResize_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

1 channel 16-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.3 `NppStatus nppiResize_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

3 channel 16-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image

oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.4 **NppStatus nppiResize_16u_C4R** (const Npp16u * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp16u * *pDst*, int *nDstStep*, NppiSize *dstROISize*, double *nXFactor*, double *nYFactor*, int *eInterpolation*)

4 channel 16-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.5 **NppStatus nppiResize_16u_P3R** (const Npp16u *const *pSrc*[3], NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp16u * *pDst*[3], int *nDstStep*, NppiSize *dstROISize*, double *nXFactor*, double *nYFactor*, int *eInterpolation*)

3 channel 16-bit unsigned planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

dstROISize Size in pixels of the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.6 `NppStatus nppiResize_16u_P4R (const Npp16u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u *pDst[4], int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 16-bit unsigned planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

dstROISize Size in pixels of the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.7 `NppStatus nppiResize_32f_AC4R (const Npp32f *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f *pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 32-bit floating point image resize not affecting alpha.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of interpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.8 `NppStatus nppiResize_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

1 channel 32-bit floating point image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.9 `NppStatus nppiResize_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

3 channel 32-bit floating point image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.10 `NppStatus nppiResize_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 32-bit floating point image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.11 `NppStatus nppiResize_32f_P3R (const Npp32f *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

3 channel 32-bit floating point planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).
nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

dstROISize Size in pixels of the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.12 `NppStatus nppiResize_32f_P4R (const Npp32f *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f *pDst[4], int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 32-bit floating point planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

dstROISize Size in pixels of the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.13 `NppStatus nppiResize_8u_AC4R (const Npp8u *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u *pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 8-bit unsigned image resize not affecting alpha.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of interpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.14 `NppStatus nppiResize_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

1 channel 8-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.15 `NppStatus nppiResize_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

3 channel 8-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.16 `NppStatus nppiResize_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 8-bit unsigned image resize.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
dstROISize Size in pixels of the destination image.
nXFactor Factor by which x dimension is changed.
nYFactor Factor by which y dimension is changed.
eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.17 `NppStatus nppiResize_8u_P3R (const Npp8u *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

3 channel 8-bit unsigned planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).
nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

dstROISize Size in pixels of the destination image

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.75.3.18 `NppStatus nppiResize_8u_P4R (const Npp8u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 8-bit unsigned planar image resize.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pDst [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

nDstStep [Destination-Image Line Step](#).

dstROISize Size in pixels of the destination image.

nXFactor Factor by which x dimension is changed.

nYFactor Factor by which y dimension is changed.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76 Remap

Remap supports the following interpolation modes:.

Remap

Remaps images.

- `NppStatus nppiRemap_8u_C1R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
1 channel 8-bit unsigned image remap.
- `NppStatus nppiRemap_8u_C3R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
3 channel 8-bit unsigned image remap.
- `NppStatus nppiRemap_8u_C4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
4 channel 8-bit unsigned image remap.
- `NppStatus nppiRemap_8u_AC4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
4 channel 8-bit unsigned image remap not affecting alpha.
- `NppStatus nppiRemap_8u_P3R` (const `Npp8u` *const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp8u` *pDst[3], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
3 channel 8-bit unsigned planar image remap.
- `NppStatus nppiRemap_8u_P4R` (const `Npp8u` *const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp8u` *pDst[4], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
4 channel 8-bit unsigned planar image remap.
- `NppStatus nppiRemap_16u_C1R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
1 channel 16-bit unsigned image remap.
- `NppStatus nppiRemap_16u_C3R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
3 channel 16-bit unsigned image remap.
- `NppStatus nppiRemap_16u_C4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

4 channel 16-bit unsigned image remap.

- `NppStatus nppiRemap_16u_AC4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

4 channel 16-bit unsigned image remap not affecting alpha.

- `NppStatus nppiRemap_16u_P3R` (const `Npp16u` *const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16u` *pDst[3], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

3 channel 16-bit unsigned planar image remap.

- `NppStatus nppiRemap_16u_P4R` (const `Npp16u` *const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16u` *pDst[4], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

4 channel 16-bit unsigned planar image remap.

- `NppStatus nppiRemap_16s_C1R` (const `Npp16s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

1 channel 16-bit signed image remap.

- `NppStatus nppiRemap_16s_C3R` (const `Npp16s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

3 channel 16-bit signed image remap.

- `NppStatus nppiRemap_16s_C4R` (const `Npp16s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

4 channel 16-bit signed image remap.

- `NppStatus nppiRemap_16s_AC4R` (const `Npp16s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

4 channel 16-bit signed image remap not affecting alpha.

- `NppStatus nppiRemap_16s_P3R` (const `Npp16s` *const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16s` *pDst[3], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

3 channel 16-bit signed planar image remap.

- `NppStatus nppiRemap_16s_P4R` (const `Npp16s` *const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp16s` *pDst[4], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

4 channel 16-bit signed planar image remap.

- `NppStatus nppiRemap_32f_C1R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

1 channel 32-bit floating point image remap.

- `NppStatus nppiRemap_32f_C3R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
3 channel 32-bit floating point image remap.
- `NppStatus nppiRemap_32f_C4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
4 channel 32-bit floating point image remap.
- `NppStatus nppiRemap_32f_AC4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
4 channel 32-bit floating point image remap not affecting alpha.
- `NppStatus nppiRemap_32f_P3R` (const `Npp32f` *const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp32f` *pDst[3], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
3 channel 32-bit floating point planar image remap.
- `NppStatus nppiRemap_32f_P4R` (const `Npp32f` *const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` *pXMap, int nXMapStep, const `Npp32f` *pYMap, int nYMapStep, `Npp32f` *pDst[4], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
4 channel 32-bit floating point planar image remap.
- `NppStatus nppiRemap_64f_C1R` (const `Npp64f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp64f` *pXMap, int nXMapStep, const `Npp64f` *pYMap, int nYMapStep, `Npp64f` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
1 channel 64-bit floating point image remap.
- `NppStatus nppiRemap_64f_C3R` (const `Npp64f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp64f` *pXMap, int nXMapStep, const `Npp64f` *pYMap, int nYMapStep, `Npp64f` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
3 channel 64-bit floating point image remap.
- `NppStatus nppiRemap_64f_C4R` (const `Npp64f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp64f` *pXMap, int nXMapStep, const `Npp64f` *pYMap, int nYMapStep, `Npp64f` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
4 channel 64-bit floating point image remap.
- `NppStatus nppiRemap_64f_AC4R` (const `Npp64f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp64f` *pXMap, int nXMapStep, const `Npp64f` *pYMap, int nYMapStep, `Npp64f` *pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
4 channel 64-bit floating point image remap not affecting alpha.
- `NppStatus nppiRemap_64f_P3R` (const `Npp64f` *const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp64f` *pXMap, int nXMapStep, const `Npp64f` *pYMap, int nYMapStep, `Npp64f` *pDst[3], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)
3 channel 64-bit floating point planar image remap.

- **NppStatus nppiRemap_64f_P4R** (const **Npp64f** *const pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp64f** *pXMap, int nXMapStep, const **Npp64f** *pYMap, int nYMapStep, **Npp64f** *pDst[4], int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)

4 channel 64-bit floating point planar image remap.

7.76.1 Detailed Description

Remap supports the following interpolation modes:.

NPPI_INTER_NN NPPI_INTER_LINEAR NPPI_INTER_CUBIC NPPI_INTER_CUBIC2P_BSPLINE
NPPI_INTER_CUBIC2P_CATMULLROM NPPI_INTER_CUBIC2P_B05C03 NPPI_INTER_-
LANCZOS

Remap chooses source pixels using pixel coordinates explicitly supplied in two 2D device memory image arrays pointed to by the pXMap and pYMap pointers. The pXMap array contains the X coordinated and the pYMap array contains the Y coordinate of the corresponding source image pixel to use as input. These coordinates are in floating point format so fraction pixel positions can be used. The coordinates of the source pixel to sample are determined as follows:

$nSrcX = pXMap[nDstX, nDstY]$ $nSrcY = pYMap[nDstX, nDstY]$

In the Remap functions below source image clip checking is handled as follows:

If the source pixel fractional x and y coordinates are greater than or equal to oSizeROI.x and less than oSizeROI.x + oSizeROI.width and greater than or equal to oSizeROI.y and less than oSizeROI.y + oSizeROI.height then the source pixel is considered to be within the source image clip rectangle and the source image is sampled. Otherwise the source image is not sampled and a destination pixel is not written to the destination image.

7.76.2 Error Codes

The remap primitives return the following error codes:

- **NPP_WRONG_INTERSECTION_ROI_ERROR** indicates an error condition if srcROIRect has no intersection with the source image.
- **NPP_INTERPOLATION_ERROR** if eInterpolation has an illegal value.

7.76.3 Function Documentation

7.76.3.1 NppStatus nppiRemap_16s_AC4R (const **Npp16s** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp32f** *pXMap, int nXMapStep, const **Npp32f** *pYMap, int nYMapStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)

4 channel 16-bit signed image remap not affecting alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.2 `NppStatus nppiRemap_16s_C1R (const Npp16s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp16s * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

1 channel 16-bit signed image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.3 NppStatus nppiRemap_16s_C3R (const Npp16s * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, const Npp32f * *pXMap*, int *nXMapStep*, const Npp32f * *pYMap*, int *nYMapStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *eInterpolation*)

3 channel 16-bit signed image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep *pXMap* image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep *pYMap* image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of *eInterpolation* to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.4 NppStatus nppiRemap_16s_C4R (const Npp16s * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, const Npp32f * *pXMap*, int *nXMapStep*, const Npp32f * *pYMap*, int *nYMapStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *eInterpolation*)

4 channel 16-bit signed image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep *pXMap* image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep *pYMap* image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.5 `NppStatus nppiRemap_16s_P3R (const Npp16s *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp16s *pDst[3], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 16-bit signed planar image remap.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Planar-Image Pointer Array](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.6 `NppStatus nppiRemap_16s_P4R (const Npp16s *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp16s *pDst[4], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 16-bit signed planar image remap.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Planar-Image Pointer Array](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.7 NppStatus nppiRemap_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)

4 channel 16-bit unsigned image remap not affecting alpha.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of interpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.8 `NppStatus nppiRemap_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

1 channel 16-bit unsigned image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.9 `NppStatus nppiRemap_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 16-bit unsigned image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.10 `NppStatus nppiRemap_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 16-bit unsigned image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.11 `NppStatus nppiRemap_16u_P3R (const Npp16u *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp16u * pDst[3], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 16-bit unsigned planar image remap.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep *pXMap* image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep *pYMap* image array line step in bytes.

pDst [Destination-Planar-Image Pointer Array](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of *eInterpolation* to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.12 `NppStatus nppiRemap_16u_P4R (const Npp16u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp16u *pDst[4], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 16-bit unsigned planar image remap.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep *pXMap* image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep *pYMap* image array line step in bytes.

pDst [Destination-Planar-Image Pointer Array](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of *eInterpolation* to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.13 `NppStatus nppiRemap_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 32-bit floating point image remap not affecting alpha.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.14 `NppStatus nppiRemap_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

1 channel 32-bit floating point image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.15 `NppStatus nppiRemap_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 32-bit floating point image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.16 `NppStatus nppiRemap_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 32-bit floating point image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep *pXMap* image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep *pYMap* image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of *eInterpolation* to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.17 `NppStatus nppiRemap_32f_P3R (const Npp32f *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp32f *pDst[3], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 32-bit floating point planar image remap.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep *pXMap* image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep *pYMap* image array line step in bytes.

pDst [Destination-Planar-Image Pointer Array](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of *eInterpolation* to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.18 `NppStatus nppiRemap_32f_P4R (const Npp32f *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp32f *pDst[4], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 32-bit floating point planar image remap.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep *pXMap* image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep *pYMap* image array line step in bytes.

pDst [Destination-Planar-Image Pointer Array](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of *eInterpolation* to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.19 `NppStatus nppiRemap_64f_AC4R (const Npp64f *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep, const Npp64f *pYMap, int nYMapStep, Npp64f *pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 64-bit floating point image remap not affecting alpha.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep *pXMap* image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep *pYMap* image array line step in bytes.

pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Region of interest size in the destination image.
eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.20 `NppStatus nppiRemap_64f_C1R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f * pXMap, int nXMapStep, const Npp64f * pYMap, int nYMapStep, Npp64f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

1 channel 64-bit floating point image remap.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.
oSrcROI Region of interest in the source image.
pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.
nXMapStep pXMap image array line step in bytes.
pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.
nYMapStep pYMap image array line step in bytes.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstSizeROI Region of interest size in the destination image.
eInterpolation The type of eInterpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.21 `NppStatus nppiRemap_64f_C3R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f * pXMap, int nXMapStep, const Npp64f * pYMap, int nYMapStep, Npp64f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 64-bit floating point image remap.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.22 `NppStatus nppiRemap_64f_C4R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f * pXMap, int nXMapStep, const Npp64f * pYMap, int nYMapStep, Npp64f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 64-bit floating point image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.23 `NppStatus nppiRemap_64f_P3R (const Npp64f *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep, const Npp64f *pYMap, int nYMapStep, Npp64f *pDst[3], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 64-bit floating point planar image remap.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Planar-Image Pointer Array](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.24 `NppStatus nppiRemap_64f_P4R (const Npp64f *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep, const Npp64f *pYMap, int nYMapStep, Npp64f *pDst[4], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 64-bit floating point planar image remap.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst Destination-Planar-Image Pointer Array.

nDstStep Destination-Image Line Step.

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.25 `NppStatus nppiRemap_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 8-bit unsigned image remap not affecting alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of interpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.26 `NppStatus nppiRemap_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

1 channel 8-bit unsigned image remap.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.27 `NppStatus nppiRemap_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 8-bit unsigned image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.28 `NppStatus nppiRemap_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 8-bit unsigned image remap.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.29 `NppStatus nppiRemap_8u_P3R (const Npp8u *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp8u * pDst[3], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 8-bit unsigned planar image remap.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Planar-Image Pointer Array](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.76.3.30 `NppStatus nppiRemap_8u_P4R (const Npp8u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp8u *pDst[4], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 8-bit unsigned planar image remap.

Parameters:

pSrc [Source-Planar-Image Pointer Array](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image.

oSrcROI Region of interest in the source image.

pXMap Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

nXMapStep pXMap image array line step in bytes.

pYMap Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

nYMapStep pYMap image array line step in bytes.

pDst [Destination-Planar-Image Pointer Array](#).

nDstStep [Destination-Image Line Step](#).

oDstSizeROI Region of interest size in the destination image.

eInterpolation The type of eInterpolation to perform resampling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

7.77 Rotate

Rotates an image around the origin (0,0) and then shifts it.

Utility Functions

- **NppStatus** **nppiGetRotateQuad** (**NppiRect** oSrcROI, double aQuad[4][2], double nAngle, double nShiftX, double nShiftY)
Compute shape of rotated image.
- **NppStatus** **nppiGetRotateBound** (**NppiRect** oSrcROI, double aBoundingBox[2][2], double nAngle, double nShiftX, double nShiftY)
Compute bounding-box of rotated image.

Rotate

- **NppStatus** **nppiRotate_8u_C1R** (const **Npp8u** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** *pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)
8-bit unsigned image rotate.
- **NppStatus** **nppiRotate_8u_C3R** (const **Npp8u** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** *pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)
3 channel 8-bit unsigned image rotate.
- **NppStatus** **nppiRotate_8u_C4R** (const **Npp8u** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** *pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)
4 channel 8-bit unsigned image rotate.
- **NppStatus** **nppiRotate_8u_AC4R** (const **Npp8u** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** *pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)
4 channel 8-bit unsigned image rotate ignoring alpha channel.
- **NppStatus** **nppiRotate_16u_C1R** (const **Npp16u** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** *pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)
16-bit unsigned image rotate.
- **NppStatus** **nppiRotate_16u_C3R** (const **Npp16u** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** *pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)
3 channel 16-bit unsigned image rotate.
- **NppStatus** **nppiRotate_16u_C4R** (const **Npp16u** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** *pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

4 channel 16-bit unsigned image rotate.

- `NppStatus nppiRotate_16u_AC4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

4 channel 16-bit unsigned image rotate ignoring alpha channel.

- `NppStatus nppiRotate_32f_C1R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

32-bit float image rotate.

- `NppStatus nppiRotate_32f_C3R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

3 channel 32-bit float image rotate.

- `NppStatus nppiRotate_32f_C4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

4 channel 32-bit float image rotate.

- `NppStatus nppiRotate_32f_AC4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

4 channel 32-bit float image rotate ignoring alpha channel.

7.77.1 Detailed Description

Rotates an image around the origin (0,0) and then shifts it.

7.77.2 Rotate Error Codes

- `NPP_INTERPOLATION_ERROR` if eInterpolation has an illegal value.
- `NPP_RECTANGLE_ERROR` Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1.
- `NPP_WRONG_INTERSECTION_ROI_ERROR` indicates an error condition if srcROIrect has no intersection with the source image.
- `NPP_WRONG_INTERSECTION_QUAD_WARNING` indicates a warning that no operation is performed if the transformed source ROI does not intersect the destination ROI.

7.77.3 Function Documentation

7.77.3.1 `NppStatus nppiGetRotateBound` (`NppiRect` oSrcROI, double aBoundingBox[2][2], double nAngle, double nShiftX, double nShiftY)

Compute bounding-box of rotated image.

Parameters:

oSrcROI Region-of-interest of the source image.

aBoundingBox Two 2D points representing the bounding-box of the rotated image. All four points from `nppiGetRotateQuad` are contained inside the axis-aligned rectangle spanned by the two points of this bounding box.

nAngle The rotation angle.

nShiftX Post-rotation shift in x-direction.

nShiftY Post-rotation shift in y-direction.

Returns:

[ROI Related Error Codes.](#)

7.77.3.2 `NppStatus nppiGetRotateQuad (NppiRect oSrcROI, double aQuad[4][2], double nAngle, double nShiftX, double nShiftY)`

Compute shape of rotated image.

Parameters:

oSrcROI Region-of-interest of the source image.

aQuad Array of 2D points. These points are the locations of the corners of the rotated ROI.

nAngle The rotation nAngle.

nShiftX Post-rotation shift in x-direction

nShiftY Post-rotation shift in y-direction

Returns:

[ROI Related Error Codes.](#)

7.77.3.3 `NppStatus nppiRotate_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 16-bit unsigned image rotate ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer.](#)

nSrcStep [Source-Image Line Step.](#)

oSrcSize Size in pixels of the source image

oSrcROI Region of interest in the source image.

pDst [Destination-Image Pointer.](#)

nDstStep [Destination-Image Line Step.](#)

oDstROI Region of interest in the destination image.

nAngle The angle of rotation in degrees.

nShiftX Shift along horizontal axis

nShiftY Shift along vertical axis

eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.77.3.4 `NppStatus nppiRotate_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

16-bit unsigned image rotate.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image

oSrcROI Region of interest in the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nAngle The angle of rotation in degrees.

nShiftX Shift along horizontal axis

nShiftY Shift along vertical axis

eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.77.3.5 `NppStatus nppiRotate_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

3 channel 16-bit unsigned image rotate.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image

oSrcROI Region of interest in the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nAngle The angle of rotation in degrees.

nShiftX Shift along horizontal axis

nShiftY Shift along vertical axis

eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.77.3.6 `NppStatus nppiRotate_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 16-bit unsigned image rotate.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image

oSrcROI Region of interest in the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nAngle The angle of rotation in degrees.

nShiftX Shift along horizontal axis

nShiftY Shift along vertical axis

eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.77.3.7 `NppStatus nppiRotate_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 32-bit float image rotate ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Size in pixels of the source image

oSrcROI Region of interest in the source image.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.

nAngle The angle of rotation in degrees.
nShiftX Shift along horizontal axis
nShiftY Shift along vertical axis
eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.77.3.8 `NppStatus nppiRotate_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

32-bit float image rotate.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nAngle The angle of rotation in degrees.
nShiftX Shift along horizontal axis
nShiftY Shift along vertical axis
eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.77.3.9 `NppStatus nppiRotate_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

3 channel 32-bit float image rotate.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

oDstROI Region of interest in the destination image.
nAngle The angle of rotation in degrees.
nShiftX Shift along horizontal axis
nShiftY Shift along vertical axis
eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.77.3.10 `NppStatus nppiRotate_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 32-bit float image rotate.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nAngle The angle of rotation in degrees.
nShiftX Shift along horizontal axis
nShiftY Shift along vertical axis
eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.77.3.11 `NppStatus nppiRotate_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 8-bit unsigned image rotate ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.
oDstROI Region of interest in the destination image.
nAngle The angle of rotation in degrees.
nShiftX Shift along horizontal axis
nShiftY Shift along vertical axis
eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.77.3.12 `NppStatus nppiRotate_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

8-bit unsigned image rotate.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oDstROI Region of interest in the destination image.
nAngle The angle of rotation in degrees.
nShiftX Shift along horizontal axis
nShiftY Shift along vertical axis
eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.77.3.13 `NppStatus nppiRotate_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

3 channel 8-bit unsigned image rotate.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.

pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nAngle The angle of rotation in degrees.
nShiftX Shift along horizontal axis
nShiftY Shift along vertical axis
eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.77.3.14 `NppStatus nppiRotate_8u_C4R (const Npp8u *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u *pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 8-bit unsigned image rotate.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Size in pixels of the source image
oSrcROI Region of interest in the source image.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Region of interest in the destination image.
nAngle The angle of rotation in degrees.
nShiftX Shift along horizontal axis
nShiftY Shift along vertical axis
eInterpolation The type of interpolation to perform resampling

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

7.78 Mirror

Mirror

Mirrors images horizontally, vertically and diagonally.

- `NppStatus nppiMirror_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
1 channel 8-bit unsigned image mirror.
- `NppStatus nppiMirror_8u_C1IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)
1 channel 8-bit unsigned in place image mirror.
- `NppStatus nppiMirror_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
3 channel 8-bit unsigned image mirror.
- `NppStatus nppiMirror_8u_C3IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)
3 channel 8-bit unsigned in place image mirror.
- `NppStatus nppiMirror_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
4 channel 8-bit unsigned image mirror.
- `NppStatus nppiMirror_8u_C4IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)
4 channel 8-bit unsigned in place image mirror.
- `NppStatus nppiMirror_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
4 channel 8-bit unsigned image mirror not affecting alpha.
- `NppStatus nppiMirror_8u_AC4IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)
4 channel 8-bit unsigned in place image mirror not affecting alpha.
- `NppStatus nppiMirror_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
1 channel 16-bit unsigned image mirror.
- `NppStatus nppiMirror_16u_C1IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)
1 channel 16-bit unsigned in place image mirror.
- `NppStatus nppiMirror_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
3 channel 16-bit unsigned image mirror.

- **NppStatus nppiMirror_16u_C3IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)
3 channel 16-bit unsigned in place image mirror.
- **NppStatus nppiMirror_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)
4 channel 16-bit unsigned image mirror.
- **NppStatus nppiMirror_16u_C4IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)
4 channel 16-bit unsigned in place image mirror.
- **NppStatus nppiMirror_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)
4 channel 16-bit unsigned image mirror not affecting alpha.
- **NppStatus nppiMirror_16u_AC4IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)
4 channel 16-bit unsigned in place image mirror not affecting alpha.
- **NppStatus nppiMirror_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)
1 channel 16-bit signed image mirror.
- **NppStatus nppiMirror_16s_C1IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)
1 channel 16-bit signed in place image mirror.
- **NppStatus nppiMirror_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)
3 channel 16-bit signed image mirror.
- **NppStatus nppiMirror_16s_C3IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)
3 channel 16-bit signed in place image mirror.
- **NppStatus nppiMirror_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)
4 channel 16-bit signed image mirror.
- **NppStatus nppiMirror_16s_C4IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)
4 channel 16-bit signed in place image mirror.
- **NppStatus nppiMirror_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)
4 channel 16-bit signed image mirror not affecting alpha.
- **NppStatus nppiMirror_16s_AC4IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)
4 channel 16-bit signed in place image mirror not affecting alpha.

- `NppStatus nppiMirror_32s_C1R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
1 channel 32-bit image mirror.
- `NppStatus nppiMirror_32s_C1IR` (`Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)
1 channel 32-bit signed in place image mirror.
- `NppStatus nppiMirror_32s_C3R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
3 channel 32-bit image mirror.
- `NppStatus nppiMirror_32s_C3IR` (`Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)
3 channel 32-bit signed in place image mirror.
- `NppStatus nppiMirror_32s_C4R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
4 channel 32-bit image mirror.
- `NppStatus nppiMirror_32s_C4IR` (`Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)
4 channel 32-bit signed in place image mirror.
- `NppStatus nppiMirror_32s_AC4R` (const `Npp32s` *pSrc, int nSrcStep, `Npp32s` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
4 channel 32-bit image mirror not affecting alpha.
- `NppStatus nppiMirror_32s_AC4IR` (`Npp32s` *pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)
4 channel 32-bit signed in place image mirror not affecting alpha.
- `NppStatus nppiMirror_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
1 channel 32-bit float image mirror.
- `NppStatus nppiMirror_32f_C1IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)
1 channel 32-bit float in place image mirror.
- `NppStatus nppiMirror_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)
3 channel 32-bit float image mirror.
- `NppStatus nppiMirror_32f_C3IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)
3 channel 32-bit float in place image mirror.
- `NppStatus nppiMirror_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)

4 channel 32-bit float image mirror.

- **NppStatus** **nppiMirror_32f_C4IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)

4 channel 32-bit float in place image mirror.

- **NppStatus** **nppiMirror_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)

4 channel 32-bit float image mirror not affecting alpha.

- **NppStatus** **nppiMirror_32f_AC4IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)

4 channel 32-bit float in place image mirror not affecting alpha.

7.78.1 Detailed Description

7.78.2 Mirror Error Codes

- **NPP_MIRROR_FLIP_ERR** if flip has an illegal value.

7.78.3 Function Documentation

7.78.3.1 **NppStatus** **nppiMirror_16s_AC4IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)

4 channel 16-bit signed in place image mirror not affecting alpha.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

7.78.3.2 **NppStatus** **nppiMirror_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)

4 channel 16-bit signed image mirror not affecting alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Distance in bytes between starts of consecutive lines of the destination image.

oROI [Region-of-Interest \(ROI\)](#).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.3 NppStatus nppiMirror_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oROI, NppiAxis flip)

1 channel 16-bit signed in place image mirror.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.4 NppStatus nppiMirror_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oROI, NppiAxis flip)

1 channel 16-bit signed image mirror.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.5 NppStatus nppiMirror_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oROI, NppiAxis flip)

3 channel 16-bit signed in place image mirror.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

7.78.3.6 NppStatus nppiMirror_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oROI, NppiAxis flip)

3 channel 16-bit signed image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

7.78.3.7 NppStatus nppiMirror_16s_C4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oROI, NppiAxis flip)

4 channel 16-bit signed in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

7.78.3.8 NppStatus nppiMirror_16s_C4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oROI, NppiAxis flip)

4 channel 16-bit signed image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Distance in bytes between starts of consecutive lines of the destination image.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

7.78.3.9 NppStatus nppiMirror_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oROI, NppiAxis flip)

4 channel 16-bit unsigned in place image mirror not affecting alpha.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

7.78.3.10 NppStatus nppiMirror_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oROI, NppiAxis flip)

4 channel 16-bit unsigned image mirror not affecting alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Distance in bytes between starts of consecutive lines of the destination image.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

7.78.3.11 NppStatus nppiMirror_16u_C1IR (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 16-bit unsigned in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.12 NppStatus nppiMirror_16u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 16-bit unsigned image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.13 NppStatus nppiMirror_16u_C3IR (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 16-bit unsigned in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.14 **NppStatus nppiMirror_16u_C3R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 16-bit unsigned image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.15 **NppStatus nppiMirror_16u_C4IR** (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 16-bit unsigned in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.16 **NppStatus nppiMirror_16u_C4R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 16-bit unsigned image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Distance in bytes between starts of consecutive lines of the destination image.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.17 NppStatus nppiMirror_32f_AC4IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit float in place image mirror not affecting alpha.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.18 NppStatus nppiMirror_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit float image mirror not affecting alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Distance in bytes between starts of consecutive lines of the destination image.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.19 NppStatus nppiMirror_32f_C1IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 32-bit float in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.20 **NppStatus nppiMirror_32f_C1R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 32-bit float image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.21 **NppStatus nppiMirror_32f_C3IR** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 32-bit float in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.22 **NppStatus nppiMirror_32f_C3R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 32-bit float image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.23 NppStatus nppiMirror_32f_C4IR (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit float in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.24 NppStatus nppiMirror_32f_C4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit float image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Distance in bytes between starts of consecutive lines of the destination image.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.25 NppStatus nppiMirror_32s_AC4IR (Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit signed in place image mirror not affecting alpha.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.26 **NppStatus nppiMirror_32s_AC4R** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit image mirror not affecting alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Distance in bytes between starts of consecutive lines of the destination image.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.27 **NppStatus nppiMirror_32s_C11R** (Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 32-bit signed in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.28 **NppStatus nppiMirror_32s_C1R** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 32-bit image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.29 NppStatus nppiMirror_32s_C3IR (Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 32-bit signed in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.30 NppStatus nppiMirror_32s_C3R (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 32-bit image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.31 NppStatus nppiMirror_32s_C4IR (Npp32s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit signed in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.32 **NppStatus nppiMirror_32s_C4R** (const Npp32s * *pSrc*, int *nSrcStep*, Npp32s * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Distance in bytes between starts of consecutive lines of the destination image.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.33 **NppStatus nppiMirror_8u_AC4IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 8-bit unsigned in place image mirror not affecting alpha.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.34 **NppStatus nppiMirror_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 8-bit unsigned image mirror not affecting alpha.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Distance in bytes between starts of consecutive lines of the destination image.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.35 NppStatus nppiMirror_8u_C1IR (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 8-bit unsigned in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.36 NppStatus nppiMirror_8u_C1R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 8-bit unsigned image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.37 NppStatus nppiMirror_8u_C3IR (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 8-bit unsigned in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.38 NppStatus nppiMirror_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 8-bit unsigned image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.39 NppStatus nppiMirror_8u_C4IR (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 8-bit unsigned in place image mirror.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.78.3.40 NppStatus nppiMirror_8u_C4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 8-bit unsigned image mirror.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Distance in bytes between starts of consecutive lines of the destination image.

oROI Region-of-Interest (ROI).

flip Specifies the axis about which the image is to be mirrored.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

7.79 Affine Transforms

Utility Functions

- **NppStatus** **nppiGetAffineTransform** (**NppiRect** oSrcROI, const double aQuad[4][2], double aCoeffs[2][3])

Computes affine transform coefficients based on source ROI and destination quadrilateral.

- **NppStatus** **nppiGetAffineQuad** (**NppiRect** oSrcROI, double aQuad[4][2], const double aCoeffs[2][3])

Compute shape of transformed image.

- **NppStatus** **nppiGetAffineBound** (**NppiRect** oSrcROI, double aBound[2][2], const double aCoeffs[2][3])

Compute bounding-box of transformed image.

Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is given as a 2×3 matrix C . A pixel location (x, y) in the source image is mapped to the location (x', y') in the destination image. The destination image coordinates are computed as follows:

$$x' = c_{00} * x + c_{01} * y + c_{02} \quad y' = c_{10} * x + c_{11} * y + c_{12} \quad C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \end{bmatrix}$$

Affine transforms can be understood as a linear transformation (traditional matrix multiplication) and a shift operation. The 2×2 matrix

$$L = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix}$$

represents the linear transform portion of the affine transformation. The vector

$$v = \begin{pmatrix} c_{02} \\ c_{12} \end{pmatrix}$$

represents the post-transform shift, i.e. after the pixel location is transformed by L it is translated by v .

- **NppStatus** **nppiWarpAffine_8u_C1R** (const **Npp8u** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** *pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[2][3], int eInterpolation)

Single-channel 8-bit unsigned affine warp.

- **NppStatus** **nppiWarpAffine_8u_C3R** (const **Npp8u** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** *pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel 8-bit unsigned affine warp.

- **NppStatus** **nppiWarpAffine_8u_C4R** (const **Npp8u** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** *pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 8-bit unsigned affine warp.

- `NppStatus nppiWarpAffine_8u_AC4R` (const `Npp8u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

Four-channel 8-bit unsigned affine warp, ignoring alpha channel.

- `NppStatus nppiWarpAffine_8u_P3R` (const `Npp8u *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

Three-channel planar 8-bit unsigned affine warp.

- `NppStatus nppiWarpAffine_8u_P4R` (const `Npp8u *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

Four-channel planar 8-bit unsigned affine warp.

- `NppStatus nppiWarpAffine_16u_C1R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

Single-channel 16-bit unsigned affine warp.

- `NppStatus nppiWarpAffine_16u_C3R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

Three-channel 16-bit unsigned affine warp.

- `NppStatus nppiWarpAffine_16u_C4R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

Four-channel 16-bit unsigned affine warp.

- `NppStatus nppiWarpAffine_16u_AC4R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

Four-channel 16-bit unsigned affine warp, ignoring alpha channel.

- `NppStatus nppiWarpAffine_16u_P3R` (const `Npp16u *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

Three-channel planar 16-bit unsigned affine warp.

- `NppStatus nppiWarpAffine_16u_P4R` (const `Npp16u *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

Four-channel planar 16-bit unsigned affine warp.

- `NppStatus nppiWarpAffine_32s_C1R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

Single-channel 32-bit signed affine warp.

- `NppStatus nppiWarpAffine_32s_C3R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel 32-bit signed affine warp.

- `NppStatus nppiWarpAffine_32s_C4R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 32-bit signed affine warp.

- `NppStatus nppiWarpAffine_32s_AC4R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 32-bit signed affine warp, ignoring alpha channel.

- `NppStatus nppiWarpAffine_32s_P3R` (const `Npp32s` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel planar 32-bit signed affine warp.

- `NppStatus nppiWarpAffine_32s_P4R` (const `Npp32s` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel planar 32-bit signed affine warp.

- `NppStatus nppiWarpAffine_32f_C1R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Single-channel 32-bit floating-point affine warp.

- `NppStatus nppiWarpAffine_32f_C3R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel 32-bit floating-point affine warp.

- `NppStatus nppiWarpAffine_32f_C4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 32-bit floating-point affine warp.

- `NppStatus nppiWarpAffine_32f_AC4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 32-bit floating-point affine warp, ignoring alpha channel.

- `NppStatus nppiWarpAffine_32f_P3R` (const `Npp32f` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel planar 32-bit floating-point affine warp.

- `NppStatus nppiWarpAffine_32f_P4R` (const `Npp32f *pSrc[4]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f *pDst[4]`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel planar 32-bit floating-point affine warp.

- `NppStatus nppiWarpAffine_64f_C1R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Single-channel 64-bit floating-point affine warp.

- `NppStatus nppiWarpAffine_64f_C3R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel 64-bit floating-point affine warp.

- `NppStatus nppiWarpAffine_64f_C4R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 64-bit floating-point affine warp.

- `NppStatus nppiWarpAffine_64f_AC4R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 64-bit floating-point affine warp, ignoring alpha channel.

- `NppStatus nppiWarpAffine_64f_P3R` (const `Npp64f *aSrc[3]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *aDst[3]`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel planar 64-bit floating-point affine warp.

- `NppStatus nppiWarpAffine_64f_P4R` (const `Npp64f *aSrc[4]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *aDst[4]`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel planar 64-bit floating-point affine warp.

Backwards Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is given as a 2×3 matrix C . A pixel location (x, y) in the source image is mapped to the location (x', y') in the destination image. The destination image coordinates fulfill the following properties:

$$x = c_{00} * x' + c_{01} * y' + c_{02} \quad y = c_{10} * x' + c_{11} * y' + c_{12} \quad C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \end{bmatrix}$$

In other words, given matrix C the source image's shape is transformed to the destination image using the inverse matrix C^{-1} :

$$M = C^{-1} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \end{bmatrix} \quad x' = m_{00} * x + m_{01} * y + m_{02} \quad y' = m_{10} * x + m_{11} * y + m_{12}$$

- `NppStatus nppiWarpAffineBack_8u_C1R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Single-channel 8-bit unsigned integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_8u_C3R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel 8-bit unsigned integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_8u_C4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 8-bit unsigned integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_8u_AC4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 8-bit unsigned integer backwards affine warp, ignoring alpha channel.

- `NppStatus nppiWarpAffineBack_8u_P3R` (const `Npp8u` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel planar 8-bit unsigned integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_8u_P4R` (const `Npp8u` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel planar 8-bit unsigned integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_16u_C1R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Single-channel 16-bit unsigned integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_16u_C3R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel 16-bit unsigned integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_16u_C4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 16-bit unsigned integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_16u_AC4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 16-bit unsigned integer backwards affine warp, ignoring alpha channel.

- `NppStatus nppiWarpAffineBack_16u_P3R` (const `Npp16u` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel planar 16-bit unsigned integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_16u_P4R` (const `Npp16u` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel planar 16-bit unsigned integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_32s_C1R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Single-channel 32-bit signed integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_32s_C3R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel 32-bit signed integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_32s_C4R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 32-bit signed integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_32s_AC4R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 32-bit signed integer backwards affine warp, ignoring alpha channel.

- `NppStatus nppiWarpAffineBack_32s_P3R` (const `Npp32s` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel planar 32-bit signed integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_32s_P4R` (const `Npp32s` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel planar 32-bit signed integer backwards affine warp.

- `NppStatus nppiWarpAffineBack_32f_C1R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Single-channel 32-bit floating-point backwards affine warp.

- `NppStatus nppiWarpAffineBack_32f_C3R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel 32-bit floating-point backwards affine warp.

- `NppStatus nppiWarpAffineBack_32f_C4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 32-bit floating-point backwards affine warp.

- `NppStatus nppiWarpAffineBack_32f_AC4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel 32-bit floating-point backwards affine warp, ignoring alpha channel.

- `NppStatus nppiWarpAffineBack_32f_P3R` (const `Npp32f` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Three-channel planar 32-bit floating-point backwards affine warp.

- `NppStatus nppiWarpAffineBack_32f_P4R` (const `Npp32f` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

Four-channel planar 32-bit floating-point backwards affine warp.

Quad-Based Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is computed such that it maps a quadrilateral in source image space to a quadrilateral in destination image space.

An affine transform is fully determined by the mapping of 3 discrete points. The following primitives compute an affine transformation matrix that maps the first three corners of the source quad are mapped to the first three vertices of the destination image quad. If the fourth vertices do not match the transform, an `NPP_AFFINE_QUAD_INCORRECT_WARNING` is returned by the primitive.

- `NppStatus nppiWarpAffineQuad_8u_C1R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Single-channel 32-bit floating-point quad-based affine warp.

- `NppStatus nppiWarpAffineQuad_8u_C3R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel 8-bit unsigned integer quad-based affine warp.

- `NppStatus nppiWarpAffineQuad_8u_C4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 8-bit unsigned integer quad-based affine warp.

- `NppStatus nppiWarpAffineQuad_8u_AC4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 8-bit unsigned integer quad-based affine warp, ignoring alpha channel.

- `NppStatus nppiWarpAffineQuad_8u_P3R` (const `Npp8u` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel planar 8-bit unsigned integer quad-based affine warp.

- `NppStatus nppiWarpAffineQuad_8u_P4R` (const `Npp8u` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel planar 8-bit unsigned integer quad-based affine warp.

- `NppStatus nppiWarpAffineQuad_16u_C1R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Single-channel 16-bit unsigned integer quad-based affine warp.

- `NppStatus nppiWarpAffineQuad_16u_C3R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel 16-bit unsigned integer quad-based affine warp.

- `NppStatus nppiWarpAffineQuad_16u_C4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 16-bit unsigned integer quad-based affine warp.

- `NppStatus nppiWarpAffineQuad_16u_AC4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 16-bit unsigned integer quad-based affine warp, ignoring alpha channel.

- `NppStatus nppiWarpAffineQuad_16u_P3R` (const `Npp16u` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel planar 16-bit unsigned integer quad-based affine warp.

- `NppStatus nppiWarpAffineQuad_16u_P4R` (const `Npp16u` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel planar 16-bit unsigned integer quad-based affine warp.

- `NppStatus nppiWarpAffineQuad_32s_C1R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Single-channel 32-bit signed integer quad-based affine warp.

- `NppStatus nppiWarpAffineQuad_32s_C3R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel 32-bit signed integer quad-based affine warp.

- **NppStatus** **nppiWarpAffineQuad_32s_C4R** (const **Npp32s** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32s** *pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 32-bit signed integer quad-based affine warp.

- **NppStatus** **nppiWarpAffineQuad_32s_AC4R** (const **Npp32s** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32s** *pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 32-bit signed integer quad-based affine warp, ignoring alpha channel.

- **NppStatus** **nppiWarpAffineQuad_32s_P3R** (const **Npp32s** *pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32s** *pDst[3], int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel planar 32-bit signed integer quad-based affine warp.

- **NppStatus** **nppiWarpAffineQuad_32s_P4R** (const **Npp32s** *pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32s** *pDst[4], int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel planar 32-bit signed integer quad-based affine warp.

- **NppStatus** **nppiWarpAffineQuad_32f_C1R** (const **Npp32f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** *pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

Single-channel 32-bit floating-point quad-based affine warp.

- **NppStatus** **nppiWarpAffineQuad_32f_C3R** (const **Npp32f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** *pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel 32-bit floating-point quad-based affine warp.

- **NppStatus** **nppiWarpAffineQuad_32f_C4R** (const **Npp32f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** *pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 32-bit floating-point quad-based affine warp.

- **NppStatus** **nppiWarpAffineQuad_32f_AC4R** (const **Npp32f** *pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** *pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 32-bit floating-point quad-based affine warp, ignoring alpha channel.

- **NppStatus** **nppiWarpAffineQuad_32f_P3R** (const **Npp32f** *pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** *pDst[3], int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel planar 32-bit floating-point quad-based affine warp.

- **NppStatus** **nppiWarpAffineQuad_32f_P4R** (const **Npp32f** *pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** *pDst[4], int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel planar 32-bit floating-point quad-based affine warp.

7.79.1 Detailed Description

7.79.2 Affine Transform Error Codes

- **NPP_RECT_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- **NPP_WRONG_INTERSECTION_ROI_ERROR** Indicates an error condition if oSrcROI has no intersection with the source image
- **NPP_INTERPOLATION_ERROR** Indicates an error condition if interpolation has an illegal value
- **NPP_COEFF_ERROR** Indicates an error condition if coefficient values are invalid
- **NPP_WRONG_INTERSECTION_QUAD_WARNING** Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI

7.79.3 Function Documentation

7.79.3.1 NppStatus nppiGetAffineBound (NppiRect oSrcROI, double aBound[2][2], const double aCoeffs[2][3])

Compute bounding-box of transformed image.

The method effectively computes the bounding box (axis aligned rectangle) of the transformed source ROI (see [nppiGetAffineQuad\(\)](#)).

Parameters:

- oSrcROI** The source ROI.
- aBound** The resulting bounding box.
- aCoeffs** The affine transform coefficients.

Returns:

Error codes:

- **NPP_SIZE_ERROR** Indicates an error condition if any image dimension has zero or negative value
- **NPP_RECT_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- **NPP_COEFF_ERROR** Indicates an error condition if coefficient values are invalid

7.79.3.2 NppStatus nppiGetAffineQuad (NppiRect oSrcROI, double aQuad[4][2], const double aCoeffs[2][3])

Compute shape of transformed image.

This method computes the quadrilateral in the destination image that the source ROI is transformed into by the affine transformation expressed by the coefficients array (aCoeffs).

Parameters:

- oSrcROI** The source ROI.

aQuad The resulting destination quadrangle.

aCoeffs The affine transform coefficients.

Returns:

Error codes:

- [NPP_SIZE_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP_RECT_ERROR](#) Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- [NPP_COEFF_ERROR](#) Indicates an error condition if coefficient values are invalid

7.79.3.3 NppStatus nppiGetAffineTransform (NppiRect oSrcROI, const double aQuad[4][2], double aCoeffs[2][3])

Computes affine transform coefficients based on source ROI and destination quadrilateral.

The function computes the coefficients of an affine transformation that maps the given source ROI (axis aligned rectangle with integer coordinates) to a quadrilateral in the destination image.

An affine transform in 2D is fully determined by the mapping of just three vertices. This function's API allows for passing a complete quadrilateral effectively making the problem overdetermined. What this means in practice is, that for certain quadrilaterals it is not possible to find an affine transform that would map all four corners of the source ROI to the four vertices of that quadrilateral.

The function circumvents this problem by only looking at the first three vertices of the destination image quadrilateral to determine the affine transformation's coefficients. If the destination quadrilateral is indeed one that cannot be mapped using an affine transformation the function informs the user of this situation by returning a [NPP_AFFINE_QUAD_INCORRECT_WARNING](#).

Parameters:

oSrcROI The source ROI. This rectangle needs to be at least one pixel wide and high. If either width or height are less than one an [NPP_RECT_ERROR](#) is returned.

aQuad The destination quadrilateral.

aCoeffs The resulting affine transform coefficients.

Returns:

Error codes:

- [NPP_SIZE_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP_RECT_ERROR](#) Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- [NPP_COEFF_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP_AFFINE_QUAD_INCORRECT_WARNING](#) Indicates a warning when quad does not conform to the transform properties. Fourth vertex is ignored, internally computed coordinates are used instead

7.79.3.4 `NppStatus nppiWarpAffine_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 16-bit unsigned affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.5 `NppStatus nppiWarpAffine_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 16-bit unsigned affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.6 `NppStatus nppiWarpAffine_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 16-bit unsigned affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.7 `NppStatus nppiWarpAffine_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 16-bit unsigned affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.8 `NppStatus nppiWarpAffine_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 16-bit unsigned affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.9 `NppStatus nppiWarpAffine_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 16-bit unsigned affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.10 `NppStatus nppiWarpAffine_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.11 `NppStatus nppiWarpAffine_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 32-bit floating-point affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.12 `NppStatus nppiWarpAffine_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 32-bit floating-point affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.13 `NppStatus nppiWarpAffine_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.14 `NppStatus nppiWarpAffine_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit floating-point affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.15 `NppStatus nppiWarpAffine_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit floating-point affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.16 `NppStatus nppiWarpAffine_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit signed affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.17 `NppStatus nppiWarpAffine_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 32-bit signed affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.18 `NppStatus nppiWarpAffine_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 32-bit signed affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.19 `NppStatus nppiWarpAffine_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit signed affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.20 `NppStatus nppiWarpAffine_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit signed affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.21 `NppStatus nppiWarpAffine_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit signed affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.22 `NppStatus nppiWarpAffine_64f_AC4R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 64-bit floating-point affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.23 `NppStatus nppiWarpAffine_64f_C1R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 64-bit floating-point affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.24 `NppStatus nppiWarpAffine_64f_C3R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 64-bit floating-point affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.25 `NppStatus nppiWarpAffine_64f_C4R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 64-bit floating-point affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.26 `NppStatus nppiWarpAffine_64f_P3R (const Npp64f * aSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * aDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 64-bit floating-point affine warp.

Parameters:

aSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.27 `NppStatus nppiWarpAffine_64f_P4R (const Npp64f * aSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * aDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 64-bit floating-point affine warp.

Parameters:

aSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.28 `NppStatus nppiWarpAffine_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.29 `NppStatus nppiWarpAffine_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 8-bit unsigned affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.30 `NppStatus nppiWarpAffine_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 8-bit unsigned affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.31 `NppStatus nppiWarpAffine_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.32 `NppStatus nppiWarpAffine_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 8-bit unsigned affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.33 `NppStatus nppiWarpAffine_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 8-bit unsigned affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.34 `NppStatus nppiWarpAffineBack_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.35 `NppStatus nppiWarpAffineBack_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 16-bit unsigned integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.36 `NppStatus nppiWarpAffineBack_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 16-bit unsigned integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.37 `NppStatus nppiWarpAffineBack_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.38 `NppStatus nppiWarpAffineBack_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 16-bit unsigned integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.39 `NppStatus nppiWarpAffineBack_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.40 `NppStatus nppiWarpAffineBack_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.41 `NppStatus nppiWarpAffineBack_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 32-bit floating-point backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.42 `NppStatus nppiWarpAffineBack_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 32-bit floating-point backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.43 `NppStatus nppiWarpAffineBack_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.44 `NppStatus nppiWarpAffineBack_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit floating-point backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.45 `NppStatus nppiWarpAffineBack_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit floating-point backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.46 `NppStatus nppiWarpAffineBack_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit signed integer backwards affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.47 `NppStatus nppiWarpAffineBack_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 32-bit signed integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.48 `NppStatus nppiWarpAffineBack_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 32-bit signed integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.49 `NppStatus nppiWarpAffineBack_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit signed integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.50 `NppStatus nppiWarpAffineBack_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit signed integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.51 `NppStatus nppiWarpAffineBack_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit signed integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.52 `NppStatus nppiWarpAffineBack_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.53 `NppStatus nppiWarpAffineBack_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 8-bit unsigned integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.54 `NppStatus nppiWarpAffineBack_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 8-bit unsigned integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.55 `NppStatus nppiWarpAffineBack_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.56 `NppStatus nppiWarpAffineBack_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 8-bit unsigned integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.57 `NppStatus nppiWarpAffineBack_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 8-bit unsigned integer backwards affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Affine transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.58 `NppStatus nppiWarpAffineQuad_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.59 `NppStatus nppiWarpAffineQuad_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 16-bit unsigned integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.60 `NppStatus nppiWarpAffineQuad_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 16-bit unsigned integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.61 `NppStatus nppiWarpAffineQuad_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.62 `NppStatus nppiWarpAffineQuad_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 16-bit unsigned integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.63 `NppStatus nppiWarpAffineQuad_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 16-bit unsigned integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.64 `NppStatus nppiWarpAffineQuad_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.65 `NppStatus nppiWarpAffineQuad_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit floating-point quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.66 `NppStatus nppiWarpAffineQuad_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit floating-point quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.67 `NppStatus nppiWarpAffineQuad_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.68 `NppStatus nppiWarpAffineQuad_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit floating-point quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.69 `NppStatus nppiWarpAffineQuad_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit floating-point quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.70 `NppStatus nppiWarpAffineQuad_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.71 `NppStatus nppiWarpAffineQuad_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit signed integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.72 `NppStatus nppiWarpAffineQuad_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit signed integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.73 `NppStatus nppiWarpAffineQuad_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.74 `NppStatus nppiWarpAffineQuad_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit signed integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.75 `NppStatus nppiWarpAffineQuad_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit signed integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.76 `NppStatus nppiWarpAffineQuad_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based affine warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.77 `NppStatus nppiWarpAffineQuad_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit floating-point quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.78 `NppStatus nppiWarpAffineQuad_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 8-bit unsigned integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.79 `NppStatus nppiWarpAffineQuad_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.80 `NppStatus nppiWarpAffineQuad_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 8-bit unsigned integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.79.3.81 `NppStatus nppiWarpAffineQuad_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 8-bit unsigned integer quad-based affine warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

7.80 Perspective Transform

Utility Functions

- [NppStatus nppiGetPerspectiveTransform](#) ([NppiRect](#) oSrcROI, const double quad[4][2], double aCoeffs[3][3])

Calculates perspective transform coefficients given source rectangular ROI and its destination quadrangle projection.

- [NppStatus nppiGetPerspectiveQuad](#) ([NppiRect](#) oSrcROI, double quad[4][2], const double aCoeffs[3][3])

Calculates perspective transform projection of given source rectangular ROI.

- [NppStatus nppiGetPerspectiveBound](#) ([NppiRect](#) oSrcROI, double bound[2][2], const double aCoeffs[3][3])

Calculates bounding box of the perspective transform projection of the given source rectangular ROI.

Perspective Transform

Transforms (warps) an image based on a perspective transform.

The perspective transform is given as a 3×3 matrix C . A pixel location (x, y) in the source image is mapped to the location (x', y') in the destination image. The destination image coordinates are computed as follows:

$$x' = \frac{c_{00} * x + c_{01} * y + c_{02}}{c_{20} * x + c_{21} * y + c_{22}} \quad y' = \frac{c_{10} * x + c_{11} * y + c_{12}}{c_{20} * x + c_{21} * y + c_{22}}$$

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix}$$

- [NppStatus nppiWarpPerspective_8u_C1R](#) (const [Npp8u](#) *pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) *pDst, int nDstStep, [NppiRect](#) oDstROI, const double aCoeffs[3][3], int eInterpolation)

Single-channel 8-bit unsigned integer perspective warp.

- [NppStatus nppiWarpPerspective_8u_C3R](#) (const [Npp8u](#) *pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) *pDst, int nDstStep, [NppiRect](#) oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel 8-bit unsigned integer perspective warp.

- [NppStatus nppiWarpPerspective_8u_C4R](#) (const [Npp8u](#) *pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) *pDst, int nDstStep, [NppiRect](#) oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 8-bit unsigned integer perspective warp.

- [NppStatus nppiWarpPerspective_8u_AC4R](#) (const [Npp8u](#) *pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) *pDst, int nDstStep, [NppiRect](#) oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 8-bit unsigned integer perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspective_8u_P3R` (const `Npp8u` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel planar 8-bit unsigned integer perspective warp.

- `NppStatus nppiWarpPerspective_8u_P4R` (const `Npp8u` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel planar 8-bit unsigned integer perspective warp.

- `NppStatus nppiWarpPerspective_16u_C1R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Single-channel 16-bit unsigned integer perspective warp.

- `NppStatus nppiWarpPerspective_16u_C3R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel 16-bit unsigned integer perspective warp.

- `NppStatus nppiWarpPerspective_16u_C4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 16-bit unsigned integer perspective warp.

- `NppStatus nppiWarpPerspective_16u_AC4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 16-bit unsigned integer perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspective_16u_P3R` (const `Npp16u` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel planar 16-bit unsigned integer perspective warp.

- `NppStatus nppiWarpPerspective_16u_P4R` (const `Npp16u` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel planar 16-bit unsigned integer perspective warp.

- `NppStatus nppiWarpPerspective_32s_C1R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Single-channel 32-bit signed integer perspective warp.

- `NppStatus nppiWarpPerspective_32s_C3R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel 32-bit signed integer perspective warp.

- `NppStatus nppiWarpPerspective_32s_C4R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 32-bit signed integer perspective warp.

- `NppStatus nppiWarpPerspective_32s_AC4R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 32-bit signed integer perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspective_32s_P3R` (const `Npp32s` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel planar 32-bit signed integer perspective warp.

- `NppStatus nppiWarpPerspective_32s_P4R` (const `Npp32s` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel planar 32-bit signed integer perspective warp.

- `NppStatus nppiWarpPerspective_32f_C1R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Single-channel 32-bit floating-point perspective warp.

- `NppStatus nppiWarpPerspective_32f_C3R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel 32-bit floating-point perspective warp.

- `NppStatus nppiWarpPerspective_32f_C4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 32-bit floating-point perspective warp.

- `NppStatus nppiWarpPerspective_32f_AC4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 32-bit floating-point perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspective_32f_P3R` (const `Npp32f` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel planar 32-bit floating-point perspective warp.

- `NppStatus nppiWarpPerspective_32f_P4R` (const `Npp32f` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel planar 32-bit floating-point perspective warp.

Backwards Perspective Transform

Transforms (warps) an image based on a perspective transform.

The perspective transform is given as a 3×3 matrix C . A pixel location (x, y) in the source image is mapped to the location (x', y') in the destination image. The destination image coordinates fulfill the following properties:

$$x = \frac{c_{00} * x' + c_{01} * y' + c_{02}}{c_{20} * x' + c_{21} * y' + c_{22}} \quad y = \frac{c_{10} * x' + c_{11} * y' + c_{12}}{c_{20} * x' + c_{21} * y' + c_{22}}$$

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix}$$

In other words, given matrix C the source image's shape is transformed to the destination image using the inverse matrix C^{-1} :

$$M = C^{-1} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix} \quad x' = \frac{c_{00} * x + c_{01} * y + c_{02}}{c_{20} * x + c_{21} * y + c_{22}} \quad y' = \frac{c_{10} * x + c_{11} * y + c_{12}}{c_{20} * x + c_{21} * y + c_{22}}$$

- `NppStatus nppiWarpPerspectiveBack_8u_C1R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Single-channel 8-bit unsigned integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_8u_C3R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel 8-bit unsigned integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_8u_C4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 8-bit unsigned integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_8u_AC4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 8-bit unsigned integer backwards perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspectiveBack_8u_P3R` (const `Npp8u` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel planar 8-bit unsigned integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_8u_P4R` (const `Npp8u` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel planar 8-bit unsigned integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_16u_C1R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Single-channel 16-bit unsigned integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_16u_C3R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel 16-bit unsigned integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_16u_C4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 16-bit unsigned integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_16u_AC4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 16-bit unsigned integer backwards perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspectiveBack_16u_P3R` (const `Npp16u` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel planar 16-bit unsigned integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_16u_P4R` (const `Npp16u` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel planar 16-bit unsigned integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_32s_C1R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Single-channel 32-bit signed integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_32s_C3R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel 32-bit signed integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_32s_C4R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 32-bit signed integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_32s_AC4R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 32-bit signed integer backwards perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspectiveBack_32s_P3R` (const `Npp32s` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel planar 32-bit signed integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_32s_P4R` (const `Npp32s` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel planar 32-bit signed integer backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_32f_C1R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Single-channel 32-bit floating-point backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_32f_C3R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel 32-bit floating-point backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_32f_C4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 32-bit floating-point backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_32f_AC4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel 32-bit floating-point backwards perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspectiveBack_32f_P3R` (const `Npp32f` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel planar 32-bit floating-point backwards perspective warp.

- `NppStatus nppiWarpPerspectiveBack_32f_P4R` (const `Npp32f` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

Four-channel planar 32-bit floating-point backwards perspective warp.

Quad-Based Perspective Transform

Transforms (warps) an image based on an perspective transform.

The perspective transform is computed such that it maps a quadrilateral in source image space to a quadrilateral in destination image space.

- `NppStatus nppiWarpPerspectiveQuad_8u_C1R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Single-channel 8-bit unsigned integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_8u_C3R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel 8-bit unsigned integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_8u_C4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 8-bit unsigned integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_8u_AC4R` (const `Npp8u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 8-bit unsigned integer quad-based perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspectiveQuad_8u_P3R` (const `Npp8u` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel planar 8-bit unsigned integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_8u_P4R` (const `Npp8u` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel planar 8-bit unsigned integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_16u_C1R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Single-channel 16-bit unsigned integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_16u_C3R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel 16-bit unsigned integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_16u_C4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 16-bit unsigned integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_16u_AC4R` (const `Npp16u` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 16-bit unsigned integer quad-based perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspectiveQuad_16u_P3R` (const `Npp16u` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel planar 16-bit unsigned integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_16u_P4R` (const `Npp16u` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel planar 16-bit unsigned integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_32s_C1R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Single-channel 32-bit signed integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_32s_C3R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel 32-bit signed integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_32s_C4R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 32-bit signed integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_32s_AC4R` (const `Npp32s` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 32-bit signed integer quad-based perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspectiveQuad_32s_P3R` (const `Npp32s` *pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` *pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel planar 32-bit signed integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_32s_P4R` (const `Npp32s` *pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` *pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel planar 32-bit signed integer quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_32f_C1R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Single-channel 32-bit floating-point quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_32f_C3R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Three-channel 32-bit floating-point quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_32f_C4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 32-bit floating-point quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_32f_AC4R` (const `Npp32f` *pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f` *pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

Four-channel 32-bit floating-point quad-based perspective warp, ignoring alpha channel.

- `NppStatus nppiWarpPerspectiveQuad_32f_P3R` (const `Npp32f *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

Three-channel planar 32-bit floating-point quad-based perspective warp.

- `NppStatus nppiWarpPerspectiveQuad_32f_P4R` (const `Npp32f *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

Four-channel planar 32-bit floating-point quad-based perspective warp.

7.80.1 Detailed Description

7.80.2 Perspective Transform Error Codes

- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the `oSrcROI` and source image is less than or equal to 1
- `NPP_WRONG_INTERSECTION_ROI_ERROR` Indicates an error condition if `oSrcROI` has no intersection with the source image
- `NPP_INTERPOLATION_ERROR` Indicates an error condition if interpolation has an illegal value
- `NPP_COEFF_ERROR` Indicates an error condition if coefficient values are invalid
- `NPP_WRONG_INTERSECTION_QUAD_WARNING` Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI

7.80.3 Function Documentation

7.80.3.1 `NppStatus nppiGetPerspectiveBound (NppiRect oSrcROI, double bound[2][2], const double aCoeffs[3][3])`

Calculates bounding box of the perspective transform projection of the given source rectangular ROI.

Parameters:

oSrcROI Source ROI

bound Bounding box of the transformed source ROI

aCoeffs Perspective transform coefficients

Returns:

Error codes:

- `NPP_SIZE_ERROR` Indicates an error condition if any image dimension has zero or negative value
- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the `oSrcROI` and source image is less than or equal to 1
- `NPP_COEFF_ERROR` Indicates an error condition if coefficient values are invalid

7.80.3.2 NppStatus nppiGetPerspectiveQuad (NppiRect *oSrcROI*, double *quad*[4][2], const double *aCoeffs*[3][3])

Calculates perspective transform projection of given source rectangular ROI.

Parameters:

oSrcROI Source ROI
quad Destination quadrangle
aCoeffs Perspective transform coefficients

Returns:

Error codes:

- [NPP_SIZE_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP_RECT_ERROR](#) Indicates an error condition if width or height of the intersection of the *oSrcROI* and source image is less than or equal to 1
- [NPP_COEFF_ERROR](#) Indicates an error condition if coefficient values are invalid

7.80.3.3 NppStatus nppiGetPerspectiveTransform (NppiRect *oSrcROI*, const double *quad*[4][2], double *aCoeffs*[3][3])

Calculates perspective transform coefficients given source rectangular ROI and its destination quadrangle projection.

Parameters:

oSrcROI Source ROI
quad Destination quadrangle
aCoeffs Perspective transform coefficients

Returns:

Error codes:

- [NPP_SIZE_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP_RECT_ERROR](#) Indicates an error condition if width or height of the intersection of the *oSrcROI* and source image is less than or equal to 1
- [NPP_COEFF_ERROR](#) Indicates an error condition if coefficient values are invalid

7.80.3.4 NppStatus nppiWarpPerspective_16u_AC4R (const Npp16u * *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp16u * *pDst*, int *nDstStep*, NppiRect *oDstROI*, const double *aCoeffs*[3][3], int *eInterpolation*)

Four-channel 16-bit unsigned integer perspective warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels
nSrcStep [Source-Image Line Step](#).
oSrcROI Source ROI
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Destination ROI
aCoeffs Perspective transform coefficients
eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.5 `NppStatus nppiWarpPerspective_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 16-bit unsigned integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).
oSrcSize Size of source image in pixels
nSrcStep [Source-Image Line Step](#).
oSrcROI Source ROI
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oDstROI Destination ROI
aCoeffs Perspective transform coefficients
eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.6 `NppStatus nppiWarpPerspective_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 16-bit unsigned integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).
oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.7 `NppStatus nppiWarpPerspective_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 16-bit unsigned integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.8 `NppStatus nppiWarpPerspective_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 16-bit unsigned integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

7.80.3.9 `NppStatus nppiWarpPerspective_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer perspective warp.

Parameters:

pSrc Source-Image Pointer.

oSrcSize Size of source image in pixels

nSrcStep Source-Image Line Step.

oSrcROI Source ROI

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

7.80.3.10 `NppStatus nppiWarpPerspective_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point perspective warp, ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

oSrcSize Size of source image in pixels

nSrcStep Source-Image Line Step.

oSrcROI Source ROI

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

7.80.3.11 `NppStatus nppiWarpPerspective_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 32-bit floating-point perspective warp.

Parameters:

pSrc Source-Image Pointer.

oSrcSize Size of source image in pixels

nSrcStep Source-Image Line Step.

oSrcROI Source ROI

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

7.80.3.12 `NppStatus nppiWarpPerspective_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 32-bit floating-point perspective warp.

Parameters:

pSrc Source-Image Pointer.

oSrcSize Size of source image in pixels

nSrcStep Source-Image Line Step.

oSrcROI Source ROI

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

7.80.3.13 `NppStatus nppiWarpPerspective_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point perspective warp.

Parameters:

pSrc Source-Image Pointer.

oSrcSize Size of source image in pixels

nSrcStep Source-Image Line Step.

oSrcROI Source ROI

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

7.80.3.14 `NppStatus nppiWarpPerspective_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit floating-point perspective warp.

Parameters:

pSrc Source-Image Pointer.

oSrcSize Size of source image in pixels

nSrcStep Source-Image Line Step.

oSrcROI Source ROI

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.15 `NppStatus nppiWarpPerspective_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 32-bit floating-point perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.16 `NppStatus nppiWarpPerspective_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer perspective warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.17 `NppStatus nppiWarpPerspective_32s_C1R (const Npp32s *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s *pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 32-bit signed integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.18 `NppStatus nppiWarpPerspective_32s_C3R (const Npp32s *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s *pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 32-bit signed integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.19 `NppStatus nppiWarpPerspective_32s_C4R (const Npp32s *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s *pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.20 `NppStatus nppiWarpPerspective_32s_P3R (const Npp32s *pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s *pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit signed integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.21 `NppStatus nppiWarpPerspective_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 32-bit signed integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.22 `NppStatus nppiWarpPerspective_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 8-bit unsigned integer perspective warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.23 `NppStatus nppiWarpPerspective_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 8-bit unsigned integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.24 `NppStatus nppiWarpPerspective_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 8-bit unsigned integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.25 `NppStatus nppiWarpPerspective_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 8-bit unsigned integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.26 `NppStatus nppiWarpPerspective_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 8-bit unsigned integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.27 `NppStatus nppiWarpPerspective_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 8-bit unsigned integer perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.28 `NppStatus nppiWarpPerspectiveBack_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards perspective warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.29 `NppStatus nppiWarpPerspectiveBack_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 16-bit unsigned integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.30 `NppStatus nppiWarpPerspectiveBack_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 16-bit unsigned integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.31 `NppStatus nppiWarpPerspectiveBack_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.32 `NppStatus nppiWarpPerspectiveBack_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.33 `NppStatus nppiWarpPerspectiveBack_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.34 `NppStatus nppiWarpPerspectiveBack_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards perspective warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.35 `NppStatus nppiWarpPerspectiveBack_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 32-bit floating-point backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.36 `NppStatus nppiWarpPerspectiveBack_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 32-bit floating-point backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.37 `NppStatus nppiWarpPerspectiveBack_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.38 `NppStatus nppiWarpPerspectiveBack_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit floating-point backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.39 `NppStatus nppiWarpPerspectiveBack_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 32-bit floating-point backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.40 `NppStatus nppiWarpPerspectiveBack_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer backwards perspective warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.41 `NppStatus nppiWarpPerspectiveBack_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 32-bit signed integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.42 `NppStatus nppiWarpPerspectiveBack_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 32-bit signed integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.43 `NppStatus nppiWarpPerspectiveBack_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.44 `NppStatus nppiWarpPerspectiveBack_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit signed integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.45 `NppStatus nppiWarpPerspectiveBack_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 32-bit signed integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.46 `NppStatus nppiWarpPerspectiveBack_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards perspective warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.47 `NppStatus nppiWarpPerspectiveBack_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 8-bit unsigned integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.48 `NppStatus nppiWarpPerspectiveBack_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 8-bit unsigned integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.49 `NppStatus nppiWarpPerspectiveBack_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.50 `NppStatus nppiWarpPerspectiveBack_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 8-bit unsigned integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.51 `NppStatus nppiWarpPerspectiveBack_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 8-bit unsigned integer backwards perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aCoeffs Perspective transform coefficients

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.52 `NppStatus nppiWarpPerspectiveQuad_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based perspective warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.53 `NppStatus nppiWarpPerspectiveQuad_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 16-bit unsigned integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.54 `NppStatus nppiWarpPerspectiveQuad_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 16-bit unsigned integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.55 `NppStatus nppiWarpPerspectiveQuad_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.56 `NppStatus nppiWarpPerspectiveQuad_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 16-bit unsigned integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.57 `NppStatus nppiWarpPerspectiveQuad_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 16-bit unsigned integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.58 `NppStatus nppiWarpPerspectiveQuad_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based perspective warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.59 `NppStatus nppiWarpPerspectiveQuad_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit floating-point quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.60 `NppStatus nppiWarpPerspectiveQuad_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit floating-point quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.61 `NppStatus nppiWarpPerspectiveQuad_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.62 `NppStatus nppiWarpPerspectiveQuad_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit floating-point quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.63 `NppStatus nppiWarpPerspectiveQuad_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit floating-point quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.64 `NppStatus nppiWarpPerspectiveQuad_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based perspective warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.65 `NppStatus nppiWarpPerspectiveQuad_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit signed integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.66 `NppStatus nppiWarpPerspectiveQuad_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit signed integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.67 `NppStatus nppiWarpPerspectiveQuad_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.68 `NppStatus nppiWarpPerspectiveQuad_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit signed integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.69 `NppStatus nppiWarpPerspectiveQuad_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit signed integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.70 `NppStatus nppiWarpPerspectiveQuad_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based perspective warp, ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.71 `NppStatus nppiWarpPerspectiveQuad_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 8-bit unsigned integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.72 `NppStatus nppiWarpPerspectiveQuad_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 8-bit unsigned integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.73 `NppStatus nppiWarpPerspectiveQuad_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.74 `NppStatus nppiWarpPerspectiveQuad_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 8-bit unsigned integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.80.3.75 `NppStatus nppiWarpPerspectiveQuad_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 8-bit unsigned integer quad-based perspective warp.

Parameters:

pSrc [Source-Image Pointer](#).

oSrcSize Size of source image in pixels

nSrcStep [Source-Image Line Step](#).

oSrcROI Source ROI

aSrcQuad Source quad.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oDstROI Destination ROI

aDstQuad Destination quad.

eInterpolation Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

7.81 Linear Transforms

Linear image transformations.

Modules

- [Fourier Transforms](#)

7.81.1 Detailed Description

Linear image transformations.

7.82 Fourier Transforms

Functions

- **NppStatus nppiMagnitude_32fc32f_C1R** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
32-bit floating point complex to 32-bit floating point magnitude.
- **NppStatus nppiMagnitudeSqr_32fc32f_C1R** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
32-bit floating point complex to 32-bit floating point squared magnitude.

7.82.1 Function Documentation

7.82.1.1 **NppStatus nppiMagnitude_32fc32f_C1R** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)

32-bit floating point complex to 32-bit floating point magnitude.

Converts complex-number pixel image to single channel image computing the result pixels as the magnitude of the complex values.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.82.1.2 **NppStatus nppiMagnitudeSqr_32fc32f_C1R** (const **Npp32fc** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)

32-bit floating point complex to 32-bit floating point squared magnitude.

Converts complex-number pixel image to single channel image computing the result pixels as the squared magnitude of the complex values.

The squared magnitude is an intermediate result of magnitude computation and can thus be computed faster than actual magnitude. If magnitudes are required for sorting/comparing only, using this function instead of `nppiMagnitude_32fc32f_C1R` can be a worthwhile performance optimization.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.83 Morphological Operations

Morphological image operations.

Modules

- [Dilation](#)

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask.

- [Dilation with border control](#)

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask.

- [Dilate3x3](#)

Dilation using a 3x3 mask with the anchor at its center pixel.

- [Dilate3x3Border](#)

Dilation using a 3x3 mask with the anchor at its center pixel with border control.

- [Erode](#)

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask.

- [Erosion with border control](#)

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask.

- [Erode3x3](#)

Erosion using a 3x3 mask with the anchor at its center pixel.

- [Erode3x3Border](#)

Erosion using a 3x3 mask with the anchor at its center pixel with border control.

7.83.1 Detailed Description

Morphological image operations.

Morphological operations are classified as [Neighborhood Operations](#).

7.84 Dilation

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask.

Functions

- **NppStatus nppiDilate_8u_C1R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Single-channel 8-bit unsigned integer dilation.
- **NppStatus nppiDilate_8u_C3R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Three-channel 8-bit unsigned integer dilation.
- **NppStatus nppiDilate_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Four-channel 8-bit unsigned integer dilation.
- **NppStatus nppiDilate_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Four-channel 8-bit unsigned integer dilation, ignoring alpha-channel.
- **NppStatus nppiDilate_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Single-channel 16-bit unsigned integer dilation.
- **NppStatus nppiDilate_16u_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Three-channel 16-bit unsigned integer dilation.
- **NppStatus nppiDilate_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Four-channel 16-bit unsigned integer dilation.
- **NppStatus nppiDilate_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Four-channel 16-bit unsigned integer dilation, ignoring alpha-channel.
- **NppStatus nppiDilate_32f_C1R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Single-channel 32-bit floating-point dilation.
- **NppStatus nppiDilate_32f_C3R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Three-channel 32-bit floating-point dilation.
- **NppStatus nppiDilate_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Four-channel 32-bit floating-point dilation.

- **NppStatus nppiDilate_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

Four-channel 32-bit floating-point dilation, ignoring alpha-channel.

7.84.1 Detailed Description

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask.

Pixels whose corresponding mask values are zero do not participate in the maximum search.

It is the user's responsibility to avoid [Sampling Beyond Image Boundaries](#).

7.84.2 Function Documentation

- 7.84.2.1 NppStatus nppiDilate_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

Four-channel 16-bit unsigned integer dilation, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.84.2.2 NppStatus nppiDilate_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

Single-channel 16-bit unsigned integer dilation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.84.2.3 `NppStatus nppiDilate_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 16-bit unsigned integer dilation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.84.2.4 `NppStatus nppiDilate_16u_C4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 16-bit unsigned integer dilation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.84.2.5 `NppStatus nppiDilate_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 32-bit floating-point dilation, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.84.2.6 `NppStatus nppiDilate_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Single-channel 32-bit floating-point dilation.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.84.2.7 `NppStatus nppiDilate_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 32-bit floating-point dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pMask Pointer to the start address of the mask array
oMaskSize Width and Height mask array.
oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.84.2.8 `NppStatus nppiDilate_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 32-bit floating-point dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pMask Pointer to the start address of the mask array
oMaskSize Width and Height mask array.
oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.84.2.9 `NppStatus nppiDilate_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 8-bit unsigned integer dilation, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.84.2.10 `NppStatus nppiDilate_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Single-channel 8-bit unsigned integer dilation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.84.2.11 `NppStatus nppiDilate_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 8-bit unsigned integer dilation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.84.2.12 `NppStatus nppiDilate_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 8-bit unsigned integer dilation.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85 Dilation with border control

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask.

Functions

- `NppStatus nppiDilateBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single-channel 8-bit unsigned integer dilation with border control.

- `NppStatus nppiDilateBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three-channel 8-bit unsigned integer dilation with border control.

- `NppStatus nppiDilateBorder_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four-channel 8-bit unsigned integer dilation with border control.

- `NppStatus nppiDilateBorder_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four-channel 8-bit unsigned integer dilation with border control, ignoring alpha-channel.

- `NppStatus nppiDilateBorder_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single-channel 16-bit unsigned integer dilation with border control.

- `NppStatus nppiDilateBorder_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three-channel 16-bit unsigned integer dilation with border control.

- `NppStatus nppiDilateBorder_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four-channel 16-bit unsigned integer dilation with border control.

- `NppStatus nppiDilateBorder_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four-channel 16-bit unsigned integer dilation with border control, ignoring alpha-channel.

- `NppStatus nppiDilateBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single-channel 32-bit floating-point dilation with border control.

- **NppStatus nppiDilateBorder_32f_C3R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Three-channel 32-bit floating-point dilation with border control.

- **NppStatus nppiDilateBorder_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four-channel 32-bit floating-point dilation with border control.

- **NppStatus nppiDilateBorder_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four-channel 32-bit floating-point dilation with border control, ignoring alpha-channel.

7.85.1 Detailed Description

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask.

Pixels whose corresponding mask values are zero do not participate in the maximum search.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

7.85.2 Function Documentation

- 7.85.2.1 NppStatus nppiDilateBorder_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four-channel 16-bit unsigned integer dilation with border control, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85.2.2 `NppStatus nppiDilateBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single-channel 16-bit unsigned integer dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85.2.3 `NppStatus nppiDilateBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three-channel 16-bit unsigned integer dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85.2.4 NppStatus nppiDilateBorder_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Four-channel 16-bit unsigned integer dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85.2.5 NppStatus nppiDilateBorder_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Four-channel 32-bit floating-point dilation with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85.2.6 NppStatus nppiDilateBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Single-channel 32-bit floating-point dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85.2.7 NppStatus nppiDilateBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Three-channel 32-bit floating-point dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85.2.8 `NppStatus nppiDilateBorder_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four-channel 32-bit floating-point dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85.2.9 `NppStatus nppiDilateBorder_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four-channel 8-bit unsigned integer dilation with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85.2.10 `NppStatus nppiDilateBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single-channel 8-bit unsigned integer dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85.2.11 `NppStatus nppiDilateBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three-channel 8-bit unsigned integer dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pMask Pointer to the start address of the mask array
oMaskSize Width and Height mask array.
oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.85.2.12 `NppStatus nppiDilateBorder_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four-channel 8-bit unsigned integer dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pMask Pointer to the start address of the mask array
oMaskSize Width and Height mask array.
oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.86 Dilate3x3

Dilation using a 3x3 mask with the anchor at its center pixel.

Functions

- **NppStatus** **nppiDilate3x3_8u_C1R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Single-channel 8-bit unsigned integer 3x3 dilation.
- **NppStatus** **nppiDilate3x3_8u_C3R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Three-channel 8-bit unsigned integer 3x3 dilation.
- **NppStatus** **nppiDilate3x3_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 8-bit unsigned integer 3x3 dilation.
- **NppStatus** **nppiDilate3x3_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 8-bit unsigned integer 3x3 dilation, ignoring alpha-channel.
- **NppStatus** **nppiDilate3x3_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Single-channel 16-bit unsigned integer 3x3 dilation.
- **NppStatus** **nppiDilate3x3_16u_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Three-channel 16-bit unsigned integer 3x3 dilation.
- **NppStatus** **nppiDilate3x3_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 16-bit unsigned integer 3x3 dilation.
- **NppStatus** **nppiDilate3x3_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 16-bit unsigned integer 3x3 dilation, ignoring alpha-channel.
- **NppStatus** **nppiDilate3x3_32f_C1R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Single-channel 32-bit floating-point 3x3 dilation.
- **NppStatus** **nppiDilate3x3_32f_C3R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Three-channel 32-bit floating-point 3x3 dilation.
- **NppStatus** **nppiDilate3x3_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 32-bit floating-point 3x3 dilation.

- **NppStatus nppiDilate3x3_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 32-bit floating-point 3x3 dilation, ignoring alpha-channel.
- **NppStatus nppiDilate3x3_64f_C1R** (const **Npp64f** *pSrc, **Npp32s** nSrcStep, **Npp64f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Single-channel 64-bit floating-point 3x3 dilation.

7.86.1 Detailed Description

Dilation using a 3x3 mask with the anchor at its center pixel.

It is the user's responsibility to avoid [Sampling Beyond Image Boundaries](#).

7.86.2 Function Documentation

7.86.2.1 **NppStatus nppiDilate3x3_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four-channel 16-bit unsigned integer 3x3 dilation, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.86.2.2 **NppStatus nppiDilate3x3_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)

Single-channel 16-bit unsigned integer 3x3 dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.86.2.3 NppStatus nppiDilate3x3_16u_C3R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three-channel 16-bit unsigned integer 3x3 dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.86.2.4 NppStatus nppiDilate3x3_16u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit unsigned integer 3x3 dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.86.2.5 NppStatus nppiDilate3x3_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating-point 3x3 dilation, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.86.2.6 NppStatus nppiDilate3x3_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single-channel 32-bit floating-point 3x3 dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.86.2.7 NppStatus nppiDilate3x3_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit floating-point 3x3 dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.86.2.8 NppStatus nppiDilate3x3_32f_C4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating-point 3x3 dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.86.2.9 **NppStatus nppiDilate3x3_64f_C1R** (const Npp64f * *pSrc*, Npp32s *nSrcStep*, Npp64f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single-channel 64-bit floating-point 3x3 dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.86.2.10 **NppStatus nppiDilate3x3_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 8-bit unsigned integer 3x3 dilation, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.86.2.11 **NppStatus nppiDilate3x3_8u_C1R** (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single-channel 8-bit unsigned integer 3x3 dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.86.2.12 `NppStatus nppiDilate3x3_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Three-channel 8-bit unsigned integer 3x3 dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.86.2.13 `NppStatus nppiDilate3x3_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 8-bit unsigned integer 3x3 dilation.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.87 Dilate3x3Border

Dilation using a 3x3 mask with the anchor at its center pixel with border control.

Functions

- **NppStatus** **nppiDilate3x3Border_8u_C1R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)
Single-channel 8-bit unsigned integer 3x3 dilation with border control.
- **NppStatus** **nppiDilate3x3Border_8u_C3R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)
Three-channel 8-bit unsigned integer 3x3 dilation with border control.
- **NppStatus** **nppiDilate3x3Border_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)
Four-channel 8-bit unsigned integer 3x3 dilation with border control.
- **NppStatus** **nppiDilate3x3Border_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)
Four-channel 8-bit unsigned integer 3x3 dilation with border control, ignoring alpha-channel.
- **NppStatus** **nppiDilate3x3Border_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)
Single-channel 16-bit unsigned integer 3x3 dilation with border control.
- **NppStatus** **nppiDilate3x3Border_16u_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)
Three-channel 16-bit unsigned integer 3x3 dilation with border control.
- **NppStatus** **nppiDilate3x3Border_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)
Four-channel 16-bit unsigned integer 3x3 dilation with border control.
- **NppStatus** **nppiDilate3x3Border_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)
Four-channel 16-bit unsigned integer 3x3 dilation with border control, ignoring alpha-channel.
- **NppStatus** **nppiDilate3x3Border_32f_C1R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiBorderType** eBorderType)
Single-channel 32-bit floating-point 3x3 dilation with border control.

- **NppStatus nppiDilate3x3Border_32f_C3R** (const [Npp32f](#) *pSrc, [Npp32s](#) nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp32f](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiBorderType](#) eBorderType)

Three-channel 32-bit floating-point 3x3 dilation with border control.

- **NppStatus nppiDilate3x3Border_32f_C4R** (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiBorderType](#) eBorderType)

Four-channel 32-bit floating-point 3x3 dilation with border control.

- **NppStatus nppiDilate3x3Border_32f_AC4R** (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiBorderType](#) eBorderType)

Four-channel 32-bit floating-point 3x3 dilation with border control, ignoring alpha-channel.

7.87.1 Detailed Description

Dilation using a 3x3 mask with the anchor at its center pixel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

7.87.2 Function Documentation

- 7.87.2.1 NppStatus nppiDilate3x3Border_16u_AC4R** (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp16u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiBorderType](#) eBorderType)

Four-channel 16-bit unsigned integer 3x3 dilation with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.87.2.2 NppStatus nppiDilate3x3Border_16u_C1R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiBorderType *eBorderType*)

Single-channel 16-bit unsigned integer 3x3 dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset Source image starting point relative to *pSrc*.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.87.2.3 NppStatus nppiDilate3x3Border_16u_C3R (const Npp16u * *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiBorderType *eBorderType*)

Three-channel 16-bit unsigned integer 3x3 dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to *pSrc*.

oSrcOffset Source image starting point relative to *pSrc*.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.87.2.4 NppStatus nppiDilate3x3Border_16u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiBorderType *eBorderType*)

Four-channel 16-bit unsigned integer 3x3 dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.87.2.5 `NppStatus nppiDilate3x3Border_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four-channel 32-bit floating-point 3x3 dilation with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.87.2.6 `NppStatus nppiDilate3x3Border_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single-channel 32-bit floating-point 3x3 dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.87.2.7 `NppStatus nppiDilate3x3Border_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three-channel 32-bit floating-point 3x3 dilation with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.87.2.8 `NppStatus nppiDilate3x3Border_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four-channel 32-bit floating-point 3x3 dilation with border control.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.87.2.9 `NppStatus nppiDilate3x3Border_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four-channel 8-bit unsigned integer 3x3 dilation with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.87.2.10 `NppStatus nppiDilate3x3Border_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single-channel 8-bit unsigned integer 3x3 dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.87.2.11 `NppStatus nppiDilate3x3Border_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three-channel 8-bit unsigned integer 3x3 dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.87.2.12 `NppStatus nppiDilate3x3Border_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four-channel 8-bit unsigned integer 3x3 dilation with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.88 Erode

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask.

Functions

- **NppStatus nppiErode_8u_C1R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Single-channel 8-bit unsigned integer erosion.
- **NppStatus nppiErode_8u_C3R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Three-channel 8-bit unsigned integer erosion.
- **NppStatus nppiErode_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Four-channel 8-bit unsigned integer erosion.
- **NppStatus nppiErode_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Four-channel 8-bit unsigned integer erosion, ignoring alpha-channel.
- **NppStatus nppiErode_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Single-channel 16-bit unsigned integer erosion.
- **NppStatus nppiErode_16u_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Three-channel 16-bit unsigned integer erosion.
- **NppStatus nppiErode_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Four-channel 16-bit unsigned integer erosion.
- **NppStatus nppiErode_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Four-channel 16-bit unsigned integer erosion, ignoring alpha-channel.
- **NppStatus nppiErode_32f_C1R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Single-channel 32-bit floating-point erosion.
- **NppStatus nppiErode_32f_C3R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Three-channel 32-bit floating-point erosion.
- **NppStatus nppiErode_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)
Four-channel 32-bit floating-point erosion.

- **NppStatus nppiErode_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

Four-channel 32-bit floating-point erosion, ignoring alpha-channel.

7.88.1 Detailed Description

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask.

Pixels whose corresponding mask values are zero do not participate in the maximum search.

It is the user's responsibility to avoid [Sampling Beyond Image Boundaries](#).

7.88.2 Function Documentation

- 7.88.2.1 NppStatus nppiErode_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

Four-channel 16-bit unsigned integer erosion, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.88.2.2 NppStatus nppiErode_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

Single-channel 16-bit unsigned integer erosion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.88.2.3 `NppStatus nppiErode_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 16-bit unsigned integer erosion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.88.2.4 `NppStatus nppiErode_16u_C4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 16-bit unsigned integer erosion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.88.2.5 `NppStatus nppiErode_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 32-bit floating-point erosion, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.88.2.6 `NppStatus nppiErode_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Single-channel 32-bit floating-point erosion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.88.2.7 `NppStatus nppiErode_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 32-bit floating-point erosion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pMask Pointer to the start address of the mask array
oMaskSize Width and Height mask array.
oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.88.2.8 `NppStatus nppiErode_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 32-bit floating-point erosion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pMask Pointer to the start address of the mask array
oMaskSize Width and Height mask array.
oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.88.2.9 `NppStatus nppiErode_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 8-bit unsigned integer erosion, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.88.2.10 `NppStatus nppiErode_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Single-channel 8-bit unsigned integer erosion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.88.2.11 `NppStatus nppiErode_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 8-bit unsigned integer erosion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.88.2.12 `NppStatus nppiErode_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 8-bit unsigned integer erosion.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89 Erosion with border control

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask.

Functions

- `NppStatus nppiErodeBorder_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single-channel 8-bit unsigned integer erosion with border control.

- `NppStatus nppiErodeBorder_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three-channel 8-bit unsigned integer erosion with border control.

- `NppStatus nppiErodeBorder_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four-channel 8-bit unsigned integer erosion with border control.

- `NppStatus nppiErodeBorder_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four-channel 8-bit unsigned integer erosion with border control, ignoring alpha-channel.

- `NppStatus nppiErodeBorder_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single-channel 16-bit unsigned integer erosion with border control.

- `NppStatus nppiErodeBorder_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Three-channel 16-bit unsigned integer erosion with border control.

- `NppStatus nppiErodeBorder_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four-channel 16-bit unsigned integer erosion with border control.

- `NppStatus nppiErodeBorder_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Four-channel 16-bit unsigned integer erosion with border control, ignoring alpha-channel.

- `NppStatus nppiErodeBorder_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp8u` *pMask, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

Single-channel 32-bit floating-point erosion with border control.

- **NppStatus nppiErodeBorder_32f_C3R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Three-channel 32-bit floating-point erosion with border control.

- **NppStatus nppiErodeBorder_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four-channel 32-bit floating-point erosion with border control.

- **NppStatus nppiErodeBorder_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four-channel 32-bit floating-point erosion with border control, ignoring alpha-channel.

7.89.1 Detailed Description

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask.

Pixels whose corresponding mask values are zero do not participate in the minimum search.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

7.89.2 Function Documentation

- 7.89.2.1 NppStatus nppiErodeBorder_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** *pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor, **NppiBorderType** eBorderType)

Four-channel 16-bit unsigned integer erosion with border control, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89.2.2 `NppStatus nppiErodeBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single-channel 16-bit unsigned integer erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89.2.3 `NppStatus nppiErodeBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three-channel 16-bit unsigned integer erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89.2.4 NppStatus nppiErodeBorder_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Four-channel 16-bit unsigned integer erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89.2.5 NppStatus nppiErodeBorder_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Four-channel 32-bit floating-point erosion with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89.2.6 NppStatus nppiErodeBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Single-channel 32-bit floating-point erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89.2.7 NppStatus nppiErodeBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Three-channel 32-bit floating-point erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89.2.8 NppStatus nppiErodeBorder_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Four-channel 32-bit floating-point erosion with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89.2.9 NppStatus nppiErodeBorder_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)

Four-channel 8-bit unsigned integer erosion with border control, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89.2.10 `NppStatus nppiErodeBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single-channel 8-bit unsigned integer erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMask Pointer to the start address of the mask array

oMaskSize Width and Height mask array.

oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89.2.11 `NppStatus nppiErodeBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three-channel 8-bit unsigned integer erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pMask Pointer to the start address of the mask array
oMaskSize Width and Height mask array.
oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.89.2.12 `NppStatus nppiErodeBorder_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four-channel 8-bit unsigned integer erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pMask Pointer to the start address of the mask array
oMaskSize Width and Height mask array.
oAnchor X and Y offsets of the mask origin frame of reference w.r.t the source pixel.
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.90 Erode3x3

Erosion using a 3x3 mask with the anchor at its center pixel.

Functions

- **NppStatus** **nppiErode3x3_8u_C1R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Single-channel 8-bit unsigned integer 3x3 erosion.
- **NppStatus** **nppiErode3x3_8u_C3R** (const **Npp8u** *pSrc, **Npp32s** nSrcStep, **Npp8u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Three-channel 8-bit unsigned integer 3x3 erosion.
- **NppStatus** **nppiErode3x3_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 8-bit unsigned integer 3x3 erosion.
- **NppStatus** **nppiErode3x3_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 8-bit unsigned integer 3x3 erosion, ignoring alpha-channel.
- **NppStatus** **nppiErode3x3_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Single-channel 16-bit unsigned integer 3x3 erosion.
- **NppStatus** **nppiErode3x3_16u_C3R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Three-channel 16-bit unsigned integer 3x3 erosion.
- **NppStatus** **nppiErode3x3_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 16-bit unsigned integer 3x3 erosion.
- **NppStatus** **nppiErode3x3_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 16-bit unsigned integer 3x3 erosion, ignoring alpha-channel.
- **NppStatus** **nppiErode3x3_32f_C1R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Single-channel 32-bit floating-point 3x3 erosion.
- **NppStatus** **nppiErode3x3_32f_C3R** (const **Npp32f** *pSrc, **Npp32s** nSrcStep, **Npp32f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Three-channel 32-bit floating-point 3x3 erosion.
- **NppStatus** **nppiErode3x3_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 32-bit floating-point 3x3 erosion.

- **NppStatus nppiErode3x3_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI)
Four-channel 32-bit floating-point 3x3 erosion, ignoring alpha-channel.
- **NppStatus nppiErode3x3_64f_C1R** (const **Npp64f** *pSrc, **Npp32s** nSrcStep, **Npp64f** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)
Single-channel 64-bit floating-point 3x3 erosion.

7.90.1 Detailed Description

Erosion using a 3x3 mask with the anchor at its center pixel.

It is the user's responsibility to avoid [Sampling Beyond Image Boundaries](#).

7.90.2 Function Documentation

7.90.2.1 **NppStatus nppiErode3x3_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI)

Four-channel 16-bit unsigned integer 3x3 erosion, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.90.2.2 **NppStatus nppiErode3x3_16u_C1R** (const **Npp16u** *pSrc, **Npp32s** nSrcStep, **Npp16u** *pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)

Single-channel 16-bit unsigned integer 3x3 erosion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.90.2.3 **NppStatus nppiErode3x3_16u_C3R** (const Npp16u * *pSrc*, Npp32s *nSrcStep*, Npp16u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three-channel 16-bit unsigned integer 3x3 erosion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.90.2.4 **NppStatus nppiErode3x3_16u_C4R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit unsigned integer 3x3 erosion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.90.2.5 **NppStatus nppiErode3x3_32f_AC4R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating-point 3x3 erosion, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.90.2.6 NppStatus nppiErode3x3_32f_C1R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single-channel 32-bit floating-point 3x3 erosion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.90.2.7 NppStatus nppiErode3x3_32f_C3R (const Npp32f * *pSrc*, Npp32s *nSrcStep*, Npp32f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit floating-point 3x3 erosion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.90.2.8 NppStatus nppiErode3x3_32f_C4R (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating-point 3x3 erosion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.90.2.9 NppStatus nppiErode3x3_64f_C1R (const Npp64f * *pSrc*, Npp32s *nSrcStep*, Npp64f * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single-channel 64-bit floating-point 3x3 erosion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.90.2.10 NppStatus nppiErode3x3_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 8-bit unsigned integer 3x3 erosion, ignoring alpha-channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.90.2.11 NppStatus nppiErode3x3_8u_C1R (const Npp8u * *pSrc*, Npp32s *nSrcStep*, Npp8u * *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single-channel 8-bit unsigned integer 3x3 erosion.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.90.2.12 `NppStatus nppiErode3x3_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Three-channel 8-bit unsigned integer 3x3 erosion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.90.2.13 `NppStatus nppiErode3x3_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 8-bit unsigned integer 3x3 erosion.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.91 Erode3x3Border

Erosion using a 3x3 mask with the anchor at its center pixel with border control.

Functions

- `NppStatus nppiErode3x3Border_8u_C1R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)
Single-channel 8-bit unsigned integer 3x3 erosion with border control.
- `NppStatus nppiErode3x3Border_8u_C3R` (const `Npp8u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)
Three-channel 8-bit unsigned integer 3x3 erosion with border control.
- `NppStatus nppiErode3x3Border_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)
Four-channel 8-bit unsigned integer 3x3 erosion with border control.
- `NppStatus nppiErode3x3Border_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)
Four-channel 8-bit unsigned integer 3x3 erosion with border control, ignoring alpha-channel.
- `NppStatus nppiErode3x3Border_16u_C1R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)
Single-channel 16-bit unsigned integer 3x3 erosion with border control.
- `NppStatus nppiErode3x3Border_16u_C3R` (const `Npp16u` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)
Three-channel 16-bit unsigned integer 3x3 erosion with border control.
- `NppStatus nppiErode3x3Border_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)
Four-channel 16-bit unsigned integer 3x3 erosion with border control.
- `NppStatus nppiErode3x3Border_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)
Four-channel 16-bit unsigned integer 3x3 erosion with border control, ignoring alpha-channel.
- `NppStatus nppiErode3x3Border_32f_C1R` (const `Npp32f` *pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` *pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)
Single-channel 32-bit floating-point 3x3 erosion with border control.

- **NppStatus nppiErode3x3Border_32f_C3R** (const [Npp32f](#) *pSrc, [Npp32s](#) nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp32f](#) *pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, [NppiBorderType](#) eBorderType)

Three-channel 32-bit floating-point 3x3 erosion with border control.

- **NppStatus nppiErode3x3Border_32f_C4R** (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiBorderType](#) eBorderType)

Four-channel 32-bit floating-point 3x3 erosion with border control.

- **NppStatus nppiErode3x3Border_32f_AC4R** (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiBorderType](#) eBorderType)

Four-channel 32-bit floating-point 3x3 erosion with border control, ignoring alpha-channel.

7.91.1 Detailed Description

Erosion using a 3x3 mask with the anchor at its center pixel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

7.91.2 Function Documentation

- 7.91.2.1 NppStatus nppiErode3x3Border_16u_AC4R** (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp16u](#) *pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiBorderType](#) eBorderType)

Four-channel 16-bit unsigned integer 3x3 erosion with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.91.2.2 NppStatus nppiErode3x3Border_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)

Single-channel 16-bit unsigned integer 3x3 erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.91.2.3 NppStatus nppiErode3x3Border_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)

Three-channel 16-bit unsigned integer 3x3 erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.91.2.4 NppStatus nppiErode3x3Border_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)

Four-channel 16-bit unsigned integer 3x3 erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.91.2.5 `NppStatus nppiErode3x3Border_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four-channel 32-bit floating-point 3x3 erosion with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.91.2.6 `NppStatus nppiErode3x3Border_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single-channel 32-bit floating-point 3x3 erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.91.2.7 `NppStatus nppiErode3x3Border_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three-channel 32-bit floating-point 3x3 erosion with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.91.2.8 `NppStatus nppiErode3x3Border_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four-channel 32-bit floating-point 3x3 erosion with border control.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.91.2.9 `NppStatus nppiErode3x3Border_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four-channel 8-bit unsigned integer 3x3 erosion with border control, ignoring alpha-channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.91.2.10 `NppStatus nppiErode3x3Border_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single-channel 8-bit unsigned integer 3x3 erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcSize Source image width and height in pixels relative to pSrc.

oSrcOffset Source image starting point relative to pSrc.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.91.2.11 `NppStatus nppiErode3x3Border_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three-channel 8-bit unsigned integer 3x3 erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.91.2.12 `NppStatus nppiErode3x3Border_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four-channel 8-bit unsigned integer 3x3 erosion with border control.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcSize Source image width and height in pixels relative to pSrc.
oSrcOffset Source image starting point relative to pSrc.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eBorderType The border type operation to be applied at source image border boundaries.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.92 Statistical Operations

Primitives for computing the statistical properties of an image.

Modules

- [Sum](#)

Primitives for computing the sum of all the pixel values in an image.

- [Min](#)

Primitives for computing the minimal pixel value of an image.

- [MinIndx](#)

Primitives for computing the minimal value and its indices (X and Y coordinates) of an image.

- [Max](#)

Primitives for computing the maximal pixel value of an image.

- [MaxIndx](#)

Primitives for computing the maximal value and its indices (X and Y coordinates) of an image.

- [MinMax](#)

Primitives for computing both the minimal and the maximal values of an image.

- [MinMaxIndx](#)

Primitives for computing the minimal and the maximal values with their indices (X and Y coordinates) of an image.

- [Mean](#)

Primitives for computing the arithmetic mean of all the pixel values in an image.

- [Mean_StdDev](#)

Primitives for computing both the arithmetic mean and the standard deviation of an image.

- [Image Norms](#)

Primitives for computing the norms of an image, the norms of difference, and the relative errors of two images.

- [DotProd](#)

Primitives for computing the dot product of two images.

- [CountInRange.](#)

Primitives for computing the amount of pixels that fall into the specified intensity range.

- [MaxEvery](#)

Primitives for computing the maximal value of the pixel pair from two images.

- [MinEvery](#)

Primitives for computing the minimal value of the pixel pair from two images.

- [Integral](#)

Primitives for computing the integral image of a given image.

- [SqrIntegral](#)

Primitives for computing both the integral and the squared integral images of a given image.

- [RectStdDev](#)

Primitives for computing the standard deviation of the integral images.

- [HistogramEven](#)

Primitives for computing the histogram of an image with evenly distributed bins.

- [HistogramRange](#)

Primitives for computing the histogram of an image within specified ranges.

- [Image Proximity](#)

Primitives for computing the proximity measure between a source image and a template image.

- [Image Quality Index](#)

Primitives for computing the image quality index of two images.

- [MaximumError](#)

Primitives for computing the maximum error between two images.

- [AverageError](#)

Primitives for computing the average error between two images.

- [MaximumRelativeError](#)

Primitives for computing the maximum relative error between two images.

- [AverageRelativeError](#)

Primitives for computing the average relative error between two images.

NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_8u_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size for [nppiMaximumError_8u_C1R](#).

- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_8s_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size for [nppiMaximumError_8s_C1R](#).

- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_16u_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size for `nppiMaximumError_16u_C1R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_16s_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_16s_C1R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_16sc_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_16sc_C1R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_32u_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_32u_C1R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_32s_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_32s_C1R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_32sc_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_32sc_C1R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_32f_C1R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_32fc_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_32fc_C1R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_64f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_64f_C1R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_8u_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_8u_C2R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_8s_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_8s_C2R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_16u_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_16u_C2R`.

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_16s_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size for `nppiMaximumError_16s_C2R`.

- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_16sc_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_16sc_C2R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32u_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_32u_C2R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32s_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_32s_C2R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32sc_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_32sc_C2R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32f_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_32f_C2R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32fc_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_32fc_C2R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_64f_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_64f_C2R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_8u_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_8u_C3R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_8s_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_8s_C3R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_16u_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_16u_C3R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_16s_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_16s_C3R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_16sc_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_16sc_C3R](#).
- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize_32u_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#)
*hpBufferSize)
 Buffer size for [nppiMaximumError_32u_C3R](#).

- **NppStatus** **nppiMaximumErrorGetBufferHostSize_32s_C3R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_32s_C3R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_32sc_C3R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_32sc_C3R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_32f_C3R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_32f_C3R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_32fc_C3R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_32fc_C3R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_64f_C3R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_64f_C3R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_8u_C4R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_8u_C4R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_8s_C4R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_8s_C4R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_16u_C4R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_16u_C4R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_16s_C4R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_16s_C4R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_16sc_C4R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_16sc_C4R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_32u_C4R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_32u_C4R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_32s_C4R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)
 Buffer size for nppiMaximumError_32s_C4R.
- **NppStatus** **nppiMaximumErrorGetBufferHostSize_32sc_C4R** (**NppiSize** **oSizeROI**, **int**
 *hpBufferSize)

Buffer size for `nppiMaximumError_32sc_C4R`.

- `NppStatus nppiMaximumErrorGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumError_32f_C4R`.

- `NppStatus nppiMaximumErrorGetBufferHostSize_32fc_C4R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumError_32fc_C4R`.

- `NppStatus nppiMaximumErrorGetBufferHostSize_64f_C4R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumError_64f_C4R`.

NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

- `NppStatus nppiAverageErrorGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageError_8u_C1R`.

- `NppStatus nppiAverageErrorGetBufferHostSize_8s_C1R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageError_8s_C1R`.

- `NppStatus nppiAverageErrorGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageError_16u_C1R`.

- `NppStatus nppiAverageErrorGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageError_16s_C1R`.

- `NppStatus nppiAverageErrorGetBufferHostSize_16sc_C1R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageError_16sc_C1R`.

- `NppStatus nppiAverageErrorGetBufferHostSize_32u_C1R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageError_32u_C1R`.

- `NppStatus nppiAverageErrorGetBufferHostSize_32s_C1R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageError_32s_C1R`.

- `NppStatus nppiAverageErrorGetBufferHostSize_32sc_C1R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageError_32sc_C1R`.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32f_C1R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32fc_C1R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C1R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_64f_C1R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8u_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_8u_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8s_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_8s_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16u_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_16u_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16s_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_16s_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16sc_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_16sc_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32u_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32u_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32s_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32s_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32sc_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32sc_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32f_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size for `nppiAverageError_32fc_C2R`.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C2R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_64f_C2R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_8u_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8s_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_8s_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_16u_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16s_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_16s_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16sc_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_16sc_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32u_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_32u_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32s_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_32s_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32sc_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_32sc_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_32f_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_32fc_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C3R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_64f_C3R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8u_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_8u_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8s_C4R` (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)
Buffer size for `nppiAverageError_8s_C4R`.

- **NppStatus** `nppiAverageErrorGetBufferHostSize_16u_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageError_16u_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16s_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageError_16s_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16sc_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageError_16sc_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32u_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageError_32u_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32s_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageError_32s_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32sc_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageError_32sc_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageError_32f_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageError_32fc_C4R`.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C4R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiAverageError_64f_C4R`.

NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

- **NppStatus** `nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumRelativeError_8u_C1R`.
- **NppStatus** `nppiMaximumRelativeErrorGetBufferHostSize_8s_C1R` (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for `nppiMaximumRelativeError_8s_C1R`.

- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_16u_C1R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_16s_C1R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_16sc_C1R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_32u_C1R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_32s_C1R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_32sc_C1R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_32f_C1R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_32fc_C1R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_64f_C1R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_8u_C2R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_8s_C2R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMaximumRelativeError_16u_C2R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size for `nppiMaximumRelativeError_16s_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C2R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_16sc_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C2R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_32u_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C2R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_32s_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C2R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_32sc_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_32f_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C2R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_32fc_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C2R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_64f_C2R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_8u_C3R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_8s_C3R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_16u_C3R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_16s_C3R`.

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiMaximumRelativeError_16sc_C3R`.

- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_32u_C3R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_32s_C3R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_32sc_C3R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_32f_C3R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_32fc_C3R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_64f_C3R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_8u_C4R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_8s_C4R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_16u_C4R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_16s_C4R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_16sc_C4R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_32u_C4R](#).
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_32s_C4R](#).

- **NppStatus** **nppiMaximumRelativeErrorGetBufferHostSize_32sc_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_32sc_C4R](#).
- **NppStatus** **nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_32f_C4R](#).
- **NppStatus** **nppiMaximumRelativeErrorGetBufferHostSize_32fc_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_32fc_C4R](#).
- **NppStatus** **nppiMaximumRelativeErrorGetBufferHostSize_64f_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for [nppiMaximumRelativeError_64f_C4R](#).

NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_8u_C1R](#).
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize_8s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_8s_C1R](#).
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_16u_C1R](#).
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize_16s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_16s_C1R](#).
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize_16sc_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_16sc_C1R](#).
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize_32u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_32u_C1R](#).
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize_32s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_32s_C1R](#).

- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_32sc_C1R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_32f_C1R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_32fc_C1R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_64f_C1R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_8u_C2R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_8s_C2R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_16u_C2R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_16s_C2R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_16sc_C2R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_32u_C2R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_32s_C2R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiAverageRelativeError_32sc_C2R](#).
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size for `nppiAverageRelativeError_32f_C2R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C2R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_32fc_C2R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C2R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_64f_C2R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_8u_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_8s_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_16u_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_16s_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_16sc_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_32u_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_32s_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_32sc_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_32f_C3R`.

- `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C3R (NppiSize oSizeROI, int *hpBufferSize)`

Buffer size for `nppiAverageRelativeError_32fc_C3R`.

- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_64f_C3R](#).
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_8u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_8u_C4R](#).
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_8s_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_8s_C4R](#).
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_16u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_16u_C4R](#).
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_16s_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_16s_C4R](#).
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_16sc_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_16sc_C4R](#).
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_32u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_32u_C4R](#).
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_32s_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_32s_C4R](#).
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_32sc_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_32sc_C4R](#).
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_32f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_32f_C4R](#).
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_32fc_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_32fc_C4R](#).
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize_64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiAverageRelativeError_64f_C4R](#).

7.92.1 Detailed Description

Primitives for computing the statistical properties of an image.

Some statistical primitives also require scratch buffer during the computation. For details, please refer to [Scratch Buffer and Host Pointer](#).

7.92.2 Function Documentation

7.92.2.1 `NppStatus nppiAverageErrorGetBufferHostSize_16s_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageError_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.2 `NppStatus nppiAverageErrorGetBufferHostSize_16s_C2R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageError_16s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.3 `NppStatus nppiAverageErrorGetBufferHostSize_16s_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageError_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.4 NppStatus nppiAverageErrorGetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.5 NppStatus nppiAverageErrorGetBufferHostSize_16sc_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16sc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.6 NppStatus nppiAverageErrorGetBufferHostSize_16sc_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16sc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.7 NppStatus nppiAverageErrorGetBufferHostSize_16sc_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16sc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.8 NppStatus nppiAverageErrorGetBufferHostSize_16sc_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16sc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.9 NppStatus nppiAverageErrorGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.10 NppStatus nppiAverageErrorGetBufferHostSize_16u_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.11 NppStatus nppiAverageErrorGetBufferHostSize_16u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.12 NppStatus nppiAverageErrorGetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.13 NppStatus nppiAverageErrorGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.14 NppStatus nppiAverageErrorGetBufferHostSize_32f_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_32f_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.15 NppStatus nppiAverageErrorGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.16 NppStatus nppiAverageErrorGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.17 NppStatus nppiAverageErrorGetBufferHostSize_32fc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32fc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.18 NppStatus nppiAverageErrorGetBufferHostSize_32fc_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_32fc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.19 NppStatus nppiAverageErrorGetBufferHostSize_32fc_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_32fc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.20 NppStatus nppiAverageErrorGetBufferHostSize_32fc_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_32fc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.21 NppStatus nppiAverageErrorGetBufferHostSize_32s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_32s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.22 NppStatus nppiAverageErrorGetBufferHostSize_32s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.23 NppStatus nppiAverageErrorGetBufferHostSize_32s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.24 NppStatus nppiAverageErrorGetBufferHostSize_32s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.25 NppStatus nppiAverageErrorGetBufferHostSize_32sc_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_32sc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.26 NppStatus nppiAverageErrorGetBufferHostSize_32sc_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_32sc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.27 NppStatus nppiAverageErrorGetBufferHostSize_32sc_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_32sc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.28 NppStatus nppiAverageErrorGetBufferHostSize_32sc_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_32sc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.29 NppStatus nppiAverageErrorGetBufferHostSize_32u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.30 NppStatus nppiAverageErrorGetBufferHostSize_32u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.31 NppStatus nppiAverageErrorGetBufferHostSize_32u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_32u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.32 NppStatus nppiAverageErrorGetBufferHostSize_32u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_32u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.33 NppStatus nppiAverageErrorGetBufferHostSize_64f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_64f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.34 NppStatus nppiAverageErrorGetBufferHostSize_64f_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_64f_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.35 NppStatus nppiAverageErrorGetBufferHostSize_64f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_64f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.36 NppStatus nppiAverageErrorGetBufferHostSize_64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_64f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.37 NppStatus nppiAverageErrorGetBufferHostSize_8s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_8s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.38 NppStatus nppiAverageErrorGetBufferHostSize_8s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_8s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.39 NppStatus nppiAverageErrorGetBufferHostSize_8s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_8s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.40 NppStatus nppiAverageErrorGetBufferHostSize_8s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_8s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.41 NppStatus nppiAverageErrorGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.42 NppStatus nppiAverageErrorGetBufferHostSize_8u_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageError_8u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.43 NppStatus nppiAverageErrorGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.44 NppStatus nppiAverageErrorGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageError_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.45 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.46 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_16s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.47 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.48 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.49 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_16sc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.50 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16sc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.51 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16sc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.52 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_16sc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.53 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.54 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_16u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.55 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.56 NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.57 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.58 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32f_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.59 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.60 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.61 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError_32fc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.62 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C2R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError_32fc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.63 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError_32fc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.64 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32fc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.65 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.66 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.67 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_32s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.68 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_32s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.69 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_32sc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.70 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_32sc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.71 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32sc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.72 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32sc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.73 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_32u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.74 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_32u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.75 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_32u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.76 NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_32u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.77 NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_64f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.78 NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_64f_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.79 NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_64f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.80 NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_64f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.81 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError_8s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.82 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C2R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError_8s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.83 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError_8s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.84 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError_8s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.85 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.86 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_8u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.87 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiAverageRelativeError_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.88 NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiAverageRelativeError_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.89 NppStatus nppiMaximumErrorGetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.90 NppStatus nppiMaximumErrorGetBufferHostSize_16s_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_16s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.91 NppStatus nppiMaximumErrorGetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.92 NppStatus nppiMaximumErrorGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.93 NppStatus nppiMaximumErrorGetBufferHostSize_16sc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16sc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.94 NppStatus nppiMaximumErrorGetBufferHostSize_16sc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16sc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.95 NppStatus nppiMaximumErrorGetBufferHostSize_16sc_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_16sc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.96 NppStatus nppiMaximumErrorGetBufferHostSize_16sc_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_16sc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.97 NppStatus nppiMaximumErrorGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.98 NppStatus nppiMaximumErrorGetBufferHostSize_16u_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_16u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.99 NppStatus nppiMaximumErrorGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.100 NppStatus nppiMaximumErrorGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.101 NppStatus nppiMaximumErrorGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.102 NppStatus nppiMaximumErrorGetBufferHostSize_32f_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32f_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.103 NppStatus nppiMaximumErrorGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.104 NppStatus nppiMaximumErrorGetBufferHostSize_32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.105 NppStatus nppiMaximumErrorGetBufferHostSize_32fc_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32fc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.106 NppStatus nppiMaximumErrorGetBufferHostSize_32fc_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32fc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.107 NppStatus nppiMaximumErrorGetBufferHostSize_32fc_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32fc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.108 NppStatus nppiMaximumErrorGetBufferHostSize_32fc_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32fc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.109 NppStatus nppiMaximumErrorGetBufferHostSize_32s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.110 NppStatus nppiMaximumErrorGetBufferHostSize_32s_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.111 NppStatus nppiMaximumErrorGetBufferHostSize_32s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.112 NppStatus nppiMaximumErrorGetBufferHostSize_32s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.113 NppStatus nppiMaximumErrorGetBufferHostSize_32sc_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32sc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.114 NppStatus nppiMaximumErrorGetBufferHostSize_32sc_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32sc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.115 NppStatus nppiMaximumErrorGetBufferHostSize_32sc_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32sc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.116 NppStatus nppiMaximumErrorGetBufferHostSize_32sc_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32sc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.117 NppStatus nppiMaximumErrorGetBufferHostSize_32u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.118 NppStatus nppiMaximumErrorGetBufferHostSize_32u_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.119 NppStatus nppiMaximumErrorGetBufferHostSize_32u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_32u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.120 NppStatus nppiMaximumErrorGetBufferHostSize_32u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_32u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.121 NppStatus nppiMaximumErrorGetBufferHostSize_64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_64f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.122 NppStatus nppiMaximumErrorGetBufferHostSize_64f_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumError_64f_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.123 NppStatus nppiMaximumErrorGetBufferHostSize_64f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_64f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.124 NppStatus nppiMaximumErrorGetBufferHostSize_64f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_64f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.125 NppStatus nppiMaximumErrorGetBufferHostSize_8s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_8s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.126 NppStatus nppiMaximumErrorGetBufferHostSize_8s_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_8s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.127 NppStatus nppiMaximumErrorGetBufferHostSize_8s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_8s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.128 NppStatus nppiMaximumErrorGetBufferHostSize_8s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_8s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.129 NppStatus nppiMaximumErrorGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.130 NppStatus nppiMaximumErrorGetBufferHostSize_8u_C2R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_8u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.131 NppStatus nppiMaximumErrorGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.132 NppStatus nppiMaximumErrorGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumError_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.133 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumRelativeError_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.134 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.135 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.136 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.137 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16sc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.138 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16sc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.139 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16sc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.140 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16sc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.141 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.142 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.143 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.144 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.145 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.146 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32f_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.147 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.148 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.149 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32fc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.150 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32fc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.151 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32fc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.152 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32fc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.153 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.154 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.155 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.156 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.157 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32sc_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.158 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32sc_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.159 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32sc_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.160 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32sc_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.161 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.162 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.163 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.164 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_32u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.165 **NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C1R** (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumRelativeError_64f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.166 **NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C2R** (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumRelativeError_64f_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.167 **NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C3R** (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumRelativeError_64f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.168 **NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C4R** (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMaximumRelativeError_64f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.169 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.170 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8s_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.171 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.172 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.173 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.174 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C2R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8u_C2R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.175 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.92.2.176 NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMaximumRelativeError_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.93 Sum

Primitives for computing the sum of all the pixel values in an image.

Sum

Given an image $pSrc$ with width W and height H , the sum will be computed as

$$Sum = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

All the results are stored in a 64-bit double precision format, except for two primitives `npSum_8u64s_C1R` and `npSum_8u64s_C4R`.

The sum functions require additional scratch buffer for computations.

- `NppStatus npSum_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pSum)
One-channel 8-bit unsigned image sum.
- `NppStatus npSum_8u64s_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64s` *pSum)
One-channel 8-bit unsigned image sum.
- `NppStatus npSum_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pSum)
One-channel 16-bit unsigned image sum.
- `NppStatus npSum_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pSum)
One-channel 16-bit signed image sum.
- `NppStatus npSum_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pSum)
One-channel 32-bit floating point image sum.
- `NppStatus npSum_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Three-channel 8-bit unsigned image sum.
- `NppStatus npSum_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Three-channel 16-bit unsigned image sum.
- `NppStatus npSum_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Three-channel 16-bit signed image sum.
- `NppStatus npSum_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aSum[3])
Three-channel 32-bit floating point image sum.

- **NppStatus** **nppiSum_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp64f** aSum[3])
Four-channel 8-bit unsigned image sum ignoring alpha channel.
- **NppStatus** **nppiSum_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp64f** aSum[3])
Four-channel 16-bit unsigned image sum ignoring alpha channel.
- **NppStatus** **nppiSum_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp64f** aSum[3])
Four-channel 16-bit signed image sum ignoring alpha channel.
- **NppStatus** **nppiSum_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp64f** aSum[3])
Four-channel 32-bit floating point image sum ignoring alpha channel.
- **NppStatus** **nppiSum_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp64f** aSum[4])
Four-channel 8-bit unsigned image sum.
- **NppStatus** **nppiSum_8u64s_C4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp64s** aSum[4])
Four-channel 8-bit unsigned image sum.
- **NppStatus** **nppiSum_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp64f** aSum[4])
Four-channel 16-bit unsigned image sum.
- **NppStatus** **nppiSum_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp64f** aSum[4])
Four-channel 16-bit signed image sum.
- **NppStatus** **nppiSum_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp64f** aSum[4])
Four-channel 32-bit floating point image sum.

SumGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the sum primitives.

- **NppStatus** **nppiSumGetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiSum_8u_C1R**.*
- **NppStatus** **nppiSumGetBufferHostSize_8u64s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiSum_8u64s_C1R**.*
- **NppStatus** **nppiSumGetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiSum_16u_C1R**.*

- [NppStatus nppiSumGetBufferHostSize_16s_C1R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_16s_C1R](#).
- [NppStatus nppiSumGetBufferHostSize_32f_C1R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_32f_C1R](#).
- [NppStatus nppiSumGetBufferHostSize_8u_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_8u_C3R](#).
- [NppStatus nppiSumGetBufferHostSize_16u_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_16u_C3R](#).
- [NppStatus nppiSumGetBufferHostSize_16s_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_16s_C3R](#).
- [NppStatus nppiSumGetBufferHostSize_32f_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_32f_C3R](#).
- [NppStatus nppiSumGetBufferHostSize_8u_AC4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_8u_AC4R](#).
- [NppStatus nppiSumGetBufferHostSize_16u_AC4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_16u_AC4R](#).
- [NppStatus nppiSumGetBufferHostSize_16s_AC4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_16s_AC4R](#).
- [NppStatus nppiSumGetBufferHostSize_32f_AC4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_32f_AC4R](#).
- [NppStatus nppiSumGetBufferHostSize_8u64s_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_8u64s_C4R](#).
- [NppStatus nppiSumGetBufferHostSize_8u_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_8u_C4R](#).
- [NppStatus nppiSumGetBufferHostSize_16u_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_16u_C4R](#).
- [NppStatus nppiSumGetBufferHostSize_16s_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_16s_C4R](#).
- [NppStatus nppiSumGetBufferHostSize_32f_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for [nppiSum_32f_C4R](#).

7.93.1 Detailed Description

Primitives for computing the sum of all the pixel values in an image.

7.93.2 Function Documentation

7.93.2.1 NppStatus nppiSum_16s_AC4R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f *aSum*[3])

Four-channel 16-bit signed image sum ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSumGetBufferHostSize_16s_AC4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel (alpha channel is not computed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.2 NppStatus nppiSum_16s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f * *pSum*)

One-channel 16-bit signed image sum.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSumGetBufferHostSize_16s_C1R](#) to determine the minium number of bytes required.

pSum Pointer to the computed sum.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.3 NppStatus nppiSum_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f *aSum*[3])

Three-channel 16-bit signed image sum.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSumGetBufferHostSize_16s_C3R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.4 NppStatus nppiSum_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[4])

Four-channel 16-bit signed image sum.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize_16s_C4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.5 NppStatus nppiSum_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])

Four-channel 16-bit unsigned image sum ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize_16u_AC4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel (alpha channel is not computed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.6 NppStatus nppiSum_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pSum)

One-channel 16-bit unsigned image sum.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSumGetBufferHostSize_16u_C1R](#) to determine the minium number of bytes required.

pSum Pointer to the computed sum.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.7 NppStatus nppiSum_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])

Three-channel 16-bit unsigned image sum.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSumGetBufferHostSize_16u_C3R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.8 NppStatus nppiSum_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[4])

Four-channel 16-bit unsigned image sum.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use
[nppiSumGetBufferHostSize_16u_C4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.9 NppStatus nppiSum_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f *aSum*[3])

Four-channel 32-bit floating point image sum ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize_32f_AC4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel (alpha channel is not computed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.10 NppStatus nppiSum_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f * *pSum*)

One-channel 32-bit floating point image sum.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize_32f_C1R](#) to determine the minium number of bytes required.

pSum Pointer to the computed sum.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.11 NppStatus nppiSum_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f *aSum*[3])

Three-channel 32-bit floating point image sum.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize_32f_C3R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.12 **NppStatus nppiSum_32f_C4R** (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f *aSum*[4])

Four-channel 32-bit floating point image sum.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize_32f_C4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.13 **NppStatus nppiSum_8u64s_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64s * *pSum*)

One-channel 8-bit unsigned image sum.

The result is 64-bit long long integer.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize_8u64s_C1R](#) to determine the minium number of bytes required.

pSum Pointer to the computed sum.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.14 **NppStatus nppiSum_8u64s_C4R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64s *aSum*[4])

Four-channel 8-bit unsigned image sum.

The result is 64-bit long long integer.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize_8u64s_C4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.15 `NppStatus nppiSum_8u_AC4R (const Npp8u *pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u *pDeviceBuffer, Npp64f aSum[3])`

Four-channel 8-bit unsigned image sum ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize_8u_AC4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel (alpha channel is not computed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.16 `NppStatus nppiSum_8u_C1R (const Npp8u *pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u *pDeviceBuffer, Npp64f *pSum)`

One-channel 8-bit unsigned image sum.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize_8u_C1R](#) to determine the minium number of bytes required.

pSum Pointer to the computed sum.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.17 `NppStatus nppiSum_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])`

Three-channel 8-bit unsigned image sum.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiSumGetBufferHostSize_8u_C3R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.18 `NppStatus nppiSum_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[4])`

Four-channel 8-bit unsigned image sum.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use

[nppiSumGetBufferHostSize_8u_C4R](#) to determine the minium number of bytes required.

aSum Array that contains computed sum for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.93.2.19 `NppStatus nppiSumGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiSum_16s_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.20 NppStatus nppiSumGetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiSum_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.21 NppStatus nppiSumGetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiSum_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.22 NppStatus nppiSumGetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiSum_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.23 NppStatus nppiSumGetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiSum_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.93.2.24 NppStatus nppiSumGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.93.2.25 NppStatus nppiSumGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.93.2.26 NppStatus nppiSumGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.27 NppStatus nppiSumGetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiSum_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.28 NppStatus nppiSumGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiSum_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.29 NppStatus nppiSumGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiSum_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.30 NppStatus nppiSumGetBufferHostSize_32f_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiSum_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.93.2.31 NppStatus nppiSumGetBufferHostSize_8u64s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_8u64s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.93.2.32 NppStatus nppiSumGetBufferHostSize_8u64s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_8u64s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.93.2.33 NppStatus nppiSumGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiSum_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.34 NppStatus nppiSumGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiSum_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.35 NppStatus nppiSumGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiSum_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.36 NppStatus nppiSumGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiSum_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94 Min

Primitives for computing the minimal pixel value of an image.

Min

The scratch buffer is required by the min functions.

- `NppStatus nppiMin_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` *pMin)
One-channel 8-bit unsigned image min.
- `NppStatus nppiMin_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` *pMin)
One-channel 16-bit unsigned image min.
- `NppStatus nppiMin_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` *pMin)
One-channel 16-bit signed image min.
- `NppStatus nppiMin_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` *pMin)
One-channel 32-bit floating point image min.
- `NppStatus nppiMin_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` aMin[3])
Three-channel 8-bit unsigned image min.
- `NppStatus nppiMin_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` aMin[3])
Three-channel 16-bit unsigned image min.
- `NppStatus nppiMin_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` aMin[3])
Three-channel 16-bit signed image min.
- `NppStatus nppiMin_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` aMin[3])
Three-channel 32-bit floating point image min.
- `NppStatus nppiMin_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` aMin[4])
Four-channel 8-bit unsigned image min.
- `NppStatus nppiMin_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` aMin[4])
Four-channel 16-bit unsigned image min.
- `NppStatus nppiMin_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` aMin[4])

Four-channel 16-bit signed image min.

- `NppStatus nppiMin_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` aMin[4])

Four-channel 32-bit floating point image min.

- `NppStatus nppiMin_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` aMin[3])

Four-channel 8-bit unsigned image min ignoring alpha channel.

- `NppStatus nppiMin_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` aMin[3])

Four-channel 16-bit unsigned image min ignoring alpha channel.

- `NppStatus nppiMin_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` aMin[3])

Four-channel 16-bit signed image min ignoring alpha channel.

- `NppStatus nppiMin_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` aMin[3])

Four-channel 32-bit floating point image min ignoring alpha channel.

MinGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the min primitives.

- `NppStatus nppiMinGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMin_8u_C1R`.
- `NppStatus nppiMinGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMin_16u_C1R`.
- `NppStatus nppiMinGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMin_16s_C1R`.
- `NppStatus nppiMinGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMin_32f_C1R`.
- `NppStatus nppiMinGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMin_8u_C3R`.
- `NppStatus nppiMinGetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMin_16u_C3R`.
- `NppStatus nppiMinGetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMin_16s_C3R`.
- `NppStatus nppiMinGetBufferHostSize_32f_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMin_32f_C3R`.

- [NppStatus nppiMinGetBufferHostSize_8u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMin_8u_C4R](#).
- [NppStatus nppiMinGetBufferHostSize_16u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMin_16u_C4R](#).
- [NppStatus nppiMinGetBufferHostSize_16s_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMin_16s_C4R](#).
- [NppStatus nppiMinGetBufferHostSize_32f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMin_32f_C4R](#).
- [NppStatus nppiMinGetBufferHostSize_8u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMin_8u_AC4R](#).
- [NppStatus nppiMinGetBufferHostSize_16u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMin_16u_AC4R](#).
- [NppStatus nppiMinGetBufferHostSize_16s_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMin_16s_AC4R](#).
- [NppStatus nppiMinGetBufferHostSize_32f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMin_32f_AC4R](#).

7.94.1 Detailed Description

Primitives for computing the minimal pixel value of an image.

7.94.2 Function Documentation

7.94.2.1 [NppStatus nppiMin_16s_AC4R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp16s](#) aMin[3])

Four-channel 16-bit signed image min ignoring alpha channel.

Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_16s_AC4R](#) to determine the minium number of bytes required.

[aMin](#) Array that contains the computed minimum results for each channel (alpha channel is not processed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.2 NppStatus nppiMin_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s * pMin)

One-channel 16-bit signed image min.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_16s_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed minimum result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.3 NppStatus nppiMin_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMin[3])

Three-channel 16-bit signed image min.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_16s_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.4 NppStatus nppiMin_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMin[4])

Four-channel 16-bit signed image min.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_16s_C4R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.5 NppStatus nppiMin_16u_AC4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16u *aMin*[3])

Four-channel 16-bit unsigned image min ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_16u_AC4R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel (alpha channel is not processed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.6 NppStatus nppiMin_16u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16u * *pMin*)

One-channel 16-bit unsigned image min.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_16u_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed minimum result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.7 NppStatus nppiMin_16u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16u *aMin*[3])

Three-channel 16-bit unsigned image min.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_16u_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.8 NppStatus nppiMin_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[4])

Four-channel 16-bit unsigned image min.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_16u_C4R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.9 NppStatus nppiMin_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[3])

Four-channel 32-bit floating point image min ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_32f_AC4R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel (alpha channel is not processed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.10 NppStatus nppiMin_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f * pMin)

One-channel 32-bit floating point image min.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_32f_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed minimum result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.11 `NppStatus nppiMin_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[3])`

Three-channel 32-bit floating point image min.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_32f_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.12 `NppStatus nppiMin_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[4])`

Four-channel 32-bit floating point image min.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_32f_C4R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.13 **NppStatus nppiMin_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp8u *aMin*[3])

Four-channel 8-bit unsigned image min ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_8u_AC4R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel (alpha channel is not processed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.14 **NppStatus nppiMin_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp8u * *pMin*)

One-channel 8-bit unsigned image min.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_8u_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed minimum result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.15 **NppStatus nppiMin_8u_C3R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp8u *aMin*[3])

Three-channel 8-bit unsigned image min.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_8u_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.16 `NppStatus nppiMin_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[4])`

Four-channel 8-bit unsigned image min.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize_8u_C4R](#) to determine the minimum number of bytes required.

aMin Array that contains the computed minimum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.94.2.17 `NppStatus nppiMinGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMin_16s_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.18 `NppStatus nppiMinGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMin_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.19 NppStatus nppiMinGetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMin_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.20 NppStatus nppiMinGetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMin_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.21 NppStatus nppiMinGetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMin_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.22 NppStatus nppiMinGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMin_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.23 NppStatus nppiMinGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.24 NppStatus nppiMinGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.25 NppStatus nppiMinGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.26 NppStatus nppiMinGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMin_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.27 NppStatus nppiMinGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMin_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.28 NppStatus nppiMinGetBufferHostSize_32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMin_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.29 NppStatus nppiMinGetBufferHostSize_8u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMin_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.30 NppStatus nppiMinGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.31 NppStatus nppiMinGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.94.2.32 NppStatus nppiMinGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMin_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.95 MinIndx

Primitives for computing the minimal value and its indices (X and Y coordinates) of an image.

MinIndx

If there are several minima in the selected ROI, the function returns one on the top leftmost position.

The scratch buffer is required by the functions.

- **NppStatus nppiMinIndx_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** *pMin, int *pIndexX, int *pIndexY)
One-channel 8-bit unsigned image MinIndx.
- **NppStatus nppiMinIndx_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** *pMin, int *pIndexX, int *pIndexY)
One-channel 16-bit unsigned image MinIndx.
- **NppStatus nppiMinIndx_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** *pMin, int *pIndexX, int *pIndexY)
One-channel 16-bit signed image MinIndx.
- **NppStatus nppiMinIndx_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** *pMin, int *pIndexX, int *pIndexY)
One-channel 32-bit floating point image MinIndx.
- **NppStatus nppiMinIndx_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** aMin[3], int aIndexX[3], int aIndexY[3])
Three-channel 8-bit unsigned image MinIndx.
- **NppStatus nppiMinIndx_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** aMin[3], int aIndexX[3], int aIndexY[3])
Three-channel 16-bit unsigned image MinIndx.
- **NppStatus nppiMinIndx_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** aMin[3], int aIndexX[3], int aIndexY[3])
Three-channel 16-bit signed image MinIndx.
- **NppStatus nppiMinIndx_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** aMin[3], int aIndexX[3], int aIndexY[3])
Three-channel 32-bit floating point image MinIndx.
- **NppStatus nppiMinIndx_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** aMin[4], int aIndexX[4], int aIndexY[4])
Four-channel 8-bit unsigned image MinIndx.
- **NppStatus nppiMinIndx_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** aMin[4], int aIndexX[4], int aIndexY[4])
Four-channel 16-bit unsigned image MinIndx.

- **NppStatus nppiMinIndx_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** aMin[4], int aIndexX[4], int aIndexY[4])
Four-channel 16-bit signed image MinIndx.
- **NppStatus nppiMinIndx_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** aMin[4], int aIndexX[4], int aIndexY[4])
Four-channel 32-bit floating point image MinIndx.
- **NppStatus nppiMinIndx_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** aMin[3], int aIndexX[3], int aIndexY[3])
Four-channel 8-bit unsigned image MinIndx ignoring alpha channel.
- **NppStatus nppiMinIndx_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** aMin[3], int aIndexX[3], int aIndexY[3])
Four-channel 16-bit unsigned image MinIndx ignoring alpha channel.
- **NppStatus nppiMinIndx_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** aMin[3], int aIndexX[3], int aIndexY[3])
Four-channel 16-bit signed image MinIndx ignoring alpha channel.
- **NppStatus nppiMinIndx_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** aMin[3], int aIndexX[3], int aIndexY[3])
Four-channel 32-bit floating point image MinIndx ignoring alpha channel.

MinIndxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MinIndx primitives.

- **NppStatus nppiMinIndxGetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_8u_C1R.
- **NppStatus nppiMinIndxGetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_16u_C1R.
- **NppStatus nppiMinIndxGetBufferHostSize_16s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_16s_C1R.
- **NppStatus nppiMinIndxGetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_32f_C1R.
- **NppStatus nppiMinIndxGetBufferHostSize_8u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_8u_C3R.
- **NppStatus nppiMinIndxGetBufferHostSize_16u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_16u_C3R.
- **NppStatus nppiMinIndxGetBufferHostSize_16s_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIndx_16s_C3R.

- [NppStatus nppiMinIdxGetBufferHostSize_32f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIdx_32f_C3R.
- [NppStatus nppiMinIdxGetBufferHostSize_8u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIdx_8u_C4R.
- [NppStatus nppiMinIdxGetBufferHostSize_16u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIdx_16u_C4R.
- [NppStatus nppiMinIdxGetBufferHostSize_16s_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIdx_16s_C4R.
- [NppStatus nppiMinIdxGetBufferHostSize_32f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIdx_32f_C4R.
- [NppStatus nppiMinIdxGetBufferHostSize_8u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIdx_8u_AC4R.
- [NppStatus nppiMinIdxGetBufferHostSize_16u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIdx_16u_AC4R.
- [NppStatus nppiMinIdxGetBufferHostSize_16s_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIdx_16s_AC4R.
- [NppStatus nppiMinIdxGetBufferHostSize_32f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMinIdx_32f_AC4R.

7.95.1 Detailed Description

Primitives for computing the minimal value and its indices (X and Y coordinates) of an image.

7.95.2 Function Documentation

7.95.2.1 [NppStatus nppiMinIdx_16s_AC4R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp16s](#) aMin[3], int aIndexX[3], int aIndexY[3])

Four-channel 16-bit signed image MinIdx ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize_16s_AC4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.2 NppStatus nppiMinIndx_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s * pMin, int * pIndexX, int * pIndexY)

One-channel 16-bit signed image MinIndx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_16s_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed min result.

pIndexX Pointer to the X coordinate of the image min value.

pIndexY Pointer to the Y coordinate of the image min value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.3 NppStatus nppiMinIndx_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMin[3], int aIndexX[3], int aIndexY[3])

Three-channel 16-bit signed image MinIndx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_16s_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.4 NppStatus nppiMinIndx_16s_C4R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16s *aMin*[4], int *aIndexX*[4], int *aIndexY*[4])

Four-channel 16-bit signed image MinIndx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_16s_C4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.5 NppStatus nppiMinIndx_16u_AC4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16u *aMin*[3], int *aIndexX*[3], int *aIndexY*[3])

Four-channel 16-bit unsigned image MinIndx ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_16u_AC4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.6 NppStatus nppiMinIndx_16u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16u * *pMin*, int * *pIndexX*, int * *pIndexY*)

One-channel 16-bit unsigned image MinIndx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_16u_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed min result.

pIndexX Pointer to the X coordinate of the image min value.

pIndexY Pointer to the Y coordinate of the image min value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.7 NppStatus nppiMinIndx_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[3], int aIndexX[3], int aIndexY[3])

Three-channel 16-bit unsigned image MinIndx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_16u_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.8 NppStatus nppiMinIndx_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[4], int aIndexX[4], int aIndexY[4])

Four-channel 16-bit unsigned image MinIndx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_16u_C4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.9 NppStatus nppiMinIndx_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp32f *aMin*[3], int *aIndexX*[3], int *aIndexY*[3])

Four-channel 32-bit floating point image MinIndx ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_32f_AC4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.10 NppStatus nppiMinIndx_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp32f * *pMin*, int * *pIndexX*, int * *pIndexY*)

One-channel 32-bit floating point image MinIndx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_32f_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed min result.

pIndexX Pointer to the X coordinate of the image min value.

pIndexY Pointer to the Y coordinate of the image min value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.11 NppStatus nppiMinIndx_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp32f *aMin*[3], int *aIndexX*[3], int *aIndexY*[3])

Three-channel 32-bit floating point image MinIndx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_32f_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.12 NppStatus nppiMinIndx_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[4], int aIndexX[4], int aIndexY[4])

Four-channel 32-bit floating point image MinIndx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_32f_C4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.13 NppStatus nppiMinIndx_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[3], int aIndexX[3], int aIndexY[3])

Four-channel 8-bit unsigned image MinIndx ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_8u_AC4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.14 **NppStatus nppiMinIdx_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp8u * *pMin*, int * *pIndexX*, int * *pIndexY*)

One-channel 8-bit unsigned image MinIdx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize_8u_C1R](#) to determine the minium number of bytes required.

pMin Pointer to the computed min result.

pIndexX Pointer to the X coordinate of the image min value.

pIndexY Ppointer to the Y coordinate of the image min value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.15 **NppStatus nppiMinIdx_8u_C3R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp8u *aMin*[3], int *aIndexX*[3], int *aIndexY*[3])

Three-channel 8-bit unsigned image MinIdx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize_8u_C3R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.16 **NppStatus nppiMinIdx_8u_C4R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp8u *aMin*[4], int *aIndexX*[4], int *aIndexY*[4])

Four-channel 8-bit unsigned image MinIdx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize_8u_C4R](#) to determine the minium number of bytes required.

aMin Array that contains the min values.

aIndexX Array that contains the X coordinates of the image min values.

aIndexY Array that contains the Y coordinates of the image min values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.95.2.17 NppStatus nppiMinIndxGetBufferHostSize_16s_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the dvice scratch buffer size (in bytes) for nppiMinIndx_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.18 NppStatus nppiMinIndxGetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the dvice scratch buffer size (in bytes) for nppiMinIndx_16s_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.19 NppStatus nppiMinIndxGetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the dvice scratch buffer size (in bytes) for nppiMinIndx_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.20 NppStatus nppiMinIdxGetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiMinIdx_16s_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.21 NppStatus nppiMinIdxGetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiMinIdx_8u_AC4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.22 NppStatus nppiMinIdxGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiMinIdx_16u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.23 NppStatus nppiMinIndxGetBufferHostSize_16u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_16u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.24 NppStatus nppiMinIndxGetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_16u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.25 NppStatus nppiMinIndxGetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.26 NppStatus nppiMinIndxGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_32f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.27 NppStatus nppiMinIdxGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMinIdx_32f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.28 NppStatus nppiMinIdxGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMinIdx_32f_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.95.2.29 NppStatus nppiMinIdxGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiMinIdx_8u_AC4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.30 NppStatus nppiMinIndxGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMinIndx_8u_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.31 NppStatus nppiMinIndxGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMinIndx_8u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.32 NppStatus nppiMinIndxGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMinIndx_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96 Max

Primitives for computing the maximal pixel value of an image.

Max

The scratch buffer is required by the functions.

- `NppStatus nppiMax_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` *pMax)
One-channel 8-bit unsigned image Max.
- `NppStatus nppiMax_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` *pMax)
One-channel 16-bit unsigned image Max.
- `NppStatus nppiMax_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` *pMax)
One-channel 16-bit signed image Max.
- `NppStatus nppiMax_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` *pMax)
One-channel 32-bit floating point image Max.
- `NppStatus nppiMax_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` aMax[3])
Three-channel 8-bit unsigned image Max.
- `NppStatus nppiMax_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` aMax[3])
Three-channel 16-bit unsigned image Max.
- `NppStatus nppiMax_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` aMax[3])
Three-channel 16-bit signed image Max.
- `NppStatus nppiMax_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` aMax[3])
Three-channel 32-bit floating point image Max.
- `NppStatus nppiMax_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` aMax[4])
Four-channel 8-bit unsigned image Max.
- `NppStatus nppiMax_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` aMax[4])
Four-channel 16-bit unsigned image Max.
- `NppStatus nppiMax_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` aMax[4])

Four-channel 16-bit signed image Max.

- `NppStatus nppiMax_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` aMax[4])

Four-channel 32-bit floating point image Max.

- `NppStatus nppiMax_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp8u` aMax[3])

Four-channel 8-bit unsigned image Max ignoring alpha channel.

- `NppStatus nppiMax_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16u` aMax[3])

Four-channel 16-bit unsigned image Max ignoring alpha channel.

- `NppStatus nppiMax_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp16s` aMax[3])

Four-channel 16-bit signed image Max ignoring alpha channel.

- `NppStatus nppiMax_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp32f` aMax[3])

Four-channel 32-bit floating point image Max ignoring alpha channel.

MaxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Max primitives.

- `NppStatus nppiMaxGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiMax_8u_C1R`.

- `NppStatus nppiMaxGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiMax_16u_C1R`.

- `NppStatus nppiMaxGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiMax_16s_C1R`.

- `NppStatus nppiMaxGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiMax_32f_C1R`.

- `NppStatus nppiMaxGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiMax_8u_C3R`.

- `NppStatus nppiMaxGetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiMax_16u_C3R`.

- `NppStatus nppiMaxGetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiMax_16s_C3R`.

- `NppStatus nppiMaxGetBufferHostSize_32f_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiMax_32f_C3R`.

- [NppStatus nppiMaxGetBufferHostSize_8u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMax_8u_C4R](#).
- [NppStatus nppiMaxGetBufferHostSize_16u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMax_16u_C4R](#).
- [NppStatus nppiMaxGetBufferHostSize_16s_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMax_16s_C4R](#).
- [NppStatus nppiMaxGetBufferHostSize_32f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMax_32f_C4R](#).
- [NppStatus nppiMaxGetBufferHostSize_8u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMax_8u_AC4R](#).
- [NppStatus nppiMaxGetBufferHostSize_16u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMax_16u_AC4R](#).
- [NppStatus nppiMaxGetBufferHostSize_16s_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMax_16s_AC4R](#).
- [NppStatus nppiMaxGetBufferHostSize_32f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMax_32f_AC4R](#).

7.96.1 Detailed Description

Primitives for computing the maximal pixel value of an image.

7.96.2 Function Documentation

7.96.2.1 [NppStatus nppiMax_16s_AC4R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp16s](#) aMax[3])

Four-channel 16-bit signed image Max ignoring alpha channel.

Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_16s_AC4R](#) to determine the maximum number of bytes required.

[aMax](#) Array that contains the computed maximum results for each channel (alpha channel is not processed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.2 NppStatus nppiMax_16s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16s * *pMax*)

One-channel 16-bit signed image Max.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_16s_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed maximum result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.3 NppStatus nppiMax_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16s *aMax*[3])

Three-channel 16-bit signed image Max.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_16s_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.4 NppStatus nppiMax_16s_C4R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16s *aMax*[4])

Four-channel 16-bit signed image Max.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_16s_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.5 NppStatus nppiMax_16u_AC4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16u *aMax*[3])

Four-channel 16-bit unsigned image Max ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_16u_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel (alpha channel is not processed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.6 NppStatus nppiMax_16u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16u * *pMax*)

One-channel 16-bit unsigned image Max.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_16u_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed maximum result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.7 NppStatus nppiMax_16u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp16u *aMax*[3])

Three-channel 16-bit unsigned image Max.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_16u_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.8 NppStatus nppiMax_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[4])

Four-channel 16-bit unsigned image Max.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_16u_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.9 NppStatus nppiMax_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[3])

Four-channel 32-bit floating point image Max ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_32f_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel (alpha channel is not processed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.10 NppStatus nppiMax_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f * pMax)

One-channel 32-bit floating point image Max.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_32f_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed maximum result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.11 `NppStatus nppiMax_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[3])`

Three-channel 32-bit floating point image Max.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_32f_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.12 `NppStatus nppiMax_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[4])`

Four-channel 32-bit floating point image Max.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_32f_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.13 NppStatus nppiMax_8u_AC4R (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp8u *aMax*[3])

Four-channel 8-bit unsigned image Max ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_8u_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel (alpha channel is not processed).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.14 NppStatus nppiMax_8u_C1R (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp8u * *pMax*)

One-channel 8-bit unsigned image Max.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_8u_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed maximum result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.15 NppStatus nppiMax_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp8u *aMax*[3])

Three-channel 8-bit unsigned image Max.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_8u_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.16 `NppStatus nppiMax_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[4])`

Four-channel 8-bit unsigned image Max.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize_8u_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the computed maximum results for each channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.96.2.17 `NppStatus nppiMaxGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMax_16s_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.18 `NppStatus nppiMaxGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMax_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.19 NppStatus nppiMaxGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.20 NppStatus nppiMaxGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.21 NppStatus nppiMaxGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.22 NppStatus nppiMaxGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.23 NppStatus nppiMaxGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.24 NppStatus nppiMaxGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.25 NppStatus nppiMaxGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.26 NppStatus nppiMaxGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.27 NppStatus nppiMaxGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.28 NppStatus nppiMaxGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.29 NppStatus nppiMaxGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.30 NppStatus nppiMaxGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.31 NppStatus nppiMaxGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.96.2.32 NppStatus nppiMaxGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMax_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.97 MaxIdx

Primitives for computing the maximal value and its indices (X and Y coordinates) of an image.

MaxIdx

If there are several maxima in the selected region of interest, the function returns one on the top leftmost position.

The scratch buffer is required by the functions.

- **NppStatus nppiMaxIdx_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** *pMax, int *pIndexX, int *pIndexY)
One-channel 8-bit unsigned image MaxIdx.
- **NppStatus nppiMaxIdx_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** *pMax, int *pIndexX, int *pIndexY)
One-channel 16-bit unsigned image MaxIdx.
- **NppStatus nppiMaxIdx_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** *pMax, int *pIndexX, int *pIndexY)
One-channel 16-bit signed image MaxIdx.
- **NppStatus nppiMaxIdx_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** *pMax, int *pIndexX, int *pIndexY)
One-channel 32-bit floating point image MaxIdx.
- **NppStatus nppiMaxIdx_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** aMax[3], int aIndexX[3], int aIndexY[3])
Three-channel 8-bit unsigned image MaxIdx.
- **NppStatus nppiMaxIdx_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** aMax[3], int aIndexX[3], int aIndexY[3])
Three-channel 16-bit unsigned image MaxIdx.
- **NppStatus nppiMaxIdx_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** aMax[3], int aIndexX[3], int aIndexY[3])
Three-channel 16-bit signed image MaxIdx.
- **NppStatus nppiMaxIdx_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** aMax[3], int aIndexX[3], int aIndexY[3])
Three-channel 32-bit floating point image MaxIdx.
- **NppStatus nppiMaxIdx_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** aMax[4], int aIndexX[4], int aIndexY[4])
Four-channel 8-bit unsigned image MaxIdx.
- **NppStatus nppiMaxIdx_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** aMax[4], int aIndexX[4], int aIndexY[4])
Four-channel 16-bit unsigned image MaxIdx.

- **NppStatus nppiMaxIdx_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** aMax[4], int aIndexX[4], int aIndexY[4])
Four-channel 16-bit signed image MaxIdx.
- **NppStatus nppiMaxIdx_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** aMax[4], int aIndexX[4], int aIndexY[4])
Four-channel 32-bit floating point image MaxIdx.
- **NppStatus nppiMaxIdx_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp8u** aMax[3], int aIndexX[3], int aIndexY[3])
Four-channel 8-bit unsigned image MaxIdx ignoring alpha channel.
- **NppStatus nppiMaxIdx_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16u** aMax[3], int aIndexX[3], int aIndexY[3])
Four-channel 16-bit unsigned image MaxIdx ignoring alpha channel.
- **NppStatus nppiMaxIdx_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp16s** aMax[3], int aIndexX[3], int aIndexY[3])
Four-channel 16-bit signed image MaxIdx ignoring alpha channel.
- **NppStatus nppiMaxIdx_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** *pDeviceBuffer, **Npp32f** aMax[3], int aIndexX[3], int aIndexY[3])
Four-channel 32-bit floating point image MaxIdx ignoring alpha channel.

MaxIdxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MaxIdx primitives.

- **NppStatus nppiMaxIdxGetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_8u_C1R.
- **NppStatus nppiMaxIdxGetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16u_C1R.
- **NppStatus nppiMaxIdxGetBufferHostSize_16s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16s_C1R.
- **NppStatus nppiMaxIdxGetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_32f_C1R.
- **NppStatus nppiMaxIdxGetBufferHostSize_8u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_8u_C3R.
- **NppStatus nppiMaxIdxGetBufferHostSize_16u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16u_C3R.
- **NppStatus nppiMaxIdxGetBufferHostSize_16s_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16s_C3R.

- [NppStatus nppiMaxIndxGetBufferHostSize_32f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_32f_C3R.
- [NppStatus nppiMaxIndxGetBufferHostSize_8u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_C4R.
- [NppStatus nppiMaxIndxGetBufferHostSize_16u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16u_C4R.
- [NppStatus nppiMaxIndxGetBufferHostSize_16s_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16s_C4R.
- [NppStatus nppiMaxIndxGetBufferHostSize_32f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_32f_C4R.
- [NppStatus nppiMaxIndxGetBufferHostSize_8u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_AC4R.
- [NppStatus nppiMaxIndxGetBufferHostSize_16u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16u_AC4R.
- [NppStatus nppiMaxIndxGetBufferHostSize_16s_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16s_AC4R.
- [NppStatus nppiMaxIndxGetBufferHostSize_32f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiMaxIndx_32f_AC4R.

7.97.1 Detailed Description

Primitives for computing the maximal value and its indices (X and Y coordinates) of an image.

7.97.2 Function Documentation

7.97.2.1 [NppStatus nppiMaxIndx_16s_AC4R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp16s](#) aMax[3], int aIndexX[3], int aIndexY[3])

Four-channel 16-bit signed image MaxIndx ignoring alpha channel.

Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMaxIndxGetBufferHostSize_16s_AC4R](#) to determine the maximum number of bytes required.

[aMax](#) Array that contains the max values.

[aIndexX](#) Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.2 `NppStatus nppiMaxIndx_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s * pMax, int * pIndexX, int * pIndexY)`

One-channel 16-bit signed image MaxIndx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize_16s_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed max result.

pIndexX Pointer to the X coordinate of the image max value.

pIndexY Pointer to the Y coordinate of the image max value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.3 `NppStatus nppiMaxIndx_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMax[3], int aIndexX[3], int aIndexY[3])`

Three-channel 16-bit signed image MaxIndx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize_16s_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.4 NppStatus nppiMaxIndx_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMax[4], int aIndexX[4], int aIndexY[4])

Four-channel 16-bit signed image MaxIndx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize_16s_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.5 NppStatus nppiMaxIndx_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[3], int aIndexX[3], int aIndexY[3])

Four-channel 16-bit unsigned image MaxIndx ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize_16u_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.6 NppStatus nppiMaxIndx_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u * pMax, int * pIndexX, int * pIndexY)

One-channel 16-bit unsigned image MaxIndx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize_16u_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed max result.

pIndexX Pointer to the X coordinate of the image max value.

pIndexY Pointer to the Y coordinate of the image max value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.7 `NppStatus nppiMaxIdx_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[3], int aIndexX[3], int aIndexY[3])`

Three-channel 16-bit unsigned image MaxIdx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize_16u_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.8 `NppStatus nppiMaxIdx_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[4], int aIndexX[4], int aIndexY[4])`

Four-channel 16-bit unsigned image MaxIdx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize_16u_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.9 NppStatus nppiMaxIndx_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp32f *aMax*[3], int *aIndexX*[3], int *aIndexY*[3])

Four-channel 32-bit floating point image MaxIndx ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize_32f_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.10 NppStatus nppiMaxIndx_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp32f * *pMax*, int * *pIndexX*, int * *pIndexY*)

One-channel 32-bit floating point image MaxIndx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize_32f_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed max result.

pIndexX Pointer to the X coordinate of the image max value.

pIndexY Pointer to the Y coordinate of the image max value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.11 NppStatus nppiMaxIndx_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp32f *aMax*[3], int *aIndexX*[3], int *aIndexY*[3])

Three-channel 32-bit floating point image MaxIndx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize_32f_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.12 NppStatus nppiMaxIdx_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[4], int aIndexX[4], int aIndexY[4])

Four-channel 32-bit floating point image MaxIdx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize_32f_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.13 NppStatus nppiMaxIdx_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[3], int aIndexX[3], int aIndexY[3])

Four-channel 8-bit unsigned image MaxIdx ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize_8u_AC4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.14 `NppStatus nppiMaxIdx_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u * pMax, int * pIndexX, int * pIndexY)`

One-channel 8-bit unsigned image MaxIdx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize_8u_C1R](#) to determine the maximum number of bytes required.

pMax Pointer to the computed max result.

pIndexX Pointer to the X coordinate of the image max value.

pIndexY Pointer to the Y coordinate of the image max value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.15 `NppStatus nppiMaxIdx_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[3], int aIndexX[3], int aIndexY[3])`

Three-channel 8-bit unsigned image MaxIdx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize_8u_C3R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.16 `NppStatus nppiMaxIdx_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[4], int aIndexX[4], int aIndexY[4])`

Four-channel 8-bit unsigned image MaxIdx.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize_8u_C4R](#) to determine the maximum number of bytes required.

aMax Array that contains the max values.

aIndexX Array that contains the X coordinates of the image max values.

aIndexY Array that contains the Y coordinates of the image max values.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.97.2.17 NppStatus nppiMaxIndxGetBufferHostSize_16s_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.97.2.18 NppStatus nppiMaxIndxGetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16s_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.97.2.19 NppStatus nppiMaxIndxGetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.97.2.20 NppStatus nppiMaxIndxGetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx_16s_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.97.2.21 NppStatus nppiMaxIndxGetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx_8u_AC4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.97.2.22 NppStatus nppiMaxIndxGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx_16u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.97.2.23 NppStatus nppiMaxIndxGetBufferHostSize_16u_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.97.2.24 NppStatus nppiMaxIndxGetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.97.2.25 NppStatus nppiMaxIndxGetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.97.2.26 NppStatus nppiMaxIndxGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_32f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.97.2.27 NppStatus nppiMaxIndxGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_32f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.97.2.28 NppStatus nppiMaxIndxGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_32f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.97.2.29 NppStatus nppiMaxIndxGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.97.2.30 NppStatus nppiMaxIndxGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.97.2.31 NppStatus nppiMaxIndxGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.97.2.32 NppStatus nppiMaxIndxGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.98 MinMax

Primitives for computing both the minimal and the maximal values of an image.

MinMax

The functions require the device scratch buffer.

- `NppStatus nppiMinMax_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pMin, `Npp8u` *pMax, `Npp8u` *pDeviceBuffer)
One-channel 8-bit unsigned image MinMax.
- `NppStatus nppiMinMax_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` *pMin, `Npp16u` *pMax, `Npp8u` *pDeviceBuffer)
One-channel 16-bit unsigned image MinMax.
- `NppStatus nppiMinMax_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16s` *pMin, `Npp16s` *pMax, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed image MinMax.
- `NppStatus nppiMinMax_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` *pMin, `Npp32f` *pMax, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point image MinMax.
- `NppStatus nppiMinMax_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` aMin[3], `Npp8u` aMax[3], `Npp8u` *pDeviceBuffer)
Three-channel 8-bit unsigned image MinMax.
- `NppStatus nppiMinMax_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` aMin[3], `Npp16u` aMax[3], `Npp8u` *pDeviceBuffer)
Three-channel 16-bit unsigned image MinMax.
- `NppStatus nppiMinMax_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16s` aMin[3], `Npp16s` aMax[3], `Npp8u` *pDeviceBuffer)
Three-channel 16-bit signed image MinMax.
- `NppStatus nppiMinMax_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` aMin[3], `Npp32f` aMax[3], `Npp8u` *pDeviceBuffer)
Three-channel 32-bit floating point image MinMax.
- `NppStatus nppiMinMax_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` aMin[3], `Npp8u` aMax[3], `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image MinMax ignoring alpha channel.
- `NppStatus nppiMinMax_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` aMin[3], `Npp16u` aMax[3], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit unsigned image MinMax ignoring alpha channel.
- `NppStatus nppiMinMax_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16s` aMin[3], `Npp16s` aMax[3], `Npp8u` *pDeviceBuffer)

Four-channel 16-bit signed image MinMax ignoring alpha channel.

- `NppStatus nppiMinMax_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` aMin[3], `Npp32f` aMax[3], `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image MinMax ignoring alpha channel.

- `NppStatus nppiMinMax_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` aMin[4], `Npp8u` aMax[4], `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image MinMax.

- `NppStatus nppiMinMax_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` aMin[4], `Npp16u` aMax[4], `Npp8u` *pDeviceBuffer)

Four-channel 16-bit unsigned image MinMax.

- `NppStatus nppiMinMax_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16s` aMin[4], `Npp16s` aMax[4], `Npp8u` *pDeviceBuffer)

Four-channel 16-bit signed image MinMax.

- `NppStatus nppiMinMax_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` aMin[4], `Npp32f` aMax[4], `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image MinMax.

MinMaxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MinMax primitives.

- `NppStatus nppiMinMaxGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMinMax_8u_C1R`.
- `NppStatus nppiMinMaxGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMinMax_16u_C1R`.
- `NppStatus nppiMinMaxGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMinMax_16s_C1R`.
- `NppStatus nppiMinMaxGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMinMax_32f_C1R`.
- `NppStatus nppiMinMaxGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMinMax_8u_C3R`.
- `NppStatus nppiMinMaxGetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMinMax_16u_C3R`.
- `NppStatus nppiMinMaxGetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMinMax_16s_C3R`.
- `NppStatus nppiMinMaxGetBufferHostSize_32f_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMinMax_32f_C3R`.

- [NppStatus nppiMinMaxGetBufferHostSize_8u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMinMax_8u_AC4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_16u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMinMax_16u_AC4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_16s_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMinMax_16s_AC4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_32f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMinMax_32f_AC4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_8u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMinMax_8u_C4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_16u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMinMax_16u_C4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_16s_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMinMax_16s_C4R](#).
- [NppStatus nppiMinMaxGetBufferHostSize_32f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Buffer size for [nppiMinMax_32f_C4R](#).

7.98.1 Detailed Description

Primitives for computing both the minimal and the maximal values of an image.

7.98.2 Function Documentation

7.98.2.1 [NppStatus nppiMinMax_16s_AC4R](#) (const [Npp16s](#) * *pSrc*, int *nSrcStep*, [NppiSize](#) *oSizeROI*, [Npp16s](#) *aMin*[3], [Npp16s](#) *aMax*[3], [Npp8u](#) * *pDeviceBuffer*)

Four-channel 16-bit signed image MinMax ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16s_AC4R](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.2 NppStatus nppiMinMax_16s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp16s * *pMin*, Npp16s * *pMax*, Npp8u * *pDeviceBuffer*)

One-channel 16-bit signed image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pMin Pointer to the computed minimal result.

pMax Pointer to the computed maximal result.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16s_C1R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.3 NppStatus nppiMinMax_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp16s *aMin*[3], Npp16s *aMax*[3], Npp8u * *pDeviceBuffer*)

Three-channel 16-bit signed image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16s_C3R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.4 NppStatus nppiMinMax_16s_C4R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp16s *aMin*[4], Npp16s *aMax*[4], Npp8u * *pDeviceBuffer*)

Four-channel 16-bit signed image MinMax.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16s_C4R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.5 NppStatus nppiMinMax_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u aMin[3], Npp16u aMax[3], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image MinMax ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16u_AC4R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.6 NppStatus nppiMinMax_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u * pMin, Npp16u * pMax, Npp8u * pDeviceBuffer)

One-channel 16-bit unsigned image MinMax.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMin Pointer to the computed minimal result.

pMax Pointer to the computed maximal result.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16u_C1R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.7 NppStatus nppiMinMax_16u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp16u *aMin*[3], Npp16u *aMax*[3], Npp8u * *pDeviceBuffer*)

Three-channel 16-bit unsigned image MinMax.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16u_C3R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.8 NppStatus nppiMinMax_16u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp16u *aMin*[4], Npp16u *aMax*[4], Npp8u * *pDeviceBuffer*)

Four-channel 16-bit unsigned image MinMax.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_16u_C4R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.9 NppStatus nppiMinMax_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32f *aMin*[3], Npp32f *aMax*[3], Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image MinMax ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_32f_AC4R](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.10 NppStatus nppiMinMax_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f * pMin, Npp32f * pMax, Npp8u * pDeviceBuffer)

One-channel 32-bit floating point image MinMax.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMin Pointer to the computed minimal result.

pMax Pointer to the computed maximal result.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_32f_C1R](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.11 NppStatus nppiMinMax_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f aMin[3], Npp32f aMax[3], Npp8u * pDeviceBuffer)

Three-channel 32-bit floating point image MinMax.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_32f_C3R](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.12 **NppStatus nppiMinMax_32f_C4R** (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32f *aMin*[4], Npp32f *aMax*[4], Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image MinMax.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_32f_C4R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.13 **NppStatus nppiMinMax_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u *aMin*[3], Npp8u *aMax*[3], Npp8u * *pDeviceBuffer*)

Four-channel 8-bit unsigned image MinMax ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_8u_AC4R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.14 **NppStatus nppiMinMax_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pMin*, Npp8u * *pMax*, Npp8u * *pDeviceBuffer*)

One-channel 8-bit unsigned image MinMax.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMin Pointer to the computed minimal result.

pMax Pointer to the computed maximal result.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_8u_C1R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.15 `NppStatus nppiMinMax_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[3], Npp8u aMax[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image MinMax.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_8u_C3R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.98.2.16 `NppStatus nppiMinMax_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[4], Npp8u aMax[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image MinMax.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aMin Array that contains the minima.

aMax Array that contains the maxima.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize_8u_C4R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.98.2.17 NppStatus nppiMinMaxGetBufferHostSize_16s_AC4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMinMax_16s_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.18 NppStatus nppiMinMaxGetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMinMax_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.19 NppStatus nppiMinMaxGetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMinMax_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.20 NppStatus nppiMinMaxGetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMinMax_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.98.2.21 NppStatus nppiMinMaxGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.98.2.22 NppStatus nppiMinMaxGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.98.2.23 NppStatus nppiMinMaxGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.98.2.24 NppStatus nppiMinMaxGetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMax_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.98.2.25 NppStatus nppiMinMaxGetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMax_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.98.2.26 NppStatus nppiMinMaxGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMax_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.98.2.27 NppStatus nppiMinMaxGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMax_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.98.2.28 NppStatus nppiMinMaxGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.98.2.29 NppStatus nppiMinMaxGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.98.2.30 NppStatus nppiMinMaxGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMax_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.31 NppStatus nppiMinMaxGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMinMax_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.32 NppStatus nppiMinMaxGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMinMax_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99 MinMaxIndx

Primitives for computing the minimal and the maximal values with their indices (X and Y coordinates) of an image.

MinMaxIndx

If there are several minima and maxima in the selected region of interest, the function returns ones on the top leftmost position.

The scratch buffer is required by the functions.

- `NppStatus nppiMinMaxIndx_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pMinValue, `Npp8u` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit unsigned char image.

- `NppStatus nppiMinMaxIndx_8s_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8s` *pMinValue, `Npp8s` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit signed char image.

- `NppStatus nppiMinMaxIndx_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` *pMinValue, `Npp16u` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 16-bit unsigned short image.

- `NppStatus nppiMinMaxIndx_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` *pMinValue, `Npp32f` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 32-bit floating point image.

Masked MinMaxIndx

See [Masked Operation](#).

- `NppStatus nppiMinMaxIndx_8u_C1MR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pMinValue, `Npp8u` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)

Masked one-channel 8-bit unsigned image MinMaxIndx.

- `NppStatus nppiMinMaxIndx_8s_C1MR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8s` *pMinValue, `Npp8s` *pMaxValue, `NppiPoint` *pMinIndex, `NppiPoint` *pMaxIndex, `Npp8u` *pDeviceBuffer)

Masked one-channel 8-bit signed image MinMaxIndx.

- [NppStatus nppiMinMaxIdx_16u_C1MR](#) (const [Npp16u](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp16u](#) *pMinValue, [Npp16u](#) *pMaxValue, [NppiPoint](#) *pMinIndex, [NppiPoint](#) *pMaxIndex, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 16-bit unsigned image MinMaxIdx.

- [NppStatus nppiMinMaxIdx_32f_C1MR](#) (const [Npp32f](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp32f](#) *pMinValue, [Npp32f](#) *pMaxValue, [NppiPoint](#) *pMinIndex, [NppiPoint](#) *pMaxIndex, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 32-bit floating point image MinMaxIdx.

Channel MinMaxIdx

See [Channel-of-Interest API](#).

- [NppStatus nppiMinMaxIdx_8u_C3CR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) *pMinValue, [Npp8u](#) *pMaxValue, [NppiPoint](#) *pMinIndex, [NppiPoint](#) *pMaxIndex, [Npp8u](#) *pDeviceBuffer)

Three-channel 8-bit unsigned image MinMaxIdx affecting only single channel.

- [NppStatus nppiMinMaxIdx_8s_C3CR](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8s](#) *pMinValue, [Npp8s](#) *pMaxValue, [NppiPoint](#) *pMinIndex, [NppiPoint](#) *pMaxIndex, [Npp8u](#) *pDeviceBuffer)

Three-channel 8-bit signed image MinMaxIdx affecting only single channel.

- [NppStatus nppiMinMaxIdx_16u_C3CR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp16u](#) *pMinValue, [Npp16u](#) *pMaxValue, [NppiPoint](#) *pMinIndex, [NppiPoint](#) *pMaxIndex, [Npp8u](#) *pDeviceBuffer)

Three-channel 16-bit unsigned image MinMaxIdx affecting only single channel.

- [NppStatus nppiMinMaxIdx_32f_C3CR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp32f](#) *pMinValue, [Npp32f](#) *pMaxValue, [NppiPoint](#) *pMinIndex, [NppiPoint](#) *pMaxIndex, [Npp8u](#) *pDeviceBuffer)

Three-channel 32-bit floating point image MinMaxIdx affecting only single channel.

Masked Channel MinMaxIdx

See [Masked Operation](#) and [Channel-of-Interest API](#).

- [NppStatus nppiMinMaxIdx_8u_C3CMR](#) (const [Npp8u](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) *pMinValue, [Npp8u](#) *pMaxValue, [NppiPoint](#) *pMinIndex, [NppiPoint](#) *pMaxIndex, [Npp8u](#) *pDeviceBuffer)

Masked three-channel 8-bit unsigned image MinMaxIdx affecting only single channel.

- [NppStatus nppiMinMaxIdx_8s_C3CMR](#) (const [Npp8s](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8s](#) *pMinValue, [Npp8s](#) *pMaxValue, [NppiPoint](#) *pMinIndex, [NppiPoint](#) *pMaxIndex, [Npp8u](#) *pDeviceBuffer)

Masked three-channel 8-bit signed image MinMaxIdx affecting only single channel.

- **NppStatus** **nppiMinMaxIndx_16u_C3CMR** (const **Npp16u** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp16u** *pMinValue, **Npp16u** *pMaxValue, **NppiPoint** *pMinIndex, **NppiPoint** *pMaxIndex, **Npp8u** *pDeviceBuffer)

Masked three-channel 16-bit unsigned image MinMaxIndx affecting only single channel.

- **NppStatus** **nppiMinMaxIndx_32f_C3CMR** (const **Npp32f** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp32f** *pMinValue, **Npp32f** *pMaxValue, **NppiPoint** *pMinIndex, **NppiPoint** *pMaxIndex, **Npp8u** *pDeviceBuffer)

Masked three-channel 32-bit floating point image MinMaxIndx affecting only single channel.

MinMaxIndxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MinMaxIndx primitives.

- **NppStatus** **nppiMinMaxIndxGetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiMinMaxIndx_8u_C1R**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize_8s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiMinMaxIndx_8s_C1R**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiMinMaxIndx_16u_C1R**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiMinMaxIndx_32f_C1R**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize_8u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiMinMaxIndx_8u_C1MR**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize_8s_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiMinMaxIndx_8s_C1MR**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize_16u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiMinMaxIndx_16u_C1MR**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize_32f_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiMinMaxIndx_32f_C1MR**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize_8u_C3CR** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiMinMaxIndx_8u_C3CR**.*

- **NppStatus** [nppiMinMaxIdxGetBufferHostSize_8s_C3CR](#) ([NppiSize](#) [oSizeROI](#), [int](#) *[hpBufferSize](#))
Buffer size for [nppiMinMaxIdx_8s_C3CR](#).
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize_16u_C3CR](#) ([NppiSize](#) [oSizeROI](#), [int](#) *[hpBufferSize](#))
Buffer size for [nppiMinMaxIdx_16u_C3CR](#).
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize_32f_C3CR](#) ([NppiSize](#) [oSizeROI](#), [int](#) *[hpBufferSize](#))
Buffer size for [nppiMinMaxIdx_32f_C3CR](#).
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize_8u_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) *[hpBufferSize](#))
Buffer size for [nppiMinMaxIdx_8u_C3CMR](#).
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize_8s_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) *[hpBufferSize](#))
Buffer size for [nppiMinMaxIdx_8s_C3CMR](#).
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize_16u_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) *[hpBufferSize](#))
Buffer size for [nppiMinMaxIdx_16u_C3CMR](#).
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize_32f_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) *[hpBufferSize](#))
Buffer size for [nppiMinMaxIdx_32f_C3CMR](#).

7.99.1 Detailed Description

Primitives for computing the minimal and the maximal values with their indices (X and Y coordinates) of an image.

7.99.2 Function Documentation

7.99.2.1 **NppStatus** [nppiMinMaxIdx_16u_C1MR](#) ([const](#) [Npp16u](#) * [pSrc](#), [int](#) [nSrcStep](#), [const](#) [Npp8u](#) * [pMask](#), [int](#) [nMaskStep](#), [NppiSize](#) [oSizeROI](#), [Npp16u](#) * [pMinValue](#), [Npp16u](#) * [pMaxValue](#), [NppiPoint](#) * [pMinIndex](#), [NppiPoint](#) * [pMaxIndex](#), [Npp8u](#) * [pDeviceBuffer](#))

Masked one-channel 16-bit unsigned image MinMaxIdx.

Parameters:

- [pSrc](#) [Source-Image Pointer](#).
- [nSrcStep](#) [Source-Image Line Step](#).
- [pMask](#) [Mask-Image Pointer](#).
- [nMaskStep](#) [Mask-Image Line Step](#).
- [oSizeROI](#) [Region-of-Interest \(ROI\)](#).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_16u_C1MR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e., *pMinIndex* = {0, 0}, *pMaxIndex* = {0, 0}, *pMinValue* = 0, *pMaxValue* = 0. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.99.2.2 NppStatus nppiMinMaxIndx_16u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp16u * *pMinValue*, Npp16u * *pMaxValue*, NppiPoint * *pMinIndex*, NppiPoint * *pMaxIndex*, Npp8u * *pDeviceBuffer*)

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 16-bit unsigned short image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_16u_C1R](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.99.2.3 NppStatus nppiMinMaxIndx_16u_C3CMR (const Npp16u * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp16u * *pMinValue*, Npp16u * *pMaxValue*, NppiPoint * *pMinIndex*, NppiPoint * *pMaxIndex*, Npp8u * *pDeviceBuffer*)

Masked three-channel 16-bit unsigned image MinMaxIndx affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_16u_C3CMR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

7.99.2.4 `NppStatus nppiMinMaxIndx_16u_C3CR (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp16u * pMinValue, Npp16u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image MinMaxIndx affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_16u_C3CR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

7.99.2.5 `NppStatus nppiMinMaxIndx_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image MinMaxIndx.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pMinValue Pointer to the minimum value.
pMaxValue Pointer to the maximum value.
pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.
pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.
pDeviceBuffer Buffer to a scratch memory. Use `nppiMinMaxIndxGetBufferHostSize_32f_C1MR` to determine the minimum number of bytes required.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

7.99.2.6 `NppStatus nppiMinMaxIndx_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 32-bit floating point image.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pMinValue Pointer to the minimum value.
pMaxValue Pointer to the maximum value.
pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.
pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.
pDeviceBuffer Buffer to a scratch memory. Use `nppiMinMaxIndxGetBufferHostSize_32f_C1R` to determine the minimum number of bytes required.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

7.99.2.7 `NppStatus nppiMinMaxIndx_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image MinMaxIndx affecting only single channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nCOI Channel_of_Interest Number.
pMinValue Pointer to the minimum value.
pMaxValue Pointer to the maximum value.
pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.
pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.
pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_32f_C3CMR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_COI_ERROR if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., pMinIndex = {0, 0}, pMaxIndex = {0, 0}, pMinValue = 0, pMaxValue = 0. If any of pMinValue, pMaxValue, pMinIndex, or pMaxIndex is not needed, zero pointer must be passed correspondingly.

7.99.2.8 `NppStatus nppiMinMaxIndx_32f_C3CR (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image MinMaxIndx affecting only single channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nCOI Channel_of_Interest Number.
pMinValue Pointer to the minimum value.
pMaxValue Pointer to the maximum value.
pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.
pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.
pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_32f_C3CR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_COI_ERROR if an invalid channel of interest is specified. If any of pMinValue, pMaxValue, pMinIndex, or pMaxIndex is not needed, zero pointer must be passed correspondingly.

7.99.2.9 `NppStatus nppiMinMaxIndx_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image MinMaxIndx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8s_C1MR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e., pMinIndex = {0, 0}, pMaxIndex = {0, 0}, pMinValue = 0, pMaxValue = 0. If any of pMinValue, pMaxValue, pMinIndex, or pMaxIndex is not needed, zero pointer must be passed correspondingly.

7.99.2.10 `NppStatus nppiMinMaxIndx_8s_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit signed char image.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8s_C1R](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.99.2.11 `NppStatus nppiMinMaxIndx_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image MinMaxIndx affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8s_C3CMR](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., *pMinIndex* = {0, 0}, *pMaxIndex* = {0, 0}, *pMinValue* = 0, *pMaxValue* = 0. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.99.2.12 `NppStatus nppiMinMaxIndx_8s_C3CR (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image MinMaxIndx affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8s_C3CR](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.99.2.13 `NppStatus nppiMinMaxIndx_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image MinMaxIndx.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pMinValue Pointer to the minimum value.

pMaxValue Pointer to the maximum value.

pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.

pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.

pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8u_C1MR](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e., *pMinIndex* = {0, 0}, *pMaxIndex* = {0, 0}, *pMinValue* = 0, *pMaxValue* = 0. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.99.2.14 `NppStatus nppiMinMaxIndx_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit unsigned char image.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pMinValue Pointer to the minimum value.
pMaxValue Pointer to the maximum value.
pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.
pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.
pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8u_C1R](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.99.2.15 `NppStatus nppiMinMaxIndx_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image MinMaxIndx affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pMask [Mask-Image Pointer](#).
nMaskStep [Mask-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nCOI [Channel_of_Interest Number](#).
pMinValue Pointer to the minimum value.
pMaxValue Pointer to the maximum value.
pMinIndex Pointer to the indices (X and Y coordinates) of the minimum value.
pMaxIndex Pointer to the indices (X and Y coordinates) of the maximum value.
pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8u_C3CMR](#) to determine the minimum number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., *pMinIndex* = {0, 0}, *pMaxIndex* = {0, 0}, *pMinValue* = 0, *pMaxValue* = 0. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.99.2.16 `NppStatus nppiMinMaxIndx_8u_C3CR (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image MinMaxIndx affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nCOI [Channel_of_Interest Number](#).
pMinValue Pointer to the minimum value.
pMaxValue Pointer to the maximum value.
pMinIndex Pointer to the indicies (X and Y coordinates) of the minimum value.
pMaxIndex Pointer to the indicies (X and Y coordinates) of the maximum value.
pDeviceBuffer Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize_8u_C3CR](#) to determine the minium number of bytes required.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

7.99.2.17 `NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMinMaxIndx_16u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).
hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.18 `NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMinMaxIndx_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).
hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.19 NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_16u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.20 NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C3CR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_16u_C3CR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.21 NppStatus nppiMinMaxIndxGetBufferHostSize_32f_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_32f_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.22 NppStatus nppiMinMaxIndxGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.23 NppStatus nppiMinMaxIndxGetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_32f_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.24 NppStatus nppiMinMaxIndxGetBufferHostSize_32f_C3CR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_32f_C3CR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.25 NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_8s_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.26 NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_8s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.27 NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_8s_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.28 NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C3CR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_8s_C3CR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.29 NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMinMaxIndx_8u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.30 NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.31 NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_8u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.99.2.32 NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C3CR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMinMaxIndx_8u_C3CR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.100 Mean

Primitives for computing the arithmetic mean of all the pixel values in an image.

Mean

Given an image $pSrc$ with width W and height H , the arithmetic mean will be computed as

$$Mean = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

The mean functions require additional scratch buffer for computations.

- `NppStatus nppiMean_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
One-channel 8-bit unsigned image Mean.
- `NppStatus nppiMean_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
One-channel 16-bit unsigned image Mean.
- `NppStatus nppiMean_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
One-channel 16-bit signed image Mean.
- `NppStatus nppiMean_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
One-channel 32-bit floating point image Mean.
- `NppStatus nppiMean_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Three-channel 8-bit unsigned image Mean.
- `NppStatus nppiMean_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Three-channel 16-bit unsigned image Mean.
- `NppStatus nppiMean_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Three-channel 16-bit signed image Mean.
- `NppStatus nppiMean_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[3])
Three-channel 32-bit floating point image Mean.
- `NppStatus nppiMean_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` aMean[4])
Four-channel 8-bit unsigned image Mean.

- [NppStatus](#) [nppiMean_16u_C4R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) aMean[4])
Four-channel 16-bit unsigned image Mean.
- [NppStatus](#) [nppiMean_16s_C4R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) aMean[4])
Four-channel 16-bit signed image Mean.
- [NppStatus](#) [nppiMean_32f_C4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) aMean[4])
Four-channel 32-bit floating point image Mean.
- [NppStatus](#) [nppiMean_8u_AC4R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) aMean[3])
Four-channel 8-bit unsigned image Mean ignoring alpha channel.
- [NppStatus](#) [nppiMean_16u_AC4R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) aMean[3])
Four-channel 16-bit unsigned image Mean ignoring alpha channel.
- [NppStatus](#) [nppiMean_16s_AC4R](#) (const [Npp16s](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) aMean[3])
Four-channel 16-bit signed image Mean ignoring alpha channel.
- [NppStatus](#) [nppiMean_32f_AC4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) aMean[3])
Four-channel 32-bit floating point image Mean ignoring alpha channel.

Masked Mean

See [Masked Operation](#).

- [NppStatus](#) [nppiMean_8u_C1MR](#) (const [Npp8u](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean)
Masked one-channel 8-bit unsigned image Mean.
- [NppStatus](#) [nppiMean_8s_C1MR](#) (const [Npp8s](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean)
Masked one-channel 8-bit signed image Mean.
- [NppStatus](#) [nppiMean_16u_C1MR](#) (const [Npp16u](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean)
Masked one-channel 16-bit unsigned image Mean.
- [NppStatus](#) [nppiMean_32f_C1MR](#) (const [Npp32f](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean)
Masked one-channel 32-bit floating point image Mean.

Masked Channel Mean

See [Channel-of-Interest API](#) and [Masked Operation](#).

- `NppStatus nppiMean_8u_C3CMR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked three-channel 8-bit unsigned image Mean affecting only single channel.
- `NppStatus nppiMean_8s_C3CMR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked three-channel 8-bit signed image Mean affecting only single channel.
- `NppStatus nppiMean_16u_C3CMR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked three-channel 16-bit unsigned image Mean affecting only single channel.
- `NppStatus nppiMean_32f_C3CMR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean)
Masked three-channel 32-bit floating point image Mean affecting only single channel.

MeanGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean primitives.

- `NppStatus nppiMeanGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_8u_C1R`.
- `NppStatus nppiMeanGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16u_C1R`.
- `NppStatus nppiMeanGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16s_C1R`.
- `NppStatus nppiMeanGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_32f_C1R`.
- `NppStatus nppiMeanGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_8u_C3R`.
- `NppStatus nppiMeanGetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16u_C3R`.
- `NppStatus nppiMeanGetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16s_C3R`.
- `NppStatus nppiMeanGetBufferHostSize_32f_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_32f_C3R`.
- `NppStatus nppiMeanGetBufferHostSize_8u_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiMean_8u_AC4R`.

- `NppStatus nppiMeanGetBufferHostSize_16u_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16u_AC4R`.
- `NppStatus nppiMeanGetBufferHostSize_16s_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16s_AC4R`.
- `NppStatus nppiMeanGetBufferHostSize_32f_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_32f_AC4R`.
- `NppStatus nppiMeanGetBufferHostSize_8u_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_8u_C4R`.
- `NppStatus nppiMeanGetBufferHostSize_16u_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16u_C4R`.
- `NppStatus nppiMeanGetBufferHostSize_16s_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16s_C4R`.
- `NppStatus nppiMeanGetBufferHostSize_32f_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_32f_C4R`.
- `NppStatus nppiMeanGetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_8u_C1MR`.
- `NppStatus nppiMeanGetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_8s_C1MR`.
- `NppStatus nppiMeanGetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16u_C1MR`.
- `NppStatus nppiMeanGetBufferHostSize_32f_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_32f_C1MR`.
- `NppStatus nppiMeanGetBufferHostSize_8u_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_8u_C3CMR`.
- `NppStatus nppiMeanGetBufferHostSize_8s_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_8s_C3CMR`.
- `NppStatus nppiMeanGetBufferHostSize_16u_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_16u_C3CMR`.
- `NppStatus nppiMeanGetBufferHostSize_32f_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiMean_32f_C3CMR`.

7.100.1 Detailed Description

Primitives for computing the arithmetic mean of all the pixel values in an image.

7.100.2 Function Documentation

7.100.2.1 NppStatus nppiMean_16s_AC4R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f *aMean*[3])

Four-channel 16-bit signed image Mean ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16s_AC4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.2 NppStatus nppiMean_16s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*)

One-channel 16-bit signed image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16s_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.3 NppStatus nppiMean_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f *aMean*[3])

Three-channel 16-bit signed image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16s_C3R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.4 `NppStatus nppiMean_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[4])`

Four-channel 16-bit signed image Mean.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16s_C4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.5 `NppStatus nppiMean_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])`

Four-channel 16-bit unsigned image Mean ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16u_AC4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.6 `NppStatus nppiMean_16u_C1MR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked one-channel 16-bit unsigned image Mean.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16u_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.100.2.7 `NppStatus nppiMean_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

One-channel 16-bit unsigned image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16u_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.8 `NppStatus nppiMean_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked three-channel 16-bit unsigned image Mean affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16u_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.100.2.9 NppStatus nppiMean_16u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f *aMean*[3])

Three-channel 16-bit unsigned image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16u_C3R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.10 NppStatus nppiMean_16u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f *aMean*[4])

Four-channel 16-bit unsigned image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_16u_C4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.11 NppStatus nppiMean_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f *aMean*[3])

Four-channel 32-bit floating point image Mean ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_32f_AC4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.100.2.12 `NppStatus nppiMean_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked one-channel 32-bit floating point image Mean.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_32f_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.100.2.13 `NppStatus nppiMean_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

One-channel 32-bit floating point image Mean.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_32f_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.100.2.14 `NppStatus nppiMean_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked three-channel 32-bit floating point image Mean affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_32f_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_COI_ERROR if an invalid channel of interest is specified.

7.100.2.15 `NppStatus nppiMean_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])`

Three-channel 32-bit floating point image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_32f_C3R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.100.2.16 `NppStatus nppiMean_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[4])`

Four-channel 32-bit floating point image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_32f_C4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.100.2.17 `NppStatus nppiMean_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked one-channel 8-bit signed image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8s_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.100.2.18 `NppStatus nppiMean_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked three-channel 8-bit signed image Mean affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8s_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified.

7.100.2.19 NppStatus nppiMean_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])

Four-channel 8-bit unsigned image Mean ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8u_AC4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.20 NppStatus nppiMean_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)

Masked one-channel 8-bit unsigned image Mean.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8u_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.100.2.21 **NppStatus nppiMean_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*)

One-channel 8-bit unsigned image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8u_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.22 **NppStatus nppiMean_8u_C3CMR** (const Npp8u * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*)

Masked three-channel 8-bit unsigned image Mean affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8u_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean result.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified.

7.100.2.23 **NppStatus nppiMean_8u_C3R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f *aMean*[3])

Three-channel 8-bit unsigned image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8u_C3R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.24 `NppStatus nppiMean_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[4])`

Four-channel 8-bit unsigned image Mean.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize_8u_C4R](#) to determine the minium number of bytes required.

aMean Array that contains the computed mean results.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.100.2.25 `NppStatus nppiMeanGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMean_16s_AC4R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.26 `NppStatus nppiMeanGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMean_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.27 NppStatus nppiMeanGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.28 NppStatus nppiMeanGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.29 NppStatus nppiMeanGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.30 NppStatus nppiMeanGetBufferHostSize_16u_C1MR (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMean_16u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.31 NppStatus nppiMeanGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMean_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.32 NppStatus nppiMeanGetBufferHostSize_16u_C3CMR (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMean_16u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.33 NppStatus nppiMeanGetBufferHostSize_16u_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMean_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.34 NppStatus nppiMeanGetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.35 NppStatus nppiMeanGetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.36 NppStatus nppiMeanGetBufferHostSize_32f_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_32f_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.37 NppStatus nppiMeanGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMean_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.38 NppStatus nppiMeanGetBufferHostSize_32f_C3CMR (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMean_32f_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.39 NppStatus nppiMeanGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMean_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.40 NppStatus nppiMeanGetBufferHostSize_32f_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMean_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.41 NppStatus nppiMeanGetBufferHostSize_8s_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_8s_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.42 NppStatus nppiMeanGetBufferHostSize_8s_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_8s_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.43 NppStatus nppiMeanGetBufferHostSize_8u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.44 NppStatus nppiMeanGetBufferHostSize_8u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_8u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.45 NppStatus nppiMeanGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.46 NppStatus nppiMeanGetBufferHostSize_8u_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_8u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.100.2.47 NppStatus nppiMeanGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.48 NppStatus nppiMeanGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiMean_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101 Mean_StdDev

Primitives for computing both the arithmetic mean and the standard deviation of an image.

Mean_StdDev

Given an image $pSrc$ with width W and height H , the mean and the standard deviation will be computed as

$$Mean = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

$$StdDev = \sqrt{\frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} (pSrc(j, i) - Mean)^2}$$

The Mean_StdDev primitives require additional scratch buffer for computations.

- `NppStatus nppiMean_StdDev_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
One-channel 8-bit unsigned image Mean_StdDev.
- `NppStatus nppiMean_StdDev_8s_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
One-channel 8-bit signed image Mean_StdDev.
- `NppStatus nppiMean_StdDev_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
One-channel 16-bit unsigned image Mean_StdDev.
- `NppStatus nppiMean_StdDev_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
One-channel 32-bit floating point image Mean_StdDev.

Masked Mean_StdDev

See [Masked Operation](#).

- `NppStatus nppiMean_StdDev_8u_C1MR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
Masked one-channel 8-bit unsigned image Mean_StdDev.
- `NppStatus nppiMean_StdDev_8s_C1MR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
Masked one-channel 8-bit signed image Mean_StdDev.
- `NppStatus nppiMean_StdDev_16u_C1MR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` *pDeviceBuffer, `Npp64f` *pMean, `Npp64f` *pStdDev)
Masked one-channel 16-bit unsigned image Mean_StdDev.

- [NppStatus nppiMean_StdDev_32f_C1MR](#) (const [Npp32f](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean, [Npp64f](#) *pStdDev)

Masked one-channel 32-bit floating point image Mean_StdDev.

Channel Mean_StdDev

See [Channel-of-Interest API](#).

- [NppStatus nppiMean_StdDev_8u_C3CR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean, [Npp64f](#) *pStdDev)

Three-channel 8-bit unsigned image Mean_StdDev affecting only single channel.

- [NppStatus nppiMean_StdDev_8s_C3CR](#) (const [Npp8s](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean, [Npp64f](#) *pStdDev)

Three-channel 8-bit signed image Mean_StdDev affecting only single channel.

- [NppStatus nppiMean_StdDev_16u_C3CR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean, [Npp64f](#) *pStdDev)

Three-channel 16-bit unsigned image Mean_StdDev affecting only single channel.

- [NppStatus nppiMean_StdDev_32f_C3CR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean, [Npp64f](#) *pStdDev)

Three-channel 32-bit floating point image Mean_StdDev affecting only single channel.

Masked Channel Mean_StdDev

See [Masked Operation](#) and [Channel-of-Interest API](#).

- [NppStatus nppiMean_StdDev_8u_C3CMR](#) (const [Npp8u](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean, [Npp64f](#) *pStdDev)

Masked three-channel 8-bit unsigned image Mean_StdDev.

- [NppStatus nppiMean_StdDev_8s_C3CMR](#) (const [Npp8s](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean, [Npp64f](#) *pStdDev)

Masked three-channel 8-bit signed image Mean_StdDev.

- [NppStatus nppiMean_StdDev_16u_C3CMR](#) (const [Npp16u](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean, [Npp64f](#) *pStdDev)

Masked three-channel 16-bit unsigned image Mean_StdDev.

- [NppStatus nppiMean_StdDev_32f_C3CMR](#) (const [Npp32f](#) *pSrc, int nSrcStep, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) *pDeviceBuffer, [Npp64f](#) *pMean, [Npp64f](#) *pStdDev)

Masked three-channel 32-bit floating point image Mean_StdDev.

MeanStdDevGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean_StdDev primitives.

- **NppStatus** **nppiMeanStdDevGetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_8u_C1R*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize_8s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_8s_C1R*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_16u_C1R*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_32f_C1R*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize_8u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_8u_C1MR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize_8s_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_8s_C1MR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize_16u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_16u_C1MR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize_32f_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_32f_C1MR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize_8u_C3CR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_8u_C3CR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize_8s_C3CR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_8s_C3CR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize_16u_C3CR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_16u_C3CR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize_32f_C3CR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for *nppiMean_StdDev_32f_C3CR*.

- [NppStatus](#) [nppiMeanStdDevGetBufferHostSize_8u_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMean_StdDev_8u_C3CMR](#).
- [NppStatus](#) [nppiMeanStdDevGetBufferHostSize_8s_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMean_StdDev_8s_C3CMR](#).
- [NppStatus](#) [nppiMeanStdDevGetBufferHostSize_16u_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMean_StdDev_16u_C3CMR](#).
- [NppStatus](#) [nppiMeanStdDevGetBufferHostSize_32f_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Buffer size for [nppiMean_StdDev_32f_C3CMR](#).

7.101.1 Detailed Description

Primitives for computing both the arithmetic mean and the standard deviation of an image.

7.101.2 Function Documentation

7.101.2.1 [NppStatus](#) [nppiMean_StdDev_16u_C1MR](#) ([const](#) [Npp16u](#) * [pSrc](#), [int](#) [nSrcStep](#), [const](#) [Npp8u](#) * [pMask](#), [int](#) [nMaskStep](#), [NppiSize](#) [oSizeROI](#), [Npp8u](#) * [pDeviceBuffer](#), [Npp64f](#) * [pMean](#), [Npp64f](#) * [pStdDev](#))

Masked one-channel 16-bit unsigned image Mean_StdDev.

Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[pMask](#) [Mask-Image Pointer](#).

[nMaskStep](#) [Mask-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_16u_C1MR](#) to determine the minium number of bytes required.

[pMean](#) Pointer to the computed mean.

[pStdDev](#) Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.101.2.2 NppStatus nppiMean_StdDev_16u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

One-channel 16-bit unsigned image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize_16u_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.101.2.3 NppStatus nppiMean_StdDev_16u_C3CMR (const Npp16u * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

Masked three-channel 16-bit unsigned image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize_16u_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified.

7.101.2.4 NppStatus nppiMean_StdDev_16u_C3CR (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

Three-channel 16-bit unsigned image Mean_StdDev affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_16u_C3CR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified.

7.101.2.5 NppStatus nppiMean_StdDev_32f_C1MR (const Npp32f * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

Masked one-channel 32-bit floating point image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_32f_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.101.2.6 NppStatus nppiMean_StdDev_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

One-channel 32-bit floating point image Mean_StdDev.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize_32f_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.101.2.7 NppStatus nppiMean_StdDev_32f_C3CMR (const Npp32f * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

Masked three-channel 32-bit floating point image Mean_StdDev.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize_32f_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_COI_ERROR if an invalid channel of interest is specified.

7.101.2.8 NppStatus nppiMean_StdDev_32f_C3CR (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

Three-channel 32-bit floating point image Mean_StdDev affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_32f_C3CR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_COI_ERROR if an invalid channel of interest is specified.

7.101.2.9 NppStatus nppiMean_StdDev_8s_C1MR (const Npp8s * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

Masked one-channel 8-bit signed image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8s_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.101.2.10 NppStatus nppiMean_StdDev_8s_C1R (const Npp8s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

One-channel 8-bit signed image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8s_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.101.2.11 NppStatus nppiMean_StdDev_8s_C3CMR (const Npp8s * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

Masked three-channel 8-bit signed image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8s_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified.

7.101.2.12 **NppStatus nppiMean_StdDev_8s_C3CR** (const Npp8s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

Three-channel 8-bit signed image Mean_StdDev affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8s_C3CR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified.

7.101.2.13 **NppStatus nppiMean_StdDev_8u_C1MR** (const Npp8u * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

Masked one-channel 8-bit unsigned image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8u_C1MR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.101.2.14 NppStatus nppiMean_StdDev_8u_C1R (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

One-channel 8-bit unsigned image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8u_C1R](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.101.2.15 NppStatus nppiMean_StdDev_8u_C3CMR (const Npp8u * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

Masked three-channel 8-bit unsigned image Mean_StdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8u_C3CMR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified.

7.101.2.16 NppStatus nppiMean_StdDev_8u_C3CR (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u * *pDeviceBuffer*, Npp64f * *pMean*, Npp64f * *pStdDev*)

Three-channel 8-bit unsigned image Mean_StdDev affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)
Use [nppiMeanStdDevGetBufferHostSize_8u_C3CR](#) to determine the minium number of bytes required.

pMean Pointer to the computed mean.

pStdDev Pointer to the computed standard deviation.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified.

7.101.2.17 NppStatus nppiMeanStdDevGetBufferHostSize_16u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_StdDev_16u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.18 NppStatus nppiMeanStdDevGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_StdDev_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.19 NppStatus nppiMeanStdDevGetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_16u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.20 NppStatus nppiMeanStdDevGetBufferHostSize_16u_C3CR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_16u_C3CR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.21 NppStatus nppiMeanStdDevGetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_32f_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.22 NppStatus nppiMeanStdDevGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.23 NppStatus nppiMeanStdDevGetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_32f_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.24 NppStatus nppiMeanStdDevGetBufferHostSize_32f_C3CR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_32f_C3CR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.25 NppStatus nppiMeanStdDevGetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_8s_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.26 NppStatus nppiMeanStdDevGetBufferHostSize_8s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_StdDev_8s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.27 NppStatus nppiMeanStdDevGetBufferHostSize_8s_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_StdDev_8s_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.28 NppStatus nppiMeanStdDevGetBufferHostSize_8s_C3CR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_StdDev_8s_C3CR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.29 NppStatus nppiMeanStdDevGetBufferHostSize_8u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiMean_StdDev_8u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.30 NppStatus nppiMeanStdDevGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.31 NppStatus nppiMeanStdDevGetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_8u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.101.2.32 NppStatus nppiMeanStdDevGetBufferHostSize_8u_C3CR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiMean_StdDev_8u_C3CR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.102 Image Norms

Primitives for computing the norms of an image, the norms of difference, and the relative errors of two images.

Modules

- [Norm_Inf](#)

Primitives for computing the infinity norm of an image.

- [Norm_L1](#)

Primitives for computing the L1 norm of an image.

- [Norm_L2](#)

Primitives for computing the L2 norm of an image.

- [NormDiff_Inf](#)

Primitives for computing the infinity norm of difference of pixels between two images.

- [NormDiff_L1](#)

Primitives for computing the L1 norm of difference of pixels between two images.

- [NormDiff_L2](#)

Primitives for computing the L2 norm of difference of pixels between two images.

- [NormRel_Inf](#)

Primitives for computing the relative error of infinity norm between two images.

- [NormRel_L1](#)

Primitives for computing the relative error of L1 norm between two images.

- [NormRel_L2](#)

Primitives for computing the relative error of L2 norm between two images.

7.102.1 Detailed Description

Primitives for computing the norms of an image, the norms of difference, and the relative errors of two images.

Given an image $pSrc$ with width W and height H ,

1. The infinity norm ([Norm_Inf](#)) is defined as the largest absolute pixel value of the image.
2. The L1 norm ([Norm_L1](#)) is defined as the sum of the absolute pixel value of the image, i.e.,

$$Norm_L1 = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc(j, i)|$$

3. The L2 norm (Norm_L2) is defined as the square root of the sum of the squared absolute pixel value of the image, i.e.,

$$Norm_L2 = \sqrt{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc(j, i)|^2}$$

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*,

1. The infinity norm of difference (NormDiff_Inf) is defined as the largest absolute difference between pixels of two images.
2. The L1 norm of difference (NormDiff_L1) is defined as the sum of the absolute difference between pixels of two images, i.e.,

$$NormDiff_L1 = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|$$

3. The L2 norm of difference (NormDiff_L2) is defined as the squared root of the sum of the squared absolute difference between pixels of two images, i.e.,

$$NormDiff_L2 = \sqrt{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|^2}$$

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*,

1. The relative error for the infinity norm of difference (NormRel_Inf) is defined as NormDiff_Inf divided by the infinity norm of the second image, i.e.,

$$NormRel_Inf = \frac{NormDiff_Inf}{Norm_Inf_{src2}}$$

2. The relative error for the L1 norm of difference (NormRel_L1) is defined as NormDiff_L1 divided by the L1 norm of the second image, i.e.,

$$NormRel_L1 = \frac{NormDiff_L1}{Norm_L1_{src2}}$$

3. The relative error for the L2 norm of difference (NormRel_L2) is defined as NormDiff_L2 divided by the L2 norm of the second image, i.e.,

$$NormRel_L2 = \frac{NormDiff_L2}{Norm_L2_{src2}}$$

The norm functions require the addition device scratch buffer for the computations.

7.103 Norm_Inf

Primitives for computing the infinity norm of an image.

Basic Norm_Inf

- **NppStatus nppiNorm_Inf_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image Norm_Inf.
- **NppStatus nppiNorm_Inf_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image Norm_Inf.
- **NppStatus nppiNorm_Inf_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image Norm_Inf.
- **NppStatus nppiNorm_Inf_32s_C1R** (const **Npp32s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
One-channel 32-bit signed image Norm_Inf.
- **NppStatus nppiNorm_Inf_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image Norm_Inf.
- **NppStatus nppiNorm_Inf_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image Norm_Inf.
- **NppStatus nppiNorm_Inf_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image Norm_Inf.
- **NppStatus nppiNorm_Inf_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image Norm_Inf.
- **NppStatus nppiNorm_Inf_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image Norm_Inf.
- **NppStatus nppiNorm_Inf_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image Norm_Inf ignoring alpha channel.
- **NppStatus nppiNorm_Inf_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image Norm_Inf ignoring alpha channel.

- `NppStatus nppiNorm_Inf_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit signed image Norm_Inf ignoring alpha channel.
- `NppStatus nppiNorm_Inf_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit floating point image Norm_Inf ignoring alpha channel.
- `NppStatus nppiNorm_Inf_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image Norm_Inf.
- `NppStatus nppiNorm_Inf_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit unsigned image Norm_Inf.
- `NppStatus nppiNorm_Inf_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit signed image Norm_Inf.
- `NppStatus nppiNorm_Inf_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit floating point image Norm_Inf.

Masked Norm_Inf

See [Masked Operation](#).

- `NppStatus nppiNorm_Inf_8u_C1MR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 8-bit unsigned image Norm_Inf.
- `NppStatus nppiNorm_Inf_8s_C1MR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 8-bit signed image Norm_Inf.
- `NppStatus nppiNorm_Inf_16u_C1MR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 16-bit unsigned image Norm_Inf.
- `NppStatus nppiNorm_Inf_32f_C1MR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 32-bit floating point image Norm_Inf.

Masked Channel Norm_Inf

See [Channel-of-Interest API](#) and [Masked Operation](#).

- **NppStatus nppiNorm_Inf_8u_C3CMR** (const **Npp8u** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
Masked three-channel 8-bit unsigned image Norm_Inf affecting only single channel.
- **NppStatus nppiNorm_Inf_8s_C3CMR** (const **Npp8s** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
Masked three-channel 8-bit signed image Norm_Inf affecting only single channel.
- **NppStatus nppiNorm_Inf_16u_C3CMR** (const **Npp16u** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
Masked three-channel 16-bit unsigned image Norm_Inf affecting only single channel.
- **NppStatus nppiNorm_Inf_32f_C3CMR** (const **Npp32f** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
Masked three-channel 32-bit floating point image Norm_Inf affecting only single channel.

NormInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm_Inf primitives.

- **NppStatus nppiNormInfGetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_8u_C1R.
- **NppStatus nppiNormInfGetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_16u_C1R.
- **NppStatus nppiNormInfGetBufferHostSize_16s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_16s_C1R.
- **NppStatus nppiNormInfGetBufferHostSize_32s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_32s_C1R.
- **NppStatus nppiNormInfGetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_32f_C1R.
- **NppStatus nppiNormInfGetBufferHostSize_8u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_8u_C1MR.
- **NppStatus nppiNormInfGetBufferHostSize_8s_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_8s_C1MR.
- **NppStatus nppiNormInfGetBufferHostSize_16u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_16u_C1MR.
- **NppStatus nppiNormInfGetBufferHostSize_32f_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
Buffer size for nppiNorm_Inf_32f_C1MR.
- **NppStatus nppiNormInfGetBufferHostSize_8u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size for *nppiNorm_Inf_8u_C3R*.

- [NppStatus nppiNormInfGetBufferHostSize_16u_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_16u_C3R*.
- [NppStatus nppiNormInfGetBufferHostSize_16s_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_16s_C3R*.
- [NppStatus nppiNormInfGetBufferHostSize_32f_C3R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_32f_C3R*.
- [NppStatus nppiNormInfGetBufferHostSize_8u_AC4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_8u_AC4R*.
- [NppStatus nppiNormInfGetBufferHostSize_16u_AC4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_16u_AC4R*.
- [NppStatus nppiNormInfGetBufferHostSize_16s_AC4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_16s_AC4R*.
- [NppStatus nppiNormInfGetBufferHostSize_32f_AC4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_32f_AC4R*.
- [NppStatus nppiNormInfGetBufferHostSize_8u_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_8u_C4R*.
- [NppStatus nppiNormInfGetBufferHostSize_16u_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_16u_C4R*.
- [NppStatus nppiNormInfGetBufferHostSize_16s_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_16s_C4R*.
- [NppStatus nppiNormInfGetBufferHostSize_32f_C4R](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_32f_C4R*.
- [NppStatus nppiNormInfGetBufferHostSize_8u_C3CMR](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_8u_C3CMR*.
- [NppStatus nppiNormInfGetBufferHostSize_8s_C3CMR](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_8s_C3CMR*.
- [NppStatus nppiNormInfGetBufferHostSize_16u_C3CMR](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_16u_C3CMR*.
- [NppStatus nppiNormInfGetBufferHostSize_32f_C3CMR](#) (NppiSize oSizeROI, int *hpBufferSize)
Buffer size for *nppiNorm_Inf_32f_C3CMR*.

7.103.1 Detailed Description

Primitives for computing the infinity norm of an image.

7.103.2 Function Documentation

7.103.2.1 NppStatus nppiNorm_Inf_16s_AC4R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u * *pDeviceBuffer*)

Four-channel 16-bit signed image Norm_Inf ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.2 NppStatus nppiNorm_Inf_16s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

One-channel 16-bit signed image Norm_Inf.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.3 NppStatus nppiNorm_Inf_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u * *pDeviceBuffer*)

Three-channel 16-bit signed image Norm_Inf.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.4 NppStatus nppiNorm_Inf_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit signed image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.5 NppStatus nppiNorm_Inf_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_Inf ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.6 `NppStatus nppiNorm_Inf_16u_C1MR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.7 `NppStatus nppiNorm_Inf_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.8 `NppStatus nppiNorm_Inf_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image Norm_Inf affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.103.2.9 NppStatus nppiNorm_Inf_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.10 NppStatus nppiNorm_Inf_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.11 **NppStatus nppiNorm_Inf_32f_AC4R** (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image Norm_Inf ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.12 **NppStatus nppiNorm_Inf_32f_C1MR** (const Npp32f * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

Masked one-channel 32-bit floating point image Norm_Inf.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.13 **NppStatus nppiNorm_Inf_32f_C1R** (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

One-channel 32-bit floating point image Norm_Inf.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.14 `NppStatus nppiNorm_Inf_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image Norm_Inf affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_COI_ERROR if an invalid channel of interest is specified.

7.103.2.15 `NppStatus nppiNorm_Inf_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Norm_Inf.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.16 NppStatus nppiNorm_Inf_32f_C4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[4], Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.17 NppStatus nppiNorm_Inf_32s_C1R (const Npp32s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

One-channel 32-bit signed image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_32s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.18 NppStatus nppiNorm_Inf_8s_C1MR (const Npp8s * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

Masked one-channel 8-bit signed image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.19 `NppStatus nppiNorm_Inf_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image Norm_Inf affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.103.2.20 `NppStatus nppiNorm_Inf_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Norm_Inf ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.21 `NppStatus nppiNorm_Inf_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormInfGetBufferHostSize_8u_C1MR` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.22 `NppStatus nppiNorm_Inf_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormInfGetBufferHostSize_8u_C1R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.23 `NppStatus nppiNorm_Inf_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image Norm_Inf affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified.

7.103.2.24 NppStatus nppiNorm_Inf_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Three-channel 8-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.25 NppStatus nppiNorm_Inf_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 8-bit unsigned image Norm_Inf.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormInfGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.103.2.26 NppStatus nppiNormInfGetBufferHostSize_16s_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_16s_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.27 NppStatus nppiNormInfGetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.28 NppStatus nppiNormInfGetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.29 NppStatus nppiNormInfGetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.30 NppStatus nppiNormInfGetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.31 NppStatus nppiNormInfGetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.32 NppStatus nppiNormInfGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.33 NppStatus nppiNormInfGetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.34 NppStatus nppiNormInfGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.35 NppStatus nppiNormInfGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.36 NppStatus nppiNormInfGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.37 NppStatus nppiNormInfGetBufferHostSize_32f_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_32f_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.38 NppStatus nppiNormInfGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.39 NppStatus nppiNormInfGetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_Inf_32f_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.40 NppStatus nppiNormInfGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.41 NppStatus nppiNormInfGetBufferHostSize_32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.42 NppStatus nppiNormInfGetBufferHostSize_32s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_32s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.43 NppStatus nppiNormInfGetBufferHostSize_8s_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_8s_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.44 NppStatus nppiNormInfGetBufferHostSize_8s_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_8s_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.45 NppStatus nppiNormInfGetBufferHostSize_8u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.46 NppStatus nppiNormInfGetBufferHostSize_8u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_8u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.47 NppStatus nppiNormInfGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.48 NppStatus nppiNormInfGetBufferHostSize_8u_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_8u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.49 NppStatus nppiNormInfGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.103.2.50 NppStatus nppiNormInfGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_Inf_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104 Norm_L1

Primitives for computing the L1 norm of an image.

Basic Norm_L1

- `NppStatus nppiNorm_L1_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
One-channel 8-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
One-channel 16-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed image Norm_L1.
- `NppStatus nppiNorm_L1_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point image Norm_L1.
- `NppStatus nppiNorm_L1_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Three-channel 8-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Three-channel 16-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Three-channel 16-bit signed image Norm_L1.
- `NppStatus nppiNorm_L1_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Three-channel 32-bit floating point image Norm_L1.
- `NppStatus nppiNorm_L1_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image Norm_L1 ignoring alpha channel.
- `NppStatus nppiNorm_L1_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit unsigned image Norm_L1 ignoring alpha channel.
- `NppStatus nppiNorm_L1_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit signed image Norm_L1 ignoring alpha channel.

- `NppStatus nppiNorm_L1_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit floating point image Norm_L1 ignoring alpha channel.
- `NppStatus nppiNorm_L1_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit signed image Norm_L1.
- `NppStatus nppiNorm_L1_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit floating point image Norm_L1.

Masked Norm_L1

See [Masked Operation](#).

- `NppStatus nppiNorm_L1_8u_C1MR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 8-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_8s_C1MR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 8-bit signed image Norm_L1.
- `NppStatus nppiNorm_L1_16u_C1MR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 16-bit unsigned image Norm_L1.
- `NppStatus nppiNorm_L1_32f_C1MR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked one-channel 32-bit floating point image Norm_L1.

Masked Channel Norm_L1

See [Channel-of-Interest API](#) and [Masked Operation](#).

- `NppStatus nppiNorm_L1_8u_C3CMR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
Masked three-channel 8-bit unsigned image Norm_L1 affecting only single channel.

- **NppStatus** **nppiNorm_L1_8s_C3CMR** (const **Npp8s** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
Masked three-channel 8-bit signed image Norm_L1 affecting only single channel.
- **NppStatus** **nppiNorm_L1_16u_C3CMR** (const **Npp16u** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
Masked three-channel 16-bit unsigned image Norm_L1 affecting only single channel.
- **NppStatus** **nppiNorm_L1_32f_C3CMR** (const **Npp32f** *pSrc, int nSrcStep, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNorm, **Npp8u** *pDeviceBuffer)
Masked three-channel 32-bit floating point image Norm_L1 affecting only single channel.

NormL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm_L1 primitives.

- **NppStatus** **nppiNormL1GetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNorm_L1_8u_C1R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNorm_L1_16u_C1R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize_16s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNorm_L1_16s_C1R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNorm_L1_32f_C1R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize_8u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNorm_L1_8u_C1MR**.*
- **NppStatus** **nppiNormL1GetBufferHostSize_8s_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNorm_L1_8s_C1MR**.*
- **NppStatus** **nppiNormL1GetBufferHostSize_16u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNorm_L1_16u_C1MR**.*
- **NppStatus** **nppiNormL1GetBufferHostSize_32f_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNorm_L1_32f_C1MR**.*
- **NppStatus** **nppiNormL1GetBufferHostSize_8u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNorm_L1_8u_C3R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize_16u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNorm_L1_16u_C3R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize_16s_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNorm_L1_16s_C3R**.*

- `NppStatus nppiNormL1GetBufferHostSize_32f_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_32f_C3R`.
- `NppStatus nppiNormL1GetBufferHostSize_8u_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_8u_AC4R`.
- `NppStatus nppiNormL1GetBufferHostSize_16u_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_16u_AC4R`.
- `NppStatus nppiNormL1GetBufferHostSize_16s_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_16s_AC4R`.
- `NppStatus nppiNormL1GetBufferHostSize_32f_AC4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_32f_AC4R`.
- `NppStatus nppiNormL1GetBufferHostSize_8u_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_8u_C4R`.
- `NppStatus nppiNormL1GetBufferHostSize_16u_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_16u_C4R`.
- `NppStatus nppiNormL1GetBufferHostSize_16s_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_16s_C4R`.
- `NppStatus nppiNormL1GetBufferHostSize_32f_C4R` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_32f_C4R`.
- `NppStatus nppiNormL1GetBufferHostSize_8u_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_8u_C3CMR`.
- `NppStatus nppiNormL1GetBufferHostSize_8s_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_8s_C3CMR`.
- `NppStatus nppiNormL1GetBufferHostSize_16u_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_16u_C3CMR`.
- `NppStatus nppiNormL1GetBufferHostSize_32f_C3CMR` (`NppiSize` oSizeROI, int *hpBufferSize)
Buffer size for `nppiNorm_L1_32f_C3CMR`.

7.104.1 Detailed Description

Primitives for computing the L1 norm of an image.

7.104.2 Function Documentation

7.104.2.1 NppStatus nppiNorm_L1_16s_AC4R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u * *pDeviceBuffer*)

Four-channel 16-bit signed image Norm_L1 ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.2 NppStatus nppiNorm_L1_16s_C1R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

One-channel 16-bit signed image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.3 NppStatus nppiNorm_L1_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u * *pDeviceBuffer*)

Three-channel 16-bit signed image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.4 NppStatus nppiNorm_L1_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit signed image Norm_L1.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.5 NppStatus nppiNorm_L1_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_L1 ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.6 NppStatus nppiNorm_L1_16u_C1MR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

Masked one-channel 16-bit unsigned image Norm_L1.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNorm Pointer to the norm value.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormL1GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.7 NppStatus nppiNorm_L1_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

One-channel 16-bit unsigned image Norm_L1.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNorm Pointer to the norm value.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormL1GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.8 NppStatus nppiNorm_L1_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

Masked three-channel 16-bit unsigned image Norm_L1 affecting only single channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nCOI Channel_of_Interest Number.
pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.104.2.9 NppStatus nppiNorm_L1_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image Norm_L1.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.10 NppStatus nppiNorm_L1_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_L1.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.11 NppStatus nppiNorm_L1_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Four-channel 32-bit floating point image Norm_L1 ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNorm Array that contains the norm values of Three-channels.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormL1GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.12 `NppStatus nppiNorm_L1_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image Norm_L1.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNorm Pointer to the norm value.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormL1GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.13 `NppStatus nppiNorm_L1_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Norm_L1.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNorm Pointer to the norm value.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormL1GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.14 NppStatus nppiNorm_L1_32f_C3CMR (const Npp32f * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

Masked three-channel 32-bit floating point image Norm_L1 affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if the step of the source image cannot be divided by 4, or NPP_COI_ERROR if an invalid channel of interest is specified.

7.104.2.15 NppStatus nppiNorm_L1_32f_C3R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u * *pDeviceBuffer*)

Three-channel 32-bit floating point image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.16 NppStatus nppiNorm_L1_32f_C4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[4], Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.17 `NppStatus nppiNorm_L1_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.18 `NppStatus nppiNorm_L1_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image Norm_L1 affecting only single channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.104.2.19 **NppStatus nppiNorm_L1_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u * *pDeviceBuffer*)

Four-channel 8-bit unsigned image Norm_L1 ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.20 **NppStatus nppiNorm_L1_8u_C1MR** (const Npp8u * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

Masked one-channel 8-bit unsigned image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.21 **NppStatus nppiNorm_L1_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

One-channel 8-bit unsigned image Norm_L1.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.22 `NppStatus nppiNorm_L1_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image Norm_L1 affecting only single channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.104.2.23 `NppStatus nppiNorm_L1_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Norm_L1.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.24 **NppStatus nppiNorm_L1_8u_C4R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[4], Npp8u * *pDeviceBuffer*)

Four-channel 8-bit unsigned image Norm_L1.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL1GetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.104.2.25 **NppStatus nppiNormL1GetBufferHostSize_16s_AC4R** (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_16s_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.26 **NppStatus nppiNormL1GetBufferHostSize_16s_C1R** (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.27 NppStatus nppiNormL1GetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.28 NppStatus nppiNormL1GetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.29 NppStatus nppiNormL1GetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.30 NppStatus nppiNormL1GetBufferHostSize_16u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_16u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.31 NppStatus nppiNormL1GetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.32 NppStatus nppiNormL1GetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_16u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.33 NppStatus nppiNormL1GetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.34 NppStatus nppiNormL1GetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.35 NppStatus nppiNormL1GetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.36 NppStatus nppiNormL1GetBufferHostSize_32f_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_32f_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.37 NppStatus nppiNormL1GetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.38 NppStatus nppiNormL1GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_32f_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.39 NppStatus nppiNormL1GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.40 NppStatus nppiNormL1GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.41 NppStatus nppiNormL1GetBufferHostSize_8s_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_8s_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.42 NppStatus nppiNormL1GetBufferHostSize_8s_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_8s_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.43 NppStatus nppiNormL1GetBufferHostSize_8u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.44 NppStatus nppiNormL1GetBufferHostSize_8u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L1_8u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.45 NppStatus nppiNormL1GetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.46 NppStatus nppiNormL1GetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_8u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.104.2.47 NppStatus nppiNormL1GetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L1_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.48 NppStatus nppiNormL1GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiNorm_L1_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105 Norm_L2

Primitives for computing the L2 norm of an image.

Basic Norm_L2

Computes the L2 norm of an image.

- `NppStatus nppiNorm_L2_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
One-channel 8-bit unsigned image Norm_L2.
- `NppStatus nppiNorm_L2_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
One-channel 16-bit unsigned image Norm_L2.
- `NppStatus nppiNorm_L2_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed image Norm_L2.
- `NppStatus nppiNorm_L2_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point image Norm_L2.
- `NppStatus nppiNorm_L2_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Three-channel 8-bit unsigned image Norm_L2.
- `NppStatus nppiNorm_L2_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Three-channel 16-bit unsigned image Norm_L2.
- `NppStatus nppiNorm_L2_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Three-channel 16-bit signed image Norm_L2.
- `NppStatus nppiNorm_L2_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Three-channel 32-bit floating point image Norm_L2.
- `NppStatus nppiNorm_L2_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image Norm_L2 ignoring alpha channel.
- `NppStatus nppiNorm_L2_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)
Four-channel 16-bit unsigned image Norm_L2 ignoring alpha channel.
- `NppStatus nppiNorm_L2_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)

Four-channel 16-bit signed image Norm_L2 ignoring alpha channel.

- `NppStatus nppiNorm_L2_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image Norm_L2 ignoring alpha channel.

- `NppStatus nppiNorm_L2_8u_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_16u_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)

Four-channel 16-bit signed image Norm_L2.

- `NppStatus nppiNorm_L2_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image Norm_L2.

Masked Norm_L2

See [Masked Operation](#).

- `NppStatus nppiNorm_L2_8u_C1MR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)

Masked one-channel 8-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_8s_C1MR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)

Masked one-channel 8-bit signed image Norm_L2.

- `NppStatus nppiNorm_L2_16u_C1MR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)

Masked one-channel 16-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_32f_C1MR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)

Masked one-channel 32-bit floating point image Norm_L2.

Masked Channel Norm_L2

See [Channel-of-Interest API](#) and [Masked Operation](#).

- `NppStatus nppiNorm_L2_8u_C3CMR` (const `Npp8u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)

Masked three-channel 8-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_8s_C3CMR` (const `Npp8s` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)

Masked three-channel 8-bit signed image Norm_L2.

- `NppStatus nppiNorm_L2_16u_C3CMR` (const `Npp16u` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)

Masked three-channel 16-bit unsigned image Norm_L2.

- `NppStatus nppiNorm_L2_32f_C3CMR` (const `Npp32f` *pSrc, int nSrcStep, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNorm, `Npp8u` *pDeviceBuffer)

Masked three-channel 32-bit floating point image Norm_L2.

NormL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm_L2 primitives.

- `NppStatus nppiNormL2GetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_8u_C1R`.

- `NppStatus nppiNormL2GetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_16u_C1R`.

- `NppStatus nppiNormL2GetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_16s_C1R`.

- `NppStatus nppiNormL2GetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_32f_C1R`.

- `NppStatus nppiNormL2GetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_8u_C1MR`.

- `NppStatus nppiNormL2GetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_8s_C1MR`.

- `NppStatus nppiNormL2GetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_16u_C1MR`.

- `NppStatus nppiNormL2GetBufferHostSize_32f_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_32f_C1MR`.

- `NppStatus nppiNormL2GetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_8u_C3R`.

- `NppStatus nppiNormL2GetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_16u_C3R`.

- `NppStatus nppiNormL2GetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)

Buffer size for `nppiNorm_L2_16s_C3R`.

- `NppStatus nppiNormL2GetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_32f_C3R`.
- `NppStatus nppiNormL2GetBufferHostSize_8u_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_8u_AC4R`.
- `NppStatus nppiNormL2GetBufferHostSize_16u_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_16u_AC4R`.
- `NppStatus nppiNormL2GetBufferHostSize_16s_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_16s_AC4R`.
- `NppStatus nppiNormL2GetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_32f_AC4R`.
- `NppStatus nppiNormL2GetBufferHostSize_8u_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_8u_C4R`.
- `NppStatus nppiNormL2GetBufferHostSize_16u_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_16u_C4R`.
- `NppStatus nppiNormL2GetBufferHostSize_16s_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_16s_C4R`.
- `NppStatus nppiNormL2GetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_32f_C4R`.
- `NppStatus nppiNormL2GetBufferHostSize_8u_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_8u_C3CMR`.
- `NppStatus nppiNormL2GetBufferHostSize_8s_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_8s_C3CMR`.
- `NppStatus nppiNormL2GetBufferHostSize_16u_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_16u_C3CMR`.
- `NppStatus nppiNormL2GetBufferHostSize_32f_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)
Buffer size for `nppiNorm_L2_32f_C3CMR`.

7.105.1 Detailed Description

Primitives for computing the L2 norm of an image.

7.105.2 Function Documentation

7.105.2.1 `NppStatus nppiNorm_L2_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image Norm_L2 ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.2 `NppStatus nppiNorm_L2_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.3 `NppStatus nppiNorm_L2_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.4 NppStatus nppiNorm_L2_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit signed image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.5 NppStatus nppiNorm_L2_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_L2 ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.6 NppStatus nppiNorm_L2_16u_C1MR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)

Masked one-channel 16-bit unsigned image Norm_L2.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNorm Pointer to the norm value.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormL2GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.7 `NppStatus nppiNorm_L2_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image Norm_L2.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNorm Pointer to the norm value.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormL2GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.8 `NppStatus nppiNorm_L2_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image Norm_L2.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nCOI Channel_of_Interest Number.
pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified.

7.105.2.9 NppStatus nppiNorm_L2_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.10 NppStatus nppiNorm_L2_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.11 NppStatus nppiNorm_L2_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)

Four-channel 32-bit floating point image Norm_L2 ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aNorm Array that contains the norm values of Three-channels.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormL2GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.12 `NppStatus nppiNorm_L2_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pMask [Mask-Image Pointer](#).
nMaskStep [Mask-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pNorm Pointer to the norm value.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormL2GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if the step of the source image cannot be divided by 4.

7.105.2.13 `NppStatus nppiNorm_L2_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pNorm Pointer to the norm value.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormL2GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.14 **NppStatus nppiNorm_L2_32f_C3CMR** (const Npp32f * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

Masked three-channel 32-bit floating point image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if the step of the source image cannot be divided by 4, or NPP_COI_ERROR if an invalid channel of interest is specified.

7.105.2.15 **NppStatus nppiNorm_L2_32f_C3R** (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u * *pDeviceBuffer*)

Three-channel 32-bit floating point image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.16 **NppStatus nppiNorm_L2_32f_C4R** (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[4], Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.17 `NppStatus nppiNorm_L2_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.18 `NppStatus nppiNorm_L2_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.105.2.19 **NppStatus nppiNorm_L2_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u * *pDeviceBuffer*)

Four-channel 8-bit unsigned image Norm_L2 ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.20 **NppStatus nppiNorm_L2_8u_C1MR** (const Npp8u * *pSrc*, int *nSrcStep*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

Masked one-channel 8-bit unsigned image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.21 **NppStatus nppiNorm_L2_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

One-channel 8-bit unsigned image Norm_L2.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.22 `NppStatus nppiNorm_L2_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNorm Pointer to the norm value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.105.2.23 `NppStatus nppiNorm_L2_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Three-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.24 NppStatus nppiNorm_L2_8u_C4R (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[4], Npp8u * *pDeviceBuffer*)

Four-channel 8-bit unsigned image Norm_L2.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNorm Array that contains the norm values of Four-channels.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormL2GetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.105.2.25 NppStatus nppiNormL2GetBufferHostSize_16s_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_16s_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.26 NppStatus nppiNormL2GetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.27 NppStatus nppiNormL2GetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.28 NppStatus nppiNormL2GetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.29 NppStatus nppiNormL2GetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.30 NppStatus nppiNormL2GetBufferHostSize_16u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_16u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.31 NppStatus nppiNormL2GetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.32 NppStatus nppiNormL2GetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_16u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.33 NppStatus nppiNormL2GetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.34 NppStatus nppiNormL2GetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.35 NppStatus nppiNormL2GetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.36 NppStatus nppiNormL2GetBufferHostSize_32f_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_32f_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.37 NppStatus nppiNormL2GetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.38 NppStatus nppiNormL2GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_32f_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.39 NppStatus nppiNormL2GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.40 NppStatus nppiNormL2GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.41 NppStatus nppiNormL2GetBufferHostSize_8s_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_8s_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.42 NppStatus nppiNormL2GetBufferHostSize_8s_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_8s_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.43 NppStatus nppiNormL2GetBufferHostSize_8u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.44 NppStatus nppiNormL2GetBufferHostSize_8u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNorm_L2_8u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.45 NppStatus nppiNormL2GetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.46 NppStatus nppiNormL2GetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_8u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.105.2.47 NppStatus nppiNormL2GetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNorm_L2_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.48 NppStatus nppiNormL2GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Buffer size for [nppiNorm_L2_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106 NormDiff_Inf

Primitives for computing the infinity norm of difference of pixels between two images.

Basic NormDiff_Inf

- **NppStatus nppiNormDiff_Inf_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image NormDiff_Inf.
- **NppStatus nppiNormDiff_Inf_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image NormDiff_Inf.
- **NppStatus nppiNormDiff_Inf_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image NormDiff_Inf.
- **NppStatus nppiNormDiff_Inf_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image NormDiff_Inf.
- **NppStatus nppiNormDiff_Inf_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image NormDiff_Inf.
- **NppStatus nppiNormDiff_Inf_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image NormDiff_Inf.
- **NppStatus nppiNormDiff_Inf_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image NormDiff_Inf.
- **NppStatus nppiNormDiff_Inf_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image NormDiff_Inf.
- **NppStatus nppiNormDiff_Inf_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image NormDiff_Inf ignoring alpha channel.
- **NppStatus nppiNormDiff_Inf_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image NormDiff_Inf ignoring alpha channel.
- **NppStatus nppiNormDiff_Inf_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image NormDiff_Inf ignoring alpha channel.

- [NppStatus nppiNormDiff_Inf_32f_AC4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[3], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormDiff_Inf ignoring alpha channel.
- [NppStatus nppiNormDiff_Inf_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image NormDiff_Inf.
- [NppStatus nppiNormDiff_Inf_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit unsigned image NormDiff_Inf.
- [NppStatus nppiNormDiff_Inf_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit signed image NormDiff_Inf.
- [NppStatus nppiNormDiff_Inf_32f_C4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormDiff_Inf.

Masked NormDiff_Inf

See [Masked Operation](#).

- [NppStatus nppiNormDiff_Inf_8u_C1MR](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 8-bit unsigned images NormDiff_Inf.
- [NppStatus nppiNormDiff_Inf_8s_C1MR](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 8-bit signed images NormDiff_Inf.
- [NppStatus nppiNormDiff_Inf_16u_C1MR](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 16-bit unsigned images NormDiff_Inf.
- [NppStatus nppiNormDiff_Inf_32f_C1MR](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 32-bit floating point images NormDiff_Inf.

Masked Channel Mean

See [Masked Operation](#) and [Channel-of-Interest API](#).

- **NppStatus** **nppiNormDiff_Inf_8u_C3CMR** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)

Masked three-channel 8-bit unsigned image NormDiff_Inf affecting only single channel.

- **NppStatus** **nppiNormDiff_Inf_8s_C3CMR** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)

Masked three-channel 8-bit signed image NormDiff_Inf affecting only single channel.

- **NppStatus** **nppiNormDiff_Inf_16u_C3CMR** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)

Masked three-channel 16-bit unsigned image NormDiff_Inf affecting only single channel.

- **NppStatus** **nppiNormDiff_Inf_32f_C3CMR** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)

Masked three-channel 32-bit floating point image NormDiff_Inf affecting only single channel.

NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

- **NppStatus** **nppiNormDiffInfGetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNormDiff_Inf_8u_C1R**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNormDiff_Inf_16u_C1R**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize_16s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNormDiff_Inf_16s_C1R**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNormDiff_Inf_32f_C1R**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize_8u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNormDiff_Inf_8u_C1MR**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize_8s_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size for **nppiNormDiff_Inf_8s_C1MR**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize_16u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_16u_C1MR*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_32f_C1MR](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_32f_C1MR*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_8u_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_8u_C3R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_16u_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_16u_C3R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_16s_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_16s_C3R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_32f_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_32f_C3R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_8u_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_8u_C4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_16u_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_16u_C4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_16s_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_16s_C4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_32f_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_32f_C4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_8u_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_8u_AC4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_16u_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_16u_AC4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_16s_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_16s_AC4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize_32f_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) *hpBufferSize)

Buffer size for *nppiNormDiff_Inf_32f_AC4R*.

- [NppStatus](#) [nppiNormDiffInfGetBufferHostSize_8u_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size for [nppiNormDiff_Inf_8u_C3CMR](#).

- [NppStatus](#) [nppiNormDiffInfGetBufferHostSize_8s_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size for [nppiNormDiff_Inf_8s_C3CMR](#).

- [NppStatus](#) [nppiNormDiffInfGetBufferHostSize_16u_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size for [nppiNormDiff_Inf_16u_C3CMR](#).

- [NppStatus](#) [nppiNormDiffInfGetBufferHostSize_32f_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size for [nppiNormDiff_Inf_32f_C3CMR](#).

7.106.1 Detailed Description

Primitives for computing the infinity norm of difference of pixels between two images.

7.106.2 Function Documentation

7.106.2.1 [NppStatus nppiNormDiff_Inf_16s_AC4R](#) ([const Npp16s *pSrc1](#), [int nSrc1Step](#), [const Npp16s *pSrc2](#), [int nSrc2Step](#), [NppiSize oSizeROI](#), [Npp64f aNormDiff\[3\]](#), [Npp8u *pDeviceBuffer](#))

Four-channel 16-bit signed image NormDiff_Inf ignoring alpha channel.

Parameters:

[pSrc1](#) [Source-Image Pointer](#).

[nSrc1Step](#) [Source-Image Line Step](#).

[pSrc2](#) [Source-Image Pointer](#).

[nSrc2Step](#) [Source-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[aNormDiff](#) Array that contains computed Inf-norm of differences.

[pDeviceBuffer](#) [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.2 `NppStatus nppiNormDiff_Inf_16s_C1R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.3 `NppStatus nppiNormDiff_Inf_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.4 `NppStatus nppiNormDiff_Inf_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNormDiff Array that contains computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffInfGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.5 `NppStatus nppiNormDiff_Inf_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff_Inf ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNormDiff Array that contains computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffInfGetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.6 `NppStatus nppiNormDiff_Inf_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned images NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.7 `NppStatus nppiNormDiff_Inf_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.8 `NppStatus nppiNormDiff_Inf_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormDiff_Inf affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI [Channel_of_Interest](#) Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.106.2.9 NppStatus nppiNormDiff_Inf_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)

Three-channel 16-bit unsigned image NormDiff_Inf.

Parameters:

pSrc1 [Source-Image](#) Pointer.

nSrc1Step [Source-Image](#) Line Step.

pSrc2 [Source-Image](#) Pointer.

nSrc2Step [Source-Image](#) Line Step.

oSizeROI [Region-of-Interest](#) (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.10 NppStatus nppiNormDiff_Inf_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)

Four-channel 16-bit unsigned image NormDiff_Inf.

Parameters:

pSrc1 [Source-Image](#) Pointer.

nSrc1Step [Source-Image](#) Line Step.

pSrc2 [Source-Image](#) Pointer.

nSrc2Step [Source-Image](#) Line Step.

oSizeROI [Region-of-Interest](#) (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.11 `NppStatus nppiNormDiff_Inf_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff_Inf ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.106.2.12 `NppStatus nppiNormDiff_Inf_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point images NormDiff_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.106.2.13 `NppStatus nppiNormDiff_Inf_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.106.2.14 `NppStatus nppiNormDiff_Inf_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormDiff_Inf affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.106.2.15 `NppStatus nppiNormDiff_Inf_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormDiff_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.106.2.16 `NppStatus nppiNormDiff_Inf_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.106.2.17 `NppStatus nppiNormDiff_Inf_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed images NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNormDiff Pointer to the computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffInfGetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.18 `NppStatus nppiNormDiff_Inf_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormDiff_Inf affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nCOI Channel_of_Interest Number.
pNormDiff Pointer to the computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffInfGetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.106.2.19 `NppStatus nppiNormDiff_Inf_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff_Inf ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aNormDiff Array that contains computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffInfGetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.20 `NppStatus nppiNormDiff_Inf_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned images NormDiff_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
pMask [Mask-Image Pointer](#).
nMaskStep [Mask-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pNormDiff Pointer to the computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffInfGetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.21 `NppStatus nppiNormDiff_Inf_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormDiff_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.22 `NppStatus nppiNormDiff_Inf_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormDiff_Inf affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.106.2.23 `NppStatus nppiNormDiff_Inf_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.24 `NppStatus nppiNormDiff_Inf_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffInfGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.106.2.25 `NppStatus nppiNormDiffInfGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiNormDiff_Inf_16s_AC4R](#).

Parameters:

oSizeROI Region-of-Interest (ROI).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.26 NppStatus nppiNormDiffInfGetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.27 NppStatus nppiNormDiffInfGetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.28 NppStatus nppiNormDiffInfGetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.29 NppStatus nppiNormDiffInfGetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.30 NppStatus nppiNormDiffInfGetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_16u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.31 NppStatus nppiNormDiffInfGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.32 NppStatus nppiNormDiffInfGetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Buffer size for [nppiNormDiff_Inf_16u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.33 NppStatus nppiNormDiffInfGetBufferHostSize_16u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.34 NppStatus nppiNormDiffInfGetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.35 NppStatus nppiNormDiffInfGetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.36 NppStatus nppiNormDiffInfGetBufferHostSize_32f_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_32f_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.37 NppStatus nppiNormDiffInfGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.38 NppStatus nppiNormDiffInfGetBufferHostSize_32f_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_32f_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.39 NppStatus nppiNormDiffInfGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.40 NppStatus nppiNormDiffInfGetBufferHostSize_32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.41 NppStatus nppiNormDiffInfGetBufferHostSize_8s_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_8s_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.42 NppStatus nppiNormDiffInfGetBufferHostSize_8s_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_8s_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.43 NppStatus nppiNormDiffInfGetBufferHostSize_8u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.44 NppStatus nppiNormDiffInfGetBufferHostSize_8u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_8u_C1MR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.45 NppStatus nppiNormDiffInfGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.46 NppStatus nppiNormDiffInfGetBufferHostSize_8u_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_8u_C3CMR](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.47 NppStatus nppiNormDiffInfGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.106.2.48 NppStatus nppiNormDiffInfGetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size for [nppiNormDiff_Inf_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107 NormDiff_L1

Primitives for computing the L1 norm of difference of pixels between two images.

Basic NormDiff_L1

- **NppStatus nppiNormDiff_L1_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image NormDiff_L1.
- **NppStatus nppiNormDiff_L1_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image NormDiff_L1 ignoring alpha channel.
- **NppStatus nppiNormDiff_L1_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image NormDiff_L1 ignoring alpha channel.
- **NppStatus nppiNormDiff_L1_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image NormDiff_L1 ignoring alpha channel.

- [NppStatus nppiNormDiff_L1_32f_AC4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[3], [Npp8u](#) *pDeviceBuffer)

Four-channel 32-bit floating point image NormDiff_L1 ignoring alpha channel.

- [NppStatus nppiNormDiff_L1_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 8-bit unsigned image NormDiff_L1.

- [NppStatus nppiNormDiff_L1_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit unsigned image NormDiff_L1.

- [NppStatus nppiNormDiff_L1_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit signed image NormDiff_L1.

- [NppStatus nppiNormDiff_L1_32f_C4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 32-bit floating point image NormDiff_L1.

Masked NormDiff_L1

See [Masked Operation](#).

- [NppStatus nppiNormDiff_L1_8u_C1MR](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 8-bit unsigned image NormDiff_L1.

- [NppStatus nppiNormDiff_L1_8s_C1MR](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 8-bit signed image NormDiff_L1.

- [NppStatus nppiNormDiff_L1_16u_C1MR](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 16-bit unsigned image NormDiff_L1.

- [NppStatus nppiNormDiff_L1_32f_C1MR](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 32-bit floating point image NormDiff_L1.

Masked Channel NormDiff_L1

See [Masked Operation](#) and [Channel-of-Interest API](#).

- [NppStatus nppiNormDiff_L1_8u_C3CMR](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)

Masked three-channel 8-bit unsigned image NormDiff_L1 affecting only single channel.

- [NppStatus nppiNormDiff_L1_8s_C3CMR](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)

Masked three-channel 8-bit signed image NormDiff_L1 affecting only single channel.

- [NppStatus nppiNormDiff_L1_16u_C3CMR](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)

Masked three-channel 16-bit unsigned image NormDiff_L1 affecting only single channel.

- [NppStatus nppiNormDiff_L1_32f_C3CMR](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)

Masked three-channel 32-bit floating point image NormDiff_L1 affecting only single channel.

NormDiffL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_L1 primitives.

- [NppStatus nppiNormDiffL1GetBufferHostSize_8u_C1R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C1R.
- [NppStatus nppiNormDiffL1GetBufferHostSize_16u_C1R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C1R.
- [NppStatus nppiNormDiffL1GetBufferHostSize_16s_C1R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C1R.
- [NppStatus nppiNormDiffL1GetBufferHostSize_32f_C1R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C1R.
- [NppStatus nppiNormDiffL1GetBufferHostSize_8u_C1MR](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C1MR.
- [NppStatus nppiNormDiffL1GetBufferHostSize_8s_C1MR](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8s_C1MR.
- [NppStatus nppiNormDiffL1GetBufferHostSize_16u_C1MR](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C1MR.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_32f_C1MR** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C1MR.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_8u_C3R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C3R.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_16u_C3R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C3R.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_16s_C3R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C3R.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_32f_C3R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C3R.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_8u_C4R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C4R.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_16u_C4R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C4R.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_16s_C4R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C4R.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_32f_C4R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C4R.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_8u_AC4R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_AC4R.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_16u_AC4R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_AC4R.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_16s_AC4R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_AC4R.

- **NppStatus** **nppiNormDiffL1GetBufferHostSize_32f_AC4R** (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_AC4R.

- **NppStatus** [nppiNormDiffL1GetBufferHostSize_8u_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for [nppiNormDiff_L1_8u_C3CMR](#).
- **NppStatus** [nppiNormDiffL1GetBufferHostSize_8s_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for [nppiNormDiff_L1_8s_C3CMR](#).
- **NppStatus** [nppiNormDiffL1GetBufferHostSize_16u_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for [nppiNormDiff_L1_16u_C3CMR](#).
- **NppStatus** [nppiNormDiffL1GetBufferHostSize_32f_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for [nppiNormDiff_L1_32f_C3CMR](#).

7.107.1 Detailed Description

Primitives for computing the L1 norm of difference of pixels between two images.

7.107.2 Function Documentation

7.107.2.1 **NppStatus** [nppiNormDiff_L1_16s_AC4R](#) ([const](#) [Npp16s](#) * [pSrc1](#), [int](#) [nSrc1Step](#), [const](#) [Npp16s](#) * [pSrc2](#), [int](#) [nSrc2Step](#), [NppiSize](#) [oSizeROI](#), [Npp64f](#) [aNormDiff](#)[3], [Npp8u](#) * [pDeviceBuffer](#))

Four-channel 16-bit signed image NormDiff_L1 ignoring alpha channel.

Parameters:

- [pSrc1](#) [Source-Image Pointer](#).
- [nSrc1Step](#) [Source-Image Line Step](#).
- [pSrc2](#) [Source-Image Pointer](#).
- [nSrc2Step](#) [Source-Image Line Step](#).
- [oSizeROI](#) [Region-of-Interest \(ROI\)](#).
- [aNormDiff](#) Array that contains computed Inf-norm of differences.
- [pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.2 **NppStatus** [nppiNormDiff_L1_16s_C1R](#) ([const](#) [Npp16s](#) * [pSrc1](#), [int](#) [nSrc1Step](#), [const](#) [Npp16s](#) * [pSrc2](#), [int](#) [nSrc2Step](#), [NppiSize](#) [oSizeROI](#), [Npp64f](#) * [pNormDiff](#), [Npp8u](#) * [pDeviceBuffer](#))

One-channel 16-bit signed image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNormDiff Pointer to the computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffL1GetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.3 `NppStatus nppiNormDiff_L1_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNormDiff Array that contains computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffL1GetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.4 `NppStatus nppiNormDiff_L1_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.5 `NppStatus nppiNormDiff_L1_16u_AC4R (const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u *pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff_L1 ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.6 `NppStatus nppiNormDiff_L1_16u_C1MR (const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, Npp64f *pNormDiff, Npp8u *pDeviceBuffer)`

Masked one-channel 16-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.7 `NppStatus nppiNormDiff_L1_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormDiff [Pointer to the computed Inf-norm of differences](#).

pDeviceBuffer [Pointer to the required device memory allocation, \[Scratch Buffer and Host Pointer\]\(#\)](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.8 `NppStatus nppiNormDiff_L1_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormDiff_L1 affecting only single channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNormDiff [Pointer to the computed Inf-norm of differences](#).

pDeviceBuffer [Pointer to the required device memory allocation, \[Scratch Buffer and Host Pointer\]\(#\)](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.107.2.9 `NppStatus nppiNormDiff_L1_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.10 `NppStatus nppiNormDiff_L1_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.11 `NppStatus nppiNormDiff_L1_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff_L1 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNormDiff Array that contains computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffL1GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.107.2.12 `NppStatus nppiNormDiff_L1_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNormDiff Pointer to the computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffL1GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.107.2.13 `NppStatus nppiNormDiff_L1_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.107.2.14 `NppStatus nppiNormDiff_L1_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormDiff_L1 affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.107.2.15 `NppStatus nppiNormDiff_L1_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.107.2.16 `NppStatus nppiNormDiff_L1_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.107.2.17 `NppStatus nppiNormDiff_L1_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image NormDiff_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.18 `NppStatus nppiNormDiff_L1_8s_C3CMR (const Npp8s *pSrc1, int nSrc1Step, const Npp8s *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f *pNormDiff, Npp8u *pDeviceBuffer)`

Masked three-channel 8-bit signed image NormDiff_L1 affecting only single channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.107.2.19 `NppStatus nppiNormDiff_L1_8u_AC4R (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u *pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff_L1 ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.20 `NppStatus nppiNormDiff_L1_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.21 `NppStatus nppiNormDiff_L1_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.22 `NppStatus nppiNormDiff_L1_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormDiff_L1 affecting only single channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNormDiff [Pointer to the computed Inf-norm of differences](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.107.2.23 `NppStatus nppiNormDiff_L1_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff [Array that contains computed Inf-norm of differences](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.24 `NppStatus nppiNormDiff_L1_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL1GetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.107.2.25 `NppStatus nppiNormDiffL1GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.26 `NppStatus nppiNormDiffL1GetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.27 NppStatus nppiNormDiffL1GetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.28 NppStatus nppiNormDiffL1GetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.29 NppStatus nppiNormDiffL1GetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.30 NppStatus nppiNormDiffL1GetBufferHostSize_16u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.31 NppStatus nppiNormDiffL1GetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_16u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.32 NppStatus nppiNormDiffL1GetBufferHostSize_16u_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_16u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.33 NppStatus nppiNormDiffL1GetBufferHostSize_16u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_16u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.34 NppStatus nppiNormDiffL1GetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.35 NppStatus nppiNormDiffL1GetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.36 NppStatus nppiNormDiffL1GetBufferHostSize_32f_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.37 NppStatus nppiNormDiffL1GetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.38 NppStatus nppiNormDiffL1GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_32f_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.39 NppStatus nppiNormDiffL1GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_32f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.40 NppStatus nppiNormDiffL1GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_32f_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.41 NppStatus nppiNormDiffL1GetBufferHostSize_8s_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8s_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.42 NppStatus nppiNormDiffL1GetBufferHostSize_8s_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8s_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.43 NppStatus nppiNormDiffL1GetBufferHostSize_8u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.44 NppStatus nppiNormDiffL1GetBufferHostSize_8u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.45 NppStatus nppiNormDiffL1GetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_8u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.46 NppStatus nppiNormDiffL1GetBufferHostSize_8u_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_8u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.107.2.47 NppStatus nppiNormDiffL1GetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L1_8u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.48 NppStatus nppiNormDiffL1GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108 NormDiff_L2

Primitives for computing the L2 norm of difference of pixels between two images.

Basic NormDiff_L2

- **NppStatus nppiNormDiff_L2_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image NormDiff_L2.
- **NppStatus nppiNormDiff_L2_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image NormDiff_L2 ignoring alpha channel.
- **NppStatus nppiNormDiff_L2_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image NormDiff_L2 ignoring alpha channel.
- **NppStatus nppiNormDiff_L2_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image NormDiff_L2 ignoring alpha channel.

- [NppStatus nppiNormDiff_L2_32f_AC4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[3], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormDiff_L2 ignoring alpha channel.
- [NppStatus nppiNormDiff_L2_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image NormDiff_L2.
- [NppStatus nppiNormDiff_L2_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit unsigned image NormDiff_L2.
- [NppStatus nppiNormDiff_L2_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit signed image NormDiff_L2.
- [NppStatus nppiNormDiff_L2_32f_C4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormDiff_L2.

Masked NormDiff_L2

See [Masked Operation](#).

- [NppStatus nppiNormDiff_L2_8u_C1MR](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 8-bit unsigned image NormDiff_L2.
- [NppStatus nppiNormDiff_L2_8s_C1MR](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 8-bit signed image NormDiff_L2.
- [NppStatus nppiNormDiff_L2_16u_C1MR](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 16-bit unsigned image NormDiff_L2.
- [NppStatus nppiNormDiff_L2_32f_C1MR](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormDiff, [Npp8u](#) *pDeviceBuffer)
Masked one-channel 32-bit floating point image NormDiff_L2.

Masked Channel NormDiff_L2

See [Masked Operation](#) and [Channel-of-Interest API](#).

- **NppStatus** **nppiNormDiff_L2_8u_C3CMR** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)

Masked three-channel 8-bit unsigned image NormDiff_L2 affecting only single channel.

- **NppStatus** **nppiNormDiff_L2_8s_C3CMR** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)

Masked three-channel 8-bit signed image NormDiff_L2 affecting only single channel.

- **NppStatus** **nppiNormDiff_L2_16u_C3CMR** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)

Masked three-channel 16-bit unsigned image NormDiff_L2 affecting only single channel.

- **NppStatus** **nppiNormDiff_L2_32f_C3CMR** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormDiff, **Npp8u** *pDeviceBuffer)

Masked three-channel 32-bit floating point image NormDiff_L2 affecting only single channel.

NormDiffL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_L2 primitives.

- **NppStatus** **nppiNormDiffL2GetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C1R.
- **NppStatus** **nppiNormDiffL2GetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C1R.
- **NppStatus** **nppiNormDiffL2GetBufferHostSize_16s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C1R.
- **NppStatus** **nppiNormDiffL2GetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C1R.
- **NppStatus** **nppiNormDiffL2GetBufferHostSize_8u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C1MR.
- **NppStatus** **nppiNormDiffL2GetBufferHostSize_8s_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8s_C1MR.
- **NppStatus** **nppiNormDiffL2GetBufferHostSize_16u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C1MR.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_32f_C1MR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C1MR.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_8u_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C3R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_16u_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C3R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_16s_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C3R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_32f_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C3R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_8u_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C4R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_16u_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C4R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_16s_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C4R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_32f_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C4R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_8u_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_AC4R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_16u_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_AC4R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_16s_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_AC4R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_32f_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_AC4R.

- **NppStatus** [nppiNormDiffL2GetBufferHostSize_8u_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for [nppiNormDiff_L2_8u_C3CMR](#).
- **NppStatus** [nppiNormDiffL2GetBufferHostSize_8s_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for [nppiNormDiff_L2_8s_C3CMR](#).
- **NppStatus** [nppiNormDiffL2GetBufferHostSize_16u_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for [nppiNormDiff_L2_16u_C3CMR](#).
- **NppStatus** [nppiNormDiffL2GetBufferHostSize_32f_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Computes the device scratch buffer size (in bytes) for [nppiNormDiff_L2_32f_C3CMR](#).

7.108.1 Detailed Description

Primitives for computing the L2 norm of difference of pixels between two images.

7.108.2 Function Documentation

7.108.2.1 **NppStatus** [nppiNormDiff_L2_16s_AC4R](#) ([const](#) [Npp16s](#) * [pSrc1](#), [int](#) [nSrc1Step](#), [const](#) [Npp16s](#) * [pSrc2](#), [int](#) [nSrc2Step](#), [NppiSize](#) [oSizeROI](#), [Npp64f](#) [aNormDiff](#)[3], [Npp8u](#) * [pDeviceBuffer](#))

Four-channel 16-bit signed image NormDiff_L2 ignoring alpha channel.

Parameters:

- [pSrc1](#) [Source-Image Pointer](#).
- [nSrc1Step](#) [Source-Image Line Step](#).
- [pSrc2](#) [Source-Image Pointer](#).
- [nSrc2Step](#) [Source-Image Line Step](#).
- [oSizeROI](#) [Region-of-Interest \(ROI\)](#).
- [aNormDiff](#) Array that contains computed Inf-norm of differences.
- [pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.2 **NppStatus** [nppiNormDiff_L2_16s_C1R](#) ([const](#) [Npp16s](#) * [pSrc1](#), [int](#) [nSrc1Step](#), [const](#) [Npp16s](#) * [pSrc2](#), [int](#) [nSrc2Step](#), [NppiSize](#) [oSizeROI](#), [Npp64f](#) * [pNormDiff](#), [Npp8u](#) * [pDeviceBuffer](#))

One-channel 16-bit signed image NormDiff_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pNormDiff Pointer to the computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffL2GetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.3 `NppStatus nppiNormDiff_L2_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormDiff_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aNormDiff Array that contains computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffL2GetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.4 `NppStatus nppiNormDiff_L2_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormDiff_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.5 `NppStatus nppiNormDiff_L2_16u_AC4R (const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u *pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff_L2 ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.6 `NppStatus nppiNormDiff_L2_16u_C1MR (const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, Npp64f *pNormDiff, Npp8u *pDeviceBuffer)`

Masked one-channel 16-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.7 `NppStatus nppiNormDiff_L2_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.8 `NppStatus nppiNormDiff_L2_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormDiff_L2 affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.108.2.9 `NppStatus nppiNormDiff_L2_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.10 `NppStatus nppiNormDiff_L2_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.11 `NppStatus nppiNormDiff_L2_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff_L2 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNormDiff Array that contains computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffL2GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.108.2.12 `NppStatus nppiNormDiff_L2_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNormDiff Pointer to the computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormDiffL2GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.108.2.13 `NppStatus nppiNormDiff_L2_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.108.2.14 `NppStatus nppiNormDiff_L2_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormDiff_L2 affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.108.2.15 `NppStatus nppiNormDiff_L2_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.108.2.16 `NppStatus nppiNormDiff_L2_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.108.2.17 `NppStatus nppiNormDiff_L2_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.18 `NppStatus nppiNormDiff_L2_8s_C3CMR (const Npp8s *pSrc1, int nSrc1Step, const Npp8s *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f *pNormDiff, Npp8u *pDeviceBuffer)`

Masked three-channel 8-bit signed image NormDiff_L2 affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.108.2.19 `NppStatus nppiNormDiff_L2_8u_AC4R (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u *pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff_L2 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.20 `NppStatus nppiNormDiff_L2_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.21 `NppStatus nppiNormDiff_L2_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.22 `NppStatus nppiNormDiff_L2_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormDiff_L2 affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormDiff Pointer to the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

7.108.2.23 `NppStatus nppiNormDiff_L2_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.24 `NppStatus nppiNormDiff_L2_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormDiff Array that contains computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormDiffL2GetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.108.2.25 `NppStatus nppiNormDiffL2GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.26 `NppStatus nppiNormDiffL2GetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.27 NppStatus nppiNormDiffL2GetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.28 NppStatus nppiNormDiffL2GetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.29 NppStatus nppiNormDiffL2GetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.30 NppStatus nppiNormDiffL2GetBufferHostSize_16u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.31 NppStatus nppiNormDiffL2GetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_16u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.32 NppStatus nppiNormDiffL2GetBufferHostSize_16u_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_16u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.33 NppStatus nppiNormDiffL2GetBufferHostSize_16u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_16u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.34 NppStatus nppiNormDiffL2GetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.35 NppStatus nppiNormDiffL2GetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.36 NppStatus nppiNormDiffL2GetBufferHostSize_32f_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.37 NppStatus nppiNormDiffL2GetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.38 NppStatus nppiNormDiffL2GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_32f_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.39 NppStatus nppiNormDiffL2GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_32f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.40 NppStatus nppiNormDiffL2GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_32f_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.41 NppStatus nppiNormDiffL2GetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8s_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.42 NppStatus nppiNormDiffL2GetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8s_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.43 NppStatus nppiNormDiffL2GetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.44 NppStatus nppiNormDiffL2GetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.45 NppStatus nppiNormDiffL2GetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_8u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.46 NppStatus nppiNormDiffL2GetBufferHostSize_8u_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_8u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.108.2.47 NppStatus nppiNormDiffL2GetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormDiff_L2_8u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.108.2.48 NppStatus nppiNormDiffL2GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109 NormRel_Inf

Primitives for computing the relative error of infinity norm between two images.

Basic NormRel_Inf

- **NppStatus nppiNormRel_Inf_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image NormRel_Inf.
- **NppStatus nppiNormRel_Inf_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image NormRel_Inf ignoring alpha channel.
- **NppStatus nppiNormRel_Inf_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image NormRel_Inf ignoring alpha channel.
- **NppStatus nppiNormRel_Inf_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image NormRel_Inf ignoring alpha channel.

- [NppStatus nppiNormRel_Inf_32f_AC4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[3], [Npp8u](#) *pDeviceBuffer)

Four-channel 32-bit floating point image NormRel_Inf ignoring alpha channel.

- [NppStatus nppiNormRel_Inf_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 8-bit unsigned image NormRel_Inf.

- [NppStatus nppiNormRel_Inf_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit unsigned image NormRel_Inf.

- [NppStatus nppiNormRel_Inf_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit signed image NormRel_Inf.

- [NppStatus nppiNormRel_Inf_32f_C4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 32-bit floating point image NormRel_Inf.

Masked NormRel_Inf

See [Masked Operation](#).

- [NppStatus nppiNormRel_Inf_8u_C1MR](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 8-bit unsigned image NormRel_Inf.

- [NppStatus nppiNormRel_Inf_8s_C1MR](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 8-bit signed image NormRel_Inf.

- [NppStatus nppiNormRel_Inf_16u_C1MR](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 16-bit unsigned image NormRel_Inf.

- [NppStatus nppiNormRel_Inf_32f_C1MR](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 32-bit floating point image NormRel_Inf.

Masked Channel NormRel_Inf

See [Masked Operation](#) and [Channel-of-Interest API](#).

- `NppStatus nppiNormRel_Inf_8u_C3CMR` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)

Masked three-channel 8-bit unsigned image NormRel_Inf affecting only single channel.

- `NppStatus nppiNormRel_Inf_8s_C3CMR` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)

Masked three-channel 8-bit signed image NormRel_Inf affecting only single channel.

- `NppStatus nppiNormRel_Inf_16u_C3CMR` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)

Masked three-channel 16-bit unsigned image NormRel_Inf affecting only single channel.

- `NppStatus nppiNormRel_Inf_32f_C3CMR` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)

Masked three-channel 32-bit floating point image NormRel_Inf affecting only single channel.

NormRelInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel_Inf primitives.

- `NppStatus nppiNormRelInfGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C1R.
- `NppStatus nppiNormRelInfGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C1R.
- `NppStatus nppiNormRelInfGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16s_C1R.
- `NppStatus nppiNormRelInfGetBufferHostSize_32s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32s_C1R.
- `NppStatus nppiNormRelInfGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C1R.
- `NppStatus nppiNormRelInfGetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C1MR.
- `NppStatus nppiNormRelInfGetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8s_C1MR.
- `NppStatus nppiNormRelInfGetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C1MR.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_32f_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C1MR.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_8u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C3R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_16u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C3R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_16s_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16s_C3R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_32f_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C3R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_8u_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C4R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_16u_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C4R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_16s_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16s_C4R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_32f_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C4R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_8u_AC4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_AC4R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_16u_AC4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_AC4R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_16s_AC4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16s_AC4R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_32f_AC4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_AC4R.

- **NppStatus** **nppiNormRelInfGetBufferHostSize_8u_C3CMR** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C3CMR.

- **NppStatus** [nppiNormRelInfGetBufferHostSize_8s_C3CMR](#) (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for [nppiNormRel_Inf_8s_C3CMR](#).
- **NppStatus** [nppiNormRelInfGetBufferHostSize_16u_C3CMR](#) (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for [nppiNormRel_Inf_16u_C3CMR](#).
- **NppStatus** [nppiNormRelInfGetBufferHostSize_32f_C3CMR](#) (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for [nppiNormRel_Inf_32f_C3CMR](#).

7.109.1 Detailed Description

Primitives for computing the relative error of infinity norm between two images.

7.109.2 Function Documentation

7.109.2.1 **NppStatus** [nppiNormRel_Inf_16s_AC4R](#) (const **Npp16s** * *pSrc1*, int *nSrc1Step*, const **Npp16s** * *pSrc2*, int *nSrc2Step*, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** * *pDeviceBuffer*)

Four-channel 16-bit signed image NormRel_Inf ignoring alpha channel.

Parameters:

- pSrc1* [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aNormRel Array that contains the computed relative error for the infinity norm of two images.
pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelInfGetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.109.2.2 **NppStatus** [nppiNormRel_Inf_16s_C1R](#) (const **Npp16s** * *pSrc1*, int *nSrc1Step*, const **Npp16s** * *pSrc2*, int *nSrc2Step*, **NppiSize** oSizeROI, **Npp64f** * *pNormRel*, **Npp8u** * *pDeviceBuffer*)

One-channel 16-bit signed image NormRel_Inf.

Parameters:

- pSrc1* [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.3 `NppStatus nppiNormRel_Inf_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.4 `NppStatus nppiNormRel_Inf_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.5 `NppStatus nppiNormRel_Inf_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel_Inf ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.6 `NppStatus nppiNormRel_Inf_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.7 `NppStatus nppiNormRel_Inf_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.8 `NppStatus nppiNormRel_Inf_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormRel_Inf affecting only single channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.9 `NppStatus nppiNormRel_Inf_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aNormRel Array that contains the computed relative error for the infinity norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.10 `NppStatus nppiNormRel_Inf_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aNormRel Array that contains the computed relative error for the infinity norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.11 NppStatus nppiNormRel_Inf_32f_AC4R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aNormRel*[3], Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image NormRel_Inf ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.12 NppStatus nppiNormRel_Inf_32f_C1MR (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, const Npp8u * *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp64f * *pNormRel*, Npp8u * *pDeviceBuffer*)

Masked one-channel 32-bit floating point image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.13 `NppStatus nppiNormRel_Inf_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.14 `NppStatus nppiNormRel_Inf_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormRel_Inf affecting only single channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.15 `NppStatus nppiNormRel_Inf_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormRel_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.109.2.16 `NppStatus nppiNormRel_Inf_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.109.2.17 `NppStatus nppiNormRel_Inf_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image NormRel_Inf.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pMask* Mask-Image Pointer.
- nMaskStep* Mask-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pNormRel* Pointer to the computed relative error for the infinity norm of two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.109.2.18 `NppStatus nppiNormRel_Inf_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormRel_Inf affecting only single channel.

Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pMask* Mask-Image Pointer.
- nMaskStep* Mask-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nCOI* Channel_of_Interest Number.
- pNormRel* Pointer to the computed relative error for the infinity norm of two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.109.2.19 `NppStatus nppiNormRel_Inf_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel_Inf ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.109.2.20 `NppStatus nppiNormRel_Inf_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.109.2.21 `NppStatus nppiNormRel_Inf_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.22 `NppStatus nppiNormRel_Inf_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormRel_Inf affecting only single channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNormRel Pointer to the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_DIVISOR_ERROR if the infinity norm of the second image is zero.

7.109.2.23 `NppStatus nppiNormRel_Inf_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.109.2.24 `NppStatus nppiNormRel_Inf_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel_Inf.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the infinity norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelInfGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

7.109.2.25 `NppStatus nppiNormRelInfGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16s_AC4R`.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.26 NppStatus nppiNormRelInfGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_16s_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.27 NppStatus nppiNormRelInfGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_16s_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.28 NppStatus nppiNormRelInfGetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_16s_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.29 NppStatus nppiNormRelInfGetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.30 NppStatus nppiNormRelInfGetBufferHostSize_16u_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.31 NppStatus nppiNormRelInfGetBufferHostSize_16u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.32 NppStatus nppiNormRelInfGetBufferHostSize_16u_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.33 NppStatus nppiNormRelInfGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_16u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.34 NppStatus nppiNormRelInfGetBufferHostSize_16u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_16u_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.35 NppStatus nppiNormRelInfGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_32f_AC4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.36 NppStatus nppiNormRelInfGetBufferHostSize_32f_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.37 NppStatus nppiNormRelInfGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.38 NppStatus nppiNormRelInfGetBufferHostSize_32f_C3CMR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.39 NppStatus nppiNormRelInfGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.40 NppStatus nppiNormRelInfGetBufferHostSize_32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_32f_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.41 NppStatus nppiNormRelInfGetBufferHostSize_32s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_32s_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.42 NppStatus nppiNormRelInfGetBufferHostSize_8s_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_8s_C1MR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.43 NppStatus nppiNormRelInfGetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8s_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.44 NppStatus nppiNormRelInfGetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.45 NppStatus nppiNormRelInfGetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.46 NppStatus nppiNormRelInfGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.47 NppStatus nppiNormRelInfGetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_8u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.48 NppStatus nppiNormRelInfGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_8u_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.109.2.49 NppStatus nppiNormRelInfGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_Inf_8u_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110 NormRel_L1

Primitives for computing the relative error of L1 norm between two images.

Basic NormRel_L1

- **NppStatus nppiNormRel_L1_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image NormRel_L1.
- **NppStatus nppiNormRel_L1_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image NormRel_L1.
- **NppStatus nppiNormRel_L1_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image NormRel_L1.
- **NppStatus nppiNormRel_L1_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image NormRel_L1.
- **NppStatus nppiNormRel_L1_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image NormRel_L1.
- **NppStatus nppiNormRel_L1_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image NormRel_L1.
- **NppStatus nppiNormRel_L1_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image NormRel_L1.
- **NppStatus nppiNormRel_L1_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image NormRel_L1.
- **NppStatus nppiNormRel_L1_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit signed image NormRel_L1 ignoring alpha channel.
- **NppStatus nppiNormRel_L1_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image NormRel_L1 ignoring alpha channel.
- **NppStatus nppiNormRel_L1_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image NormRel_L1 ignoring alpha channel.

- [NppStatus nppiNormRel_L1_32f_AC4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[3], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormRel_L1 ignoring alpha channel.
- [NppStatus nppiNormRel_L1_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image NormRel_L1.
- [NppStatus nppiNormRel_L1_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit unsigned image NormRel_L1.
- [NppStatus nppiNormRel_L1_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit signed image NormRel_L1.
- [NppStatus nppiNormRel_L1_32f_C4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)
Four-channel 32-bit floating point image NormRel_L1.

Masked NormRel_L1

See [Masked Operation](#).

- [NppStatus nppiNormRel_L1_8u_C1MR](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)
One-channel 8-bit unsigned image NormRel_L1.
- [NppStatus nppiNormRel_L1_8s_C1MR](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)
One-channel 8-bit signed image NormRel_L1.
- [NppStatus nppiNormRel_L1_16u_C1MR](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)
One-channel 16-bit unsigned image NormRel_L1.
- [NppStatus nppiNormRel_L1_32f_C1MR](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)
One-channel 32-bit floating point image NormRel_L1.

Masked Channel NormRel_L1

See [Masked Operation](#) and [Channel-of-Interest API](#).

- **NppStatus** **nppiNormRel_L1_8u_C3CMR** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)

Masked three-channel 8-bit unsigned image NormRel_L1 affecting only single channel.

- **NppStatus** **nppiNormRel_L1_8s_C3CMR** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)

Masked three-channel 8-bit signed image NormRel_L1 affecting only single channel.

- **NppStatus** **nppiNormRel_L1_16u_C3CMR** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)

Masked three-channel 16-bit unsigned image NormRel_L1 affecting only single channel.

- **NppStatus** **nppiNormRel_L1_32f_C3CMR** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, const **Npp8u** *pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)

Masked three-channel 32-bit floating point image NormRel_L1 affecting only single channel.

NormRelL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel_L1 primitives.

- **NppStatus** **nppiNormRelL1GetBufferHostSize_8u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C1R.

- **NppStatus** **nppiNormRelL1GetBufferHostSize_16u_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C1R.

- **NppStatus** **nppiNormRelL1GetBufferHostSize_16s_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C1R.

- **NppStatus** **nppiNormRelL1GetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C1R.

- **NppStatus** **nppiNormRelL1GetBufferHostSize_8u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C1MR.

- **NppStatus** **nppiNormRelL1GetBufferHostSize_8s_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8s_C1MR.

- **NppStatus** **nppiNormRelL1GetBufferHostSize_16u_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C1MR.

- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_32f_C1MR](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C1MR.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_8u_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C3R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_16u_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C3R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_16s_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C3R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_32f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C3R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_8u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C4R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_16u_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C4R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_16s_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C4R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_32f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C4R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_8u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_AC4R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_16u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_AC4R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_16s_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_AC4R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_32f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_AC4R.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_8u_C3CMR](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C3CMR.
- [NppStatus](#) [nppiNormRelL1GetBufferHostSize_8s_C3CMR](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8s_C3CMR.

- **NppStatus** [nppiNormRelL1GetBufferHostSize_16u_C3CMR](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C3CMR.

- **NppStatus** [nppiNormRelL1GetBufferHostSize_32f_C3CMR](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C3CMR.

7.110.1 Detailed Description

Primitives for computing the relative error of L1 norm between two images.

7.110.2 Function Documentation

7.110.2.1 **NppStatus** [nppiNormRel_L1_16s_AC4R](#) (**const** **Npp16s** * *pSrc1*, **int** *nSrc1Step*, **const** **Npp16s** * *pSrc2*, **int** *nSrc2Step*, **NppiSize** *oSizeROI*, **Npp64f** *aNormRel*[3], **Npp8u** * *pDeviceBuffer*)

Four-channel 16-bit signed image NormRel_L1 ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.110.2.2 **NppStatus** [nppiNormRel_L1_16s_C1R](#) (**const** **Npp16s** * *pSrc1*, **int** *nSrc1Step*, **const** **Npp16s** * *pSrc2*, **int** *nSrc2Step*, **NppiSize** *oSizeROI*, **Npp64f** * *pNormRel*, **Npp8u** * *pDeviceBuffer*)

One-channel 16-bit signed image NormRel_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormRelL1GetBufferHostSize_16s_C1R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.110.2.3 `NppStatus nppiNormRel_L1_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use `nppiNormRelL1GetBufferHostSize_16s_C3R` to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.110.2.4 `NppStatus nppiNormRel_L1_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.5 `NppStatus nppiNormRel_L1_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel_L1 ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.6 `NppStatus nppiNormRel_L1_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormRel_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.110.2.7 `NppStatus nppiNormRel_L1_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormRel_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.110.2.8 `NppStatus nppiNormRel_L1_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormRel_L1 affecting only single channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.110.2.9 `NppStatus nppiNormRel_L1_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL1GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.110.2.10 `NppStatus nppiNormRel_L1_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL1GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

7.110.2.11 `NppStatus nppiNormRel_L1_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel_L1 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNormRel Array that contains the computed relative error for the L1 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL1GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.12 `NppStatus nppiNormRel_L1_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNormRel Pointer to the computed relative error for the L1 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL1GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.13 `NppStatus nppiNormRel_L1_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNormRel Pointer to the computed relative error for the L1 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL1GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.14 `NppStatus nppiNormRel_L1_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormRel_L1 affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nCOI Channel_of_Interest Number.
pNormRel Pointer to the computed relative error for the L1 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL1GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified, NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.15 `NppStatus nppiNormRel_L1_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormRel_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aNormRel Array that contains the computed relative error for the L1 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL1GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.16 `NppStatus nppiNormRel_L1_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aNormRel Array that contains the computed relative error for the L1 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL1GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.17 `NppStatus nppiNormRel_L1_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image NormRel_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL1GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.18 `NppStatus nppiNormRel_L1_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormRel_L1 affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL1GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.19 `NppStatus nppiNormRel_L1_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image NormRel_L1 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNormRel Array that contains the computed relative error for the L1 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL1GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.20 `NppStatus nppiNormRel_L1_8u_C1MR (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, Npp64f *pNormRel, Npp8u *pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNormRel Pointer to the computed relative error for the L1 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL1GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.21 `NppStatus nppiNormRel_L1_8u_C1R (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pNormRel, Npp8u *pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.22 `NppStatus nppiNormRel_L1_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormRel_L1 affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.23 `NppStatus nppiNormRel_L1_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormRel_L1.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.24 `NppStatus nppiNormRel_L1_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel_L1.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the L1 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL1GetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L1 norm of the second image is zero.

7.110.2.25 `NppStatus nppiNormRelL1GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.26 NppStatus nppiNormRelL1GetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.27 NppStatus nppiNormRelL1GetBufferHostSize_16s_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.28 NppStatus nppiNormRelL1GetBufferHostSize_16s_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.29 NppStatus nppiNormRelL1GetBufferHostSize_16u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.30 NppStatus nppiNormRelL1GetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_16u_C1MR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.31 NppStatus nppiNormRelL1GetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_16u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.32 NppStatus nppiNormRelL1GetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_16u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.33 NppStatus nppiNormRelL1GetBufferHostSize_16u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.34 NppStatus nppiNormRelL1GetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.35 NppStatus nppiNormRelL1GetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.36 NppStatus nppiNormRelL1GetBufferHostSize_32f_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.37 NppStatus nppiNormRelL1GetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_32f_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.38 NppStatus nppiNormRelL1GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_32f_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.39 NppStatus nppiNormRelL1GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_32f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.40 NppStatus nppiNormRelL1GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.41 NppStatus nppiNormRelL1GetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8s_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.42 NppStatus nppiNormRelL1GetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8s_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.43 NppStatus nppiNormRelL1GetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.44 NppStatus nppiNormRelL1GetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_8u_C1MR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.45 NppStatus nppiNormRelL1GetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_8u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.46 NppStatus nppiNormRelL1GetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L1_8u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.47 NppStatus nppiNormRelL1GetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.110.2.48 NppStatus nppiNormRelL1GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111 NormRel_L2

Primitives for computing the relative error of L2 norm between two images.

Basic NormRel_L2

- **NppStatus nppiNormRel_L2_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image NormRel_L2.
- **NppStatus nppiNormRel_L2_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image NormRel_L2.
- **NppStatus nppiNormRel_L2_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image NormRel_L2.
- **NppStatus nppiNormRel_L2_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pNormRel, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image NormRel_L2.
- **NppStatus nppiNormRel_L2_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image NormRel_L2.
- **NppStatus nppiNormRel_L2_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image NormRel_L2.
- **NppStatus nppiNormRel_L2_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image NormRel_L2.
- **NppStatus nppiNormRel_L2_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image NormRel_L2.
- **NppStatus nppiNormRel_L2_8u_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image NormRel_L2 ignoring alpha channel.
- **NppStatus nppiNormRel_L2_16u_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image NormRel_L2 ignoring alpha channel.
- **NppStatus nppiNormRel_L2_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image NormRel_L2 ignoring alpha channel.

- [NppStatus nppiNormRel_L2_32f_AC4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[3], [Npp8u](#) *pDeviceBuffer)

Four-channel 32-bit floating point image NormRel_L2 ignoring alpha channel.

- [NppStatus nppiNormRel_L2_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 8-bit unsigned image NormRel_L2.

- [NppStatus nppiNormRel_L2_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit unsigned image NormRel_L2.

- [NppStatus nppiNormRel_L2_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit signed image NormRel_L2.

- [NppStatus nppiNormRel_L2_32f_C4R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) *pDeviceBuffer)

Four-channel 32-bit floating point image NormRel_L2.

Masked NormRel_L2

See [Masked Operation](#).

- [NppStatus nppiNormRel_L2_8u_C1MR](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 8-bit unsigned image NormRel_L2.

- [NppStatus nppiNormRel_L2_8s_C1MR](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 8-bit signed image NormRel_L2.

- [NppStatus nppiNormRel_L2_16u_C1MR](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 16-bit unsigned image NormRel_L2.

- [NppStatus nppiNormRel_L2_32f_C1MR](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, const [Npp8u](#) *pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) *pNormRel, [Npp8u](#) *pDeviceBuffer)

Masked one-channel 32-bit floating point image NormRel_L2.

Masked Channel NormRel_L2

See [Masked Operation](#) and [Channel-of-Interest API](#).

- `NppStatus nppiNormRel_L2_8u_C3CMR` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)

Masked three-channel 8-bit unsigned image NormRel_L2 affecting only single channel.

- `NppStatus nppiNormRel_L2_8s_C3CMR` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)

Masked three-channel 8-bit signed image NormRel_L2 affecting only single channel.

- `NppStatus nppiNormRel_L2_16u_C3CMR` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)

Masked three-channel 16-bit unsigned image NormRel_L2 affecting only single channel.

- `NppStatus nppiNormRel_L2_32f_C3CMR` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, const `Npp8u` *pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` *pNormRel, `Npp8u` *pDeviceBuffer)

Masked three-channel 32-bit floating point image NormRel_L2 affecting only single channel.

NormRelL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel_L2 primitives.

- `NppStatus nppiNormRelL2GetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C1R.

- `NppStatus nppiNormRelL2GetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C1R.

- `NppStatus nppiNormRelL2GetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C1R.

- `NppStatus nppiNormRelL2GetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C1R.

- `NppStatus nppiNormRelL2GetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C1MR.

- `NppStatus nppiNormRelL2GetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8s_C1MR.

- `NppStatus nppiNormRelL2GetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C1MR.

- **NppStatus** **nppiNormRelL2GetBufferHostSize_32f_C1MR** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C1MR.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_8u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C3R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_16u_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C3R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_16s_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C3R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_32f_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C3R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_8u_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C4R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_16u_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C4R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_16s_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C4R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_32f_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C4R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_8u_AC4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_AC4R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_16u_AC4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_AC4R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_16s_AC4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_AC4R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_32f_AC4R** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_AC4R.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_8u_C3CMR** (**NppiSize** oSizeROI, int *hpBufferSize)
Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C3CMR.
- **NppStatus** **nppiNormRelL2GetBufferHostSize_8s_C3CMR** (**NppiSize** oSizeROI, int *hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8s_C3CMR.

- **NppStatus** [nppiNormRelL2GetBufferHostSize_16u_C3CMR](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C3CMR.

- **NppStatus** [nppiNormRelL2GetBufferHostSize_32f_C3CMR](#) (**NppiSize** **oSizeROI**, **int** ***hpBufferSize**)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C3CMR.

7.111.1 Detailed Description

Primitives for computing the relative error of L2 norm between two images.

7.111.2 Function Documentation

7.111.2.1 **NppStatus** [nppiNormRel_L2_16s_AC4R](#) (**const** **Npp16s** * **pSrc1**, **int** **nSrc1Step**, **const** **Npp16s** * **pSrc2**, **int** **nSrc2Step**, **NppiSize** **oSizeROI**, **Npp64f** **aNormRel[3]**, **Npp8u** * **pDeviceBuffer**)

Four-channel 16-bit signed image NormRel_L2 ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16s_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or **NPP_DIVISOR_ERROR** if the L2 norm of the second image is zero.

7.111.2.2 **NppStatus** [nppiNormRel_L2_16s_C1R](#) (**const** **Npp16s** * **pSrc1**, **int** **nSrc1Step**, **const** **Npp16s** * **pSrc2**, **int** **nSrc2Step**, **NppiSize** **oSizeROI**, **Npp64f** * **pNormRel**, **Npp8u** * **pDeviceBuffer**)

One-channel 16-bit signed image NormRel_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.3 `NppStatus nppiNormRel_L2_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.4 `NppStatus nppiNormRel_L2_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.5 `NppStatus nppiNormRel_L2_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel_L2 ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.6 `NppStatus nppiNormRel_L2_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned image NormRel_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.7 `NppStatus nppiNormRel_L2_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormRel_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.8 `NppStatus nppiNormRel_L2_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormRel_L2 affecting only single channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pMask [Mask-Image Pointer](#).

nMaskStep [Mask-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nCOI [Channel_of_Interest Number](#).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_16u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.9 NppStatus nppiNormRel_L2_16u_C3R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aNormRel*[3], Npp8u * *pDeviceBuffer*)

Three-channel 16-bit unsigned image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL2GetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.10 NppStatus nppiNormRel_L2_16u_C4R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aNormRel*[4], Npp8u * *pDeviceBuffer*)

Four-channel 16-bit unsigned image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL2GetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.11 NppStatus nppiNormRel_L2_32f_AC4R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aNormRel*[3], Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image NormRel_L2 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNormRel Array that contains the computed relative error for the L2 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL2GetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.12 `NppStatus nppiNormRel_L2_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNormRel Pointer to the computed relative error for the L2 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL2GetBufferHostSize_32f_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.13 `NppStatus nppiNormRel_L2_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNormRel Pointer to the computed relative error for the L2 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL2GetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.14 `NppStatus nppiNormRel_L2_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormRel_L2 affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nCOI Channel_of_Interest Number.
pNormRel Pointer to the computed relative error for the L2 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL2GetBufferHostSize_32f_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.15 `NppStatus nppiNormRel_L2_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormRel_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.16 `NppStatus nppiNormRel_L2_32f_C4R (const Npp32f *pSrc1, int nSrc1Step, const Npp32f *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u *pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.17 `NppStatus nppiNormRel_L2_8s_C1MR (const Npp8s *pSrc1, int nSrc1Step, const Npp8s *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, Npp64f *pNormRel, Npp8u *pDeviceBuffer)`

Masked one-channel 8-bit signed image NormRel_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL2GetBufferHostSize_8s_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.18 `NppStatus nppiNormRel_L2_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormRel_L2 affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL2GetBufferHostSize_8s_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_COI_ERROR if an invalid channel of interest is specified, or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.19 `NppStatus nppiNormRel_L2_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel_L2 ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aNormRel Array that contains the computed relative error for the L2 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL2GetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.20 `NppStatus nppiNormRel_L2_8u_C1MR (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, Npp64f *pNormRel, Npp8u *pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pMask Mask-Image Pointer.
nMaskStep Mask-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pNormRel Pointer to the computed relative error for the L2 norm of two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiNormRelL2GetBufferHostSize_8u_C1MR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.21 `NppStatus nppiNormRel_L2_8u_C1R (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pNormRel, Npp8u *pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.111.2.22 `NppStatus nppiNormRel_L2_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormRel_L2 affecting only single channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pMask Mask-Image Pointer.

nMaskStep Mask-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nCOI Channel_of_Interest Number.

pNormRel Pointer to the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_8u_C3CMR](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

7.111.2.23 `NppStatus nppiNormRel_L2_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormRel_L2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.24 `NppStatus nppiNormRel_L2_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel_L2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aNormRel Array that contains the computed relative error for the L2 norm of two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiNormRelL2GetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_DIVISOR_ERROR if the L2 norm of the second image is zero.

7.111.2.25 `NppStatus nppiNormRelL2GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.26 NppStatus nppiNormRelL2GetBufferHostSize_16s_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.27 NppStatus nppiNormRelL2GetBufferHostSize_16s_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.28 NppStatus nppiNormRelL2GetBufferHostSize_16s_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.29 NppStatus nppiNormRelL2GetBufferHostSize_16u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.30 NppStatus nppiNormRelL2GetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_16u_C1MR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.31 NppStatus nppiNormRelL2GetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_16u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.32 NppStatus nppiNormRelL2GetBufferHostSize_16u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_16u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.33 NppStatus nppiNormRelL2GetBufferHostSize_16u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.34 NppStatus nppiNormRelL2GetBufferHostSize_16u_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.35 NppStatus nppiNormRelL2GetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.36 NppStatus nppiNormRelL2GetBufferHostSize_32f_C1MR (NppiSize *oSizeROI*, int * *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.37 NppStatus nppiNormRelL2GetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_32f_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.38 NppStatus nppiNormRelL2GetBufferHostSize_32f_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_32f_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.39 NppStatus nppiNormRelL2GetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_32f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.40 NppStatus nppiNormRelL2GetBufferHostSize_32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.41 NppStatus nppiNormRelL2GetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8s_C1MR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.42 NppStatus nppiNormRelL2GetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8s_C3CMR.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.43 NppStatus nppiNormRelL2GetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.44 NppStatus nppiNormRelL2GetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_8u_C1MR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.45 NppStatus nppiNormRelL2GetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_8u_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.111.2.46 NppStatus nppiNormRelL2GetBufferHostSize_8u_C3CMR (NppiSize oSizeROI, int * hpBufferSize)

Computes the device scratch buffer size (in bytes) for *nppiNormRel_L2_8u_C3CMR*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.111.2.47 NppStatus nppiNormRelL2GetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.111.2.48 NppStatus nppiNormRelL2GetBufferHostSize_8u_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112 DotProd

Primitives for computing the dot product of two images.

DotProd

Given two images $pSrc1$ and $pSrc2$ both with width W and height H , the dot product will be computed as

$$DotProd = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [pSrc1(j, i) \cdot pSrc2(j, i)]$$

The functions require additional scratch buffer for computations.

- **NppStatus** **nppiDotProd_8u64f_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image DotProd.
- **NppStatus** **nppiDotProd_8s64f_C1R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 8-bit signed image DotProd.
- **NppStatus** **nppiDotProd_16u64f_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image DotProd.
- **NppStatus** **nppiDotProd_16s64f_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image DotProd.
- **NppStatus** **nppiDotProd_32u64f_C1R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 32-bit unsigned image DotProd.
- **NppStatus** **nppiDotProd_32s64f_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 32-bit signed image DotProd.
- **NppStatus** **nppiDotProd_32f64f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pDp, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image DotProd.
- **NppStatus** **nppiDotProd_8u64f_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image DotProd.
- **NppStatus** **nppiDotProd_8s64f_C3R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)
Three-channel 8-bit signed image DotProd.
- **NppStatus** **nppiDotProd_16u64f_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)

Three-channel 16-bit unsigned image DotProd.

- `NppStatus nppiDotProd_16s64f_C3R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)

Three-channel 16-bit signed image DotProd.

- `NppStatus nppiDotProd_32u64f_C3R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)

Three-channel 32-bit unsigned image DotProd.

- `NppStatus nppiDotProd_32s64f_C3R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)

Three-channel 32-bit signed image DotProd.

- `NppStatus nppiDotProd_32f64f_C3R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)

Three-channel 32-bit floating point image DotProd.

- `NppStatus nppiDotProd_8u64f_C4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image DotProd.

- `NppStatus nppiDotProd_8s64f_C4R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 8-bit signed image DotProd.

- `NppStatus nppiDotProd_16u64f_C4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 16-bit unsigned image DotProd.

- `NppStatus nppiDotProd_16s64f_C4R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 16-bit signed image DotProd.

- `NppStatus nppiDotProd_32u64f_C4R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 32-bit unsigned image DotProd.

- `NppStatus nppiDotProd_32s64f_C4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 32-bit signed image DotProd.

- `NppStatus nppiDotProd_32f64f_C4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image DotProd.

- `NppStatus nppiDotProd_8u64f_AC4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image DotProd ignoring alpha channel.

- **NppStatus** **nppiDotProd_8s64f_AC4R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)
Four-channel 8-bit signed image DotProd ignoring alpha channel.
- **NppStatus** **nppiDotProd_16u64f_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image DotProd ignoring alpha channel.
- **NppStatus** **nppiDotProd_16s64f_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)
Four-channel 16-bit signed image DotProd ignoring alpha channel.
- **NppStatus** **nppiDotProd_32u64f_AC4R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)
Four-channel 32-bit unsigned image DotProd ignoring alpha channel.
- **NppStatus** **nppiDotProd_32s64f_AC4R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)
Four-channel 32-bit signed image DotProd ignoring alpha channel.
- **NppStatus** **nppiDotProd_32f64f_AC4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** *pDeviceBuffer)
Four-channel 32-bit floating point image DotProd ignoring alpha channel.

DotProdGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean_StdDev primitives.

- **NppStatus** **nppiDotProdGetBufferHostSize_8u64f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C1R.
- **NppStatus** **nppiDotProdGetBufferHostSize_8s64f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8s64f_C1R.
- **NppStatus** **nppiDotProdGetBufferHostSize_16u64f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C1R.
- **NppStatus** **nppiDotProdGetBufferHostSize_16s64f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16s64f_C1R.
- **NppStatus** **nppiDotProdGetBufferHostSize_32u64f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C1R.
- **NppStatus** **nppiDotProdGetBufferHostSize_32s64f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32s64f_C1R.
- **NppStatus** **nppiDotProdGetBufferHostSize_32f64f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32f64f_C1R.

- [NppStatus nppiDotProdGetBufferHostSize_8u64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_8s64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8s64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_16u64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_16s64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16s64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_32u64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_32s64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32s64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_32f64f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32f64f_C3R.
- [NppStatus nppiDotProdGetBufferHostSize_8u64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_8s64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8s64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_16u64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_16s64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_16s64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_32u64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_32s64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32s64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_32f64f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_32f64f_C4R.
- [NppStatus nppiDotProdGetBufferHostSize_8u64f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8u64f_AC4R.
- [NppStatus nppiDotProdGetBufferHostSize_8s64f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiDotProd_8s64f_AC4R.

- [NppStatus](#) [nppiDotProdGetBufferHostSize_16u64f_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Device scratch buffer size (in bytes) for nppiDotProd_16u64f_AC4R.
- [NppStatus](#) [nppiDotProdGetBufferHostSize_16s64f_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Device scratch buffer size (in bytes) for nppiDotProd_16s64f_AC4R.
- [NppStatus](#) [nppiDotProdGetBufferHostSize_32u64f_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Device scratch buffer size (in bytes) for nppiDotProd_32u64f_AC4R.
- [NppStatus](#) [nppiDotProdGetBufferHostSize_32s64f_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Device scratch buffer size (in bytes) for nppiDotProd_32s64f_AC4R.
- [NppStatus](#) [nppiDotProdGetBufferHostSize_32f64f_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))
Device scratch buffer size (in bytes) for nppiDotProd_32f64f_AC4R.

7.112.1 Detailed Description

Primitives for computing the dot product of two images.

7.112.2 Function Documentation

7.112.2.1 [NppStatus](#) [nppiDotProd_16s64f_AC4R](#) ([const](#) [Npp16s](#) * [pSrc1](#), [int](#) [nSrc1Step](#), [const](#) [Npp16s](#) * [pSrc2](#), [int](#) [nSrc2Step](#), [NppiSize](#) [oSizeROI](#), [Npp64f](#) [aDp](#)[3], [Npp8u](#) * [pDeviceBuffer](#))

Four-channel 16-bit signed image DotProd ignoring alpha channel.

Parameters:

[pSrc1](#) [Source-Image Pointer](#).

[nSrc1Step](#) [Source-Image Line Step](#).

[pSrc2](#) [Source-Image Pointer](#).

[nSrc2Step](#) [Source-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[aDp](#) Array that contains the computed dot product of the two images.

[pDeviceBuffer](#) [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
 Use [nppiDotProdGetBufferHostSize_16s64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.2 `NppStatus nppiDotProd_16s64f_C1R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16s64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.3 `NppStatus nppiDotProd_16s64f_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16s64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.4 `NppStatus nppiDotProd_16s64f_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aDp Array that contains the computed dot product of the two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiDotProdGetBufferHostSize_16s64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.5 `NppStatus nppiDotProd_16u64f_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image DotProd ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aDp Array that contains the computed Inf-norm of differences.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiDotProdGetBufferHostSize_16u64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.6 `NppStatus nppiDotProd_16u64f_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16u64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.7 `NppStatus nppiDotProd_16u64f_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDp Array that contains the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16u64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.8 `NppStatus nppiDotProd_16u64f_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDp Array that contains the computed Inf-norm of differences.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_16u64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.9 `NppStatus nppiDotProd_32f64f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image DotProd ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32f64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.10 `NppStatus nppiDotProd_32f64f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32f64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.11 `NppStatus nppiDotProd_32f64f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aDp Array that contains the computed dot product of the two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiDotProdGetBufferHostSize_32f64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.12 `NppStatus nppiDotProd_32f64f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aDp Array that contains the computed dot product of the two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiDotProdGetBufferHostSize_32f64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.13 `NppStatus nppiDotProd_32s64f_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image DotProd ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32s64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.14 `NppStatus nppiDotProd_32s64f_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32s64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.15 `NppStatus nppiDotProd_32s64f_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32s64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.16 `NppStatus nppiDotProd_32s64f_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32s64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.17 `NppStatus nppiDotProd_32u64f_AC4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image DotProd ignoring alpha channel.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_32u64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.18 `NppStatus nppiDotProd_32u64f_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image DotProd.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pDp Pointer to the computed dot product of the two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiDotProdGetBufferHostSize_32u64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.19 `NppStatus nppiDotProd_32u64f_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image DotProd.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
aDp Array that contains the computed dot product of the two images.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiDotProdGetBufferHostSize_32u64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.20 `NppStatus nppiDotProd_32u64f_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image DotProd.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferSize_32u64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.21 `NppStatus nppiDotProd_8s64f_AC4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image DotProd ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferSize_8s64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.22 `NppStatus nppiDotProd_8s64f_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferSize_8s64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.23 NppStatus nppiDotProd_8s64f_C3R (const Npp8s * *pSrc1*, int *nSrc1Step*, const Npp8s * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aDp*[3], Npp8u * *pDeviceBuffer*)

Three-channel 8-bit signed image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_8s64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.24 NppStatus nppiDotProd_8s64f_C4R (const Npp8s * *pSrc1*, int *nSrc1Step*, const Npp8s * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aDp*[4], Npp8u * *pDeviceBuffer*)

Four-channel 8-bit signed image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_8s64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.25 NppStatus nppiDotProd_8u64f_AC4R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aDp*[3], Npp8u * *pDeviceBuffer*)

Four-channel 8-bit unsigned image DotProd ignoring alpha channel.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_8u64f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.26 `NppStatus nppiDotProd_8u64f_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pDp Pointer to the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_8u64f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.27 `NppStatus nppiDotProd_8u64f_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image DotProd.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_8u64f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.28 `NppStatus nppiDotProd_8u64f_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image DotProd.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aDp Array that contains the computed dot product of the two images.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiDotProdGetBufferHostSize_8u64f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.112.2.29 `NppStatus nppiDotProdGetBufferHostSize_16s64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for nppiDotProd_16s64f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.30 `NppStatus nppiDotProdGetBufferHostSize_16s64f_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for nppiDotProd_16s64f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.31 NppStatus nppiDotProdGetBufferHostSize_16s64f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for *nppiDotProd_16s64f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.32 NppStatus nppiDotProdGetBufferHostSize_16s64f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for *nppiDotProd_16s64f_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.33 NppStatus nppiDotProdGetBufferHostSize_16u64f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for *nppiDotProd_16u64f_AC4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.34 NppStatus nppiDotProdGetBufferHostSize_16u64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.35 NppStatus nppiDotProdGetBufferHostSize_16u64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.36 NppStatus nppiDotProdGetBufferHostSize_16u64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.37 NppStatus nppiDotProdGetBufferHostSize_32f64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_32f64f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.38 NppStatus nppiDotProdGetBufferHostSize_32f64f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for *nppiDotProd_32f64f_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.39 NppStatus nppiDotProdGetBufferHostSize_32f64f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for *nppiDotProd_32f64f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.40 NppStatus nppiDotProdGetBufferHostSize_32f64f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for *nppiDotProd_32f64f_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.41 NppStatus nppiDotProdGetBufferHostSize_32s64f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiDotProd_32s64f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.42 NppStatus nppiDotProdGetBufferHostSize_32s64f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiDotProd_32s64f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.43 NppStatus nppiDotProdGetBufferHostSize_32s64f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiDotProd_32s64f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.44 NppStatus nppiDotProdGetBufferHostSize_32s64f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiDotProd_32s64f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.45 NppStatus nppiDotProdGetBufferHostSize_32u64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for *nppiDotProd_32u64f_AC4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.46 NppStatus nppiDotProdGetBufferHostSize_32u64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for *nppiDotProd_32u64f_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.47 NppStatus nppiDotProdGetBufferHostSize_32u64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for *nppiDotProd_32u64f_C3R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.48 NppStatus nppiDotProdGetBufferHostSize_32u64f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.49 NppStatus nppiDotProdGetBufferHostSize_8s64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_8s64f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.50 NppStatus nppiDotProdGetBufferHostSize_8s64f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_8s64f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.112.2.51 NppStatus nppiDotProdGetBufferHostSize_8s64f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiDotProd_8s64f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.112.2.52 NppStatus nppiDotProdGetBufferHostSize_8s64f_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiDotProd_8s64f_C4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.112.2.53 NppStatus nppiDotProdGetBufferHostSize_8u64f_AC4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiDotProd_8u64f_AC4R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.112.2.54 NppStatus nppiDotProdGetBufferHostSize_8u64f_C1R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiDotProd_8u64f_C1R*.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.112.2.55 NppStatus nppiDotProdGetBufferHostSize_8u64f_C3R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.112.2.56 NppStatus nppiDotProdGetBufferHostSize_8u64f_C4R (NppiSize *oSizeROI*, int *
hpBufferSize)**

Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.113 CountInRange.

Primitives for computing the amount of pixels that fall into the specified intensity range.

CountInRange

The lower bound and the upper bound are inclusive.

The functions require additional scratch buffer for computations.

- `NppStatus nppiCountInRange_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int *pCounts, `Npp8u` nLowerBound, `Npp8u` nUpperBound, `Npp8u` *pDeviceBuffer)
One-channel 8-bit unsigned image CountInRange.
- `NppStatus nppiCountInRange_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int *pCounts, `Npp32f` nLowerBound, `Npp32f` nUpperBound, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point image CountInRange.
- `NppStatus nppiCountInRange_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp8u` aLowerBound[3], `Npp8u` aUpperBound[3], `Npp8u` *pDeviceBuffer)
Three-channel 8-bit unsigned image CountInRange.
- `NppStatus nppiCountInRange_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp32f` aLowerBound[3], `Npp32f` aUpperBound[3], `Npp8u` *pDeviceBuffer)
Three-channel 32-bit floating point image CountInRange.
- `NppStatus nppiCountInRange_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp8u` aLowerBound[3], `Npp8u` aUpperBound[3], `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image CountInRange ignoring alpha channel.
- `NppStatus nppiCountInRange_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp32f` aLowerBound[3], `Npp32f` aUpperBound[3], `Npp8u` *pDeviceBuffer)
Four-channel 32-bit floating point image CountInRange ignoring alpha channel.

CountInRangeGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CountInRange primitives.

- `NppStatus nppiCountInRangeGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiCountInRange_8u_C1R.
- `NppStatus nppiCountInRangeGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppiCountInRange_32f_C1R.
- `NppStatus nppiCountInRangeGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_C3R.

- [NppStatus](#) [nppiCountInRangeGetBufferHostSize_32f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_C3R.

- [NppStatus](#) [nppiCountInRangeGetBufferHostSize_8u_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_AC4R.

- [NppStatus](#) [nppiCountInRangeGetBufferHostSize_32f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_AC4R.

7.113.1 Detailed Description

Primitives for computing the amount of pixels that fall into the specified intensity range.

7.113.2 Function Documentation

7.113.2.1 [NppStatus nppiCountInRange_32f_AC4R](#) (const [Npp32f](#) * *pSrc*, int *nSrcStep*, [NppiSize](#) oSizeROI, int *aCounts*[3], [Npp32f](#) *aLowerBound*[3], [Npp32f](#) *aUpperBound*[3], [Npp8u](#) * *pDeviceBuffer*)

Four-channel 32-bit floating point image CountInRange ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aCounts Array that contains the number of pixels that fall into the specified range for Three-channels.

aLowerBound Fixed size array of the lower bound of the specified range, one per channel.

aUpperBound Fixed size array of the upper bound of the specified range, one per channel.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiCountInRangeGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or [NPP_RANGE_ERROR](#) if the lower bound is larger than the upper bound.

7.113.2.2 [NppStatus nppiCountInRange_32f_C1R](#) (const [Npp32f](#) * *pSrc*, int *nSrcStep*, [NppiSize](#) oSizeROI, int * *pCounts*, [Npp32f](#) *nLowerBound*, [Npp32f](#) *nUpperBound*, [Npp8u](#) * *pDeviceBuffer*)

One-channel 32-bit floating point image CountInRange.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pCounts Pointer to the number of pixels that fall into the specified range.
nLowerBound Lower bound of the specified range.
nUpperBound Upper bound of the specified range.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiCountInRangeGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_RANGE_ERROR if the lower bound is larger than the upper bound.

7.113.2.3 `NppStatus nppiCountInRange_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, int aCounts[3], Npp32f aLowerBound[3], Npp32f aUpperBound[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image CountInRange.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
aCounts Array that contains the number of pixels that fall into the specified range for Three-channels.
aLowerBound Fixed size array of the lower bound of the specified range, one per channel.
aUpperBound Fixed size array of the upper bound of the specified range, one per channel.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiCountInRangeGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_RANGE_ERROR if the lower bound is larger than the upper bound.

7.113.2.4 `NppStatus nppiCountInRange_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int aCounts[3], Npp8u aLowerBound[3], Npp8u aUpperBound[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CountInRange ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).

aCounts Array that contains the number of pixels that fall into the specified range for Three-channels.

aLowerBound Fixed size array of the lower bound of the specified range, one per channel.

aUpperBound Fixed size array of the upper bound of the specified range, one per channel.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiCountInRangeGetBufferHostSize_8u_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_RANGE_ERROR if the lower bound is larger than the upper bound.

7.113.2.5 `NppStatus nppiCountInRange_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int * pCounts, Npp8u nLowerBound, Npp8u nUpperBound, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CountInRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pCounts Pointer to the number of pixels that fall into the specified range.

nLowerBound Lower bound of the specified range.

nUpperBound Upper bound of the specified range.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiCountInRangeGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_RANGE_ERROR if the lower bound is larger than the upper bound.

7.113.2.6 `NppStatus nppiCountInRange_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int aCounts[3], Npp8u aLowerBound[3], Npp8u aUpperBound[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CountInRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

aCounts Array that contains the number of pixels that fall into the specified range for Three-channels.

aLowerBound Fixed size array of the lower bound of the specified range, one per channel.

aUpperBound Fixed size array of the upper bound of the specified range, one per channel.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiCountInRangeGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_RANGE_ERROR if the lower bound is larger than the upper bound.

7.113.2.7 NppStatus nppiCountInRangeGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.113.2.8 NppStatus nppiCountInRangeGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.113.2.9 NppStatus nppiCountInRangeGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.113.2.10 NppStatus nppiCountInRangeGetBufferHostSize_8u_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.113.2.11 NppStatus nppiCountInRangeGetBufferHostSize_8u_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.113.2.12 NppStatus nppiCountInRangeGetBufferHostSize_8u_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.114 MaxEvery

Primitives for computing the maximal value of the pixel pair from two images.

MaxEvery

The maximum is stored into the second image.

- [NppStatus nppiMaxEvery_8u_C1IR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 8-bit unsigned image MaxEvery.
- [NppStatus nppiMaxEvery_16u_C1IR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 16-bit unsigned image MaxEvery.
- [NppStatus nppiMaxEvery_16s_C1IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 16-bit signed image MaxEvery.
- [NppStatus nppiMaxEvery_32f_C1IR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 32-bit floating point image MaxEvery.
- [NppStatus nppiMaxEvery_8u_C3IR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 8-bit unsigned image MaxEvery.
- [NppStatus nppiMaxEvery_16u_C3IR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 16-bit unsigned image MaxEvery.
- [NppStatus nppiMaxEvery_16s_C3IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 16-bit signed image MaxEvery.
- [NppStatus nppiMaxEvery_32f_C3IR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 32-bit floating point image MaxEvery.
- [NppStatus nppiMaxEvery_8u_C4IR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Four-channel 8-bit unsigned image MaxEvery.
- [NppStatus nppiMaxEvery_16u_C4IR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Four-channel 16-bit unsigned image MaxEvery.
- [NppStatus nppiMaxEvery_16s_C4IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 16-bit signed image MaxEvery.

- **NppStatus nppiMaxEvery_32f_C4IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 32-bit floating point image MaxEvery.

- **NppStatus nppiMaxEvery_8u_AC4IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 8-bit unsigned image MaxEvery ignoring alpha channel.

- **NppStatus nppiMaxEvery_16u_AC4IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 16-bit unsigned image MaxEvery ignoring alpha channel.

- **NppStatus nppiMaxEvery_16s_AC4IR** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 16-bit signed image MaxEvery ignoring alpha channel.

- **NppStatus nppiMaxEvery_32f_AC4IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 32-bit floating point image MaxEvery ignoring alpha channel.

7.114.1 Detailed Description

Primitives for computing the maximal value of the pixel pair from two images.

7.114.2 Function Documentation

7.114.2.1 **NppStatus nppiMaxEvery_16s_AC4IR** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 16-bit signed image MaxEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.2 **NppStatus nppiMaxEvery_16s_C1IR** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One-channel 16-bit signed image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.3 **NppStatus nppiMaxEvery_16s_C3IR** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three-channel 16-bit signed image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.4 **NppStatus nppiMaxEvery_16s_C4IR** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit signed image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.5 NppStatus nppiMaxEvery_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four-channel 16-bit unsigned image MaxEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.6 NppStatus nppiMaxEvery_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One-channel 16-bit unsigned image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.7 NppStatus nppiMaxEvery_16u_C3IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three-channel 16-bit unsigned image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.8 **NppStatus nppiMaxEvery_16u_C4IR** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit unsigned image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.9 **NppStatus nppiMaxEvery_32f_AC4IR** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating point image MaxEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.10 **NppStatus nppiMaxEvery_32f_C1IR** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One-channel 32-bit floating point image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.11 NppStatus nppiMaxEvery_32f_C3IR (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit floating point image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.12 NppStatus nppiMaxEvery_32f_C4IR (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating point image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.13 NppStatus nppiMaxEvery_8u_AC4IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 8-bit unsigned image MaxEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.14 NppStatus nppiMaxEvery_8u_C1IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One-channel 8-bit unsigned image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.15 NppStatus nppiMaxEvery_8u_C3IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three-channel 8-bit unsigned image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.114.2.16 NppStatus nppiMaxEvery_8u_C4IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 8-bit unsigned image MaxEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115 MinEvery

Primitives for computing the minimal value of the pixel pair from two images.

MinEvery

The minimum is stored into the second image.

- [NppStatus nppiMinEvery_8u_C1IR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 8-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16u_C1IR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 16-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16s_C1IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 16-bit signed image MinEvery.
- [NppStatus nppiMinEvery_32f_C1IR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
One-channel 32-bit floating point image MinEvery.
- [NppStatus nppiMinEvery_8u_C3IR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 8-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16u_C3IR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 16-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16s_C3IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 16-bit signed image MinEvery.
- [NppStatus nppiMinEvery_32f_C3IR](#) (const [Npp32f](#) *pSrc, int nSrcStep, [Npp32f](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Three-channel 32-bit floating point image MinEvery.
- [NppStatus nppiMinEvery_8u_C4IR](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp8u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Four-channel 8-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16u_C4IR](#) (const [Npp16u](#) *pSrc, int nSrcStep, [Npp16u](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
Four-channel 16-bit unsigned image MinEvery.
- [NppStatus nppiMinEvery_16s_C4IR](#) (const [Npp16s](#) *pSrc, int nSrcStep, [Npp16s](#) *pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

Four-channel 16-bit signed image MinEvery.

- **NppStatus nppiMinEvery_32f_C4IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 32-bit floating point image MinEvery.

- **NppStatus nppiMinEvery_8u_AC4IR** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 8-bit unsigned image MinEvery ignoring alpha channel.

- **NppStatus nppiMinEvery_16u_AC4IR** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 16-bit unsigned image MinEvery ignoring alpha channel.

- **NppStatus nppiMinEvery_16s_AC4IR** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 16-bit signed image MinEvery ignoring alpha channel.

- **NppStatus nppiMinEvery_32f_AC4IR** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 32-bit floating point image MinEvery ignoring alpha channel.

7.115.1 Detailed Description

Primitives for computing the minimal value of the pixel pair from two images.

7.115.2 Function Documentation

7.115.2.1 **NppStatus nppiMinEvery_16s_AC4IR** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 16-bit signed image MinEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSrcDst In-Place Image Pointer.

nSrcDstStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.2 NppStatus nppiMinEvery_16s_C1IR (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One-channel 16-bit signed image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.3 NppStatus nppiMinEvery_16s_C3IR (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three-channel 16-bit signed image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.4 NppStatus nppiMinEvery_16s_C4IR (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit signed image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.5 **NppStatus nppiMinEvery_16u_AC4IR** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit unsigned image MinEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.6 **NppStatus nppiMinEvery_16u_C1IR** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One-channel 16-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.7 **NppStatus nppiMinEvery_16u_C3IR** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three-channel 16-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.8 NppStatus nppiMinEvery_16u_C4IR (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.9 NppStatus nppiMinEvery_32f_AC4IR (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating point image MinEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.10 NppStatus nppiMinEvery_32f_C1IR (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One-channel 32-bit floating point image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.11 NppStatus nppiMinEvery_32f_C3IR (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit floating point image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.12 NppStatus nppiMinEvery_32f_C4IR (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating point image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.13 NppStatus nppiMinEvery_8u_AC4IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 8-bit unsigned image MinEvery ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.14 NppStatus nppiMinEvery_8u_C1IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One-channel 8-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.15 NppStatus nppiMinEvery_8u_C3IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three-channel 8-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.115.2.16 NppStatus nppiMinEvery_8u_C4IR (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four-channel 8-bit unsigned image MinEvery.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pSrcDst In-Place Image Pointer.
nSrcDstStep Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.116 Integral

Primitives for computing the integral image of a given image.

Integral

Given an input image $pSrc$ and the specified value $nVal$, the pixel value of the integral image $pDst$ at coordinate (i, j) will be computed as

$$pDst(j, i) = nVal + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)$$

If the size of the input image is $W \times H$, the size of the integral image will be $(W + 1) \times (H + 1)$.

- [NppStatus nppiIntegral_8u32s_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp32s](#) *pDst, int nDstStep, [NppiSize](#) oROI, [Npp32s](#) nVal)

One-channel 8-bit unsigned image Integral with 32-bit signed output.

- [NppStatus nppiIntegral_8u32f_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oROI, [Npp32f](#) nVal)

One-channel 8-bit unsigned image Integral with 32-bit floating point output.

7.116.1 Detailed Description

Primitives for computing the integral image of a given image.

7.116.2 Function Documentation

7.116.2.1 [NppStatus nppiIntegral_8u32f_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [Npp32f](#) *pDst, int nDstStep, [NppiSize](#) oROI, [Npp32f](#) nVal)

One-channel 8-bit unsigned image Integral with 32-bit floating point output.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oROI [Region-of-Interest \(ROI\)](#).

nVal The value to add to pDst image pixels

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.116.2.2 `NppStatus nppiIntegral_8u32s_C1R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oROI, Npp32s nVal)`

One-channel 8-bit unsigned image Integral with 32-bit signed output.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oROI Region-of-Interest (ROI).

nVal The value to add to pDst image pixels

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.117 SqrIntegral

Primitives for computing both the integral and the squared integral images of a given image.

SqrIntegral

Given an input image $pSrc$ and the specified value $nVal$, the pixel value of the integral image $pDst$ at coordinate (i, j) will be computed as

$$pDst(j, i) = nVal + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)$$

Given an input image $pSrc$ and the specified value $nValSqr$, the pixel value of the squared integral image $pSqr$ at coordinate (i, j) will be computed as

$$pSqr(j, i) = nValSqr + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)^2$$

If the size of the input image is $W \times H$, the size of the squared integral image will be $(W + 1) \times (H + 1)$.

- **NppStatus nppiSqrIntegral_8u32s_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **Npp32s** *pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32s** nVal, **Npp32s** nValSqr)
One-channel 8-bit unsigned image SqrIntegral.
- **NppStatus nppiSqrIntegral_8u32s64f_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp32s** *pDst, int nDstStep, **Npp64f** *pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32s** nVal, **Npp64f** nValSqr)
One-channel 8-bit unsigned image SqrIntegral.
- **NppStatus nppiSqrIntegral_8u32f64f_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **Npp64f** *pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32f** nVal, **Npp64f** nValSqr)
One-channel 8-bit unsigned image SqrIntegral.

7.117.1 Detailed Description

Primitives for computing both the integral and the squared integral images of a given image.

7.117.2 Function Documentation

7.117.2.1 NppStatus nppiSqrIntegral_8u32f64f_C1R (const **Npp8u** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **Npp64f** *pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32f** nVal, **Npp64f** nValSqr)

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image is 32-bit floating point. Destination square integral image is 64-bit double floating point.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pSqr Destination-Image Pointer.
nSqrStep Destination-Image Line Step.
oSrcROI Region-of-Interest (ROI).
nVal The value to add to pDst image pixels
nValSqr The value to add to pSqr image pixels

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.117.2.2 `NppStatus nppiSqrIntegral_8u32s64f_C1R (const Npp8u *pSrc, int nSrcStep, Npp32s *pDst, int nDstStep, Npp64f *pSqr, int nSqrStep, NppiSize oSrcROI, Npp32s nVal, Npp64f nValSqr)`

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image is 32-bit signed int. Destination square integral image is 64-bit double floating point.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
pSqr Destination-Image Pointer.
nSqrStep Destination-Image Line Step.
oSrcROI Region-of-Interest (ROI).
nVal The value to add to pDst image pixels
nValSqr The value to add to pSqr image pixels

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.117.2.3 `NppStatus nppiSqrIntegral_8u32s_C1R (const Npp8u *pSrc, int nSrcStep, Npp32s *pDst, int nDstStep, Npp32s *pSqr, int nSqrStep, NppiSize oSrcROI, Npp32s nVal, Npp32s nValSqr)`

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image and square integral image are 32-bit signed int.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pSqr Destination-Image Pointer.

nSqrStep Destination-Image Line Step.

oSrcROI Region-of-Interest (ROI).

nVal The value to add to pDst image pixels

nValSqr The value to add to pSqr image pixels

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.118 RectStdDev

Primitives for computing the standard deviation of the integral images.

RectStdDev

- **NppStatus** **npapiRectStdDev_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, const **Npp64f** *pSqr, int nSqrStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiRect** oRect)
One-channel 32-bit floating point image RectStdDev.
- **NppStatus** **npapiRectStdDev_32s_C1RSfs** (const **Npp32s** *pSrc, int nSrcStep, const **Npp32s** *pSqr, int nSqrStep, **Npp32s** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiRect** oRect, int nScaleFactor)
One-channel 32-bit signed image RectStdDev, scaled by $2^l - nScaleFactor$.
- **NppStatus** **npapiRectStdDev_32s32f_C1R** (const **Npp32s** *pSrc, int nSrcStep, const **Npp64f** *pSqr, int nSqrStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiRect** oRect)
One-channel 32-bit signed image RectStdDev.

7.118.1 Detailed Description

Primitives for computing the standard deviation of the integral images.

The function computes the standard deviation of the pixel in the rectangular window with the integral image *pSrc* and the squared integral image *pSqr*, which can be obtained by calling [Integral](#) and [SqrIntegral](#).

The standard deviation of the pixel (*j*, *i*) can be computed using the formula:

$$pDst(j, i) = \sqrt{\max(0, \frac{\sum(SqrIntegral) \cdot N - (\sum(Integral))^2}{N^2})}$$

where $\sum(SqrIntegral) = pSqr[j + oRect.y + oRect.height, i + oRect.x + oRect.width] - pSqr[j + oRect.y, i + oRect.x + oRect.width] - pSqr[j + oRect.y + oRect.height, i + oRect.x] + pSqr[j + oRect.y, i + oRect.x]$, $\sum(Integral) = pSrc[j + oRect.y + oRect.height, i + oRect.x + oRect.width] - pSrc[j + oRect.y, i + oRect.x + oRect.width] - pSrc[j + oRect.y + oRect.height, i + oRect.x] + pSrc[j + oRect.y, i + oRect.x]$, $N = oRect.width \cdot oRect.height$.

The size of the *pSrc* and *pSqr* should be (*oSizeROI.width* + *oRect.x* + *oRect.width*, *oSizeROI.height* + *oRect.y* + *oRect.height*).

7.118.2 Function Documentation

7.118.2.1 NppStatus npapiRectStdDev_32f_C1R (const **Npp32f** *pSrc, int nSrcStep, const **Npp64f** *pSqr, int nSqrStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppiRect** oRect)

One-channel 32-bit floating point image RectStdDev.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- pSqr* [Destination-Image Pointer](#).

nSqrStep Destination-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oRect rectangular window

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.118.2.2 `NppStatus nppiRectStdDev_32s32f_C1R (const Npp32s * pSrc, int nSrcStep, const Npp64f * pSqr, int nSqrStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiRect oRect)`

One-channel 32-bit signed image RectStdDev.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSqr Destination-Image Pointer.

nSqrStep Destination-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oRect rectangular window

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.118.2.3 `NppStatus nppiRectStdDev_32s_C1RSfs (const Npp32s * pSrc, int nSrcStep, const Npp32s * pSqr, int nSqrStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, NppiRect oRect, int nScaleFactor)`

One-channel 32-bit signed image RectStdDev, scaled by $2^{(-nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pSqr Destination-Image Pointer.

nSqrStep Destination-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

oRect rectangular window

nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.119 HistogramEven

Primitives for computing the histogram of an image with evenly distributed bins.

HistogramEven

The *nLowerLevel* (inclusive) and *nUpperLevel* (exclusive) define the boundaries of the range, which are evenly segmented into *nLevel* - 1 bins.

The computed histogram is stored in *pHist*. The levels are calculated by another primitive [nppiEvenLevelsHost_32s](#) and are stored in a host pointer *hpLevels*. The number of levels is also *nLevel* - 1. The histogram *pHist[k]* is defined as the total number of pixels that fall into the range: *hpLevels[k]* <= *pSrc(j,i)* < *hpLevels[k + 1]*. The functions require additional scratch buffer for computations.

- [NppStatus nppiEvenLevelsHost_32s](#) ([Npp32s](#) *hpLevels, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel)
Compute levels with even distribution.
- [NppStatus nppiHistogramEven_8u_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) *pHist, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel, [Npp8u](#) *pBuffer)
One-channel 8-bit unsigned HistogramEven.
- [NppStatus nppiHistogramEven_8u_C3R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) *pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) *pBuffer)
Three-channel 8-bit unsigned HistogramEven.
- [NppStatus nppiHistogramEven_8u_C4R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) *pHist[4], int nLevels[4], [Npp32s](#) nLowerLevel[4], [Npp32s](#) nUpperLevel[4], [Npp8u](#) *pBuffer)
Four-channel 8-bit unsigned HistogramEven.
- [NppStatus nppiHistogramEven_8u_AC4R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) *pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) *pBuffer)
Four-channel 8-bit unsigned HistogramEven ignoring alpha channel.
- [NppStatus nppiHistogramEven_16u_C1R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) *pHist, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel, [Npp8u](#) *pBuffer)
One-channel 16-bit unsigned HistogramEven.
- [NppStatus nppiHistogramEven_16u_C3R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) *pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) *pBuffer)
Three-channel 16-bit unsigned HistogramEven.
- [NppStatus nppiHistogramEven_16u_C4R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) *pHist[4], int nLevels[4], [Npp32s](#) nLowerLevel[4], [Npp32s](#) nUpperLevel[4], [Npp8u](#) *pBuffer)
Four-channel 16-bit unsigned HistogramEven.

- `NppStatus nppiHistogramEven_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` *pHist[3], int nLevels[3], `Npp32s` nLowerLevel[3], `Npp32s` nUpperLevel[3], `Npp8u` *pBuffer)

Four-channel 16-bit unsigned HistogramEven ignoring alpha channel.

- `NppStatus nppiHistogramEven_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` *pHist, int nLevels, `Npp32s` nLowerLevel, `Npp32s` nUpperLevel, `Npp8u` *pBuffer)

One-channel 16-bit signed HistogramEven.

- `NppStatus nppiHistogramEven_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` *pHist[3], int nLevels[3], `Npp32s` nLowerLevel[3], `Npp32s` nUpperLevel[3], `Npp8u` *pBuffer)

Three-channel 16-bit signed HistogramEven.

- `NppStatus nppiHistogramEven_16s_C4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` *pHist[4], int nLevels[4], `Npp32s` nLowerLevel[4], `Npp32s` nUpperLevel[4], `Npp8u` *pBuffer)

Four-channel 16-bit signed HistogramEven.

- `NppStatus nppiHistogramEven_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` *pHist[3], int nLevels[3], `Npp32s` nLowerLevel[3], `Npp32s` nUpperLevel[3], `Npp8u` *pBuffer)

Four-channel 16-bit signed HistogramEven ignoring alpha channel.

HistogramEvenGetBufferSize

Companion primitives for computing the device buffer size (in bytes) required by the HistogramEven primitives.

- `NppStatus nppiHistogramEvenGetBufferSize_8u_C1R` (`NppiSize` oSizeROI, int nLevels, int *hpBufferSize)

Buffer size for `nppiHistogramEven_8u_C1R`.

- `NppStatus nppiHistogramEvenGetBufferSize_8u_C3R` (`NppiSize` oSizeROI, int nLevels[3], int *hpBufferSize)

Buffer size for `nppiHistogramEven_8u_C3R`.

- `NppStatus nppiHistogramEvenGetBufferSize_8u_C4R` (`NppiSize` oSizeROI, int nLevels[4], int *hpBufferSize)

Buffer size for `nppiHistogramEven_8u_C4R`.

- `NppStatus nppiHistogramEvenGetBufferSize_8u_AC4R` (`NppiSize` oSizeROI, int nLevels[3], int *hpBufferSize)

Buffer size for `nppiHistogramEven_8u_AC4R`.

- `NppStatus nppiHistogramEvenGetBufferSize_16u_C1R` (`NppiSize` oSizeROI, int nLevels, int *hpBufferSize)

Buffer size for `nppiHistogramEven_16u_C1R`.

- **NppStatus** `nppiHistogramEvenGetBufferSize_16u_C3R` (**NppiSize** `oSizeROI`, **int** `nLevels[3]`, **int** `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16u_C3R`.
- **NppStatus** `nppiHistogramEvenGetBufferSize_16u_C4R` (**NppiSize** `oSizeROI`, **int** `nLevels[4]`, **int** `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16u_C4R`.
- **NppStatus** `nppiHistogramEvenGetBufferSize_16u_AC4R` (**NppiSize** `oSizeROI`, **int** `nLevels[3]`, **int** `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16u_AC4R`.
- **NppStatus** `nppiHistogramEvenGetBufferSize_16s_C1R` (**NppiSize** `oSizeROI`, **int** `nLevels`, **int** `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16s_C1R`.
- **NppStatus** `nppiHistogramEvenGetBufferSize_16s_C3R` (**NppiSize** `oSizeROI`, **int** `nLevels[3]`, **int** `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16s_C3R`.
- **NppStatus** `nppiHistogramEvenGetBufferSize_16s_C4R` (**NppiSize** `oSizeROI`, **int** `nLevels[4]`, **int** `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16s_C4R`.
- **NppStatus** `nppiHistogramEvenGetBufferSize_16s_AC4R` (**NppiSize** `oSizeROI`, **int** `nLevels[3]`, **int** `*hpBufferSize`)
Buffer size for `nppiHistogramEven_16s_AC4R`.

7.119.1 Detailed Description

Primitives for computing the histogram of an image with evenly distributed bins.

7.119.2 Function Documentation

7.119.2.1 **NppStatus** `nppiEvenLevelsHost_32s` (**Npp32s** `*hpLevels`, **int** `nLevels`, **Npp32s** `nLowerLevel`, **Npp32s** `nUpperLevel`)

Compute levels with even distribution.

Parameters:

- hpLevels*** A host pointer to array which receives the levels being computed. The array needs to be of size `nLevels`.
- nLevels*** The number of levels being computed. `nLevels` must be at least 2.
- nLowerLevel*** Lower boundary value of the lowest level.
- nUpperLevel*** Upper boundary value of the greatest level.

Returns:

image_data_error_codes, or NPP_HISTO_NUMBER_OF_LEVELS_ERROR if an invalid nLevels is specified.

7.119.2.2 NppStatus nppiHistogramEven_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)

Four-channel 16-bit signed HistogramEven ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.

nLevels Array containing number of levels per color channel.

nLowerLevel Array containing lower-level of lowest bin per color channel.

nUpperLevel Array containing upper-level of highest bin per color channel.

pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_16s_AC4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.119.2.3 NppStatus nppiHistogramEven_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)

One-channel 16-bit signed HistogramEven.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Pointer to array that receives the computed histogram. The array must be of size nLevels-1.

nLevels Number of levels.

nLowerLevel Lower boundary of lowest level bin.

nUpperLevel Upper boundary of highest level bin.

pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_16s_C1R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.119.2.4 NppStatus nppiHistogramEven_16s_C3R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*[3], int *nLevels*[3], Npp32s *nLowerLevel*[3], Npp32s *nUpperLevel*[3], Npp8u * *pBuffer*)

Three-channel 16-bit signed HistogramEven.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist*[*i*] be of size *nLevels*[*i*]-1.

nLevels Array containing number of levels per color channel.

nLowerLevel Array containing lower-level of lowest bin per color channel.

nUpperLevel Array containing upper-level of highest bin per color channel.

pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_16s_C3R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.119.2.5 NppStatus nppiHistogramEven_16s_C4R (const Npp16s * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*[4], int *nLevels*[4], Npp32s *nLowerLevel*[4], Npp32s *nUpperLevel*[4], Npp8u * *pBuffer*)

Four-channel 16-bit signed HistogramEven.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist*[*i*] be of size *nLevels*[*i*]-1.

nLevels Array containing number of levels per color channel.

nLowerLevel Array containing lower-level of lowest bin per color channel.

nUpperLevel Array containing upper-level of highest bin per color channel.

pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_16s_C4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.119.2.6 NppStatus nppiHistogramEven_16u_AC4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*[3], int *nLevels*[3], Npp32s *nLowerLevel*[3], Npp32s *nUpperLevel*[3], Npp8u * *pBuffer*)

Four-channel 16-bit unsigned HistogramEven ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pHist Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist*[*i*] be of size *nLevels*[*i*]-1.

nLevels Array containing number of levels per color channel.

nLowerLevel Array containing lower-level of lowest bin per color channel.

nUpperLevel Array containing upper-level of highest bin per color channel.

pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_16u_AC4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.119.2.7 NppStatus nppiHistogramEven_16u_C1R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*, int *nLevels*, Npp32s *nLowerLevel*, Npp32s *nUpperLevel*, Npp8u * *pBuffer*)

One-channel 16-bit unsigned HistogramEven.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pHist Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

nLevels Number of levels.

nLowerLevel Lower boundary of lowest level bin.

nUpperLevel Upper boundary of highest level bin.

pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_16u_C1R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.119.2.8 NppStatus nppiHistogramEven_16u_C3R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*[3], int *nLevels*[3], Npp32s *nLowerLevel*[3], Npp32s *nUpperLevel*[3], Npp8u * *pBuffer*)

Three-channel 16-bit unsigned HistogramEven.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist*[*i*] be of size *nLevels*[*i*]-1.

nLevels Array containing number of levels per color channel.

nLowerLevel Array containing lower-level of lowest bin per color channel.

nUpperLevel Array containing upper-level of highest bin per color channel.

pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_16u_C3R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.119.2.9 NppStatus nppiHistogramEven_16u_C4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*[4], int *nLevels*[4], Npp32s *nLowerLevel*[4], Npp32s *nUpperLevel*[4], Npp8u * *pBuffer*)

Four-channel 16-bit unsigned HistogramEven.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist*[*i*] be of size *nLevels*[*i*]-1.

nLevels Array containing number of levels per color channel.

nLowerLevel Array containing lower-level of lowest bin per color channel.

nUpperLevel Array containing upper-level of highest bin per color channel.

pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_16u_C4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.119.2.10 `NppStatus nppiHistogramEven_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

Four-channel 8-bit unsigned HistogramEven ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist[i]* be of size *nLevels[i]*-1.

nLevels Array containing number of levels per color channel.

nLowerLevel Array containing lower-level of lowest bin per color channel.

nUpperLevel Array containing upper-level of highest bin per color channel.

pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_8u_AC4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.119.2.11 `NppStatus nppiHistogramEven_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)`

One-channel 8-bit unsigned HistogramEven.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

nLevels Number of levels.

nLowerLevel Lower boundary of lowest level bin.

nUpperLevel Upper boundary of highest level bin.

pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_8u_C1R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.119.2.12 `NppStatus nppiHistogramEven_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

Three-channel 8-bit unsigned HistogramEven.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pHist Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.
nLevels Array containing number of levels per color channel.
nLowerLevel Array containing lower-level of lowest bin per color channel.
nUpperLevel Array containing upper-level of highest bin per color channel.
pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_8u_C3R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.119.2.13 `NppStatus nppiHistogramEven_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], int nLevels[4], Npp32s nLowerLevel[4], Npp32s nUpperLevel[4], Npp8u * pBuffer)`

Four-channel 8-bit unsigned HistogramEven.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pHist Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.
nLevels Array containing number of levels per color channel.
nLowerLevel Array containing lower-level of lowest bin per color channel.
nUpperLevel Array containing upper-level of highest bin per color channel.
pBuffer Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize_8u_C4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.119.2.14 `NppStatus nppiHistogramEvenGetBufferSize_16s_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)`

Buffer size for [nppiHistogramEven_16s_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).
nLevels Array containing number of levels per color channel.
hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

7.119.2.15 NppStatus nppiHistogramEvenGetBufferSize_16s_C1R (NppiSize *oSizeROI*, int *nLevels*, int * *hpBufferSize*)

Buffer size for [nppiHistogramEven_16s_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.119.2.16 NppStatus nppiHistogramEvenGetBufferSize_16s_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int * *hpBufferSize*)

Buffer size for [nppiHistogramEven_16s_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.119.2.17 NppStatus nppiHistogramEvenGetBufferSize_16s_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int * *hpBufferSize*)

Buffer size for [nppiHistogramEven_16s_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.119.2.18 NppStatus nppiHistogramEvenGetBufferSize_16u_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Buffer size for [nppiHistogramEven_16u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.119.2.19 NppStatus nppiHistogramEvenGetBufferSize_16u_C1R (NppiSize oSizeROI, int nLevels, int * hpBufferSize)

Buffer size for [nppiHistogramEven_16u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.119.2.20 NppStatus nppiHistogramEvenGetBufferSize_16u_C3R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Buffer size for [nppiHistogramEven_16u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.119.2.21 NppStatus nppiHistogramEvenGetBufferSize_16u_C4R (NppiSize oSizeROI, int nLevels[4], int * hpBufferSize)

Buffer size for [nppiHistogramEven_16u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.119.2.22 NppStatus nppiHistogramEvenGetBufferSize_8u_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Buffer size for [nppiHistogramEven_8u_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.119.2.23 NppStatus nppiHistogramEvenGetBufferSize_8u_C1R (NppiSize oSizeROI, int nLevels, int * hpBufferSize)

Buffer size for [nppiHistogramEven_8u_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.119.2.24 NppStatus nppiHistogramEvenGetBufferSize_8u_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int * *hpBufferSize*)

Buffer size for [nppiHistogramEven_8u_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.119.2.25 NppStatus nppiHistogramEvenGetBufferSize_8u_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int * *hpBufferSize*)

Buffer size for [nppiHistogramEven_8u_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.120 HistogramRange

Primitives for computing the histogram of an image within specified ranges.

HistogramEven

The histogram is computed according to the ranges provided in *pLevels*.

The histogram $pHist[k]$ is defined as the total number of pixels that fall into the range: $pLevels[k] \leq pSrc(j, i) < pLevels[k + 1]$. The number of the histogram bins is $nLevel - 1$. The functions require additional scratch buffer for computations.

- **NppStatus nppiHistogramRange_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist, const **Npp32s** *pLevels, int nLevels, **Npp8u** *pBuffer)
One-channel 8-bit unsigned HistogramRange.
- **NppStatus nppiHistogramRange_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[3], const **Npp32s** *pLevels[3], int nLevels[3], **Npp8u** *pBuffer)
Three-channel 8-bit unsigned HistogramRange.
- **NppStatus nppiHistogramRange_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[4], const **Npp32s** *pLevels[4], int nLevels[4], **Npp8u** *pBuffer)
Four-channel 8-bit unsigned HistogramRange.
- **NppStatus nppiHistogramRange_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[3], const **Npp32s** *pLevels[3], int nLevels[3], **Npp8u** *pBuffer)
Four-channel 8-bit unsigned HistogramRange ignoring alpha channel.
- **NppStatus nppiHistogramRange_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist, const **Npp32s** *pLevels, int nLevels, **Npp8u** *pBuffer)
One-channel 16-bit unsigned HistogramRange.
- **NppStatus nppiHistogramRange_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[3], const **Npp32s** *pLevels[3], int nLevels[3], **Npp8u** *pBuffer)
Three-channel 16-bit unsigned HistogramRange.
- **NppStatus nppiHistogramRange_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[4], const **Npp32s** *pLevels[4], int nLevels[4], **Npp8u** *pBuffer)
Four-channel 16-bit unsigned HistogramRange.
- **NppStatus nppiHistogramRange_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[3], const **Npp32s** *pLevels[3], int nLevels[3], **Npp8u** *pBuffer)
Four-channel 16-bit unsigned HistogramRange ignoring alpha channel.
- **NppStatus nppiHistogramRange_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist, const **Npp32s** *pLevels, int nLevels, **Npp8u** *pBuffer)
One-channel 16-bit signed HistogramRange.
- **NppStatus nppiHistogramRange_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[3], const **Npp32s** *pLevels[3], int nLevels[3], **Npp8u** *pBuffer)
Three-channel 16-bit signed HistogramRange.

- **NppStatus** **nppiHistogramRange_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[4], const **Npp32s** *pLevels[4], int nLevels[4], **Npp8u** *pBuffer)
Four-channel 16-bit signed HistogramRange.
- **NppStatus** **nppiHistogramRange_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[3], const **Npp32s** *pLevels[3], int nLevels[3], **Npp8u** *pBuffer)
Four-channel 16-bit signed HistogramRange.
- **NppStatus** **nppiHistogramRange_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist, const **Npp32f** *pLevels, int nLevels, **Npp8u** *pBuffer)
One-channel 32-bit floating point HistogramRange.
- **NppStatus** **nppiHistogramRange_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[3], const **Npp32f** *pLevels[3], int nLevels[3], **Npp8u** *pBuffer)
Three-channel 32-bit floating point HistogramRange.
- **NppStatus** **nppiHistogramRange_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[4], const **Npp32f** *pLevels[4], int nLevels[4], **Npp8u** *pBuffer)
Four-channel 32-bit floating point HistogramRange.
- **NppStatus** **nppiHistogramRange_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** *pHist[3], const **Npp32f** *pLevels[3], int nLevels[3], **Npp8u** *pBuffer)
Four-channel 32-bit floating point HistogramRange ignoring alpha channel.

HistogramRangeGetBufferSize

Companion primitives for computing the device buffer size (in bytes) required by the HistogramRange primitives.

- **NppStatus** **nppiHistogramRangeGetBufferSize_8u_C1R** (**NppiSize** oSizeROI, int nLevels, int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_8u_C1R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_8u_C3R** (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_8u_C3R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_8u_C4R** (**NppiSize** oSizeROI, int nLevels[4], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_8u_C4R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_8u_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_8u_AC4R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_16u_C1R** (**NppiSize** oSizeROI, int nLevels, int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16u_C1R.

- **NppStatus** **nppiHistogramRangeGetBufferSize_16u_C3R** (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16u_C3R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_16u_C4R** (**NppiSize** oSizeROI, int nLevels[4], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16u_C4R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_16u_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16u_AC4R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_16s_C1R** (**NppiSize** oSizeROI, int nLevels, int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16s_C1R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_16s_C3R** (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16s_C3R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_16s_C4R** (**NppiSize** oSizeROI, int nLevels[4], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16s_C4R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_16s_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_16s_AC4R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_32f_C1R** (**NppiSize** oSizeROI, int nLevels, int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_32f_C1R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_32f_C3R** (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_32f_C3R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_32f_C4R** (**NppiSize** oSizeROI, int nLevels[4], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_32f_C4R.
- **NppStatus** **nppiHistogramRangeGetBufferSize_32f_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int *hpBufferSize)
Scratch-buffer size for nppiHistogramRange_32f_AC4R.

7.120.1 Detailed Description

Primitives for computing the histogram of an image within specified ranges.

7.120.2 Function Documentation

7.120.2.1 `NppStatus nppiHistogramRange_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Four-channel 16-bit signed HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].

pBuffer Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize_16s_AC4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.2 `NppStatus nppiHistogramRange_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32s * pLevels, int nLevels, Npp8u * pBuffer)`

One-channel 16-bit signed HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

pLevels Pointer to array containing the level sizes of the bins. The array must be of size *nLevels*.

nLevels Number of levels in histogram.

pBuffer Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize_16s_C1R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.3 `NppStatus nppiHistogramRange_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Three-channel 16-bit signed HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

pBuffer Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize_16s_C3R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.4 `NppStatus nppiHistogramRange_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32s * pLevels[4], int nLevels[4], Npp8u * pBuffer)`

Four-channel 16-bit signed HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

pBuffer Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize_16s_C4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.5 `NppStatus nppiHistogramRange_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Four-channel 16-bit unsigned HistogramRange ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

pBuffer Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize_16u_AC4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.6 `NppStatus nppiHistogramRange_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32s * pLevels, int nLevels, Npp8u * pBuffer)`

One-channel 16-bit unsigned HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Pointer to array that receives the computed histogram. The array must be of size nLevels-1.

pLevels Pointer to array containing the level sizes of the bins. The array must be of size nLevels.

nLevels Number of levels in histogram.

pBuffer Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize_16u_C1R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.7 `NppStatus nppiHistogramRange_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Three-channel 16-bit unsigned HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

pBuffer Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize_16u_C3R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.8 `NppStatus nppiHistogramRange_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32s * pLevels[4], int nLevels[4], Npp8u * pBuffer)`

Four-channel 16-bit unsigned HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

pBuffer Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize_16u_C4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.9 `NppStatus nppiHistogramRange_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32f * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Four-channel 32-bit floating point HistogramRange ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

pBuffer Pointer to appropriately sized (nppiHistogramRangeGetBufferSize_32f_AC4R) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.10 `NppStatus nppiHistogramRange_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32f * pLevels, int nLevels, Npp8u * pBuffer)`

One-channel 32-bit floating point HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Pointer to array that receives the computed histogram. The array must be of size nLevels-1.

pLevels Pointer to array containing the level sizes of the bins. The array must be of size nLevels.

nLevels Number of levels in histogram.

pBuffer Pointer to appropriately sized (nppiHistogramRangeGetBufferSize_32f_C1R) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.11 `NppStatus nppiHistogramRange_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32f * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Three-channel 32-bit floating point HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

pBuffer Pointer to appropriately sized (nppiHistogramRangeGetBufferSize_32f_C3R) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.12 `NppStatus nppiHistogramRange_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32f * pLevels[4], int nLevels[4], Npp8u * pBuffer)`

Four-channel 32-bit floating point HistogramRange.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

pBuffer Pointer to appropriately sized (nppiHistogramRangeGetBufferSize_32f_C4R) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.13 `NppStatus nppiHistogramRange_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Four-channel 8-bit unsigned HistogramRange ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

pBuffer Pointer to appropriately sized (nppiHistogramRangeGetBufferSize_8u_AC4R) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.14 **NppStatus nppiHistogramRange_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*, const Npp32s * *pLevels*, int *nLevels*, Npp8u * *pBuffer*)

One-channel 8-bit unsigned HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

pLevels Pointer to array containing the level sizes of the bins. The array must be of size *nLevels*.

nLevels Number of levels in histogram.

pBuffer Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize_8u_C1R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.15 **NppStatus nppiHistogramRange_8u_C3R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*[3], const Npp32s * *pLevels*[3], int *nLevels*[3], Npp8u * *pBuffer*)

Three-channel 8-bit unsigned HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].

pBuffer Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize_8u_C3R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.16 **NppStatus nppiHistogramRange_8u_C4R** (const Npp8u * *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s * *pHist*[4], const Npp32s * *pLevels*[4], int *nLevels*[4], Npp8u * *pBuffer*)

Four-channel 8-bit unsigned HistogramRange.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pHist Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

nLevels Array containing number of levels per color channel.

pLevels Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

pBuffer Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize_8u_C4R](#)) scratch buffer.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.120.2.17 NppStatus nppiHistogramRangeGetBufferSize_16s_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_16s_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.18 NppStatus nppiHistogramRangeGetBufferSize_16s_C1R (NppiSize oSizeROI, int nLevels, int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_16s_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.19 NppStatus nppiHistogramRangeGetBufferSize_16s_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_16s_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.20 NppStatus nppiHistogramRangeGetBufferSize_16s_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_16s_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.21 NppStatus nppiHistogramRangeGetBufferSize_16u_AC4R (NppiSize *oSizeROI*, int *nLevels*[3], int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_16u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.22 NppStatus nppiHistogramRangeGetBufferSize_16u_C1R (NppiSize *oSizeROI*, int *nLevels*, int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_16u_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.23 NppStatus nppiHistogramRangeGetBufferSize_16u_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_16u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.24 NppStatus nppiHistogramRangeGetBufferSize_16u_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_16u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.25 NppStatus nppiHistogramRangeGetBufferSize_32f_AC4R (NppiSize *oSizeROI*, int *nLevels*[3], int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_32f_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.26 NppStatus nppiHistogramRangeGetBufferSize_32f_C1R (NppiSize *oSizeROI*, int *nLevels*, int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_32f_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.27 NppStatus nppiHistogramRangeGetBufferSize_32f_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_32f_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.28 NppStatus nppiHistogramRangeGetBufferSize_32f_C4R (NppiSize oSizeROI, int nLevels[4], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_32f_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.29 NppStatus nppiHistogramRangeGetBufferSize_8u_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_8u_AC4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.30 NppStatus nppiHistogramRangeGetBufferSize_8u_C1R (NppiSize oSizeROI, int nLevels, int * hpBufferSize)

Scratch-buffer size for nppiHistogramRange_8u_C1R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Number of levels in the histogram.

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.31 NppStatus nppiHistogramRangeGetBufferSize_8u_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_8u_C3R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.120.2.32 NppStatus nppiHistogramRangeGetBufferSize_8u_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int * *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange_8u_C4R.

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

nLevels Array containing number of levels per color channel.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

7.121 Image Proximity

Primitives for computing the proximity measure between a source image and a template image.

Modules

- [SqrDistanceFull_Norm](#)

Primitives for computing the normalized Euclidean distance between two images with full mode.

- [SqrDistanceSame_Norm](#)

Primitives for computing the normalized Euclidean distance between two images with same mode.

- [SqrDistanceValid_Norm](#)

Primitives for computing the normalized Euclidean distance between two images with valid mode.

- [CrossCorrFull_Norm](#)

Primitives for computing the normalized cross correlation between two images with full mode.

- [CrossCorrSame_Norm](#)

Primitives for computing the normalized cross correlation between two images with same mode.

- [CrossCorrValid_Norm](#)

Primitives for computing the normalized cross correlation between two images with valid mode.

- [CrossCorrValid](#)

Primitives for computing the cross correlation between two images with valid mode.

- [CrossCorrFull_NormLevel](#)

Primitives for computing the normalized cross correlation coefficient between two images with full mode.

- [CrossCorrSame_NormLevel](#)

Primitives for computing the normalized cross correlation coefficient between two images with same mode.

- [CrossCorrValid_NormLevel](#)

Primitives for computing the normalized cross correlation coefficient between two images with valid mode.

7.121.1 Detailed Description

Primitives for computing the proximity measure between a source image and a template image.

7.121.2 General Introduction

There are basically two approaches to compute the proximity measure for template matching, Euclidean distance and the cross correlation.

1. Euclidean distance computes the sum of the squared distance (SSD) between the corresponding pixels of the source image and the template image. The smaller the distance is, the more similar the source image and the template image is around the pixel. The anchor of the template image is used during the computations, which always lies in the geometric center of the image. Given a source image $pSrc$ ($W_s \times H_s$) and a template image $pTpl$ ($W_t \times H_t$), the Euclidean distance $D_{st}(c, r)$ between two images at pixel in row r and column c is computed as (s stands for source image and t for template image for short):

$$D_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2})]^2$$

2. Cross correlation computes the sum of the product between the corresponding pixels of the source image and the template image. The cross correlation $R_{st}(c, r)$ is calculated as:

$$R_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) \cdot pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2})]$$

The larger the cross correlation value is, the more similar the source image and the template image is around the pixel.

3. The cross correlation $R_{st}(c, r)$ is affected by the brightness of the images which may vary due to the lighting and exposure conditions. Therefore, NPP computes the cross correlation coefficient to circumvent this dependence. This is typically done at every step by subtracting the mean from every pixel value, i.e.,

$$\tilde{R}_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - Mean_t] \cdot [pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2}) - Mean_s]$$

NPP computes the normalized values of Euclidean distance, cross correlation and the cross correlation coefficient.

1. The normalized Euclidean distance $\sigma_{st}(c, r)$ is defined as:

$$\sigma_{st}(c, r) = \frac{D_{st}(c, r)}{\sqrt{R_{ss}(c, r) \cdot R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

2. The normalized cross correlation $\rho_{st}(c, r)$ is defined as:

$$\rho_{st}(c, r) = \frac{R_{st}(c, r)}{\sqrt{R_{ss}(c, r) \cdot R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

The $R_{ss}(c, r)$ and $R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})$ denote the auto correlation of the source image and the template image individually. They are defined as:

$$R_{ss}(c, r) = \sum_{j=c-\frac{H_t}{2}}^{c+\frac{H_t}{2}} \sum_{i=r-\frac{W_t}{2}}^{r+\frac{W_t}{2}} pSrc(j, i)$$

$$R_{tt}(\frac{H_t}{2}, \frac{W_t}{2}) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} pTpl(j, i)$$

3. Similarly, the normalized cross correlation coefficient $\gamma_{st}(c, r)$ is calculated as:

$$\gamma_{st}(c, r) = \frac{\tilde{R}_{st}(c, r)}{\sqrt{\tilde{R}_{ss}(c, r) \cdot \tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

The $\tilde{R}_{ss}(c, r)$ and $\tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2})$ are defined as:

$$\tilde{R}_{ss}(c, r) = \sum_{j=c-\frac{H_t}{2}}^{c+\frac{H_t}{2}} \sum_{i=r-\frac{W_t}{2}}^{r+\frac{W_t}{2}} [pSrc(j, i) - Mean_s]$$

$$\tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2}) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - Mean_t]$$

7.121.3 Categorizations

The Euclidean distance and the cross correlation are categorized into three types, full, same, and valid.

1. Full mode indicates that the anchor of the template image starts from the outside of the source image, assuming the out-of-boundary pixels are zero-padded. The size of the destination image is $(W_s + W_t - 1) \times (H_s + H_t - 1)$.
2. Same mode means that the anchor of the template image starts from the top left pixel of the source image. All the out-of-boundary pixels are also zero-padded. The size of the destination image is the same as the source one, i.e., $W_s \times H_s$.
3. Valid mode indicates that there are no out-of-boundary readings from the source image. The anchor of the template image starts from the inside of the source image. The size of the destination image is $(W_s - W_t + 1) \times (H_s - H_t + 1)$.

7.122 SqrDistanceFull_Norm

Primitives for computing the normalized Euclidean distance between two images with full mode.

SqrDistanceFull_Norm

The functions compute the $\sigma_{st}(c, r)$ in [General Introduction](#) with full mode (see [Categorizations](#)).

- [NppStatus nppiSqrDistanceFull_Norm_8u_C1RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
One-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiSqrDistanceFull_Norm_8u_C3RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Three-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiSqrDistanceFull_Norm_8u_C4RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiSqrDistanceFull_Norm_8u_AC4RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiSqrDistanceFull_Norm_32f_C1R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
One-channel 32-bit floating point image SqrDistanceFull_Norm.
- [NppStatus nppiSqrDistanceFull_Norm_32f_C3R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Three-channel 32-bit floating point image SqrDistanceFull_Norm.
- [NppStatus nppiSqrDistanceFull_Norm_32f_C4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 32-bit floating point image SqrDistanceFull_Norm.
- [NppStatus nppiSqrDistanceFull_Norm_32f_AC4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 32-bit floating point image SqrDistanceFull_Norm ignoring alpha channel.
- [NppStatus nppiSqrDistanceFull_Norm_8u32f_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
One-channel 8-bit unsigned image SqrDistanceFull_Norm.
- [NppStatus nppiSqrDistanceFull_Norm_8u32f_C3R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Three-channel 8-bit unsigned image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel.

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit signed image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit signed image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image SqrDistanceFull_Norm ignoring alpha channel.

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 16-bit unsigned image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 16-bit unsigned image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceFull_Norm.

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel.

7.122.1 Detailed Description

Primitives for computing the normalized Euclidean distance between two images with full mode.

7.122.2 Function Documentation

7.122.2.1 `NppStatus nppiSqrDistanceFull_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.122.2.2 `NppStatus nppiSqrDistanceFull_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.122.2.3 `NppStatus nppiSqrDistanceFull_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.122.2.4 `NppStatus nppiSqrDistanceFull_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.122.2.5 `NppStatus nppiSqrDistanceFull_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.122.2.6 `NppStatus nppiSqrDistanceFull_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.122.2.7 `NppStatus nppiSqrDistanceFull_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.122.2.8 `NppStatus nppiSqrDistanceFull_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.122.2.9 `NppStatus nppiSqrDistanceFull_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.122.2.10 `NppStatus nppiSqrDistanceFull_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.122.2.11 `NppStatus nppiSqrDistanceFull_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.122.2.12 `NppStatus nppiSqrDistanceFull_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.122.2.13 `NppStatus nppiSqrDistanceFull_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.122.2.14 `NppStatus nppiSqrDistanceFull_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.122.2.15 `NppStatus nppiSqrDistanceFull_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.122.2.16 `NppStatus nppiSqrDistanceFull_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceFull_Norm.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.122.2.17 `NppStatus nppiSqrDistanceFull_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.122.2.18 `NppStatus nppiSqrDistanceFull_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.122.2.19 `NppStatus nppiSqrDistanceFull_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.122.2.20 `NppStatus nppiSqrDistanceFull_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.123 SqrDistanceSame_Norm

Primitives for computing the normalized Euclidean distance between two images with same mode.

SqrDistanceSame_Norm

The functions compute the $\sigma_{st}(c, r)$ in [General Introduction](#) with same mode (see [Categorizations](#)).

- `NppStatus nppiSqrDistanceSame_Norm_8u_C1RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

One-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\ell} - nScaleFactor$.

- `NppStatus nppiSqrDistanceSame_Norm_8u_C3RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

Three-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\ell} - nScaleFactor$.

- `NppStatus nppiSqrDistanceSame_Norm_8u_C4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

Four-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\ell} - nScaleFactor$.

- `NppStatus nppiSqrDistanceSame_Norm_8u_AC4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

Four-channel 8-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel, scaled by $2^{\ell} - nScaleFactor$.

- `NppStatus nppiSqrDistanceSame_Norm_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 32-bit floating point image SqrDistanceSame_Norm.

- `NppStatus nppiSqrDistanceSame_Norm_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 32-bit floating point image SqrDistanceSame_Norm.

- `NppStatus nppiSqrDistanceSame_Norm_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 32-bit floating point image SqrDistanceSame_Norm.

- `NppStatus nppiSqrDistanceSame_Norm_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 32-bit floating point image SqrDistanceSame_Norm ignoring alpha channel.

- `NppStatus nppiSqrDistanceSame_Norm_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit unsigned image SqrDistanceSame_Norm.

- `NppStatus nppiSqrDistanceSame_Norm_8u32f_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Three-channel 8-bit unsigned image SqrDistanceSame_Norm.
- `NppStatus nppiSqrDistanceSame_Norm_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Four-channel 8-bit unsigned image SqrDistanceSame_Norm.
- `NppStatus nppiSqrDistanceSame_Norm_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Four-channel 8-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel.
- `NppStatus nppiSqrDistanceSame_Norm_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
One-channel 8-bit signed image SqrDistanceSame_Norm.
- `NppStatus nppiSqrDistanceSame_Norm_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Three-channel 8-bit signed image SqrDistanceSame_Norm.
- `NppStatus nppiSqrDistanceSame_Norm_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Four-channel 8-bit signed image SqrDistanceSame_Norm.
- `NppStatus nppiSqrDistanceSame_Norm_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Four-channel 8-bit signed image SqrDistanceSame_Norm ignoring alpha channel.
- `NppStatus nppiSqrDistanceSame_Norm_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
One-channel 16-bit unsigned image SqrDistanceSame_Norm.
- `NppStatus nppiSqrDistanceSame_Norm_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Three-channel 16-bit unsigned image SqrDistanceSame_Norm.
- `NppStatus nppiSqrDistanceSame_Norm_16u32f_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
Four-channel 16-bit unsigned image SqrDistanceSame_Norm.

- **NppStatus nppiSqrDistanceSame_Norm_16u32f_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel.

7.123.1 Detailed Description

Primitives for computing the normalized Euclidean distance between two images with same mode.

7.123.2 Function Documentation

- 7.123.2.1 NppStatus nppiSqrDistanceSame_Norm_16u32f_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.123.2.2 NppStatus nppiSqrDistanceSame_Norm_16u32f_C1R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep)

One-channel 16-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.123.2.3 `NppStatus nppiSqrDistanceSame_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.123.2.4 `NppStatus nppiSqrDistanceSame_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.123.2.5 `NppStatus nppiSqrDistanceSame_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.123.2.6 `NppStatus nppiSqrDistanceSame_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.123.2.7 `NppStatus nppiSqrDistanceSame_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.123.2.8 `NppStatus nppiSqrDistanceSame_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.123.2.9 `NppStatus nppiSqrDistanceSame_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.123.2.10 `NppStatus nppiSqrDistanceSame_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image SqrDistanceSame_Norm.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.123.2.11 `NppStatus nppiSqrDistanceSame_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image SqrDistanceSame_Norm.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.123.2.12 `NppStatus nppiSqrDistanceSame_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.123.2.13 `NppStatus nppiSqrDistanceSame_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.123.2.14 `NppStatus nppiSqrDistanceSame_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.123.2.15 `NppStatus nppiSqrDistanceSame_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.123.2.16 `NppStatus nppiSqrDistanceSame_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.123.2.17 `NppStatus nppiSqrDistanceSame_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel, scaled by $2^{\lceil -nScaleFactor \rceil}$.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.123.2.18 `NppStatus nppiSqrDistanceSame_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\lceil -nScaleFactor \rceil}$.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.123.2.19 `NppStatus nppiSqrDistanceSame_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\text{--}nScaleFactor}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.123.2.20 `NppStatus nppiSqrDistanceSame_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\text{--}nScaleFactor}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124 SqrDistanceValid_Norm

Primitives for computing the normalized Euclidean distance between two images with valid mode.

SqrDistanceValid_Norm

The functions compute the $\sigma_{st}(c, r)$ in [General Introduction](#) with valid mode (see [Categorizations](#)).

- `NppStatus nppiSqrDistanceValid_Norm_8u_C1RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

One-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{\ell - nScaleFactor}$.

- `NppStatus nppiSqrDistanceValid_Norm_8u_C3RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

Three-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{\ell - nScaleFactor}$.

- `NppStatus nppiSqrDistanceValid_Norm_8u_C4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

Four-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{\ell - nScaleFactor}$.

- `NppStatus nppiSqrDistanceValid_Norm_8u_AC4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

Four-channel 8-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel, scaled by $2^{\ell - nScaleFactor}$.

- `NppStatus nppiSqrDistanceValid_Norm_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 32-bit floating point image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 32-bit floating point image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 32-bit floating point image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 32-bit floating point image SqrDistanceValid_Norm ignoring alpha channel.

- `NppStatus nppiSqrDistanceValid_Norm_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit unsigned image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_8u32f_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit unsigned image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel.

- `NppStatus nppiSqrDistanceValid_Norm_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit signed image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit signed image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image SqrDistanceValid_Norm ignoring alpha channel.

- `NppStatus nppiSqrDistanceValid_Norm_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 16-bit unsigned image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 16-bit unsigned image SqrDistanceValid_Norm.

- `NppStatus nppiSqrDistanceValid_Norm_16u32f_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceValid_Norm.

- **NppStatus nppiSqrDistanceValid_Norm_16u32f_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel.

7.124.1 Detailed Description

Primitives for computing the normalized Euclidean distance between two images with valid mode.

7.124.2 Function Documentation

- 7.124.2.1 NppStatus nppiSqrDistanceValid_Norm_16u32f_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.124.2.2 NppStatus nppiSqrDistanceValid_Norm_16u32f_C1R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep)

One-channel 16-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124.2.3 `NppStatus nppiSqrDistanceValid_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124.2.4 `NppStatus nppiSqrDistanceValid_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124.2.5 `NppStatus nppiSqrDistanceValid_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.124.2.6 `NppStatus nppiSqrDistanceValid_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.124.2.7 `NppStatus nppiSqrDistanceValid_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124.2.8 `NppStatus nppiSqrDistanceValid_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124.2.9 `NppStatus nppiSqrDistanceValid_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.124.2.10 `NppStatus nppiSqrDistanceValid_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image SqrDistanceValid_Norm.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.124.2.11 `NppStatus nppiSqrDistanceValid_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image SqrDistanceValid_Norm.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.124.2.12 `NppStatus nppiSqrDistanceValid_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.124.2.13 `NppStatus nppiSqrDistanceValid_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.124.2.14 `NppStatus nppiSqrDistanceValid_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124.2.15 `NppStatus nppiSqrDistanceValid_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124.2.16 `NppStatus nppiSqrDistanceValid_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124.2.17 `NppStatus nppiSqrDistanceValid_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel, scaled by $2^l - nScaleFactor$).

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124.2.18 `NppStatus nppiSqrDistanceValid_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^l - nScaleFactor$).

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124.2.19 `NppStatus npqiSqrDistanceValid_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{\lfloor -nScaleFactor \rfloor}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.124.2.20 `NppStatus npqiSqrDistanceValid_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{\lfloor -nScaleFactor \rfloor}$.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.125 CrossCorrFull_Norm

Primitives for computing the normalized cross correlation between two images with full mode.

CrossCorrFull_Norm

The functions compute the $\rho_{st}(c, r)$ in [General Introduction](#) with full mode (see [Categorizations](#)).

- `NppStatus nppiCrossCorrFull_Norm_8u_C1RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

One-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{\ell} - nScaleFactor$.

- `NppStatus nppiCrossCorrFull_Norm_8u_C3RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

Three-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{\ell} - nScaleFactor$.

- `NppStatus nppiCrossCorrFull_Norm_8u_C4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

Four-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{\ell} - nScaleFactor$.

- `NppStatus nppiCrossCorrFull_Norm_8u_AC4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor)

Four-channel 8-bit unsigned image CrossCorrFull_Norm ignoring alpha channel, scaled by $2^{\ell} - nScaleFactor$.

- `NppStatus nppiCrossCorrFull_Norm_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 32-bit floating point image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 32-bit floating point image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 32-bit floating point image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 32-bit floating point image CrossCorrFull_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrFull_Norm_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit unsigned image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_8u32f_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit unsigned image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrFull_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrFull_Norm_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit signed image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit signed image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrFull_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrFull_Norm_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 16-bit unsigned image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 16-bit unsigned image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_16u32f_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrFull_Norm.

- `NppStatus nppiCrossCorrFull_Norm_16u32f_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrFull_Norm ignoring alpha channel.

7.125.1 Detailed Description

Primitives for computing the normalized cross correlation between two images with full mode.

7.125.2 Function Documentation

7.125.2.1 `NppStatus nppiCrossCorrFull_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrFull_Norm ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.125.2.2 `NppStatus nppiCrossCorrFull_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.125.2.3 `NppStatus nppiCrossCorrFull_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.125.2.4 `NppStatus nppiCrossCorrFull_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.125.2.5 `NppStatus nppiCrossCorrFull_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.125.2.6 `NppStatus nppiCrossCorrFull_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.125.2.7 `NppStatus nppiCrossCorrFull_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.125.2.8 `NppStatus nppiCrossCorrFull_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrFull_Norm.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.125.2.9 `NppStatus nppiCrossCorrFull_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrFull_Norm ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.125.2.10 `NppStatus nppiCrossCorrFull_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.125.2.11 `NppStatus nppiCrossCorrFull_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.125.2.12 `NppStatus nppiCrossCorrFull_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.125.2.13 `NppStatus nppiCrossCorrFull_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrFull_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.125.2.14 `NppStatus nppiCrossCorrFull_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.125.2.15 `NppStatus nppiCrossCorrFull_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.125.2.16 `NppStatus nppiCrossCorrFull_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrFull_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.125.2.17 `NppStatus nppiCrossCorrFull_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrFull_Norm ignoring alpha channel, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl [Pointer to the template image](#).
nTplStep [Number of bytes between successive rows in the template image](#).
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.125.2.18 `NppStatus nppiCrossCorrFull_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl [Pointer to the template image](#).
nTplStep [Number of bytes between successive rows in the template image](#).
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.125.2.19 `NppStatus nppiCrossCorrFull_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{(-nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.125.2.20 `NppStatus nppiCrossCorrFull_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{(-nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.126 CrossCorrSame_Norm

Primitives for computing the normalized cross correlation between two images with same mode.

CrossCorrSame_Norm

The functions compute the $\rho_{st}(c, r)$ in [General Introduction](#) with same mode (see [Categorizations](#)).

- [NppStatus nppiCrossCorrSame_Norm_8u_C1RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
One-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiCrossCorrSame_Norm_8u_C3RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Three-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiCrossCorrSame_Norm_8u_C4RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiCrossCorrSame_Norm_8u_AC4RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image CrossCorrSame_Norm ignoring alpha channel, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiCrossCorrSame_Norm_32f_C1R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
One-channel 32-bit floating point image CrossCorrSame_Norm.
- [NppStatus nppiCrossCorrSame_Norm_32f_C3R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Three-channel 32-bit floating point image CrossCorrSame_Norm.
- [NppStatus nppiCrossCorrSame_Norm_32f_C4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 32-bit floating point image CrossCorrSame_Norm.
- [NppStatus nppiCrossCorrSame_Norm_32f_AC4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 32-bit floating point image CrossCorrSame_Norm ignoring alpha channel.
- [NppStatus nppiCrossCorrSame_Norm_8u32f_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
One-channel 8-bit unsigned image CrossCorrSame_Norm.
- [NppStatus nppiCrossCorrSame_Norm_8u32f_C3R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Three-channel 8-bit unsigned image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrSame_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrSame_Norm_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit signed image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit signed image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrSame_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrSame_Norm_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 16-bit unsigned image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 16-bit unsigned image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_16u32f_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrSame_Norm.

- `NppStatus nppiCrossCorrSame_Norm_16u32f_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrSame_Norm ignoring alpha channel.

7.126.1 Detailed Description

Primitives for computing the normalized cross correlation between two images with same mode.

7.126.2 Function Documentation

7.126.2.1 `NppStatus nppiCrossCorrSame_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrSame_Norm ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.126.2.2 `NppStatus nppiCrossCorrSame_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.126.2.3 `NppStatus nppiCrossCorrSame_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.126.2.4 `NppStatus nppiCrossCorrSame_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.126.2.5 `NppStatus nppiCrossCorrSame_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.126.2.6 `NppStatus nppiCrossCorrSame_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.126.2.7 `NppStatus nppiCrossCorrSame_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.126.2.8 `NppStatus nppiCrossCorrSame_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.126.2.9 `NppStatus nppiCrossCorrSame_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.126.2.10 `NppStatus nppiCrossCorrSame_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.126.2.11 `NppStatus nppiCrossCorrSame_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.126.2.12 `NppStatus nppiCrossCorrSame_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.126.2.13 `NppStatus nppiCrossCorrSame_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrSame_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.126.2.14 `NppStatus nppiCrossCorrSame_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.126.2.15 `NppStatus nppiCrossCorrSame_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.126.2.16 `NppStatus nppiCrossCorrSame_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrSame_Norm.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.126.2.17 `NppStatus nppiCrossCorrSame_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrSame_Norm ignoring alpha channel, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.126.2.18 `NppStatus nppiCrossCorrSame_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.126.2.19 `NppStatus nppiCrossCorrSame_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{\lfloor \cdot \rfloor - nScaleFactor}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.126.2.20 `NppStatus nppiCrossCorrSame_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{\lfloor \cdot \rfloor - nScaleFactor}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127 CrossCorrValid_Norm

Primitives for computing the normalized cross correlation between two images with valid mode.

CrossCorrValid_Norm

The functions compute the $\rho_{st}(c, r)$ in [General Introduction](#) with valid mode (see [Categorizations](#)).

- [NppStatus nppiCrossCorrValid_Norm_8u_C1RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
One-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiCrossCorrValid_Norm_8u_C3RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Three-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiCrossCorrValid_Norm_8u_C4RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiCrossCorrValid_Norm_8u_AC4RSfs](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) *pDst, int nDstStep, int nScaleFactor)
Four-channel 8-bit unsigned image CrossCorrValid_Norm ignoring alpha channel, scaled by $2^{\ell - nScaleFactor}$.
- [NppStatus nppiCrossCorrValid_Norm_32f_C1R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
One-channel 32-bit floating point image CrossCorrValid_Norm.
- [NppStatus nppiCrossCorrValid_Norm_32f_C3R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Three-channel 32-bit floating point image CrossCorrValid_Norm.
- [NppStatus nppiCrossCorrValid_Norm_32f_C4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 32-bit floating point image CrossCorrValid_Norm.
- [NppStatus nppiCrossCorrValid_Norm_32f_AC4R](#) (const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
Four-channel 32-bit floating point image CrossCorrValid_Norm ignoring alpha channel.
- [NppStatus nppiCrossCorrValid_Norm_8u32f_C1R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)
One-channel 8-bit unsigned image CrossCorrValid_Norm.
- [NppStatus nppiCrossCorrValid_Norm_8u32f_C3R](#) (const [Npp8u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep)

Three-channel 8-bit unsigned image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit unsigned image CrossCorrValid_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrValid_Norm_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 8-bit signed image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 8-bit signed image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 8-bit signed image CrossCorrValid_Norm ignoring alpha channel.

- `NppStatus nppiCrossCorrValid_Norm_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 16-bit unsigned image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Three-channel 16-bit unsigned image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_16u32f_C4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrValid_Norm.

- `NppStatus nppiCrossCorrValid_Norm_16u32f_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

Four-channel 16-bit unsigned image CrossCorrValid_Norm ignoring alpha channel.

7.127.1 Detailed Description

Primitives for computing the normalized cross correlation between two images with valid mode.

7.127.2 Function Documentation

7.127.2.1 `NppStatus nppiCrossCorrValid_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrValid_Norm ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.2 `NppStatus nppiCrossCorrValid_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.3 `NppStatus nppiCrossCorrValid_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.4 `NppStatus nppiCrossCorrValid_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.5 `NppStatus nppiCrossCorrValid_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.127.2.6 `NppStatus nppiCrossCorrValid_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.127.2.7 `NppStatus nppiCrossCorrValid_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.8 `NppStatus nppiCrossCorrValid_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrValid_Norm.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.9 `NppStatus nppiCrossCorrValid_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrValid_Norm ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.10 `NppStatus nppiCrossCorrValid_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.127.2.11 `NppStatus nppiCrossCorrValid_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.127.2.12 `NppStatus nppiCrossCorrValid_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.127.2.13 `NppStatus nppiCrossCorrValid_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrValid_Norm ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.127.2.14 `NppStatus nppiCrossCorrValid_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.15 `NppStatus nppiCrossCorrValid_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.16 `NppStatus nppiCrossCorrValid_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrValid_Norm.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.17 `NppStatus nppiCrossCorrValid_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrValid_Norm ignoring alpha channel, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl [Pointer to the template image](#).
nTplStep [Number of bytes between successive rows in the template image](#).
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.18 `NppStatus nppiCrossCorrValid_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{(- nScaleFactor)}$.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl [Pointer to the template image](#).
nTplStep [Number of bytes between successive rows in the template image](#).
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.127.2.19 `NppStatus nppiCrossCorrValid_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{\lceil -nScaleFactor \rceil}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.127.2.20 `NppStatus nppiCrossCorrValid_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{\lceil -nScaleFactor \rceil}$.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
oSrcRoiSize Region-of-Interest (ROI).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize Region-of-Interest (ROI).
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
nScaleFactor Integer Result Scaling.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.128 CrossCorrValid

Primitives for computing the cross correlation between two images with valid mode.

CrossCorrValid

The functions compute the $R_{st}(c, r)$ in [General Introduction](#) with valid mode (see [Categorizations](#)).

- `NppStatus nppiCrossCorrValid_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
One-channel 32-bit floating point images CrossCorrValid.
- `NppStatus nppiCrossCorrValid_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
One-channel 8-bit unsigned images CrossCorrValid.
- `NppStatus nppiCrossCorrValid_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
One-channel 8-bit signed images CrossCorrValid.
- `NppStatus nppiCrossCorrValid_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)
One-channel 16-bit unsigned images CrossCorrValid.

7.128.1 Detailed Description

Primitives for computing the cross correlation between two images with valid mode.

7.128.2 Function Documentation

7.128.2.1 `NppStatus nppiCrossCorrValid_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep)

One-channel 16-bit unsigned images CrossCorrValid.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcRoiSize* [Region-of-Interest \(ROI\)](#).
- pTpl* [Pointer to the template image](#).
- nTplStep* [Number of bytes between successive rows in the template image](#).
- oTplRoiSize* [Region-of-Interest \(ROI\)](#).
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.128.2.2 `NppStatus nppiCrossCorrValid_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point images CrossCorrValid.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.128.2.3 `NppStatus nppiCrossCorrValid_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed images CrossCorrValid.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.128.2.4 `NppStatus nppiCrossCorrValid_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned images CrossCorrValid.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129 CrossCorrFull_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with full mode.

CrossCorrFull_NormLevel

The functions compute the $\gamma_{st}(c, r)$ in [General Introduction](#) with full mode (see [Categorizations](#)).

The functions require additional scratch buffer for computations.

- `NppStatus nppiCrossCorrFull_NormLevel_8u_C1RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

One-channel 8-bit unsigned image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_8u_C3RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit unsigned image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_8u_C4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_8u_AC4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrFull_NormLevel_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 32-bit floating point image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 32-bit floating point image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image CrossCorrFull_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 8-bit unsigned image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit unsigned image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 8-bit signed image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit signed image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit signed image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit signed image CrossCorrFull_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrFull_NormLevel_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 16-bit unsigned image CrossCorrFull_NormLevel.

- `NppStatus nppiCrossCorrFull_NormLevel_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 16-bit unsigned image CrossCorrFull_NormLevel.

- **NppStatus** [nppiCrossCorrFull_NormLevel_16u32f_C4R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit unsigned image CrossCorrFull_NormLevel.

- **NppStatus** [nppiCrossCorrFull_NormLevel_16u32f_AC4R](#) (const [Npp16u](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

FullNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CrossCorrFull_NormLevel primitives.

- **NppStatus** [nppiFullNormLevelGetBufferHostSize_8u_C1RSfs](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u_C1RSfs](#).

- **NppStatus** [nppiFullNormLevelGetBufferHostSize_8u_C3RSfs](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u_C3RSfs](#).

- **NppStatus** [nppiFullNormLevelGetBufferHostSize_8u_C4RSfs](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u_C4RSfs](#).

- **NppStatus** [nppiFullNormLevelGetBufferHostSize_8u_AC4RSfs](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u_AC4RSfs](#).

- **NppStatus** [nppiFullNormLevelGetBufferHostSize_32f_C1R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_32f_C1R](#).

- **NppStatus** [nppiFullNormLevelGetBufferHostSize_32f_C3R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_32f_C3R](#).

- **NppStatus** [nppiFullNormLevelGetBufferHostSize_32f_C4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_32f_C4R](#).

- **NppStatus** [nppiFullNormLevelGetBufferHostSize_32f_AC4R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_32f_AC4R](#).

- **NppStatus** [nppiFullNormLevelGetBufferHostSize_8u32f_C1R](#) ([NppiSize](#) oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u32f_C1R`.

- [NppStatus](#) [nppiFullNormLevelGetBufferHostSize_8u32f_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u32f_C3R`.

- [NppStatus](#) [nppiFullNormLevelGetBufferHostSize_8u32f_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u32f_C4R`.

- [NppStatus](#) [nppiFullNormLevelGetBufferHostSize_8u32f_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u32f_AC4R`.

- [NppStatus](#) [nppiFullNormLevelGetBufferHostSize_8s32f_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8s32f_C1R`.

- [NppStatus](#) [nppiFullNormLevelGetBufferHostSize_8s32f_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8s32f_C3R`.

- [NppStatus](#) [nppiFullNormLevelGetBufferHostSize_8s32f_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8s32f_C4R`.

- [NppStatus](#) [nppiFullNormLevelGetBufferHostSize_8s32f_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8s32f_AC4R`.

- [NppStatus](#) [nppiFullNormLevelGetBufferHostSize_16u32f_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_16u32f_C1R`.

- [NppStatus](#) [nppiFullNormLevelGetBufferHostSize_16u32f_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_16u32f_C3R`.

- [NppStatus](#) [nppiFullNormLevelGetBufferHostSize_16u32f_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_16u32f_C4R`.

- [NppStatus](#) [nppiFullNormLevelGetBufferHostSize_16u32f_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [*hpBufferSize](#))

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_16u32f_AC4R`.

7.129.1 Detailed Description

Primitives for computing the normalized cross correlation coefficient between two images with full mode.

7.129.2 Function Documentation

7.129.2.1 `NppStatus nppiCrossCorrFull_NormLevel_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_16u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.2 `NppStatus nppiCrossCorrFull_NormLevel_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_16u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.3 NppStatus nppiCrossCorrFull_NormLevel_16u32f_C3R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSrcRoiSize*, const Npp16u * *pTpl*, int *nTplStep*, NppiSize *oTplRoiSize*, Npp32f * *pDst*, int *nDstStep*, Npp8u * *pDeviceBuffer*)

Three-channel 16-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_16u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.4 NppStatus nppiCrossCorrFull_NormLevel_16u32f_C4R (const Npp16u * *pSrc*, int *nSrcStep*, NppiSize *oSrcRoiSize*, const Npp16u * *pTpl*, int *nTplStep*, NppiSize *oTplRoiSize*, Npp32f * *pDst*, int *nDstStep*, Npp8u * *pDeviceBuffer*)

Four-channel 16-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_16u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.5 NppStatus nppiCrossCorrFull_NormLevel_32f_AC4R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcRoiSize*, const Npp32f * *pTpl*, int *nTplStep*, NppiSize *oTplRoiSize*, Npp32f * *pDst*, int *nDstStep*, Npp8u * *pDeviceBuffer*)

Four-channel 32-bit floating point image CrossCorrFull_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.6 NppStatus nppiCrossCorrFull_NormLevel_32f_C1R (const Npp32f * *pSrc*, int *nSrcStep*, NppiSize *oSrcRoiSize*, const Npp32f * *pTpl*, int *nTplStep*, NppiSize *oTplRoiSize*, Npp32f * *pDst*, int *nDstStep*, Npp8u * *pDeviceBuffer*)

One-channel 32-bit floating point image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.7 `NppStatus nppiCrossCorrFull_NormLevel_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.8 `NppStatus nppiCrossCorrFull_NormLevel_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.9 `NppStatus nppiCrossCorrFull_NormLevel_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrFull_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8s32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.10 `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8s32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.11 `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8s32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.12 `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8s32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.13 `NppStatus nppiCrossCorrFull_NormLevel_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.14 `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.15 `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.16 `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.17 `NppStatus nppiCrossCorrFull_NormLevel_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u_AC4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.18 `NppStatus nppiCrossCorrFull_NormLevel_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u_C1RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.19 `NppStatus nppiCrossCorrFull_NormLevel_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcRoiSize* [Region-of-Interest \(ROI\)](#).
- pTpl* [Pointer to the template image](#).
- nTplStep* [Number of bytes between successive rows in the template image](#).
- oTplRoiSize* [Region-of-Interest \(ROI\)](#).
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- nScaleFactor* [Integer Result Scaling](#).
- pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u_C3RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.20 `NppStatus nppiCrossCorrFull_NormLevel_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel.

Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcRoiSize* [Region-of-Interest \(ROI\)](#).
- pTpl* [Pointer to the template image](#).
- nTplStep* [Number of bytes between successive rows in the template image](#).
- oTplRoiSize* [Region-of-Interest \(ROI\)](#).
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- nScaleFactor* [Integer Result Scaling](#).
- pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiFullNormLevelGetBufferHostSize_8u_C4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.129.2.21 NppStatus nppiFullNormLevelGetBufferHostSize_16u32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_16u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.22 NppStatus nppiFullNormLevelGetBufferHostSize_16u32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_16u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.23 NppStatus nppiFullNormLevelGetBufferHostSize_16u32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_16u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.24 NppStatus nppiFullNormLevelGetBufferHostSize_16u32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_16u32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.25 NppStatus nppiFullNormLevelGetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.26 NppStatus nppiFullNormLevelGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.27 NppStatus nppiFullNormLevelGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.28 NppStatus nppiFullNormLevelGetBufferHostSize_32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.29 NppStatus nppiFullNormLevelGetBufferHostSize_8s32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8s32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.30 NppStatus nppiFullNormLevelGetBufferHostSize_8s32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8s32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.31 NppStatus nppiFullNormLevelGetBufferHostSize_8s32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8s32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.32 NppStatus nppiFullNormLevelGetBufferHostSize_8s32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8s32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.33 NppStatus nppiFullNormLevelGetBufferHostSize_8u32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.34 NppStatus nppiFullNormLevelGetBufferHostSize_8u32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.35 NppStatus nppiFullNormLevelGetBufferHostSize_8u32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.36 NppStatus nppiFullNormLevelGetBufferHostSize_8u32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.37 NppStatus nppiFullNormLevelGetBufferHostSize_8u_AC4RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u_AC4RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.38 NppStatus nppiFullNormLevelGetBufferHostSize_8u_C1RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u_C1RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.39 NppStatus nppiFullNormLevelGetBufferHostSize_8u_C3RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u_C3RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.129.2.40 NppStatus nppiFullNormLevelGetBufferHostSize_8u_C4RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrFull_NormLevel_8u_C4RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130 CrossCorrSame_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with same mode.

CrossCorrSame_NormLevel

The functions compute the $\gamma_{st}(c, r)$ in [General Introduction](#) with same mode (see [Categorizations](#)).

The functions require additional scratch buffer for computations.

- `NppStatus nppiCrossCorrSame_NormLevel_8u_C1RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

One-channel 8-bit unsigned image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_8u_C3RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit unsigned image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_8u_C4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_8u_AC4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrSame_NormLevel_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 32-bit floating point image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 32-bit floating point image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image CrossCorrSame_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 8-bit unsigned image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit unsigned image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 8-bit signed image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit signed image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit signed image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit signed image CrossCorrSame_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 16-bit unsigned image CrossCorrSame_NormLevel.

- `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 16-bit unsigned image CrossCorrSame_NormLevel.

- **NppStatus** **nppiCrossCorrSame_NormLevel_16u32f_C4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)

Four-channel 16-bit unsigned image CrossCorrSame_NormLevel.

- **NppStatus** **nppiCrossCorrSame_NormLevel_16u32f_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)

Four-channel 16-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

SameNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CrossCorrSame_-NormLevel primitives.

- **NppStatus** **nppiSameNormLevelGetBufferHostSize_8u_C1RSfs** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8u_C1RSfs.

- **NppStatus** **nppiSameNormLevelGetBufferHostSize_8u_C3RSfs** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8u_C3RSfs.

- **NppStatus** **nppiSameNormLevelGetBufferHostSize_8u_C4RSfs** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8u_C4RSfs.

- **NppStatus** **nppiSameNormLevelGetBufferHostSize_8u_AC4RSfs** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8u_AC4RSfs.

- **NppStatus** **nppiSameNormLevelGetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_32f_C1R.

- **NppStatus** **nppiSameNormLevelGetBufferHostSize_32f_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_32f_C3R.

- **NppStatus** **nppiSameNormLevelGetBufferHostSize_32f_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_32f_C4R.

- **NppStatus** **nppiSameNormLevelGetBufferHostSize_32f_AC4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_32f_AC4R.

- **NppStatus** **nppiSameNormLevelGetBufferHostSize_8u32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u32f_C1R`.

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8u32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u32f_C3R`.

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8u32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u32f_C4R`.

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8u32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u32f_AC4R`.

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8s32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8s32f_C1R`.

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8s32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8s32f_C3R`.

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8s32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8s32f_C4R`.

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8s32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8s32f_AC4R`.

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_16u32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_16u32f_C1R`.

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_16u32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_16u32f_C3R`.

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_16u32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_16u32f_C4R`.

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_16u32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_16u32f_AC4R`.

7.130.1 Detailed Description

Primitives for computing the normalized cross correlation coefficient between two images with same mode.

7.130.2 Function Documentation

7.130.2.1 `NppStatus nppiCrossCorrSame_NormLevel_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_16u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.2 `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_16u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.3 `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_16u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.4 `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_16u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.5 `NppStatus nppiCrossCorrSame_NormLevel_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrSame_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.6 `NppStatus nppiCrossCorrSame_NormLevel_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image CrossCorrSame_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.7 `NppStatus nppiCrossCorrSame_NormLevel_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image CrossCorrSame_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.8 `NppStatus nppiCrossCorrSame_NormLevel_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrSame_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.9 `NppStatus nppiCrossCorrSame_NormLevel_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrSame_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8s32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.10 `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image CrossCorrSame_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8s32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.11 `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image CrossCorrSame_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8s32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.12 `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrSame_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8s32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.13 `NppStatus nppiCrossCorrSame_NormLevel_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.14 `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.15 `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.16 `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.17 `NppStatus nppiCrossCorrSame_NormLevel_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8u_AC4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.18 `NppStatus nppiCrossCorrSame_NormLevel_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiSameNormLevelGetBufferHostSize_8u_C1RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.19 `NppStatus nppiCrossCorrSame_NormLevel_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
nScaleFactor [Integer Result Scaling](#).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiSameNormLevelGetBufferHostSize_8u_C3RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.20 `NppStatus nppiCrossCorrSame_NormLevel_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
nScaleFactor [Integer Result Scaling](#).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiSameNormLevelGetBufferHostSize_8u_C4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.130.2.21 NppStatus nppiSameNormLevelGetBufferHostSize_16u32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_16u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.22 NppStatus nppiSameNormLevelGetBufferHostSize_16u32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_16u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.23 NppStatus nppiSameNormLevelGetBufferHostSize_16u32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_16u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.24 NppStatus nppiSameNormLevelGetBufferHostSize_16u32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_16u32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.25 NppStatus nppiSameNormLevelGetBufferHostSize_32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.26 NppStatus nppiSameNormLevelGetBufferHostSize_32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.27 NppStatus nppiSameNormLevelGetBufferHostSize_32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.28 NppStatus nppiSameNormLevelGetBufferHostSize_32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.29 NppStatus nppiSameNormLevelGetBufferHostSize_8s32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8s32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.30 NppStatus nppiSameNormLevelGetBufferHostSize_8s32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8s32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.31 NppStatus nppiSameNormLevelGetBufferHostSize_8s32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8s32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.32 NppStatus nppiSameNormLevelGetBufferHostSize_8s32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8s32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.33 NppStatus nppiSameNormLevelGetBufferHostSize_8u32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.34 NppStatus nppiSameNormLevelGetBufferHostSize_8u32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.35 NppStatus nppiSameNormLevelGetBufferHostSize_8u32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.36 NppStatus nppiSameNormLevelGetBufferHostSize_8u32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.37 NppStatus nppiSameNormLevelGetBufferHostSize_8u_AC4RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u_AC4RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.38 NppStatus nppiSameNormLevelGetBufferHostSize_8u_C1RSfs (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u_C1RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.39 NppStatus nppiSameNormLevelGetBufferHostSize_8u_C3RSfs (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u_C3RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.130.2.40 NppStatus nppiSameNormLevelGetBufferHostSize_8u_C4RSfs (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame_NormLevel_8u_C4RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131 CrossCorrValid_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with valid mode.

CrossCorrValid_NormLevel

The functions compute the $\gamma_{st}(c, r)$ in [General Introduction](#) with valid mode (see [Categorizations](#)).

The functions require additional scratch buffer for computations.

- `NppStatus nppiCrossCorrValid_NormLevel_8u_C1RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

One-channel 8-bit unsigned image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_8u_C3RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit unsigned image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_8u_C4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_8u_AC4RSfs` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` *pDst, int nDstStep, int nScaleFactor, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrValid_NormLevel_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 32-bit floating point image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 32-bit floating point image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_32f_C4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image CrossCorrValid_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C1R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 8-bit unsigned image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C3R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit unsigned image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_8u32f_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C1R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 8-bit signed image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C3R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit signed image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit signed image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_8s32f_AC4R` (const `Npp8s` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Four-channel 8-bit signed image CrossCorrValid_NormLevel ignoring alpha channel.

- `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C1R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

One-channel 16-bit unsigned image CrossCorrValid_NormLevel.

- `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C3R` (const `Npp16u` *pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` *pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` *pDst, int nDstStep, `Npp8u` *pDeviceBuffer)

Three-channel 16-bit unsigned image CrossCorrValid_NormLevel.

- **NppStatus** **nppiCrossCorrValid_NormLevel_16u32f_C4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)

Four-channel 16-bit unsigned image CrossCorrValid_NormLevel.

- **NppStatus** **nppiCrossCorrValid_NormLevel_16u32f_AC4R** (const **Npp16u** *pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** *pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** *pDst, int nDstStep, **Npp8u** *pDeviceBuffer)

Four-channel 16-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

ValidNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CrossCorrValid_NormLevel primitives.

- **NppStatus** **nppiValidNormLevelGetBufferHostSize_8u_C1RSfs** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u_C1RSfs.

- **NppStatus** **nppiValidNormLevelGetBufferHostSize_8u_C3RSfs** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u_C3RSfs.

- **NppStatus** **nppiValidNormLevelGetBufferHostSize_8u_C4RSfs** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u_C4RSfs.

- **NppStatus** **nppiValidNormLevelGetBufferHostSize_8u_AC4RSfs** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u_AC4RSfs.

- **NppStatus** **nppiValidNormLevelGetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_32f_C1R.

- **NppStatus** **nppiValidNormLevelGetBufferHostSize_32f_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_32f_C3R.

- **NppStatus** **nppiValidNormLevelGetBufferHostSize_32f_C4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_32f_C4R.

- **NppStatus** **nppiValidNormLevelGetBufferHostSize_32f_AC4R** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_32f_AC4R.

- **NppStatus** **nppiValidNormLevelGetBufferHostSize_8u32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8u32f_C1R`.

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8u32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8u32f_C3R`.

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8u32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8u32f_C4R`.

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8u32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8u32f_AC4R`.

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8s32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8s32f_C1R`.

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8s32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8s32f_C3R`.

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8s32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8s32f_C4R`.

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8s32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8s32f_AC4R`.

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_16u32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_16u32f_C1R`.

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_16u32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_16u32f_C3R`.

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_16u32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_16u32f_C4R`.

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_16u32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_16u32f_AC4R`.

7.131.1 Detailed Description

Primitives for computing the normalized cross correlation coefficient between two images with valid mode.

7.131.2 Function Documentation

7.131.2.1 `NppStatus nppiCrossCorrValid_NormLevel_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_16u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.2 `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_16u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.3 `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_16u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.4 `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

oSrcRoiSize Region-of-Interest (ROI).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize Region-of-Interest (ROI).

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_16u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.5 `NppStatus nppiCrossCorrValid_NormLevel_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrValid_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.6 `NppStatus nppiCrossCorrValid_NormLevel_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.7 `NppStatus nppiCrossCorrValid_NormLevel_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.8 `NppStatus nppiCrossCorrValid_NormLevel_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.9 `NppStatus nppiCrossCorrValid_NormLevel_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrValid_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8s32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.10 `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8s32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.11 `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8s32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.12 `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8s32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.13 `NppStatus nppiCrossCorrValid_NormLevel_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.14 `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.15 `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.16 `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl Pointer to the template image.

nTplStep Number of bytes between successive rows in the template image.

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.17 `NppStatus nppiCrossCorrValid_NormLevel_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u_AC4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.18 `NppStatus nppiCrossCorrValid_NormLevel_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

oSrcRoiSize [Region-of-Interest \(ROI\)](#).

pTpl [Pointer to the template image](#).

nTplStep [Number of bytes between successive rows in the template image](#).

oTplRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).
Use [nppiValidNormLevelGetBufferHostSize_8u_C1RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.19 `NppStatus nppiCrossCorrValid_NormLevel_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
nScaleFactor [Integer Result Scaling](#).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiValidNormLevelGetBufferHostSize_8u_C3RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.20 `NppStatus nppiCrossCorrValid_NormLevel_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
oSrcRoiSize [Region-of-Interest \(ROI\)](#).
pTpl Pointer to the template image.
nTplStep Number of bytes between successive rows in the template image.
oTplRoiSize [Region-of-Interest \(ROI\)](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
nScaleFactor [Integer Result Scaling](#).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiValidNormLevelGetBufferHostSize_8u_C4RSfs](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.131.2.21 NppStatus nppiValidNormLevelGetBufferHostSize_16u32f_AC4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_16u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.22 NppStatus nppiValidNormLevelGetBufferHostSize_16u32f_C1R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_16u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.23 NppStatus nppiValidNormLevelGetBufferHostSize_16u32f_C3R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_16u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.24 NppStatus nppiValidNormLevelGetBufferHostSize_16u32f_C4R (NppiSize oSizeROI, int * hpBufferSize)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_16u32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.25 NppStatus nppiValidNormLevelGetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.26 NppStatus nppiValidNormLevelGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.27 NppStatus nppiValidNormLevelGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.28 NppStatus nppiValidNormLevelGetBufferHostSize_32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.29 NppStatus nppiValidNormLevelGetBufferHostSize_8s32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8s32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.30 NppStatus nppiValidNormLevelGetBufferHostSize_8s32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8s32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.31 NppStatus nppiValidNormLevelGetBufferHostSize_8s32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8s32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.32 NppStatus nppiValidNormLevelGetBufferHostSize_8s32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8s32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.33 NppStatus nppiValidNormLevelGetBufferHostSize_8u32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.34 NppStatus nppiValidNormLevelGetBufferHostSize_8u32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.35 NppStatus nppiValidNormLevelGetBufferHostSize_8u32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.36 NppStatus nppiValidNormLevelGetBufferHostSize_8u32f_C4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u32f_C4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.37 NppStatus nppiValidNormLevelGetBufferHostSize_8u_AC4RSfs (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u_AC4RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.38 NppStatus nppiValidNormLevelGetBufferHostSize_8u_C1RSfs (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u_C1RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.39 NppStatus nppiValidNormLevelGetBufferHostSize_8u_C3RSfs (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u_C3RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.131.2.40 NppStatus nppiValidNormLevelGetBufferHostSize_8u_C4RSfs (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid_NormLevel_8u_C4RSfs](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.132 Image Quality Index

Primitives for computing the image quality index of two images.

QualityIndex

Given two images M and N (both $W \times H$), the mathematical formula to calculate the image quality index Q between them is expressed as:

$$Q = \frac{4\sigma_{MN}\tilde{M}\tilde{N}}{[(\tilde{M}^2) + (\tilde{N}^2)][(\sigma_M)^2 + (\sigma_N)^2]}$$

where

$$\tilde{M} = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} M(j, i)$$

$$\tilde{N} = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} N(j, i)$$

$$\sigma_M = \sqrt{\frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [M(j, i) - \tilde{M}]^2}$$

$$\sigma_N = \sqrt{\frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [N(j, i) - \tilde{N}]^2}$$

$$\sigma_{MN} = \frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [M(j, i) - \tilde{M}][N(j, i) - \tilde{N}]$$

The functions require additional scratch buffer for computations.

- [NppStatus nppiQualityIndex_8u32f_C1R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) *pDst, [Npp8u](#) *pDeviceBuffer)

One-channel 8-bit unsigned image QualityIndex.

- [NppStatus nppiQualityIndex_16u32f_C1R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) *pDst, [Npp8u](#) *pDeviceBuffer)

One-channel 16-bit unsigned image QualityIndex.

- [NppStatus nppiQualityIndex_32f_C1R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) *pDst, [Npp8u](#) *pDeviceBuffer)

One-channel 32-bit floating point image QualityIndex.

- [NppStatus nppiQualityIndex_8u32f_C3R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) *pDst, [Npp8u](#) *pDeviceBuffer)

Three-channel 8-bit unsigned image QualityIndex.

- [NppStatus nppiQualityIndex_16u32f_C3R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) *pDst, [Npp8u](#) *pDeviceBuffer)

Three-channel 16-bit unsigned image QualityIndex.

- **NppStatus** **nppiQualityIndex_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** *pDst, **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image QualityIndex.
- **NppStatus** **nppiQualityIndex_8u32f_AC4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** *pDst, **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image QualityIndex.
- **NppStatus** **nppiQualityIndex_16u32f_AC4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** *pDst, **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image QualityIndex.
- **NppStatus** **nppiQualityIndex_32f_AC4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** *pDst, **Npp8u** *pDeviceBuffer)
Four-channel 32-bit floating point image QualityIndex.

QualityIndexGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the QualityIndex primitives.

- **NppStatus** **nppiQualityIndexGetBufferHostSize_8u32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size (in bytes) for **nppiQualityIndex_8u32f_C1R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize_16u32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size (in bytes) for **nppiQualityIndex_16u32f_C1R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize_32f_C1R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size (in bytes) for **nppiQualityIndex_32f_C1R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize_8u32f_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size (in bytes) for **nppiQualityIndex_8u32f_C3R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize_16u32f_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size (in bytes) for **nppiQualityIndex_16u32f_C3R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize_32f_C3R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size (in bytes) for **nppiQualityIndex_32f_C3R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize_8u32f_AC4R** (**NppiSize** oSizeROI, int *hpBufferSize)
*Buffer size (in bytes) for **nppiQualityIndex_8u32f_AC4R**.*

- **NppStatus** `nppiQualityIndexGetBufferHostSize_16u32f_AC4R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size (in bytes) for [nppiQualityIndex_16u32f_AC4R](#).
- **NppStatus** `nppiQualityIndexGetBufferHostSize_32f_AC4R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)
Buffer size (in bytes) for [nppiQualityIndex_32f_AC4R](#).

7.132.1 Detailed Description

Primitives for computing the image quality index of two images.

7.132.2 Function Documentation

7.132.2.1 **NppStatus** `nppiQualityIndex_16u32f_AC4R` (**const** **Npp16u** * `pSrc1`, **int** `nSrc1Step`, **const** **Npp16u** * `pSrc2`, **int** `nSrc2Step`, **NppiSize** `oRoiSize`, **Npp32f** * `pDst`, **Npp8u** * `pDeviceBuffer`)

Four-channel 16-bit unsigned image QualityIndex.

Parameters:

- `pSrc1` [Source-Image Pointer](#).
- `nSrc1Step` [Source-Image Line Step](#).
- `pSrc2` [Source-Image Pointer](#).
- `nSrc2Step` [Source-Image Line Step](#).
- `oRoiSize` [Region-of-Interest \(ROI\)](#).
- `pDst` [Pointer to the quality index](#).
- `pDeviceBuffer` [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
 Use [nppiQualityIndexGetBufferHostSize_16u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.132.2.2 **NppStatus** `nppiQualityIndex_16u32f_C1R` (**const** **Npp16u** * `pSrc1`, **int** `nSrc1Step`, **const** **Npp16u** * `pSrc2`, **int** `nSrc2Step`, **NppiSize** `oRoiSize`, **Npp32f** * `pDst`, **Npp8u** * `pDeviceBuffer`)

One-channel 16-bit unsigned image QualityIndex.

Parameters:

- `pSrc1` [Source-Image Pointer](#).
- `nSrc1Step` [Source-Image Line Step](#).
- `pSrc2` [Source-Image Pointer](#).

nSrc2Step Source-Image Line Step.

oRoiSize Region-of-Interest (ROI).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_16u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_QUALITY_INDEX_ERROR if pixels of either image are constant numberse.

7.132.2.3 `NppStatus nppiQualityIndex_16u32f_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image QualityIndex.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oRoiSize Region-of-Interest (ROI).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_16u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_QUALITY_INDEX_ERROR if pixels of either image are constant numberse.

7.132.2.4 `NppStatus nppiQualityIndex_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image QualityIndex.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oRoiSize Region-of-Interest (ROI).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_QUALITY_INDEX_ERROR if pixels of either image are constant numberse.

7.132.2.5 `NppStatus nppiQualityIndex_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image QualityIndex.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oRoiSize [Region-of-Interest \(ROI\)](#).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_QUALITY_INDEX_ERROR if pixels of either image are constant numberse.

7.132.2.6 `NppStatus nppiQualityIndex_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image QualityIndex.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oRoiSize [Region-of-Interest \(ROI\)](#).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_QUALITY_INDEX_ERROR if pixels of either image are constant numberse.

7.132.2.7 `NppStatus nppiQualityIndex_8u32f_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image QualityIndex.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oRoiSize Region-of-Interest (ROI).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_8u32f_AC4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.132.2.8 `NppStatus nppiQualityIndex_8u32f_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image QualityIndex.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oRoiSize Region-of-Interest (ROI).

pDst Pointer to the quality index.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_8u32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.132.2.9 `NppStatus nppiQualityIndex_8u32f_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image QualityIndex.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oRoiSize [Region-of-Interest \(ROI\)](#).

pDst [Pointer to the quality index](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppiQualityIndexGetBufferHostSize_8u32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

7.132.2.10 `NppStatus nppiQualityIndexGetBufferHostSize_16u32f_AC4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for [nppiQualityIndex_16u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.132.2.11 `NppStatus nppiQualityIndexGetBufferHostSize_16u32f_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size (in bytes) for [nppiQualityIndex_16u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.132.2.12 NppStatus nppiQualityIndexGetBufferHostSize_16u32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiQualityIndex_16u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.132.2.13 NppStatus nppiQualityIndexGetBufferHostSize_32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiQualityIndex_32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.132.2.14 NppStatus nppiQualityIndexGetBufferHostSize_32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiQualityIndex_32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.132.2.15 NppStatus nppiQualityIndexGetBufferHostSize_32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiQualityIndex_32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.132.2.16 NppStatus nppiQualityIndexGetBufferHostSize_8u32f_AC4R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiQualityIndex_8u32f_AC4R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.132.2.17 NppStatus nppiQualityIndexGetBufferHostSize_8u32f_C1R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiQualityIndex_8u32f_C1R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.132.2.18 NppStatus nppiQualityIndexGetBufferHostSize_8u32f_C3R (NppiSize *oSizeROI*, int * *hpBufferSize*)

Buffer size (in bytes) for [nppiQualityIndex_8u32f_C3R](#).

Parameters:

oSizeROI [Region-of-Interest \(ROI\)](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_NULL_POINTER_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

7.133 MaximumError

Primitives for computing the maximum error between two images.

Functions

- **NppStatus nppiMaximumError_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image Maximum_Error.
- **NppStatus nppiMaximumError_8s_C1R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 8-bit signed image Maximum_Error.
- **NppStatus nppiMaximumError_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image Maximum_Error.
- **NppStatus nppiMaximumError_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image Maximum_Error.
- **NppStatus nppiMaximumError_16sc_C1R** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed complex image Maximum_Error.
- **NppStatus nppiMaximumError_32u_C1R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit unsigned image Maximum_Error.
- **NppStatus nppiMaximumError_32s_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit signed image Maximum_Error.
- **NppStatus nppiMaximumError_32sc_C1R** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit signed complex image Maximum_Error.
- **NppStatus nppiMaximumError_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image Maximum_Error.
- **NppStatus nppiMaximumError_32fc_C1R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point complex image Maximum_Error.
- **NppStatus nppiMaximumError_64f_C1R** (const **Npp64f** *pSrc1, int nSrc1Step, const **Npp64f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 64-bit floating point image Maximum_Error.

- `NppStatus nppiMaximumError_8u_C2R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Two-channel 8-bit unsigned image Maximum_Error.

- `NppStatus nppiMaximumError_8s_C2R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Two-channel 8-bit signed image Maximum_Error.

- `NppStatus nppiMaximumError_16u_C2R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Two-channel 16-bit unsigned image Maximum_Error.

- `NppStatus nppiMaximumError_16s_C2R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Two-channel 16-bit signed image Maximum_Error.

- `NppStatus nppiMaximumError_16sc_C2R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Two-channel 16-bit signed complex image Maximum_Error.

- `NppStatus nppiMaximumError_32u_C2R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Two-channel 32-bit unsigned image Maximum_Error.

- `NppStatus nppiMaximumError_32s_C2R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Two-channel 32-bit signed image Maximum_Error.

- `NppStatus nppiMaximumError_32sc_C2R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Two-channel 32-bit signed complex image Maximum_Error.

- `NppStatus nppiMaximumError_32f_C2R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Two-channel 32-bit floating point image Maximum_Error.

- `NppStatus nppiMaximumError_32fc_C2R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Two-channel 32-bit floating point complex image Maximum_Error.

- `NppStatus nppiMaximumError_64f_C2R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Two-channel 64-bit floating point image Maximum_Error.

- `NppStatus nppiMaximumError_8u_C3R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit unsigned image Maximum_Error.

- `NppStatus nppiMaximumError_8s_C3R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Three-channel 8-bit signed image Maximum_Error.

- [NppStatus nppiMaximumError_16u_C3R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit unsigned image Maximum_Error.
- [NppStatus nppiMaximumError_16s_C3R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit signed image Maximum_Error.
- [NppStatus nppiMaximumError_16sc_C3R](#) (const [Npp16sc](#) *pSrc1, int nSrc1Step, const [Npp16sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit signed complex image Maximum_Error.
- [NppStatus nppiMaximumError_32u_C3R](#) (const [Npp32u](#) *pSrc1, int nSrc1Step, const [Npp32u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit unsigned image Maximum_Error.
- [NppStatus nppiMaximumError_32s_C3R](#) (const [Npp32s](#) *pSrc1, int nSrc1Step, const [Npp32s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit signed image Maximum_Error.
- [NppStatus nppiMaximumError_32sc_C3R](#) (const [Npp32sc](#) *pSrc1, int nSrc1Step, const [Npp32sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit signed complex image Maximum_Error.
- [NppStatus nppiMaximumError_32f_C3R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit floating point image Maximum_Error.
- [NppStatus nppiMaximumError_32fc_C3R](#) (const [Npp32fc](#) *pSrc1, int nSrc1Step, const [Npp32fc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit floating point complex image Maximum_Error.
- [NppStatus nppiMaximumError_64f_C3R](#) (const [Npp64f](#) *pSrc1, int nSrc1Step, const [Npp64f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 64-bit floating point image Maximum_Error.
- [NppStatus nppiMaximumError_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image Maximum_Error.
- [NppStatus nppiMaximumError_8s_C4R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit signed image Maximum_Error.
- [NppStatus nppiMaximumError_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit unsigned image Maximum_Error.
- [NppStatus nppiMaximumError_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit signed image Maximum_Error.

- `NppStatus nppiMaximumError_16sc_C4R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 16-bit signed complex image Maximum_Error.

- `NppStatus nppiMaximumError_32u_C4R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit unsigned image Maximum_Error.

- `NppStatus nppiMaximumError_32s_C4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit signed image Maximum_Error.

- `NppStatus nppiMaximumError_32sc_C4R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit signed complex image Maximum_Error.

- `NppStatus nppiMaximumError_32f_C4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image Maximum_Error.

- `NppStatus nppiMaximumError_32fc_C4R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point complex image Maximum_Error.

- `NppStatus nppiMaximumError_64f_C4R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 64-bit floating point image Maximum_Error.

7.133.1 Detailed Description

Primitives for computing the maximum error between two images.

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*, the maximum error is defined as the largest absolute difference between pixels of two images. If the image is in complex format, the absolute value of the complex number is provided.

7.133.2 Function Documentation

7.133.2.1 `NppStatus nppiMaximumError_16s_C1R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

One-channel 16-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.2 `NppStatus nppiMaximumError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.3 `NppStatus nppiMaximumError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.4 `NppStatus nppiMaximumError_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.5 `NppStatus nppiMaximumError_16sc_C1R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.6 `NppStatus nppiMaximumError_16sc_C2R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed complex image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error (absolute value).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.7 `NppStatus nppiMaximumError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed complex image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error (absolute value).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.8 `NppStatus nppiMaximumError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed complex image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.9 `NppStatus nppiMaximumError_16u_C1R (const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

One-channel 16-bit unsigned image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.10 `NppStatus nppiMaximumError_16u_C2R (const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

Two-channel 16-bit unsigned image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.11 `NppStatus nppiMaximumError_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.12 `NppStatus nppiMaximumError_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.13 `NppStatus nppiMaximumError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.133.2.14 `NppStatus nppiMaximumError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.133.2.15 `NppStatus nppiMaximumError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.133.2.16 `NppStatus nppiMaximumError_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.133.2.17 `NppStatus nppiMaximumError_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.133.2.18 NppStatus nppiMaximumError_32fc_C2R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)

Two-channel 32-bit floating point complex image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.133.2.19 NppStatus nppiMaximumError_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)

Three-channel 32-bit floating point complex image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.133.2.20 `NppStatus nppiMaximumError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.133.2.21 `NppStatus nppiMaximumError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.22 `NppStatus nppiMaximumError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.23 `NppStatus nppiMaximumError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.24 `NppStatus nppiMaximumError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.25 `NppStatus nppiMaximumError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.26 `NppStatus nppiMaximumError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.27 `NppStatus nppiMaximumError_32sc_C3R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.28 `NppStatus nppiMaximumError_32sc_C4R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed complex image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.29 `NppStatus nppiMaximumError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.30 `NppStatus nppiMaximumError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.31 `NppStatus nppiMaximumError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.32 `NppStatus nppiMaximumError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.33 `NppStatus nppiMaximumError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.133.2.34 `NppStatus nppiMaximumError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.133.2.35 `NppStatus nppiMaximumError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.133.2.36 `NppStatus nppiMaximumError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.133.2.37 `NppStatus nppiMaximumError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.38 `NppStatus nppiMaximumError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiMaximumErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.39 `NppStatus nppiMaximumError_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiMaximumErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.40 `NppStatus nppiMaximumError_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiMaximumErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.41 `NppStatus nppiMaximumError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.42 `NppStatus nppiMaximumError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image Maximum_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.43 `NppStatus nppiMaximumError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.133.2.44 `NppStatus nppiMaximumError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Maximum_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134 AverageError

Primitives for computing the average error between two images.

Functions

- **NppStatus nppiAverageError_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image Average_Error.
- **NppStatus nppiAverageError_8s_C1R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 8-bit signed image Average_Error.
- **NppStatus nppiAverageError_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image Average_Error.
- **NppStatus nppiAverageError_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image Average_Error.
- **NppStatus nppiAverageError_16sc_C1R** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed complex image Average_Error.
- **NppStatus nppiAverageError_32u_C1R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit unsigned image Average_Error.
- **NppStatus nppiAverageError_32s_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit signed image Average_Error.
- **NppStatus nppiAverageError_32sc_C1R** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit signed complex image Average_Error.
- **NppStatus nppiAverageError_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image Average_Error.
- **NppStatus nppiAverageError_32fc_C1R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point complex image Average_Error.
- **NppStatus nppiAverageError_64f_C1R** (const **Npp64f** *pSrc1, int nSrc1Step, const **Npp64f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 64-bit floating point image Average_Error.

- [NppStatus nppiAverageError_8u_C2R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 8-bit unsigned image Average_Error.
- [NppStatus nppiAverageError_8s_C2R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 8-bit signed image Average_Error.
- [NppStatus nppiAverageError_16u_C2R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 16-bit unsigned image Average_Error.
- [NppStatus nppiAverageError_16s_C2R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 16-bit signed image Average_Error.
- [NppStatus nppiAverageError_16sc_C2R](#) (const [Npp16sc](#) *pSrc1, int nSrc1Step, const [Npp16sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 16-bit signed complex image Average_Error.
- [NppStatus nppiAverageError_32u_C2R](#) (const [Npp32u](#) *pSrc1, int nSrc1Step, const [Npp32u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit unsigned image Average_Error.
- [NppStatus nppiAverageError_32s_C2R](#) (const [Npp32s](#) *pSrc1, int nSrc1Step, const [Npp32s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit signed image Average_Error.
- [NppStatus nppiAverageError_32sc_C2R](#) (const [Npp32sc](#) *pSrc1, int nSrc1Step, const [Npp32sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit signed complex image Average_Error.
- [NppStatus nppiAverageError_32f_C2R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit floating point image Average_Error.
- [NppStatus nppiAverageError_32fc_C2R](#) (const [Npp32fc](#) *pSrc1, int nSrc1Step, const [Npp32fc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 32-bit floating point complex image Average_Error.
- [NppStatus nppiAverageError_64f_C2R](#) (const [Npp64f](#) *pSrc1, int nSrc1Step, const [Npp64f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Two-channel 64-bit floating point image Average_Error.
- [NppStatus nppiAverageError_8u_C3R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 8-bit unsigned image Average_Error.
- [NppStatus nppiAverageError_8s_C3R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 8-bit signed image Average_Error.

- **NppStatus** **nppiAverageError_16u_C3R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Three-channel 16-bit unsigned image Average_Error.
- **NppStatus** **nppiAverageError_16s_C3R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed image Average_Error.
- **NppStatus** **nppiAverageError_16sc_C3R** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Three-channel 16-bit signed complex image Average_Error.
- **NppStatus** **nppiAverageError_32u_C3R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Three-channel 32-bit unsigned image Average_Error.
- **NppStatus** **nppiAverageError_32s_C3R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Three-channel 32-bit signed image Average_Error.
- **NppStatus** **nppiAverageError_32sc_C3R** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Three-channel 32-bit signed complex image Average_Error.
- **NppStatus** **nppiAverageError_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point image Average_Error.
- **NppStatus** **nppiAverageError_32fc_C3R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Three-channel 32-bit floating point complex image Average_Error.
- **NppStatus** **nppiAverageError_64f_C3R** (const **Npp64f** *pSrc1, int nSrc1Step, const **Npp64f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Three-channel 64-bit floating point image Average_Error.
- **NppStatus** **nppiAverageError_8u_C4R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Four-channel 8-bit unsigned image Average_Error.
- **NppStatus** **nppiAverageError_8s_C4R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Four-channel 8-bit signed image Average_Error.
- **NppStatus** **nppiAverageError_16u_C4R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Four-channel 16-bit unsigned image Average_Error.
- **NppStatus** **nppiAverageError_16s_C4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 16-bit signed image Average_Error.

- **NppStatus** **nppiAverageError_16sc_C4R** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 16-bit signed complex image Average_Error.

- **NppStatus** **nppiAverageError_32u_C4R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 32-bit unsigned image Average_Error.

- **NppStatus** **nppiAverageError_32s_C4R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 32-bit signed image Average_Error.

- **NppStatus** **nppiAverageError_32sc_C4R** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 32-bit signed complex image Average_Error.

- **NppStatus** **nppiAverageError_32f_C4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 32-bit floating point image Average_Error.

- **NppStatus** **nppiAverageError_32fc_C4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 32-bit floating point complex image Average_Error.

- **NppStatus** **nppiAverageError_64f_C4R** (const **Npp64f** *pSrc1, int nSrc1Step, const **Npp64f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 64-bit floating point image Average_Error.

7.134.1 Detailed Description

Primitives for computing the average error between two images.

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*, the average error is defined as:

$$AverageError = \frac{1}{W \cdot H \cdot N} \sum_{n=0}^{N-1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|$$

where *N* stands for the number of channels. If the image is in complex format, the absolute value is used for computation.

7.134.2 Function Documentation

- 7.134.2.1** **NppStatus nppiAverageError_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

One-channel 16-bit signed image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.2 `NppStatus nppiAverageError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.3 `NppStatus nppiAverageError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.4 `NppStatus nppiAverageError_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.5 `NppStatus nppiAverageError_16sc_C1R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed complex image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.6 `NppStatus nppiAverageError_16sc_C2R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.7 `NppStatus nppiAverageError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.8 `NppStatus nppiAverageError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed complex image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error (absolute value).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.9 `NppStatus nppiAverageError_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.10 `NppStatus nppiAverageError_16u_C2R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit unsigned image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.11 `NppStatus nppiAverageError_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.12 `NppStatus nppiAverageError_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.13 `NppStatus nppiAverageError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.134.2.14 `NppStatus nppiAverageError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.134.2.15 `NppStatus nppiAverageError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.134.2.16 `NppStatus nppiAverageError_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.134.2.17 `NppStatus nppiAverageError_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.134.2.18 `NppStatus nppiAverageError_32fc_C2R (const Npp32fc *pSrc1, int nSrc1Step, const Npp32fc *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

Two-channel 32-bit floating point complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.134.2.19 `NppStatus nppiAverageError_32fc_C3R (const Npp32fc *pSrc1, int nSrc1Step, const Npp32fc *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

Three-channel 32-bit floating point complex image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.134.2.20 `NppStatus nppiAverageError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error (absolute value).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.134.2.21 `NppStatus nppiAverageError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.22 `NppStatus nppiAverageError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.23 `NppStatus nppiAverageError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.24 `NppStatus nppiAverageError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.25 `NppStatus nppiAverageError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error (absolute value).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.26 `NppStatus nppiAverageError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.27 `NppStatus nppiAverageError_32sc_C3R (const Npp32sc *pSrc1, int nSrc1Step, const Npp32sc *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

Three-channel 32-bit signed complex image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.28 `NppStatus nppiAverageError_32sc_C4R (const Npp32sc *pSrc1, int nSrc1Step, const Npp32sc *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

Four-channel 32-bit signed complex image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.29 `NppStatus nppiAverageError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.30 `NppStatus nppiAverageError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.31 `NppStatus nppiAverageError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.32 `NppStatus nppiAverageError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.33 `NppStatus nppiAverageError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.134.2.34 `NppStatus nppiAverageError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.134.2.35 `NppStatus nppiAverageError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.134.2.36 `NppStatus nppiAverageError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [npapiAverageErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.134.2.37 `NppStatus nppiAverageError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [npapiAverageErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.38 `NppStatus nppiAverageError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.39 `NppStatus nppiAverageError_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image Average_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.40 `NppStatus nppiAverageError_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.41 `NppStatus nppiAverageError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.42 `NppStatus nppiAverageError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.43 `NppStatus nppiAverageError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.134.2.44 `NppStatus nppiAverageError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Average_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135 MaximumRelativeError

Primitives for computing the maximum relative error between two images.

Functions

- `NppStatus nppiMaximumRelativeError_8u_C1R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 8-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_8s_C1R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 8-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16u_C1R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16s_C1R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16sc_C1R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 16-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32u_C1R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32s_C1R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32sc_C1R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32f_C1R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32fc_C1R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 32-bit floating point complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_64f_C1R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
One-channel 64-bit floating point image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_8u_C2R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 8-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_8s_C2R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 8-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16u_C2R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 16-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16s_C2R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 16-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16sc_C2R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 16-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32u_C2R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32s_C2R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32sc_C2R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32f_C2R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit floating point image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32fc_C2R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 32-bit floating point complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_64f_C2R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Two-channel 64-bit floating point image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_8u_C3R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 8-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_8s_C3R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 8-bit signed image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_16u_C3R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 16-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16s_C3R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 16-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16sc_C3R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 16-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32u_C3R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 32-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32s_C3R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 32-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32sc_C3R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 32-bit signed complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32f_C3R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 32-bit floating point image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_32fc_C3R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 32-bit floating point complex image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_64f_C3R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Three-channel 64-bit floating point image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_8u_C4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Four-channel 8-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_8s_C4R` (const `Npp8s` *pSrc1, int nSrc1Step, const `Npp8s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Four-channel 8-bit signed image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16u_C4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)
Four-channel 16-bit unsigned image MaximumRelative_Error.
- `NppStatus nppiMaximumRelativeError_16s_C4R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 16-bit signed image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_16sc_C4R` (const `Npp16sc` *pSrc1, int nSrc1Step, const `Npp16sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 16-bit signed complex image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_32u_C4R` (const `Npp32u` *pSrc1, int nSrc1Step, const `Npp32u` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit unsigned image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_32s_C4R` (const `Npp32s` *pSrc1, int nSrc1Step, const `Npp32s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit signed image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_32sc_C4R` (const `Npp32sc` *pSrc1, int nSrc1Step, const `Npp32sc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit signed complex image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_32f_C4R` (const `Npp32f` *pSrc1, int nSrc1Step, const `Npp32f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_32fc_C4R` (const `Npp32fc` *pSrc1, int nSrc1Step, const `Npp32fc` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 32-bit floating point complex image MaximumRelative_Error.

- `NppStatus nppiMaximumRelativeError_64f_C4R` (const `Npp64f` *pSrc1, int nSrc1Step, const `Npp64f` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

Four-channel 64-bit floating point image MaximumRelative_Error.

7.135.1 Detailed Description

Primitives for computing the maximum relative error between two images.

Given two images $pSrc1$ and $pSrc2$ both with width W and height H , the maximum relative error is defined as:

$$MaximumRelativeError = \max \frac{|pSrc1(j, i) - pSrc2(j, i)|}{\max(|pSrc1(j, i)|, |pSrc2(j, i)|)}$$

If the image is in complex format, the absolute value is used for computation. For multiple channles, the maximum relative error of all the channles is returned.

7.135.2 Function Documentation

- 7.135.2.1** `NppStatus nppiMaximumRelativeError_16s_C1R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` *pError, `Npp8u` *pDeviceBuffer)

One-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.2 `NppStatus nppiMaximumRelativeError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.3 `NppStatus nppiMaximumRelativeError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.4 `NppStatus nppiMaximumRelativeError_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.

pSrc2 [Source-Image Pointer](#).

nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.5 `NppStatus nppiMaximumRelativeError_16sc_C1R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.

pSrc2 [Source-Image Pointer](#).

nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.6 `NppStatus nppiMaximumRelativeError_16sc_C2R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error (absolute value).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.7 `NppStatus nppiMaximumRelativeError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error (absolute value).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.8 `NppStatus nppiMaximumRelativeError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.9 `NppStatus nppiMaximumRelativeError_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.10 `NppStatus nppiMaximumRelativeError_16u_C2R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.11 `NppStatus nppiMaximumRelativeError_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.12 `NppStatus nppiMaximumRelativeError_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.13 `NppStatus nppiMaximumRelativeError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.135.2.14 `NppStatus nppiMaximumRelativeError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.135.2.15 `NppStatus nppiMaximumRelativeError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.135.2.16 `NppStatus nppiMaximumRelativeError_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.135.2.17 `NppStatus nppiMaximumRelativeError_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.135.2.18 `NppStatus nppiMaximumRelativeError_32fc_C2R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.135.2.19 `NppStatus nppiMaximumRelativeError_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.135.2.20 `NppStatus nppiMaximumRelativeError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.135.2.21 `NppStatus nppiMaximumRelativeError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.22 `NppStatus nppiMaximumRelativeError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.23 `NppStatus nppiMaximumRelativeError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.24 `NppStatus nppiMaximumRelativeError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.25 `NppStatus nppiMaximumRelativeError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.26 `NppStatus nppiMaximumRelativeError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.

pSrc2 [Source-Image Pointer](#).

nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.27 `NppStatus nppiMaximumRelativeError_32sc_C3R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.

pSrc2 [Source-Image Pointer](#).

nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.28 `NppStatus nppiMaximumRelativeError_32sc_C4R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error (absolute value).
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.29 `NppStatus nppiMaximumRelativeError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.30 `NppStatus nppiMaximumRelativeError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.31 `NppStatus nppiMaximumRelativeError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.32 `NppStatus nppiMaximumRelativeError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.33 `NppStatus nppiMaximumRelativeError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.135.2.34 `NppStatus nppiMaximumRelativeError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.135.2.35 `NppStatus nppiMaximumRelativeError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.135.2.36 `NppStatus nppiMaximumRelativeError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.135.2.37 `NppStatus nppiMaximumRelativeError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.38 `NppStatus nppiMaximumRelativeError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.39 `NppStatus nppiMaximumRelativeError_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.40 `NppStatus nppiMaximumRelativeError_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.41 `NppStatus nppiMaximumRelativeError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.42 `NppStatus nppiMaximumRelativeError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.43 `NppStatus nppiMaximumRelativeError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.135.2.44 `NppStatus nppiMaximumRelativeError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136 AverageRelativeError

Primitives for computing the average relative error between two images.

Functions

- **NppStatus nppiAverageRelativeError_8u_C1R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 8-bit unsigned image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_8s_C1R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 8-bit signed image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_16u_C1R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 16-bit unsigned image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_16sc_C1R** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 16-bit signed complex image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_32u_C1R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit unsigned image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_32s_C1R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit signed image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_32sc_C1R** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit signed complex image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_32fc_C1R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 32-bit floating point complex image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_64f_C1R** (const **Npp64f** *pSrc1, int nSrc1Step, const **Npp64f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
One-channel 64-bit floating point image MaximumRelative_Error.

- **NppStatus nppiAverageRelativeError_8u_C2R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Two-channel 8-bit unsigned image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_8s_C2R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Two-channel 8-bit signed image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_16u_C2R** (const **Npp16u** *pSrc1, int nSrc1Step, const **Npp16u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Two-channel 16-bit unsigned image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_16s_C2R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Two-channel 16-bit signed image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_16sc_C2R** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Two-channel 16-bit signed complex image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_32u_C2R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Two-channel 32-bit unsigned image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_32s_C2R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Two-channel 32-bit signed image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_32sc_C2R** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Two-channel 32-bit signed complex image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_32f_C2R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Two-channel 32-bit floating point image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_32fc_C2R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Two-channel 32-bit floating point complex image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_64f_C2R** (const **Npp64f** *pSrc1, int nSrc1Step, const **Npp64f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Two-channel 64-bit floating point image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_8u_C3R** (const **Npp8u** *pSrc1, int nSrc1Step, const **Npp8u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Three-channel 8-bit unsigned image MaximumRelative_Error.
- **NppStatus nppiAverageRelativeError_8s_C3R** (const **Npp8s** *pSrc1, int nSrc1Step, const **Npp8s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)
Three-channel 8-bit signed image MaximumRelative_Error.

- [NppStatus nppiAverageRelativeError_16u_C3R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_16s_C3R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit signed image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_16sc_C3R](#) (const [Npp16sc](#) *pSrc1, int nSrc1Step, const [Npp16sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 16-bit signed complex image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_32u_C3R](#) (const [Npp32u](#) *pSrc1, int nSrc1Step, const [Npp32u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_32s_C3R](#) (const [Npp32s](#) *pSrc1, int nSrc1Step, const [Npp32s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit signed image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_32sc_C3R](#) (const [Npp32sc](#) *pSrc1, int nSrc1Step, const [Npp32sc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit signed complex image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_32f_C3R](#) (const [Npp32f](#) *pSrc1, int nSrc1Step, const [Npp32f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit floating point image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_32fc_C3R](#) (const [Npp32fc](#) *pSrc1, int nSrc1Step, const [Npp32fc](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 32-bit floating point complex image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_64f_C3R](#) (const [Npp64f](#) *pSrc1, int nSrc1Step, const [Npp64f](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Three-channel 64-bit floating point image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_8u_C4R](#) (const [Npp8u](#) *pSrc1, int nSrc1Step, const [Npp8u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_8s_C4R](#) (const [Npp8s](#) *pSrc1, int nSrc1Step, const [Npp8s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 8-bit signed image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_16u_C4R](#) (const [Npp16u](#) *pSrc1, int nSrc1Step, const [Npp16u](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)
Four-channel 16-bit unsigned image MaximumRelative_Error.
- [NppStatus nppiAverageRelativeError_16s_C4R](#) (const [Npp16s](#) *pSrc1, int nSrc1Step, const [Npp16s](#) *pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) *pError, [Npp8u](#) *pDeviceBuffer)

Four-channel 16-bit signed image MaximumRelative_Error.

- **NppStatus nppiAverageRelativeError_16sc_C4R** (const **Npp16sc** *pSrc1, int nSrc1Step, const **Npp16sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 16-bit signed complex image MaximumRelative_Error.

- **NppStatus nppiAverageRelativeError_32u_C4R** (const **Npp32u** *pSrc1, int nSrc1Step, const **Npp32u** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 32-bit unsigned image MaximumRelative_Error.

- **NppStatus nppiAverageRelativeError_32s_C4R** (const **Npp32s** *pSrc1, int nSrc1Step, const **Npp32s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 32-bit signed image MaximumRelative_Error.

- **NppStatus nppiAverageRelativeError_32sc_C4R** (const **Npp32sc** *pSrc1, int nSrc1Step, const **Npp32sc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 32-bit signed complex image MaximumRelative_Error.

- **NppStatus nppiAverageRelativeError_32f_C4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 32-bit floating point image MaximumRelative_Error.

- **NppStatus nppiAverageRelativeError_32fc_C4R** (const **Npp32fc** *pSrc1, int nSrc1Step, const **Npp32fc** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 32-bit floating point complex image MaximumRelative_Error.

- **NppStatus nppiAverageRelativeError_64f_C4R** (const **Npp64f** *pSrc1, int nSrc1Step, const **Npp64f** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

Four-channel 64-bit floating point image MaximumRelative_Error.

7.136.1 Detailed Description

Primitives for computing the average relative error between two images.

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*, the maximum relative error is defined as:

$$AverageRelativeError = \frac{1}{W \cdot H \cdot N} \sum_{n=0}^{N-1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} \frac{|pSrc1(j, i) - pSrc2(j, i)|}{\max(|pSrc1(j, i)|, |pSrc2(j, i)|)}$$

where *N* is the number of channels. If the image is in complex format, the absolute value is used for computation.

7.136.2 Function Documentation

- 7.136.2.1 NppStatus nppiAverageRelativeError_16s_C1R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** *pError, **Npp8u** *pDeviceBuffer)

One-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.2 `NppStatus nppiAverageRelativeError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.3 `NppStatus nppiAverageRelativeError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.4 `NppStatus nppiAverageRelativeError_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.

pSrc2 [Source-Image Pointer](#).

nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.5 `NppStatus nppiAverageRelativeError_16sc_C1R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step Source-Image Line Step.

pSrc2 [Source-Image Pointer](#).

nSrc2Step Source-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.6 `NppStatus nppiAverageRelativeError_16sc_C2R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.7 `NppStatus nppiAverageRelativeError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.8 `NppStatus nppiAverageRelativeError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.9 `NppStatus nppiAverageRelativeError_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.10 `NppStatus nppiAverageRelativeError_16u_C2R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.11 NppStatus nppiAverageRelativeError_16u_C3R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f * *pError*, Npp8u * *pDeviceBuffer*)

Three-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.12 NppStatus nppiAverageRelativeError_16u_C4R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f * *pError*, Npp8u * *pDeviceBuffer*)

Four-channel 16-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.13 `NppStatus nppiAverageRelativeError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.136.2.14 `NppStatus nppiAverageRelativeError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.136.2.15 `NppStatus nppiAverageRelativeError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

7.136.2.16 `NppStatus nppiAverageRelativeError_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.136.2.17 `NppStatus nppiAverageRelativeError_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.136.2.18 `NppStatus nppiAverageRelativeError_32fc_C2R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.136.2.19 `NppStatus nppiAverageRelativeError_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.136.2.20 `NppStatus nppiAverageRelativeError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.136.2.21 `NppStatus nppiAverageRelativeError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.22 `NppStatus nppiAverageRelativeError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.23 `NppStatus nppiAverageRelativeError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.24 `NppStatus nppiAverageRelativeError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
oSizeROI Region-of-Interest (ROI).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.25 `NppStatus nppiAverageRelativeError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.26 `NppStatus nppiAverageRelativeError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.27 `NppStatus nppiAverageRelativeError_32sc_C3R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.28 `NppStatus nppiAverageRelativeError_32sc_C4R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed complex image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error (absolute value).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16s_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.29 `NppStatus nppiAverageRelativeError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.30 `NppStatus nppiAverageRelativeError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.31 `NppStatus nppiAverageRelativeError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

oSizeROI Region-of-Interest (ROI).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.32 `NppStatus nppiAverageRelativeError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageRelativeErrorGetBufferHostSize_16u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.33 `NppStatus nppiAverageRelativeError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
 Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.136.2.34 `NppStatus nppiAverageRelativeError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.136.2.35 `NppStatus nppiAverageRelativeError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.136.2.36 `NppStatus nppiAverageRelativeError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_32f_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

7.136.2.37 `NppStatus nppiAverageRelativeError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.38 `NppStatus nppiAverageRelativeError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.39 `NppStatus nppiAverageRelativeError_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.40 `NppStatus nppiAverageRelativeError_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).
nSrc1Step [Source-Image Line Step](#).
pSrc2 [Source-Image Pointer](#).
nSrc2Step [Source-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
pError Pointer to the computed error.
pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.41 `NppStatus nppiAverageRelativeError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C1R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.42 `NppStatus nppiAverageRelativeError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C2R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.43 `NppStatus nppiAverageRelativeError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C3R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.136.2.44 `NppStatus nppiAverageRelativeError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image MaximumRelative_Error.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

pError Pointer to the computed error.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppiAverageRelativeErrorGetBufferHostSize_8u_C4R](#) to compute the required size (in bytes).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.137 Memory Management

Routines for allocating and deallocating pitched image storage.

Functions

- void [nppiFree](#) (void *pData)
Free method for any 2D allocated memory.

Image Memory Allocation

ImageAllocator methods for 2D arrays of data.

The allocators have width and height parameters to specify the size of the image data being allocated. They return a pointer to the newly created memory and return the numbers of bytes between successive lines.

If the memory allocation failed due to lack of free device memory or device memory fragmentation the routine returns 0.

All allocators return memory with line strides that are beneficial for performance. It is not mandatory to use these allocators. Any valid CUDA device-memory pointers can be used by the NPP primitives and there are no restrictions on line strides.

- [Npp8u * nppiMalloc_8u_C1](#) (int nWidthPixels, int nHeightPixels, int *pStepBytes)
8-bit unsigned image memory allocator.
- [Npp8u * nppiMalloc_8u_C2](#) (int nWidthPixels, int nHeightPixels, int *pStepBytes)
2 channel 8-bit unsigned image memory allocator.
- [Npp8u * nppiMalloc_8u_C3](#) (int nWidthPixels, int nHeightPixels, int *pStepBytes)
3 channel 8-bit unsigned image memory allocator.
- [Npp8u * nppiMalloc_8u_C4](#) (int nWidthPixels, int nHeightPixels, int *pStepBytes)
4 channel 8-bit unsigned image memory allocator.
- [Npp16u * nppiMalloc_16u_C1](#) (int nWidthPixels, int nHeightPixels, int *pStepBytes)
16-bit unsigned image memory allocator.
- [Npp16u * nppiMalloc_16u_C2](#) (int nWidthPixels, int nHeightPixels, int *pStepBytes)
2 channel 16-bit unsigned image memory allocator.
- [Npp16u * nppiMalloc_16u_C3](#) (int nWidthPixels, int nHeightPixels, int *pStepBytes)
3 channel 16-bit unsigned image memory allocator.
- [Npp16u * nppiMalloc_16u_C4](#) (int nWidthPixels, int nHeightPixels, int *pStepBytes)
4 channel 16-bit unsigned image memory allocator.
- [Npp16s * nppiMalloc_16s_C1](#) (int nWidthPixels, int nHeightPixels, int *pStepBytes)
16-bit signed image memory allocator.

- `Npp16s * nppiMalloc_16s_C2` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
2 channel 16-bit signed image memory allocator.
- `Npp16s * nppiMalloc_16s_C4` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
4 channel 16-bit signed image memory allocator.
- `Npp16sc * nppiMalloc_16sc_C1` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
1 channel 16-bit signed complex image memory allocator.
- `Npp16sc * nppiMalloc_16sc_C2` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
2 channel 16-bit signed complex image memory allocator.
- `Npp16sc * nppiMalloc_16sc_C3` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
3 channel 16-bit signed complex image memory allocator.
- `Npp16sc * nppiMalloc_16sc_C4` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
4 channel 16-bit signed complex image memory allocator.
- `Npp32s * nppiMalloc_32s_C1` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
32-bit signed image memory allocator.
- `Npp32s * nppiMalloc_32s_C3` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
3 channel 32-bit signed image memory allocator.
- `Npp32s * nppiMalloc_32s_C4` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
4 channel 32-bit signed image memory allocator.
- `Npp32sc * nppiMalloc_32sc_C1` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
32-bit integer complex image memory allocator.
- `Npp32sc * nppiMalloc_32sc_C2` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
2 channel 32-bit integer complex image memory allocator.
- `Npp32sc * nppiMalloc_32sc_C3` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
3 channel 32-bit integer complex image memory allocator.
- `Npp32sc * nppiMalloc_32sc_C4` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
4 channel 32-bit integer complex image memory allocator.
- `Npp32f * nppiMalloc_32f_C1` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
32-bit floating point image memory allocator.
- `Npp32f * nppiMalloc_32f_C2` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
2 channel 32-bit floating point image memory allocator.
- `Npp32f * nppiMalloc_32f_C3` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
3 channel 32-bit floating point image memory allocator.
- `Npp32f * nppiMalloc_32f_C4` (int nWidthPixels, int nHeightPixels, int *pStepBytes)
4 channel 32-bit floating point image memory allocator.

- `Npp32fc * nppiMalloc_32fc_C1` (int *nWidthPixels*, int *nHeightPixels*, int **pStepBytes*)
32-bit float complex image memory allocator.
- `Npp32fc * nppiMalloc_32fc_C2` (int *nWidthPixels*, int *nHeightPixels*, int **pStepBytes*)
2 channel 32-bit float complex image memory allocator.
- `Npp32fc * nppiMalloc_32fc_C3` (int *nWidthPixels*, int *nHeightPixels*, int **pStepBytes*)
3 channel 32-bit float complex image memory allocator.
- `Npp32fc * nppiMalloc_32fc_C4` (int *nWidthPixels*, int *nHeightPixels*, int **pStepBytes*)
4 channel 32-bit float complex image memory allocator.

7.137.1 Detailed Description

Routines for allocating and deallocating pitched image storage.

These methods are provided for convenience. They allocate memory that may contain additional padding bytes at the end of each line of pixels. Though padding is not necessary for any of the NPP image-processing primitives to work correctly, its absence may cause severe performance degradation compared to properly padded images.

7.137.2 Function Documentation

7.137.2.1 void nppiFree (void * *pData*)

Free method for any 2D allocated memory.

This method should be used to free memory allocated with any of the `nppiMalloc_<modifier>` methods.

Parameters:

pData A pointer to memory allocated using `nppiMalloc_<modifier>`.

7.137.2.2 Npp16s* nppiMalloc_16s_C1 (int *nWidthPixels*, int *nHeightPixels*, int **pStepBytes*)

16-bit signed image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.3 Npp16s* nppiMalloc_16s_C2 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

2 channel 16-bit signed image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.4 Npp16s* nppiMalloc_16s_C4 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

4 channel 16-bit signed image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.5 Npp16sc* nppiMalloc_16sc_C1 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

1 channel 16-bit signed complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.6 Npp16sc* nppiMalloc_16sc_C2 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

2 channel 16-bit signed complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.7 Npp16sc* nppiMalloc_16sc_C3 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

3 channel 16-bit signed complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.8 Npp16sc* nppiMalloc_16sc_C4 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

4 channel 16-bit signed complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.9 Npp16u* nppiMalloc_16u_C1 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

16-bit unsigned image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.10 Npp16u* nppiMalloc_16u_C2 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

2 channel 16-bit unsigned image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.11 Npp16u* nppiMalloc_16u_C3 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

3 channel 16-bit unsigned image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.12 Npp16u* nppiMalloc_16u_C4 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

4 channel 16-bit unsigned image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.13 Npp32f* nppiMalloc_32f_C1 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

32-bit floating point image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.14 Npp32f* nppiMalloc_32f_C2 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

2 channel 32-bit floating point image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.15 Npp32f* nppiMalloc_32f_C3 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

3 channel 32-bit floating point image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.16 Npp32f* nppiMalloc_32f_C4 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

4 channel 32-bit floating point image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.17 Npp32fc* nppiMalloc_32fc_C1 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

32-bit float complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.18 Npp32fc* nppiMalloc_32fc_C2 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

2 channel 32-bit float complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.19 Npp32fc* nppiMalloc_32fc_C3 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

3 channel 32-bit float complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.20 Npp32fc* nppiMalloc_32fc_C4 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

4 channel 32-bit float complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.21 Npp32s* nppiMalloc_32s_C1 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

32-bit signed image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.22 Npp32s* nppiMalloc_32s_C3 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

3 channel 32-bit signed image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.23 Npp32s* nppiMalloc_32s_C4 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

4 channel 32-bit signed image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.24 Npp32sc* nppiMalloc_32sc_C1 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

32-bit integer complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.25 Npp32sc* nppiMalloc_32sc_C2 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

2 channel 32-bit integer complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.26 Npp32sc* nppiMalloc_32sc_C3 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

3 channel 32-bit integer complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.27 Npp32sc* nppiMalloc_32sc_C4 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

4 channel 32-bit integer complex image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.28 Npp8u* nppiMalloc_8u_C1 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

8-bit unsigned image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.29 Npp8u* nppiMalloc_8u_C2 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

2 channel 8-bit unsigned image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.30 Npp8u* nppiMalloc_8u_C3 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

3 channel 8-bit unsigned image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.137.2.31 Npp8u* nppiMalloc_8u_C4 (int *nWidthPixels*, int *nHeightPixels*, int * *pStepBytes*)

4 channel 8-bit unsigned image memory allocator.

Parameters:

nWidthPixels Image width.

nHeightPixels Image height.

pStepBytes [Line Step](#).

Returns:

Pointer to new image data.

7.138 Threshold and Compare Operations

Methods for pixel-wise threshold and compare operations.

Modules

- [Threshold Operations](#)

Threshold image pixels.

- [Compare Operations](#)

Compare the pixels of two images and create a binary result image.

7.138.1 Detailed Description

Methods for pixel-wise threshold and compare operations.

7.139 Threshold Operations

Threshold image pixels.

Functions

- **NppStatus nppiThreshold_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** nThreshold, **NppCmpOp** eComparisonOperation)
1 channel 8-bit unsigned char threshold.
- **NppStatus nppiThreshold_8u_C1IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp8u** nThreshold, **NppCmpOp** eComparisonOperation)
1 channel 8-bit unsigned char in place threshold.
- **NppStatus nppiThreshold_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** nThreshold, **NppCmpOp** eComparisonOperation)
1 channel 16-bit unsigned short threshold.
- **NppStatus nppiThreshold_16u_C1IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16u** nThreshold, **NppCmpOp** eComparisonOperation)
1 channel 16-bit unsigned short in place threshold.
- **NppStatus nppiThreshold_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16s** nThreshold, **NppCmpOp** eComparisonOperation)
1 channel 16-bit signed short threshold.
- **NppStatus nppiThreshold_16s_C1IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16s** nThreshold, **NppCmpOp** eComparisonOperation)
1 channel 16-bit signed short in place threshold.
- **NppStatus nppiThreshold_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** nThreshold, **NppCmpOp** eComparisonOperation)
1 channel 32-bit floating point threshold.
- **NppStatus nppiThreshold_32f_C1IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** nThreshold, **NppCmpOp** eComparisonOperation)
1 channel 32-bit floating point in place threshold.
- **NppStatus nppiThreshold_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** rThresholds[3], **NppCmpOp** eComparisonOperation)
3 channel 8-bit unsigned char threshold.
- **NppStatus nppiThreshold_8u_C3IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp8u** rThresholds[3], **NppCmpOp** eComparisonOperation)
3 channel 8-bit unsigned char in place threshold.
- **NppStatus nppiThreshold_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** rThresholds[3], **NppCmpOp** eComparisonOperation)
3 channel 16-bit unsigned short threshold.

- `NppStatus nppiThreshold_16u_C3IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], `NppCmpOp` eComparisonOperation)
3 channel 16-bit unsigned short in place threshold.
- `NppStatus nppiThreshold_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], `NppCmpOp` eComparisonOperation)
3 channel 16-bit signed short threshold.
- `NppStatus nppiThreshold_16s_C3IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], `NppCmpOp` eComparisonOperation)
3 channel 16-bit signed short in place threshold.
- `NppStatus nppiThreshold_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], `NppCmpOp` eComparisonOperation)
3 channel 32-bit floating point threshold.
- `NppStatus nppiThreshold_32f_C3IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], `NppCmpOp` eComparisonOperation)
3 channel 32-bit floating point in place threshold.
- `NppStatus nppiThreshold_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], `NppCmpOp` eComparisonOperation)
4 channel 8-bit unsigned char image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_8u_AC4IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], `NppCmpOp` eComparisonOperation)
4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], `NppCmpOp` eComparisonOperation)
4 channel 16-bit unsigned short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_16u_AC4IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], `NppCmpOp` eComparisonOperation)
4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], `NppCmpOp` eComparisonOperation)
4 channel 16-bit signed short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_16s_AC4IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], `NppCmpOp` eComparisonOperation)
4 channel 16-bit signed short in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], `NppCmpOp` eComparisonOperation)
4 channel 32-bit floating point image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_32f_AC4IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], `NppCmpOp` eComparisonOperation)
4 channel 32-bit floating point in place image threshold, not affecting Alpha.

- `NppStatus nppiThreshold_GT_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold)
1 channel 8-bit unsigned char threshold.
- `NppStatus nppiThreshold_GT_8u_C1IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold)
1 channel 8-bit unsigned char in place threshold.
- `NppStatus nppiThreshold_GT_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold)
1 channel 16-bit unsigned short threshold.
- `NppStatus nppiThreshold_GT_16u_C1IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold)
1 channel 16-bit unsigned short in place threshold.
- `NppStatus nppiThreshold_GT_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold)
1 channel 16-bit signed short threshold.
- `NppStatus nppiThreshold_GT_16s_C1IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold)
1 channel 16-bit signed short in place threshold.
- `NppStatus nppiThreshold_GT_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold)
1 channel 32-bit floating point threshold.
- `NppStatus nppiThreshold_GT_32f_C1IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold)
1 channel 32-bit floating point in place threshold.
- `NppStatus nppiThreshold_GT_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])
3 channel 8-bit unsigned char threshold.
- `NppStatus nppiThreshold_GT_8u_C3IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])
3 channel 8-bit unsigned char in place threshold.
- `NppStatus nppiThreshold_GT_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])
3 channel 16-bit unsigned short threshold.
- `NppStatus nppiThreshold_GT_16u_C3IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])
3 channel 16-bit unsigned short in place threshold.
- `NppStatus nppiThreshold_GT_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])

3 channel 16-bit signed short threshold.

- `NppStatus nppiThreshold_GT_16s_C3IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])

3 channel 16-bit signed short in place threshold.

- `NppStatus nppiThreshold_GT_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])

3 channel 32-bit floating point threshold.

- `NppStatus nppiThreshold_GT_32f_C3IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])

3 channel 32-bit floating point in place threshold.

- `NppStatus nppiThreshold_GT_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

- `NppStatus nppiThreshold_GT_8u_AC4IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

- `NppStatus nppiThreshold_GT_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

- `NppStatus nppiThreshold_GT_16u_AC4IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

- `NppStatus nppiThreshold_GT_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])

4 channel 16-bit signed short image threshold, not affecting Alpha.

- `NppStatus nppiThreshold_GT_16s_AC4IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

- `NppStatus nppiThreshold_GT_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])

4 channel 32-bit floating point image threshold, not affecting Alpha.

- `NppStatus nppiThreshold_GT_32f_AC4IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

- `NppStatus nppiThreshold_LT_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold)

1 channel 8-bit unsigned char threshold.

- **NppStatus** **nppiThreshold_LT_8u_C1IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp8u** nThreshold)
1 channel 8-bit unsigned char in place threshold.
- **NppStatus** **nppiThreshold_LT_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** nThreshold)
1 channel 16-bit unsigned short threshold.
- **NppStatus** **nppiThreshold_LT_16u_C1IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16u** nThreshold)
1 channel 16-bit unsigned short in place threshold.
- **NppStatus** **nppiThreshold_LT_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16s** nThreshold)
1 channel 16-bit signed short threshold.
- **NppStatus** **nppiThreshold_LT_16s_C1IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16s** nThreshold)
1 channel 16-bit signed short in place threshold.
- **NppStatus** **nppiThreshold_LT_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** nThreshold)
1 channel 32-bit floating point threshold.
- **NppStatus** **nppiThreshold_LT_32f_C1IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** nThreshold)
1 channel 32-bit floating point in place threshold.
- **NppStatus** **nppiThreshold_LT_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** rThresholds[3])
3 channel 8-bit unsigned char threshold.
- **NppStatus** **nppiThreshold_LT_8u_C3IR** (**Npp8u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp8u** rThresholds[3])
3 channel 8-bit unsigned char in place threshold.
- **NppStatus** **nppiThreshold_LT_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, **Npp16u** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** rThresholds[3])
3 channel 16-bit unsigned short threshold.
- **NppStatus** **nppiThreshold_LT_16u_C3IR** (**Npp16u** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16u** rThresholds[3])
3 channel 16-bit unsigned short in place threshold.
- **NppStatus** **nppiThreshold_LT_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3])
3 channel 16-bit signed short threshold.
- **NppStatus** **nppiThreshold_LT_16s_C3IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3])
3 channel 16-bit signed short in place threshold.

- `NppStatus nppiThreshold_LT_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])
3 channel 32-bit floating point threshold.
- `NppStatus nppiThreshold_LT_32f_C3IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])
3 channel 32-bit floating point in place threshold.
- `NppStatus nppiThreshold_LT_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])
4 channel 8-bit unsigned char image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LT_8u_AC4IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])
4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LT_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])
4 channel 16-bit unsigned short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LT_16u_AC4IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])
4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LT_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])
4 channel 16-bit signed short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LT_16s_AC4IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])
4 channel 16-bit signed short in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LT_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])
4 channel 32-bit floating point image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LT_32f_AC4IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])
4 channel 32-bit floating point in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_Val_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue, `NppCmpOp` eComparisonOperation)
1 channel 8-bit unsigned char threshold.
- `NppStatus nppiThreshold_Val_8u_C1IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue, `NppCmpOp` eComparisonOperation)
1 channel 8-bit unsigned char in place threshold.

- `NppStatus nppiThreshold_Val_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue, `NppCmpOp` eComparisonOperation)
1 channel 16-bit unsigned short threshold.
- `NppStatus nppiThreshold_Val_16u_C1IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue, `NppCmpOp` eComparisonOperation)
1 channel 16-bit unsigned short in place threshold.
- `NppStatus nppiThreshold_Val_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue, `NppCmpOp` eComparisonOperation)
1 channel 16-bit signed short threshold.
- `NppStatus nppiThreshold_Val_16s_C1IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue, `NppCmpOp` eComparisonOperation)
1 channel 16-bit signed short in place threshold.
- `NppStatus nppiThreshold_Val_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue, `NppCmpOp` eComparisonOperation)
1 channel 32-bit floating point threshold.
- `NppStatus nppiThreshold_Val_32f_C1IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue, `NppCmpOp` eComparisonOperation)
1 channel 32-bit floating point in place threshold.
- `NppStatus nppiThreshold_Val_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3], `NppCmpOp` eComparisonOperation)
3 channel 8-bit unsigned char threshold.
- `NppStatus nppiThreshold_Val_8u_C3IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3], `NppCmpOp` eComparisonOperation)
3 channel 8-bit unsigned char in place threshold.
- `NppStatus nppiThreshold_Val_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3], `NppCmpOp` eComparisonOperation)
3 channel 16-bit unsigned short threshold.
- `NppStatus nppiThreshold_Val_16u_C3IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3], `NppCmpOp` eComparisonOperation)
3 channel 16-bit unsigned short in place threshold.
- `NppStatus nppiThreshold_Val_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3], `NppCmpOp` eComparisonOperation)
3 channel 16-bit signed short threshold.

- `NppStatus nppiThreshold_Val_16s_C3IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3], `NppCmpOp` eComparisonOperation)
3 channel 16-bit signed short in place threshold.
- `NppStatus nppiThreshold_Val_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3], `NppCmpOp` eComparisonOperation)
3 channel 32-bit floating point threshold.
- `NppStatus nppiThreshold_Val_32f_C3IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3], `NppCmpOp` eComparisonOperation)
3 channel 32-bit floating point in place threshold.
- `NppStatus nppiThreshold_Val_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3], `NppCmpOp` eComparisonOperation)
4 channel 8-bit unsigned char image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_Val_8u_AC4IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3], `NppCmpOp` eComparisonOperation)
4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_Val_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3], `NppCmpOp` eComparisonOperation)
4 channel 16-bit unsigned short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_Val_16u_AC4IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3], `NppCmpOp` eComparisonOperation)
4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_Val_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3], `NppCmpOp` eComparisonOperation)
4 channel 16-bit signed short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_Val_16s_AC4IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3], `NppCmpOp` eComparisonOperation)
4 channel 16-bit signed short in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_Val_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3], `NppCmpOp` eComparisonOperation)
4 channel 32-bit floating point image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_Val_32f_AC4IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3], `NppCmpOp` eComparisonOperation)
4 channel 32-bit floating point in place image threshold, not affecting Alpha.

- `NppStatus nppiThreshold_GTVal_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue)
1 channel 8-bit unsigned char threshold.
- `NppStatus nppiThreshold_GTVal_8u_C1IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue)
1 channel 8-bit unsigned char in place threshold.
- `NppStatus nppiThreshold_GTVal_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)
1 channel 16-bit unsigned short threshold.
- `NppStatus nppiThreshold_GTVal_16u_C1IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)
1 channel 16-bit unsigned short in place threshold.
- `NppStatus nppiThreshold_GTVal_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)
1 channel 16-bit signed short threshold.
- `NppStatus nppiThreshold_GTVal_16s_C1IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)
1 channel 16-bit signed short in place threshold.
- `NppStatus nppiThreshold_GTVal_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)
1 channel 32-bit floating point threshold.
- `NppStatus nppiThreshold_GTVal_32f_C1IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)
1 channel 32-bit floating point in place threshold.
- `NppStatus nppiThreshold_GTVal_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])
3 channel 8-bit unsigned char threshold.
- `NppStatus nppiThreshold_GTVal_8u_C3IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])
3 channel 8-bit unsigned char in place threshold.
- `NppStatus nppiThreshold_GTVal_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])
3 channel 16-bit unsigned short threshold.
- `NppStatus nppiThreshold_GTVal_16u_C3IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])
3 channel 16-bit unsigned short in place threshold.
- `NppStatus nppiThreshold_GTVal_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])
3 channel 16-bit signed short threshold.

- `NppStatus nppiThreshold_GTVVal_16s_C3IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])
3 channel 16-bit signed short in place threshold.
- `NppStatus nppiThreshold_GTVVal_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])
3 channel 32-bit floating point threshold.
- `NppStatus nppiThreshold_GTVVal_32f_C3IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])
3 channel 32-bit floating point in place threshold.
- `NppStatus nppiThreshold_GTVVal_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])
4 channel 8-bit unsigned char image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_GTVVal_8u_AC4IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])
4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_GTVVal_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])
4 channel 16-bit unsigned short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_GTVVal_16u_AC4IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])
4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_GTVVal_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])
4 channel 16-bit signed short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_GTVVal_16s_AC4IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])
4 channel 16-bit signed short in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_GTVVal_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])
4 channel 32-bit floating point image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_GTVVal_32f_AC4IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])
4 channel 32-bit floating point in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTVVal_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue)
1 channel 8-bit unsigned char threshold.
- `NppStatus nppiThreshold_LTVVal_8u_C1IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue)

1 channel 8-bit unsigned char in place threshold.

- `NppStatus nppiThreshold_LTVal_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)

1 channel 16-bit unsigned short threshold.

- `NppStatus nppiThreshold_LTVal_16u_C1IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)

1 channel 16-bit unsigned short in place threshold.

- `NppStatus nppiThreshold_LTVal_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)

1 channel 16-bit signed short threshold.

- `NppStatus nppiThreshold_LTVal_16s_C1IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)

1 channel 16-bit signed short in place threshold.

- `NppStatus nppiThreshold_LTVal_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)

1 channel 32-bit floating point threshold.

- `NppStatus nppiThreshold_LTVal_32f_C1IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)

1 channel 32-bit floating point in place threshold.

- `NppStatus nppiThreshold_LTVal_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])

3 channel 8-bit unsigned char threshold.

- `NppStatus nppiThreshold_LTVal_8u_C3IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])

3 channel 8-bit unsigned char in place threshold.

- `NppStatus nppiThreshold_LTVal_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])

3 channel 16-bit unsigned short threshold.

- `NppStatus nppiThreshold_LTVal_16u_C3IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])

3 channel 16-bit unsigned short in place threshold.

- `NppStatus nppiThreshold_LTVal_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])

3 channel 16-bit signed short threshold.

- `NppStatus nppiThreshold_LTVal_16s_C3IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])

3 channel 16-bit signed short in place threshold.

- `NppStatus nppiThreshold_LTVal_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])
3 channel 32-bit floating point threshold.
- `NppStatus nppiThreshold_LTVal_32f_C3IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])
3 channel 32-bit floating point in place threshold.
- `NppStatus nppiThreshold_LTVal_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])
4 channel 8-bit unsigned char image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTVal_8u_AC4IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])
4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTVal_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])
4 channel 16-bit unsigned short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTVal_16u_AC4IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])
4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTVal_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])
4 channel 16-bit signed short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTVal_16s_AC4IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])
4 channel 16-bit signed short in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTVal_32f_AC4R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])
4 channel 32-bit floating point image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTVal_32f_AC4IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])
4 channel 32-bit floating point in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTValGTVal_8u_C1R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThresholdLT, const `Npp8u` nValueLT, const `Npp8u` nThresholdGT, const `Npp8u` nValueGT)
1 channel 8-bit unsigned char threshold.
- `NppStatus nppiThreshold_LTValGTVal_8u_C1IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThresholdLT, const `Npp8u` nValueLT, const `Npp8u` nThresholdGT, const `Npp8u` nValueGT)
1 channel 8-bit unsigned char in place threshold.

- `NppStatus nppiThreshold_LTValGTVal_16u_C1R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThresholdLT, const `Npp16u` nValueLT, const `Npp16u` nThresholdGT, const `Npp16u` nValueGT)

1 channel 16-bit unsigned short threshold.

- `NppStatus nppiThreshold_LTValGTVal_16u_C1IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThresholdLT, const `Npp16u` nValueLT, const `Npp16u` nThresholdGT, const `Npp16u` nValueGT)

1 channel 16-bit unsigned short in place threshold.

- `NppStatus nppiThreshold_LTValGTVal_16s_C1R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThresholdLT, const `Npp16s` nValueLT, const `Npp16s` nThresholdGT, const `Npp16s` nValueGT)

1 channel 16-bit signed short threshold.

- `NppStatus nppiThreshold_LTValGTVal_16s_C1IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThresholdLT, const `Npp16s` nValueLT, const `Npp16s` nThresholdGT, const `Npp16s` nValueGT)

1 channel 16-bit signed short in place threshold.

- `NppStatus nppiThreshold_LTValGTVal_32f_C1R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThresholdLT, const `Npp32f` nValueLT, const `Npp32f` nThresholdGT, const `Npp32f` nValueGT)

1 channel 32-bit floating point threshold.

- `NppStatus nppiThreshold_LTValGTVal_32f_C1IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThresholdLT, const `Npp32f` nValueLT, const `Npp32f` nThresholdGT, const `Npp32f` nValueGT)

1 channel 32-bit floating point in place threshold.

- `NppStatus nppiThreshold_LTValGTVal_8u_C3R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholdsLT[3], const `Npp8u` rValuesLT[3], const `Npp8u` rThresholdsGT[3], const `Npp8u` rValuesGT[3])

3 channel 8-bit unsigned char threshold.

- `NppStatus nppiThreshold_LTValGTVal_8u_C3IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholdsLT[3], const `Npp8u` rValuesLT[3], const `Npp8u` rThresholdsGT[3], const `Npp8u` rValuesGT[3])

3 channel 8-bit unsigned char in place threshold.

- `NppStatus nppiThreshold_LTValGTVal_16u_C3R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholdsLT[3], const `Npp16u` rValuesLT[3], const `Npp16u` rThresholdsGT[3], const `Npp16u` rValuesGT[3])

3 channel 16-bit unsigned short threshold.

- `NppStatus nppiThreshold_LTValGTVal_16u_C3IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholdsLT[3], const `Npp16u` rValuesLT[3], const `Npp16u` rThresholdsGT[3], const `Npp16u` rValuesGT[3])

3 channel 16-bit unsigned short in place threshold.

- `NppStatus nppiThreshold_LTValGTVal_16s_C3R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholdsLT[3], const `Npp16s` rValuesLT[3], const `Npp16s` rThresholdsGT[3], const `Npp16s` rValuesGT[3])
3 channel 16-bit signed short threshold.
- `NppStatus nppiThreshold_LTValGTVal_16s_C3IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholdsLT[3], const `Npp16s` rValuesLT[3], const `Npp16s` rThresholdsGT[3], const `Npp16s` rValuesGT[3])
3 channel 16-bit signed short in place threshold.
- `NppStatus nppiThreshold_LTValGTVal_32f_C3R` (const `Npp32f` *pSrc, int nSrcStep, `Npp32f` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholdsLT[3], const `Npp32f` rValuesLT[3], const `Npp32f` rThresholdsGT[3], const `Npp32f` rValuesGT[3])
3 channel 32-bit floating point threshold.
- `NppStatus nppiThreshold_LTValGTVal_32f_C3IR` (`Npp32f` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholdsLT[3], const `Npp32f` rValuesLT[3], const `Npp32f` rThresholdsGT[3], const `Npp32f` rValuesGT[3])
3 channel 32-bit floating point in place threshold.
- `NppStatus nppiThreshold_LTValGTVal_8u_AC4R` (const `Npp8u` *pSrc, int nSrcStep, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholdsLT[3], const `Npp8u` rValuesLT[3], const `Npp8u` rThresholdsGT[3], const `Npp8u` rValuesGT[3])
4 channel 8-bit unsigned char image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTValGTVal_8u_AC4IR` (`Npp8u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholdsLT[3], const `Npp8u` rValuesLT[3], const `Npp8u` rThresholdsGT[3], const `Npp8u` rValuesGT[3])
4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTValGTVal_16u_AC4R` (const `Npp16u` *pSrc, int nSrcStep, `Npp16u` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholdsLT[3], const `Npp16u` rValuesLT[3], const `Npp16u` rThresholdsGT[3], const `Npp16u` rValuesGT[3])
4 channel 16-bit unsigned short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTValGTVal_16u_AC4IR` (`Npp16u` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholdsLT[3], const `Npp16u` rValuesLT[3], const `Npp16u` rThresholdsGT[3], const `Npp16u` rValuesGT[3])
4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTValGTVal_16s_AC4R` (const `Npp16s` *pSrc, int nSrcStep, `Npp16s` *pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholdsLT[3], const `Npp16s` rValuesLT[3], const `Npp16s` rThresholdsGT[3], const `Npp16s` rValuesGT[3])
4 channel 16-bit signed short image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTValGTVal_16s_AC4IR` (`Npp16s` *pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholdsLT[3], const `Npp16s` rValuesLT[3], const `Npp16s` rThresholdsGT[3], const `Npp16s` rValuesGT[3])
4 channel 16-bit signed short in place image threshold, not affecting Alpha.

- **NppStatus** **nppiThreshold_LTValGTVal_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, **Npp32f** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** rThresholdsLT[3], const **Npp32f** rValuesLT[3], const **Npp32f** rThresholdsGT[3], const **Npp32f** rValuesGT[3])

4 channel 32-bit floating point image threshold, not affecting Alpha.

- **NppStatus** **nppiThreshold_LTValGTVal_32f_AC4IR** (**Npp32f** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** rThresholdsLT[3], const **Npp32f** rValuesLT[3], const **Npp32f** rThresholdsGT[3], const **Npp32f** rValuesGT[3])

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

7.139.1 Detailed Description

Threshold image pixels.

7.139.2 Function Documentation

7.139.2.1 **NppStatus nppiThreshold_16s_AC4IR** (**Npp16s** *pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3], **NppCmpOp** eComparisonOperation)

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.2 **NppStatus nppiThreshold_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, **Npp16s** *pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3], **NppCmpOp** eComparisonOperation)

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.3 NppStatus nppiThreshold_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold, NppCmpOp eComparisonOperation)

1 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.4 NppStatus nppiThreshold_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, NppCmpOp eComparisonOperation)

1 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.5 NppStatus nppiThreshold_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], NppCmpOp eComparisonOperation)

3 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.6 NppStatus nppiThreshold_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], NppCmpOp eComparisonOperation)

3 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

7.139.2.7 `NppStatus nppiThreshold_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (`sourcePixel.channel OP nThreshold`) is true, the channel value is set to `nThreshold`, otherwise it is set to `sourcePixel`.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: `NPP_CMP_LESS` and `NPP_CMP_GREATER`.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

7.139.2.8 `NppStatus nppiThreshold_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (`sourcePixel.channel OP nThreshold`) is true, the channel value is set to `nThreshold`, otherwise it is set to `sourcePixel`.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: `NPP_CMP_LESS` and `NPP_CMP_GREATER`.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

7.139.2.9 **NppStatus nppiThreshold_16u_C1IR** (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *nThreshold*, NppCmpOp *eComparisonOperation*)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.10 **NppStatus nppiThreshold_16u_C1R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *nThreshold*, NppCmpOp *eComparisonOperation*)

1 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.11 **NppStatus nppiThreshold_16u_C3IR** (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3], NppCmpOp *eComparisonOperation*)

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.12 **NppStatus nppiThreshold_16u_C3R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3], NppCmpOp *eComparisonOperation*)

3 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.13 **NppStatus nppiThreshold_32f_AC4IR** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3], NppCmpOp *eComparisonOperation*)

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.14 NppStatus nppiThreshold_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], NppCmpOp eComparisonOperation)

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.15 NppStatus nppiThreshold_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold, NppCmpOp eComparisonOperation)

1 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.16 NppStatus nppiThreshold_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold, NppCmpOp eComparisonOperation)

1 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.17 NppStatus nppiThreshold_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], NppCmpOp eComparisonOperation)

3 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.18 NppStatus nppiThreshold_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], NppCmpOp eComparisonOperation)

3 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.19 NppStatus nppiThreshold_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], NppCmpOp eComparisonOperation)

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.20 **NppStatus nppiThreshold_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3], NppCmpOp *eComparisonOperation*)

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.21 **NppStatus nppiThreshold_8u_C1IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*, NppCmpOp *eComparisonOperation*)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.22 **NppStatus nppiThreshold_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*, NppCmpOp *eComparisonOperation*)

1 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.23 **NppStatus nppiThreshold_8u_C3IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3], NppCmpOp *eComparisonOperation*)

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.24 NppStatus nppiThreshold_8u_C3R (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3], NppCmpOp *eComparisonOperation*)

3 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.25 NppStatus nppiThreshold_GT_16s_AC4IR (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.26 NppStatus nppiThreshold_GT_16s_AC4R (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.27 **NppStatus nppiThreshold_GT_16s_C1IR** (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *nThreshold*)

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.28 **NppStatus nppiThreshold_GT_16s_C1R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *nThreshold*)

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.29 **NppStatus nppiThreshold_GT_16s_C3IR** (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.30 **NppStatus nppiThreshold_GT_16s_C3R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.31 **NppStatus nppiThreshold_GT_16u_AC4IR** (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.32 NppStatus nppiThreshold_GT_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3])

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.33 NppStatus nppiThreshold_GT_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.34 **NppStatus nppiThreshold_GT_16u_C1R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *nThreshold*)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.35 **NppStatus nppiThreshold_GT_16u_C3IR** (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.36 **NppStatus nppiThreshold_GT_16u_C3R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.37 NppStatus nppiThreshold_GT_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.38 NppStatus nppiThreshold_GT_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.39 **NppStatus nppiThreshold_GT_32f_C1IR** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *nThreshold*)

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.40 **NppStatus nppiThreshold_GT_32f_C1R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *nThreshold*)

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.41 **NppStatus nppiThreshold_GT_32f_C3IR** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3])

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.42 NppStatus nppiThreshold_GT_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.43 NppStatus nppiThreshold_GT_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.44 **NppStatus nppiThreshold_GT_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3])

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.45 **NppStatus nppiThreshold_GT_8u_C1IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.46 **NppStatus nppiThreshold_GT_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*)

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.47 NppStatus nppiThreshold_GT_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.48 NppStatus nppiThreshold_GT_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.49 **NppStatus nppiThreshold_GTVal_16s_AC4IR** (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3], const Npp16s *rValues*[3])

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.50 **NppStatus nppiThreshold_GTVal_16s_AC4R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3], const Npp16s *rValues*[3])

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.51 **NppStatus nppiThreshold_GTVal_16s_C1IR** (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *nThreshold*, const Npp16s *nValue*)

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.
nValue The threshold replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.52 `NppStatus nppiThreshold_GTVVal_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue)`

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.
nValue The threshold replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.53 `NppStatus nppiThreshold_GTVVal_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.
rValues The threshold replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.54 `NppStatus nppiThreshold_GTVVal_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.55 `NppStatus nppiThreshold_GTVVal_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.56 `NppStatus nppiThreshold_GTVVal_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.
rValues The threshold replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.57 **NppStatus nppiThreshold_GTVa16u_C1IR** (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *nThreshold*, const Npp16u *nValue*)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.
nValue The threshold replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.58 **NppStatus nppiThreshold_GTVa16u_C1R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *nThreshold*, const Npp16u *nValue*)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

nValue The threshold replacement value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.59 `NppStatus nppiThreshold_GTVal_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.60 `NppStatus nppiThreshold_GTVal_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.61 **NppStatus nppiThreshold_GTVAl_32f_AC4IR** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3], const Npp32f *rValues*[3])

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.62 **NppStatus nppiThreshold_GTVAl_32f_AC4R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3], const Npp32f *rValues*[3])

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.63 **NppStatus nppiThreshold_GTVAl_32f_C1IR** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *nThreshold*, const Npp32f *nValue*)

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.
nValue The threshold replacement values.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.64 `NppStatus nppiThreshold_GTVVal_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue)`

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.
nValue The threshold replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.65 `NppStatus nppiThreshold_GTVVal_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.
rValues The threshold replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.66 `NppStatus nppiThreshold_GTVa1_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.67 `NppStatus nppiThreshold_GTVa1_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.68 `NppStatus nppiThreshold_GTVa1_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.
rValues The threshold replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.69 `NppStatus nppiThreshold_GTVal_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue)`

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.
nValue The threshold replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.70 `NppStatus nppiThreshold_GTVal_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

nValue The threshold replacement value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.71 `NppStatus nppiThreshold_GTVal_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.72 `NppStatus nppiThreshold_GTVal_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.73 **NppStatus nppiThreshold_LT_16s_AC4IR** (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.74 **NppStatus nppiThreshold_LT_16s_AC4R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.75 **NppStatus nppiThreshold_LT_16s_C1IR** (Npp16s * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *nThreshold*)

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.76 NppStatus nppiThreshold_LT_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold)

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.77 NppStatus nppiThreshold_LT_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3])

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.78 **NppStatus nppiThreshold_LT_16s_C3R** (const Npp16s * *pSrc*, int *nSrcStep*, Npp16s * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.79 **NppStatus nppiThreshold_LT_16u_AC4IR** (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.80 **NppStatus nppiThreshold_LT_16u_AC4R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.81 NppStatus nppiThreshold_LT_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.82 NppStatus nppiThreshold_LT_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.83 **NppStatus nppiThreshold_LT_16u_C3IR** (Npp16u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.84 **NppStatus nppiThreshold_LT_16u_C3R** (const Npp16u * *pSrc*, int *nSrcStep*, Npp16u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.85 **NppStatus nppiThreshold_LT_32f_AC4IR** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3])

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.86 NppStatus nppiThreshold_LT_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.87 NppStatus nppiThreshold_LT_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold)

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.88 **NppStatus nppiThreshold_LT_32f_C1R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *nThreshold*)

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.89 **NppStatus nppiThreshold_LT_32f_C3IR** (Npp32f * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3])

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.90 **NppStatus nppiThreshold_LT_32f_C3R** (const Npp32f * *pSrc*, int *nSrcStep*, Npp32f * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3])

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.91 **NppStatus nppiThreshold_LT_8u_AC4IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3])

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.92 **NppStatus nppiThreshold_LT_8u_AC4R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3])

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.93 **NppStatus nppiThreshold_LT_8u_C1IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.94 **NppStatus nppiThreshold_LT_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*)

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.95 **NppStatus nppiThreshold_LT_8u_C3IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3])

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.96 `NppStatus nppiThreshold_LT_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])`

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.97 `NppStatus nppiThreshold_LTVal_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.98 `NppStatus nppiThreshold_LTVal_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.99 `NppStatus nppiThreshold_LTVal_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue)`

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

nValue The threshold replacement value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.100 `NppStatus nppiThreshold_LTVal_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue)`

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.
nValue The threshold replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.101 `NppStatus nppiThreshold_LTVal_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.
rValues The threshold replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.102 `NppStatus nppiThreshold_LTVal_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.103 `NppStatus nppiThreshold_LTVal_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.104 `NppStatus nppiThreshold_LTVal_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.105 NppStatus nppiThreshold_LTVal_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

nValue The threshold replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.106 NppStatus nppiThreshold_LTVal_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

nValue The threshold replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.107 NppStatus nppiThreshold_LTVal_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.
rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.108 `NppStatus nppiThreshold_LTVAl_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.
rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.109 `NppStatus nppiThreshold_LTVAl_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.
rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.110 `NppStatus nppiThreshold_LTVal_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.111 `NppStatus nppiThreshold_LTVal_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

nValue The threshold replacement value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.112 `NppStatus nppiThreshold_LTVal_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue)`

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThreshold The threshold value.
nValue The threshold replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.113 `NppStatus nppiThreshold_LTVAl_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.
rValues The threshold replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.114 `NppStatus nppiThreshold_LTVAl_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.115 `NppStatus nppiThreshold_LTVal_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.116 `NppStatus nppiThreshold_LTVal_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.117 **NppStatus nppiThreshold_LTVal_8u_C1IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*, const Npp8u *nValue*)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

nValue The threshold replacement value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.118 **NppStatus nppiThreshold_LTVal_8u_C1R** (const Npp8u * *pSrc*, int *nSrcStep*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*, const Npp8u *nValue*)

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

nValue The threshold replacement value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.119 **NppStatus nppiThreshold_LTVal_8u_C3IR** (Npp8u * *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3], const Npp8u *rValues*[3])

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.
rValues The threshold replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.120 `NppStatus nppiThreshold_LTVal_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholds The threshold values, one per color channel.
rValues The threshold replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.121 `NppStatus nppiThreshold_LTValGTVal_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.
nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.122 `NppStatus nppiThreshold_LTValGTVal_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])`

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.123 `NppStatus nppiThreshold_LTValGTVal_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThresholdLT, const Npp16s nValueLT, const Npp16s nThresholdGT, const Npp16s nValueGT)`

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThresholdLT The thresholdLT value.

nValueLT The thresholdLT replacement value.

nThresholdGT The thresholdGT value.

nValueGT The thresholdGT replacement value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.124 `NppStatus nppiThreshold_LTValGTVal_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThresholdLT, const Npp16s nValueLT, const Npp16s nThresholdGT, const Npp16s nValueGT)`

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThresholdLT The thresholdLT value.

nValueLT The thresholdLT replacement value.

nThresholdGT The thresholdGT value.

nValueGT The thresholdGT replacement value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.125 `NppStatus nppiThreshold_LTValGTVal_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])`

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.126 `NppStatus nppiThreshold_LTValGTVal_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])`

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.127 `NppStatus nppiThreshold_LTValGTVal_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.128 `NppStatus nppiThreshold_LTValGTVal_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.129 `NppStatus nppiThreshold_LTValGTVal_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThresholdLT, const Npp16u nValueLT, const Npp16u nThresholdGT, const Npp16u nValueGT)`

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nThresholdLT The thresholdLT value.
nValueLT The thresholdLT replacement value.
nThresholdGT The thresholdGT value.
nValueGT The thresholdGT replacement value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.130 `NppStatus nppiThreshold_LTValGTVal_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThresholdLT, const Npp16u nValueLT, const Npp16u nThresholdGT, const Npp16u nValueGT)`

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
nThresholdLT The thresholdLT value.
nValueLT The thresholdLT replacement value.
nThresholdGT The thresholdGT value.
nValueGT The thresholdGT replacement value.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

7.139.2.131 `NppStatus nppiThreshold_LTValGTVal_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.132 `NppStatus nppiThreshold_LTValGTVal_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.133 `NppStatus nppiThreshold_LTValGTVal_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.134 `NppStatus nppiThreshold_LTValGTVal_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.135 `NppStatus nppiThreshold_LTValGTVal_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThresholdLT, const Npp32f nValueLT, const Npp32f nThresholdGT, const Npp32f nValueGT)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThresholdLT The thresholdLT value.
nValueLT The thresholdLT replacement value.
nThresholdGT The thresholdGT value.
nValueGT The thresholdGT replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.136 `NppStatus nppiThreshold_LTValGTVal_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThresholdLT, const Npp32f nValueLT, const Npp32f nThresholdGT, const Npp32f nValueGT)`

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThresholdLT The thresholdLT value.
nValueLT The thresholdLT replacement value.
nThresholdGT The thresholdGT value.
nValueGT The thresholdGT replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.137 `NppStatus nppiThreshold_LTValGTVal_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.138 `NppStatus nppiThreshold_LTValGTVal_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.139 `NppStatus nppiThreshold_LTValGTVal_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.140 `NppStatus nppiThreshold_LTValGTVal_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.141 `NppStatus nppiThreshold_LTValGTVal_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThresholdLT, const Npp8u nValueLT, const Npp8u nThresholdGT, const Npp8u nValueGT)`

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep In-Place-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThresholdLT The thresholdLT value.
nValueLT The thresholdLT replacement value.
nThresholdGT The thresholdGT value.
nValueGT The thresholdGT replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.142 `NppStatus nppiThreshold_LTValGTVal_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThresholdLT, const Npp8u nValueLT, const Npp8u nThresholdGT, const Npp8u nValueGT)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nThresholdLT The thresholdLT value.
nValueLT The thresholdLT replacement value.
nThresholdGT The thresholdGT value.
nValueGT The thresholdGT replacement value.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.143 `NppStatus nppiThreshold_LTValGTVal_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrcDst Destination-Image Pointer.

nSrcDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.144 `NppStatus nppiThreshold_LTValGTVal_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholdsLT The thresholdLT values, one per color channel.

rValuesLT The thresholdLT replacement values, one per color channel.

rThresholdsGT The thresholdGT values, one per channel.

rValuesGT The thresholdGT replacement values, one per color channel.

Returns:

Image Data Related Error Codes, ROI Related Error Codes.

7.139.2.145 `NppStatus nppiThreshold_Val_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.146 `NppStatus nppiThreshold_Val_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.147 `NppStatus nppiThreshold_Val_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue, NppCmpOp eComparisonOperation)`

1 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

nValue The threshold replacement value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.148 `NppStatus nppiThreshold_Val_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue, NppCmpOp eComparisonOperation)`

1 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

nValue The threshold replacement value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.149 `NppStatus nppiThreshold_Val_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.150 `NppStatus nppiThreshold_Val_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.151 `NppStatus nppiThreshold_Val_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.152 `NppStatus npptThreshold_Val_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.153 `NppStatus npptThreshold_Val_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue, NppCmpOp eComparisonOperation)`

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

nValue The threshold replacement value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.154 `NppStatus nppiThreshold_Val_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue, NppCmpOp eComparisonOperation)`

1 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

nValue The threshold replacement value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.155 `NppStatus nppiThreshold_Val_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.156 `NppStatus nppiThreshold_Val_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.157 `NppStatus nppiThreshold_Val_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)`

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.158 `NppStatus nppiThreshold_Val_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)`

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.159 `NppStatus nppiThreshold_Val_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue, NppCmpOp eComparisonOperation)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

nValue The threshold replacement value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.160 `NppStatus nppiThreshold_Val_32f_C1R (const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue, NppCmpOp eComparisonOperation)`

1 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

nThreshold The threshold value.

nValue The threshold replacement value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.161 `NppStatus nppiThreshold_Val_32f_C3IR (Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)`

3 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.162 `NppStatus nppiThreshold_Val_32f_C3R (const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)`

3 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.163 `NppStatus nppiThreshold_Val_8u_AC4IR (Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.164 `NppStatus nppiThreshold_Val_8u_AC4R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.165 `NppStatus nppiThreshold_Val_8u_C1IR (Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue, NppCmpOp eComparisonOperation)`

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst [In-Place Image Pointer](#).

nSrcDstStep In-Place-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

nValue The threshold replacement value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.166 `NppStatus nppiThreshold_Val_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue, NppCmpOp eComparisonOperation)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nThreshold The threshold value.

nValue The threshold replacement value.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.167 `NppStatus nppiThreshold_Val_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrcDst In-Place Image Pointer.

nSrcDstStep [In-Place-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.139.2.168 `NppStatus nppiThreshold_Val_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

rThresholds The threshold values, one per color channel.

rValues The threshold replacement values, one per color channel.

eComparisonOperation The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

7.140 Compare Operations

Compare the pixels of two images and create a binary result image.

Functions

- `NppStatus nppiCompare_8u_C1R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppCmpOp` eComparisonOperation)
1 channel 8-bit unsigned char image compare.
- `NppStatus nppiCompare_8u_C3R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppCmpOp` eComparisonOperation)
3 channel 8-bit unsigned char image compare.
- `NppStatus nppiCompare_8u_C4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppCmpOp` eComparisonOperation)
4 channel 8-bit unsigned char image compare.
- `NppStatus nppiCompare_8u_AC4R` (const `Npp8u` *pSrc1, int nSrc1Step, const `Npp8u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppCmpOp` eComparisonOperation)
4 channel 8-bit unsigned char image compare, not affecting Alpha.
- `NppStatus nppiCompare_16u_C1R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppCmpOp` eComparisonOperation)
1 channel 16-bit unsigned short image compare.
- `NppStatus nppiCompare_16u_C3R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppCmpOp` eComparisonOperation)
3 channel 16-bit unsigned short image compare.
- `NppStatus nppiCompare_16u_C4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppCmpOp` eComparisonOperation)
4 channel 16-bit unsigned short image compare.
- `NppStatus nppiCompare_16u_AC4R` (const `Npp16u` *pSrc1, int nSrc1Step, const `Npp16u` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppCmpOp` eComparisonOperation)
4 channel 16-bit unsigned short image compare, not affecting Alpha.
- `NppStatus nppiCompare_16s_C1R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppCmpOp` eComparisonOperation)
1 channel 16-bit signed short image compare.
- `NppStatus nppiCompare_16s_C3R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppCmpOp` eComparisonOperation)
3 channel 16-bit signed short image compare.
- `NppStatus nppiCompare_16s_C4R` (const `Npp16s` *pSrc1, int nSrc1Step, const `Npp16s` *pSrc2, int nSrc2Step, `Npp8u` *pDst, int nDstStep, `NppiSize` oSizeROI, `NppCmpOp` eComparisonOperation)
4 channel 16-bit signed short image compare.

- **NppStatus nppiCompare_16s_AC4R** (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
4 channel 16-bit signed short image compare, not affecting Alpha.
- **NppStatus nppiCompare_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
1 channel 32-bit floating point image compare.
- **NppStatus nppiCompare_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
3 channel 32-bit floating point image compare.
- **NppStatus nppiCompare_32f_C4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
4 channel 32-bit floating point image compare.
- **NppStatus nppiCompare_32f_AC4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
4 channel 32-bit signed floating point compare, not affecting Alpha.
- **NppStatus nppiCompareC_8u_C1R** (const **Npp8u** *pSrc, int nSrcStep, const **Npp8u** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
1 channel 8-bit unsigned char image compare with constant value.
- **NppStatus nppiCompareC_8u_C3R** (const **Npp8u** *pSrc, int nSrcStep, const **Npp8u** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
3 channel 8-bit unsigned char image compare with constant value.
- **NppStatus nppiCompareC_8u_C4R** (const **Npp8u** *pSrc, int nSrcStep, const **Npp8u** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
4 channel 8-bit unsigned char image compare with constant value.
- **NppStatus nppiCompareC_8u_AC4R** (const **Npp8u** *pSrc, int nSrcStep, const **Npp8u** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
4 channel 8-bit unsigned char image compare, not affecting Alpha.
- **NppStatus nppiCompareC_16u_C1R** (const **Npp16u** *pSrc, int nSrcStep, const **Npp16u** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
1 channel 16-bit unsigned short image compare with constant value.
- **NppStatus nppiCompareC_16u_C3R** (const **Npp16u** *pSrc, int nSrcStep, const **Npp16u** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
3 channel 16-bit unsigned short image compare with constant value.
- **NppStatus nppiCompareC_16u_C4R** (const **Npp16u** *pSrc, int nSrcStep, const **Npp16u** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
4 channel 16-bit unsigned short image compare with constant value.

- **NppStatus** **nppiCompareC_16u_AC4R** (const **Npp16u** *pSrc, int nSrcStep, const **Npp16u** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
4 channel 16-bit unsigned short image compare, not affecting Alpha.
- **NppStatus** **nppiCompareC_16s_C1R** (const **Npp16s** *pSrc, int nSrcStep, const **Npp16s** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
1 channel 16-bit signed short image compare with constant value.
- **NppStatus** **nppiCompareC_16s_C3R** (const **Npp16s** *pSrc, int nSrcStep, const **Npp16s** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
3 channel 16-bit signed short image compare with constant value.
- **NppStatus** **nppiCompareC_16s_C4R** (const **Npp16s** *pSrc, int nSrcStep, const **Npp16s** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
4 channel 16-bit signed short image compare with constant value.
- **NppStatus** **nppiCompareC_16s_AC4R** (const **Npp16s** *pSrc, int nSrcStep, const **Npp16s** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
4 channel 16-bit signed short image compare, not affecting Alpha.
- **NppStatus** **nppiCompareC_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, const **Npp32f** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
1 channel 32-bit floating point image compare with constant value.
- **NppStatus** **nppiCompareC_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, const **Npp32f** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
3 channel 32-bit floating point image compare with constant value.
- **NppStatus** **nppiCompareC_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, const **Npp32f** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
4 channel 32-bit floating point image compare with constant value.
- **NppStatus** **nppiCompareC_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, const **Npp32f** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)
4 channel 32-bit signed floating point compare, not affecting Alpha.
- **NppStatus** **nppiCompareEqualEps_32f_C1R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)
1 channel 32-bit floating point image compare whether two images are equal within epsilon.
- **NppStatus** **nppiCompareEqualEps_32f_C3R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)
3 channel 32-bit floating point image compare whether two images are equal within epsilon.
- **NppStatus** **nppiCompareEqualEps_32f_C4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)
4 channel 32-bit floating point image compare whether two images are equal within epsilon.

- **NppStatus nppiCompareEqualEps_32f_AC4R** (const **Npp32f** *pSrc1, int nSrc1Step, const **Npp32f** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)
4 channel 32-bit signed floating point compare whether two images are equal within epsilon, not affecting Alpha.
- **NppStatus nppiCompareEqualEpsC_32f_C1R** (const **Npp32f** *pSrc, int nSrcStep, const **Npp32f** nConstant, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)
1 channel 32-bit floating point image compare whether image and constant are equal within epsilon.
- **NppStatus nppiCompareEqualEpsC_32f_C3R** (const **Npp32f** *pSrc, int nSrcStep, const **Npp32f** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)
3 channel 32-bit floating point image compare whether image and constant are equal within epsilon.
- **NppStatus nppiCompareEqualEpsC_32f_C4R** (const **Npp32f** *pSrc, int nSrcStep, const **Npp32f** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)
4 channel 32-bit floating point image compare whether image and constant are equal within epsilon.
- **NppStatus nppiCompareEqualEpsC_32f_AC4R** (const **Npp32f** *pSrc, int nSrcStep, const **Npp32f** *pConstants, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)
4 channel 32-bit signed floating point compare whether image and constant are equal within epsilon, not affecting Alpha.

7.140.1 Detailed Description

Compare the pixels of two images and create a binary result image.

In case of multi-channel image types, the condition must be fulfilled for all channels, otherwise the comparison is considered false. The "binary" result image is of type 8u_C1. False is represented by 0, true by NPP_MAX_8U.

7.140.2 Function Documentation

7.140.2.1 NppStatus nppiCompare_16s_AC4R (const **Npp16s** *pSrc1, int nSrc1Step, const **Npp16s** *pSrc2, int nSrc2Step, **Npp8u** *pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)

4 channel 16-bit signed short image compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

- pSrc1** Source-Image Pointer.
- nSrc1Step** Source-Image Line Step.
- pSrc2** Source-Image Pointer.
- nSrc2Step** Source-Image Line Step.
- pDst** Destination-Image Pointer.
- nDstStep** Destination-Image Line Step.
- oSizeROI** Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.2 `NppStatus nppiCompare_16s_C1R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 16-bit signed short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.3 `NppStatus nppiCompare_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

3 channel 16-bit signed short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.4 NppStatus nppiCompare_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)

4 channel 16-bit signed short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.5 NppStatus nppiCompare_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)

4 channel 16-bit unsigned short image compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.6 NppStatus nppiCompare_16u_C1R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

1 channel 16-bit unsigned short image compare.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2*.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.7 NppStatus nppiCompare_16u_C3R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

3 channel 16-bit unsigned short image compare.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2*.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.8 NppStatus nppiCompare_16u_C4R (const Npp16u * *pSrc1*, int *nSrc1Step*, const Npp16u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

4 channel 16-bit unsigned short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.9 NppStatus nppiCompare_32f_AC4R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

4 channel 32-bit signed floating point compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.10 `NppStatus nppiCompare_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.11 `NppStatus nppiCompare_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

3 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.12 NppStatus nppiCompare_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)

4 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.13 NppStatus nppiCompare_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)

4 channel 8-bit unsigned char image compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.14 NppStatus nppiCompare_8u_C1R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

1 channel 8-bit unsigned char image compare.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2*.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.15 NppStatus nppiCompare_8u_C3R (const Npp8u * *pSrc1*, int *nSrc1Step*, const Npp8u * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

3 channel 8-bit unsigned char image compare.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2*.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.16 `NppStatus nppiCompare_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

Parameters:

pSrc1 [Source-Image Pointer](#).

nSrc1Step [Source-Image Line Step](#).

pSrc2 [Source-Image Pointer](#).

nSrc2Step [Source-Image Line Step](#).

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.17 `NppStatus nppiCompareC_16s_AC4R (const Npp16s * pSrc, int nSrcStep, const Npp16s * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 16-bit signed short image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pConstants pointer to a list of constants, one per color channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.18 `NppStatus nppiCompareC_16s_C1R (const Npp16s * pSrc, int nSrcStep, const Npp16s nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
nConstant constant value.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.19 `NppStatus nppiCompareC_16s_C3R (const Npp16s * pSrc, int nSrcStep, const Npp16s * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

3 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pConstants pointer to a list of constants, one per color channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.20 `NppStatus nppiCompareC_16s_C4R (const Npp16s * pSrc, int nSrcStep, const Npp16s * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pConstants pointer to a list of constants, one per color channel.
pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.140.2.21 `NppStatus nppiCompareC_16u_AC4R (const Npp16u * pSrc, int nSrcStep, const Npp16u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pConstants pointer to a list of constants, one per color channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.140.2.22 `NppStatus nppiCompareC_16u_C1R (const Npp16u * pSrc, int nSrcStep, const Npp16u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 16-bit unsigned short image compare with constant value.

Compare pSrc's pixels with constant value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

nConstant constant value

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.140.2.23 **NppStatus nppiCompareC_16u_C3R** (const Npp16u * *pSrc*, int *nSrcStep*, const Npp16u * *pConstants*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

3 channel 16-bit unsigned short image compare with constant value.

Compare *pSrc*'s pixels with constant value.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pConstants pointer to a list of constants, one per color channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.24 **NppStatus nppiCompareC_16u_C4R** (const Npp16u * *pSrc*, int *nSrcStep*, const Npp16u * *pConstants*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

4 channel 16-bit unsigned short image compare with constant value.

Compare *pSrc*'s pixels with constant value.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pConstants pointer to a list of constants, one per color channel.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.25 **NppStatus nppiCompareC_32f_AC4R** (const Npp32f * *pSrc*, int *nSrcStep*, const Npp32f * *pConstants*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

4 channel 32-bit signed floating point compare, not affecting Alpha.

Compare *pSrc*'s pixels with constant value.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pConstants pointer to a list of constants, one per color channel.
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.26 **NppStatus nppiCompareC_32f_C1R** (const Npp32f * *pSrc*, int *nSrcStep*, const Npp32f *nConstant*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

1 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
nConstant constant value
pDst [Destination-Image Pointer](#).
nDstStep [Destination-Image Line Step](#).
oSizeROI [Region-of-Interest \(ROI\)](#).
eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.27 **NppStatus nppiCompareC_32f_C3R** (const Npp32f * *pSrc*, int *nSrcStep*, const Npp32f * *pConstants*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

3 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

Parameters:

pSrc [Source-Image Pointer](#).
nSrcStep [Source-Image Line Step](#).
pConstants pointer to a list of constants, one per color channel.
pDst [Destination-Image Pointer](#).

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.140.2.28 `NppStatus nppiCompareC_32f_C4R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pConstants pointer to a list of constants, one per color channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.140.2.29 `NppStatus nppiCompareC_8u_AC4R (const Npp8u * pSrc, int nSrcStep, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pConstants pointer to a list of constants, one per color channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.140.2.30 NppStatus nppiCompareC_8u_C1R (const Npp8u * pSrc, int nSrcStep, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)

1 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

nConstant constant value.

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.31 NppStatus nppiCompareC_8u_C3R (const Npp8u * pSrc, int nSrcStep, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)

3 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

Parameters:

pSrc [Source-Image Pointer](#).

nSrcStep [Source-Image Line Step](#).

pConstants pointer to a list of constant values, one per color channel..

pDst [Destination-Image Pointer](#).

nDstStep [Destination-Image Line Step](#).

oSizeROI [Region-of-Interest \(ROI\)](#).

eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.32 NppStatus nppiCompareC_8u_C4R (const Npp8u * pSrc, int nSrcStep, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)

4 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pConstants pointer to a list of constants, one per color channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
eComparisonOperation Specifies the comparison operation to be used in the pixel comparison.

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.33 `NppStatus nppiCompareEqualEps_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

4 channel 32-bit signed floating point compare whether two images are equal within epsilon, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nEpsilon epsilon tolerance value to compare to per color channel pixel absolute differences

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.34 `NppStatus nppiCompareEqualEps_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

1 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

Parameters:

pSrc1 Source-Image Pointer.

nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nEpsilon epsilon tolerance value to compare to pixel absolute differences

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.140.2.35 NppStatus nppiCompareEqualEps_32f_C3R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32f *nEpsilon*)

3 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2* to determine whether they are equal with a difference of epsilon.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.
pSrc2 Source-Image Pointer.
nSrc2Step Source-Image Line Step.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nEpsilon epsilon tolerance value to compare to per color channel pixel absolute differences

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.140.2.36 NppStatus nppiCompareEqualEps_32f_C4R (const Npp32f * *pSrc1*, int *nSrc1Step*, const Npp32f * *pSrc2*, int *nSrc2Step*, Npp8u * *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32f *nEpsilon*)

4 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2* to determine whether they are equal with a difference of epsilon.

Parameters:

pSrc1 Source-Image Pointer.
nSrc1Step Source-Image Line Step.

pSrc2 Source-Image Pointer.

nSrc2Step Source-Image Line Step.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nEpsilon epsilon tolerance value to compare to per color channel pixel absolute differences

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.37 `NppStatus nppiCompareEqualEpsC_32f_AC4R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

4 channel 32-bit signed floating point compare whether image and constant are equal within epsilon, not affecting Alpha.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

pConstants pointer to a list of constants, one per color channel.

pDst Destination-Image Pointer.

nDstStep Destination-Image Line Step.

oSizeROI Region-of-Interest (ROI).

nEpsilon epsilon tolerance value to compare to per color channel pixel absolute differences

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.140.2.38 `NppStatus nppiCompareEqualEpsC_32f_C1R (const Npp32f * pSrc, int nSrcStep, const Npp32f nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

1 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

Parameters:

pSrc Source-Image Pointer.

nSrcStep Source-Image Line Step.

nConstant constant value

pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nEpsilon epsilon tolerance value to compare to pixel absolute differences

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.140.2.39 `NppStatus nppiCompareEqualEpsC_32f_C3R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

3 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pConstants pointer to a list of constants, one per color channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).
nEpsilon epsilon tolerance value to compare to per color channel pixel absolute differences

Returns:

Image Data Related Error Codes, ROI Related Error Codes

7.140.2.40 `NppStatus nppiCompareEqualEpsC_32f_C4R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

4 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

Parameters:

pSrc Source-Image Pointer.
nSrcStep Source-Image Line Step.
pConstants pointer to a list of constants, one per color channel.
pDst Destination-Image Pointer.
nDstStep Destination-Image Line Step.
oSizeROI Region-of-Interest (ROI).

nEpsilon epsilon tolerance value to compare to per color channel pixel absolute differences

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

7.141 NPP Signal Processing

Modules

- [Arithmetic and Logical Operations](#)
- [Conversion Functions](#)
- [Filtering Functions](#)

Functions that provide functionality of generating output signal based on the input signal like signal integral, etc.

- [Initialization](#)
- [Statistical Functions](#)

Functions that provide global signal statistics like: sum, mean, standard deviation, min, max, etc.

- [Memory Management](#)

7.142 Arithmetic and Logical Operations

Modules

- [Arithmetic Operations](#)
- [Logical And Shift Operations](#)

7.143 Arithmetic Operations

Modules

- [AddC](#)
Adds a constant value to each sample of a signal.
- [AddProductC](#)
Adds product of a constant and each sample of a source signal to the each sample of destination signal.
- [MulC](#)
Multiplies each sample of a signal by a constant value.
- [SubC](#)
Subtracts a constant from each sample of a signal.
- [SubCRev](#)
Subtracts each sample of a signal from a constant.
- [DivC](#)
Divides each sample of a signal by a constant.
- [DivCRev](#)
Divides a constant by each sample of a signal.
- [Add](#)
Sample by sample addition of two signals.
- [AddProduct](#)
Adds sample by sample product of two signals to the destination signal.
- [Mul](#)
Sample by sample multiplication the samples of two signals.
- [Sub](#)
Sample by sample subtraction of the samples of two signals.
- [Div](#)
Sample by sample division of the samples of two signals.
- [Div_Round](#)
Sample by sample division of the samples of two signals with rounding.
- [Abs](#)
Absolute value of each sample of a signal.
- [Sqr](#)
Squares each sample of a signal.
- [Sqrt](#)

Square root of each sample of a signal.

- [Cubrt](#)

Cube root of each sample of a signal.

- [Exp](#)

E raised to the power of each sample of a signal.

- [Ln](#)

Natural logarithm of each sample of a signal.

- [10Log10](#)

Ten times the decimal logarithm of each sample of a signal.

- [SumLn](#)

Sums up the natural logarithm of each sample of a signal.

- [Arctan](#)

Inverse tangent of each sample of a signal.

- [Normalize](#)

Normalize each sample of a real or complex signal using offset and division operations.

- [Cauchy, CauchyD, and CauchyDD2](#)

Determine Cauchy robust error function and its first and second derivatives for each sample of a signal.

7.144 AddC

Adds a constant value to each sample of a signal.

Functions

- **NppStatus nppsAddC_8u_ISfs** (**Npp8u** nValue, **Npp8u** *pSrcDst, int nLength, int nScaleFactor)
8-bit unsigned char in place signal add constant, scale, then clamp to saturated value
- **NppStatus nppsAddC_8u_Sfs** (const **Npp8u** *pSrc, **Npp8u** nValue, **Npp8u** *pDst, int nLength, int nScaleFactor)
8-bit unsigned charvector add constant, scale, then clamp to saturated value.
- **NppStatus nppsAddC_16u_ISfs** (**Npp16u** nValue, **Npp16u** *pSrcDst, int nLength, int nScaleFactor)
16-bit unsigned short in place signal add constant, scale, then clamp to saturated value.
- **NppStatus nppsAddC_16u_Sfs** (const **Npp16u** *pSrc, **Npp16u** nValue, **Npp16u** *pDst, int nLength, int nScaleFactor)
16-bit unsigned short vector add constant, scale, then clamp to saturated value.
- **NppStatus nppsAddC_16s_ISfs** (**Npp16s** nValue, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)
16-bit signed short in place signal add constant, scale, then clamp to saturated value.
- **NppStatus nppsAddC_16s_Sfs** (const **Npp16s** *pSrc, **Npp16s** nValue, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal add constant, scale, then clamp to saturated value.
- **NppStatus nppsAddC_16sc_ISfs** (**Npp16sc** nValue, **Npp16sc** *pSrcDst, int nLength, int nScaleFactor)
16-bit integer complex number (16 bit real, 16 bit imaginary)signal add constant, scale, then clamp to saturated value.
- **NppStatus nppsAddC_16sc_Sfs** (const **Npp16sc** *pSrc, **Npp16sc** nValue, **Npp16sc** *pDst, int nLength, int nScaleFactor)
16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.
- **NppStatus nppsAddC_32s_ISfs** (**Npp32s** nValue, **Npp32s** *pSrcDst, int nLength, int nScaleFactor)
32-bit signed integer in place signal add constant and scale.
- **NppStatus nppsAddC_32s_Sfs** (const **Npp32s** *pSrc, **Npp32s** nValue, **Npp32s** *pDst, int nLength, int nScaleFactor)
32-bit signed integersignal add constant and scale.
- **NppStatus nppsAddC_32sc_ISfs** (**Npp32sc** nValue, **Npp32sc** *pSrcDst, int nLength, int nScaleFactor)
32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal add constant and scale.
- **NppStatus nppsAddC_32sc_Sfs** (const **Npp32sc** *pSrc, **Npp32sc** nValue, **Npp32sc** *pDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal add constant and scale.

- **NppStatus nppsAddC_32f_I** (**Npp32f** nValue, **Npp32f** *pSrcDst, int nLength)
32-bit floating point in place signal add constant.
- **NppStatus nppsAddC_32f** (const **Npp32f** *pSrc, **Npp32f** nValue, **Npp32f** *pDst, int nLength)
32-bit floating point signal add constant.
- **NppStatus nppsAddC_32fc_I** (**Npp32fc** nValue, **Npp32fc** *pSrcDst, int nLength)
32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal add constant.
- **NppStatus nppsAddC_32fc** (const **Npp32fc** *pSrc, **Npp32fc** nValue, **Npp32fc** *pDst, int nLength)
32-bit floating point complex number (32 bit real, 32 bit imaginary) signal add constant.
- **NppStatus nppsAddC_64f_I** (**Npp64f** nValue, **Npp64f** *pSrcDst, int nLength)
64-bit floating point, in place signal add constant.
- **NppStatus nppsAddC_64f** (const **Npp64f** *pSrc, **Npp64f** nValue, **Npp64f** *pDst, int nLength)
64-bit floating pointsignal add constant.
- **NppStatus nppsAddC_64fc_I** (**Npp64fc** nValue, **Npp64fc** *pSrcDst, int nLength)
64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal add constant.
- **NppStatus nppsAddC_64fc** (const **Npp64fc** *pSrc, **Npp64fc** nValue, **Npp64fc** *pDst, int nLength)
64-bit floating point complex number (64 bit real, 64 bit imaginary) signal add constant.

7.144.1 Detailed Description

Adds a constant value to each sample of a signal.

7.144.2 Function Documentation

7.144.2.1 **NppStatus nppsAddC_16s_ISfs** (**Npp16s** nValue, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal add constant, scale, then clamp to saturated value.

Parameters:

- pSrcDst** In-Place Signal Pointer.
nValue Constant value to be added to each vector element
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.144.2.2 NppStatus nppsAddC_16s_Sfs (const Npp16s * *pSrc*, Npp16s *nValue*, Npp16s * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal add constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).
nValue Constant value to be added to each vector element
pDst [Destination Signal Pointer](#).
nLength [Signal Length](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.3 NppStatus nppsAddC_16sc_ISfs (Npp16sc *nValue*, Npp16sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.

Parameters:

pSrcDst [In-Place Signal Pointer](#).
nValue Constant value to be added to each vector element
nLength [Signal Length](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.4 NppStatus nppsAddC_16sc_Sfs (const Npp16sc * *pSrc*, Npp16sc *nValue*, Npp16sc * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).
nValue Constant value to be added to each vector element
pDst [Destination Signal Pointer](#).
nLength [Signal Length](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.5 NppStatus nppsAddC_16u_ISfs (Npp16u *nValue*, Npp16u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal add constant, scale, then clamp to saturated value.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be added to each vector element

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.6 NppStatus nppsAddC_16u_Sfs (const Npp16u * *pSrc*, Npp16u *nValue*, Npp16u * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short vector add constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be added to each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.7 NppStatus nppsAddC_32f (const Npp32f * *pSrc*, Npp32f *nValue*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal add constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be added to each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.8 NppStatus nppsAddC_32f_I (Npp32f nValue, Npp32f * pSrcDst, int nLength)

32-bit floating point in place signal add constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be added to each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.9 NppStatus nppsAddC_32fc (const Npp32fc * pSrc, Npp32fc nValue, Npp32fc * pDst, int nLength)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal add constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be added to each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.10 NppStatus nppsAddC_32fc_I (Npp32fc nValue, Npp32fc * pSrcDst, int nLength)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal add constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be added to each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.11 NppStatus nppsAddC_32s_ISfs (Npp32s nValue, Npp32s * pSrcDst, int nLength, int nScaleFactor)

32-bit signed integer in place signal add constant and scale.

Parameters:

pSrcDst In-Place Signal Pointer.
nValue Constant value to be added to each vector element
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.144.2.12 NppStatus nppsAddC_32s_Sfs (const Npp32s *pSrc, Npp32s nValue, Npp32s *pDst, int nLength, int nScaleFactor)

32-bit signed integersignal add constant and scale.

Parameters:

pSrc Source Signal Pointer.
nValue Constant value to be added to each vector element
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.144.2.13 NppStatus nppsAddC_32sc_ISfs (Npp32sc nValue, Npp32sc *pSrcDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal add constant and scale.

Parameters:

pSrcDst In-Place Signal Pointer.
nValue Constant value to be added to each vector element
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.144.2.14 NppStatus nppsAddC_32sc_Sfs (const Npp32sc * *pSrc*, Npp32sc *nValue*, Npp32sc * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal add constant and scale.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be added to each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.144.2.15 NppStatus nppsAddC_64f (const Npp64f * *pSrc*, Npp64f *nValue*, Npp64f * *pDst*, int *nLength*)

64-bit floating pointsignal add constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be added to each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.144.2.16 NppStatus nppsAddC_64f_I (Npp64f *nValue*, Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point, in place signal add constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be added to each vector element

nLength Length of the vectors, number of items.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.144.2.17 NppStatus nppsAddC_64fc (const Npp64fc * *pSrc*, Npp64fc *nValue*, Npp64fc * *pDst*, int *nLength*)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal add constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be added to each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.18 NppStatus nppsAddC_64fc_I (Npp64fc *nValue*, Npp64fc * *pSrcDst*, int *nLength*)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal add constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be added to each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.19 NppStatus nppsAddC_8u_ISfs (Npp8u *nValue*, Npp8u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal add constant, scale, then clamp to saturated value

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be added to each vector element

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.144.2.20 **NppStatus nppsAddC_8u_Sfs** (**const Npp8u * pSrc**, **Npp8u nValue**, **Npp8u * pDst**, **int nLength**, **int nScaleFactor**)

8-bit unsigned charvector add constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be added to each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.145 AddProductC

Adds product of a constant and each sample of a source signal to the each sample of destination signal.

Functions

- **NppStatus** **nppsAddProductC_32f** (const **Npp32f** *pSrc, **Npp32f** nValue, **Npp32f** *pDst, int nLength)

32-bit floating point signal add product of signal times constant to destination signal.

7.145.1 Detailed Description

Adds product of a constant and each sample of a source signal to the each sample of destination signal.

7.145.2 Function Documentation

7.145.2.1 **NppStatus nppsAddProductC_32f** (const **Npp32f** *pSrc, **Npp32f** nValue, **Npp32f** *pDst, int nLength)

32-bit floating point signal add product of signal times constant to destination signal.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146 MulC

Multiplies each sample of a signal by a constant value.

Functions

- **NppStatus nppsMulC_8u_ISfs** (**Npp8u** nValue, **Npp8u** *pSrcDst, int nLength, int nScaleFactor)
8-bit unsigned char in place signal times constant, scale, then clamp to saturated value
- **NppStatus nppsMulC_8u_Sfs** (const **Npp8u** *pSrc, **Npp8u** nValue, **Npp8u** *pDst, int nLength, int nScaleFactor)
8-bit unsigned char signal times constant, scale, then clamp to saturated value.
- **NppStatus nppsMulC_16u_ISfs** (**Npp16u** nValue, **Npp16u** *pSrcDst, int nLength, int nScaleFactor)
16-bit unsigned short in place signal times constant, scale, then clamp to saturated value.
- **NppStatus nppsMulC_16u_Sfs** (const **Npp16u** *pSrc, **Npp16u** nValue, **Npp16u** *pDst, int nLength, int nScaleFactor)
16-bit unsigned short signal times constant, scale, then clamp to saturated value.
- **NppStatus nppsMulC_16s_ISfs** (**Npp16s** nValue, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)
16-bit signed short in place signal times constant, scale, then clamp to saturated value.
- **NppStatus nppsMulC_16s_Sfs** (const **Npp16s** *pSrc, **Npp16s** nValue, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal times constant, scale, then clamp to saturated value.
- **NppStatus nppsMulC_16sc_ISfs** (**Npp16sc** nValue, **Npp16sc** *pSrcDst, int nLength, int nScaleFactor)
16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.
- **NppStatus nppsMulC_16sc_Sfs** (const **Npp16sc** *pSrc, **Npp16sc** nValue, **Npp16sc** *pDst, int nLength, int nScaleFactor)
16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.
- **NppStatus nppsMulC_32s_ISfs** (**Npp32s** nValue, **Npp32s** *pSrcDst, int nLength, int nScaleFactor)
32-bit signed integer in place signal times constant and scale.
- **NppStatus nppsMulC_32s_Sfs** (const **Npp32s** *pSrc, **Npp32s** nValue, **Npp32s** *pDst, int nLength, int nScaleFactor)
32-bit signed integer signal times constant and scale.
- **NppStatus nppsMulC_32sc_ISfs** (**Npp32sc** nValue, **Npp32sc** *pSrcDst, int nLength, int nScaleFactor)
32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal times constant and scale.
- **NppStatus nppsMulC_32sc_Sfs** (const **Npp32sc** *pSrc, **Npp32sc** nValue, **Npp32sc** *pDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal times constant and scale.

- **NppStatus nppsMulC_32f_I** (**Npp32f** nValue, **Npp32f** *pSrcDst, int nLength)
32-bit floating point in place signal times constant.
- **NppStatus nppsMulC_32f** (const **Npp32f** *pSrc, **Npp32f** nValue, **Npp32f** *pDst, int nLength)
32-bit floating point signal times constant.
- **NppStatus nppsMulC_Low_32f16s** (const **Npp32f** *pSrc, **Npp32f** nValue, **Npp16s** *pDst, int nLength)
32-bit floating point signal times constant with output converted to 16-bit signed integer.
- **NppStatus nppsMulC_32f16s_Sfs** (const **Npp32f** *pSrc, **Npp32f** nValue, **Npp16s** *pDst, int nLength, int nScaleFactor)
32-bit floating point signal times constant with output converted to 16-bit signed integer with scaling and saturation of output result.
- **NppStatus nppsMulC_32fc_I** (**Npp32fc** nValue, **Npp32fc** *pSrcDst, int nLength)
32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal times constant.
- **NppStatus nppsMulC_32fc** (const **Npp32fc** *pSrc, **Npp32fc** nValue, **Npp32fc** *pDst, int nLength)
32-bit floating point complex number (32 bit real, 32 bit imaginary) signal times constant.
- **NppStatus nppsMulC_64f_I** (**Npp64f** nValue, **Npp64f** *pSrcDst, int nLength)
64-bit floating point, in place signal times constant.
- **NppStatus nppsMulC_64f** (const **Npp64f** *pSrc, **Npp64f** nValue, **Npp64f** *pDst, int nLength)
64-bit floating point signal times constant.
- **NppStatus nppsMulC_64f64s_ISfs** (**Npp64f** nValue, **Npp64s** *pDst, int nLength, int nScaleFactor)
64-bit floating point signal times constant with in place conversion to 64-bit signed integer and with scaling and saturation of output result.
- **NppStatus nppsMulC_64fc_I** (**Npp64fc** nValue, **Npp64fc** *pSrcDst, int nLength)
64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal times constant.
- **NppStatus nppsMulC_64fc** (const **Npp64fc** *pSrc, **Npp64fc** nValue, **Npp64fc** *pDst, int nLength)
64-bit floating point complex number (64 bit real, 64 bit imaginary) signal times constant.

7.146.1 Detailed Description

Multiplies each sample of a signal by a constant value.

7.146.2 Function Documentation

7.146.2.1 **NppStatus nppsMulC_16s_ISfs** (**Npp16s** nValue, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal times constant, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.
nValue Constant value to be multiplied by each vector element
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.146.2.2 NppStatus nppsMulC_16s_Sfs (const Npp16s * pSrc, Npp16s nValue, Npp16s * pDst, int nLength, int nScaleFactor)

16-bit signed short signal times constant, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.
nValue Constant value to be multiplied by each vector element
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.146.2.3 NppStatus nppsMulC_16sc_ISfs (Npp16sc nValue, Npp16sc * pSrcDst, int nLength, int nScaleFactor)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.
nValue Constant value to be multiplied by each vector element
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.146.2.4 **NppStatus nppsMulC_16sc_Sfs** (const Npp16sc * *pSrc*, Npp16sc *nValue*, Npp16sc * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.5 **NppStatus nppsMulC_16u_ISfs** (Npp16u *nValue*, Npp16u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal times constant, scale, then clamp to saturated value.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.6 **NppStatus nppsMulC_16u_Sfs** (const Npp16u * *pSrc*, Npp16u *nValue*, Npp16u * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal times constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.7 NppStatus nppsMulC_32f (const Npp32f * *pSrc*, Npp32f *nValue*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal times constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.8 NppStatus nppsMulC_32f16s_Sfs (const Npp32f * *pSrc*, Npp32f *nValue*, Npp16s * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit floating point signal times constant with output converted to 16-bit signed integer with scaling and saturation of output result.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

pDst [Destination Signal Pointer](#).

nScaleFactor [Integer Result Scaling](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.9 NppStatus nppsMulC_32f_I (Npp32f *nValue*, Npp32f * *pSrcDst*, int *nLength*)

32-bit floating point in place signal times constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.10 NppStatus nppsMulC_32fc (const Npp32fc * *pSrc*, Npp32fc *nValue*, Npp32fc * *pDst*, int *nLength*)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal times constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be multiplied by each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.146.2.11 NppStatus nppsMulC_32fc_I (Npp32fc *nValue*, Npp32fc * *pSrcDst*, int *nLength*)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal times constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be multiplied by each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.146.2.12 NppStatus nppsMulC_32s_ISfs (Npp32s *nValue*, Npp32s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer in place signal times constant and scale.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be multiplied by each vector element

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.146.2.13 NppStatus nppsMulC_32s_Sfs (const Npp32s * *pSrc*, Npp32s *nValue*, Npp32s * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal times constant and scale.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.14 NppStatus nppsMulC_32sc_ISfs (Npp32sc *nValue*, Npp32sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal times constant and scale.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.15 NppStatus nppsMulC_32sc_Sfs (const Npp32sc * *pSrc*, Npp32sc *nValue*, Npp32sc * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal times constant and scale.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.16 NppStatus nppsMulC_64f (const Npp64f * *pSrc*, Npp64f *nValue*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal times constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.17 NppStatus nppsMulC_64f64s_ISfs (Npp64f *nValue*, Npp64s * *pDst*, int *nLength*, int *nScaleFactor*)

64-bit floating point signal times constant with in place conversion to 64-bit signed integer and with scaling and saturation of output result.

Parameters:

nValue Constant value to be multiplied by each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.18 NppStatus nppsMulC_64f_I (Npp64f *nValue*, Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point, in place signal times constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be multiplied by each vector element

nLength Length of the vectors, number of items.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.146.2.19 NppStatus nppsMulC_64fc (const Npp64fc * *pSrc*, Npp64fc *nValue*, Npp64fc * *pDst*, int *nLength*)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal times constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be multiplied by each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.146.2.20 NppStatus nppsMulC_64fc_I (Npp64fc *nValue*, Npp64fc * *pSrcDst*, int *nLength*)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal times constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be multiplied by each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.146.2.21 NppStatus nppsMulC_8u_ISfs (Npp8u *nValue*, Npp8u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal times constant, scale, then clamp to saturated value

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be multiplied by each vector element

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.146.2.22 NppStatus nppsMulC_8u_Sfs (const Npp8u * *pSrc*, Npp8u *nValue*, Npp8u * *pDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char signal times constant, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be multiplied by each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.146.2.23 NppStatus nppsMulC_Low_32f16s (const Npp32f * *pSrc*, Npp32f *nValue*, Npp16s * *pDst*, int *nLength*)

32-bit floating point signal times constant with output converted to 16-bit signed integer.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be multiplied by each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.147 SubC

Subtracts a constant from each sample of a signal.

Functions

- **NppStatus nppsSubC_8u_ISfs** (**Npp8u** nValue, **Npp8u** *pSrcDst, int nLength, int nScaleFactor)
8-bit unsigned char in place signal subtract constant, scale, then clamp to saturated value
- **NppStatus nppsSubC_8u_Sfs** (const **Npp8u** *pSrc, **Npp8u** nValue, **Npp8u** *pDst, int nLength, int nScaleFactor)
8-bit unsigned char signal subtract constant, scale, then clamp to saturated value.
- **NppStatus nppsSubC_16u_ISfs** (**Npp16u** nValue, **Npp16u** *pSrcDst, int nLength, int nScaleFactor)
16-bit unsigned short in place signal subtract constant, scale, then clamp to saturated value.
- **NppStatus nppsSubC_16u_Sfs** (const **Npp16u** *pSrc, **Npp16u** nValue, **Npp16u** *pDst, int nLength, int nScaleFactor)
16-bit unsigned short signal subtract constant, scale, then clamp to saturated value.
- **NppStatus nppsSubC_16s_ISfs** (**Npp16s** nValue, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)
16-bit signed short in place signal subtract constant, scale, then clamp to saturated value.
- **NppStatus nppsSubC_16s_Sfs** (const **Npp16s** *pSrc, **Npp16s** nValue, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal subtract constant, scale, then clamp to saturated value.
- **NppStatus nppsSubC_16sc_ISfs** (**Npp16sc** nValue, **Npp16sc** *pSrcDst, int nLength, int nScaleFactor)
16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.
- **NppStatus nppsSubC_16sc_Sfs** (const **Npp16sc** *pSrc, **Npp16sc** nValue, **Npp16sc** *pDst, int nLength, int nScaleFactor)
16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.
- **NppStatus nppsSubC_32s_ISfs** (**Npp32s** nValue, **Npp32s** *pSrcDst, int nLength, int nScaleFactor)
32-bit signed integer in place signal subtract constant and scale.
- **NppStatus nppsSubC_32s_Sfs** (const **Npp32s** *pSrc, **Npp32s** nValue, **Npp32s** *pDst, int nLength, int nScaleFactor)
32-bit signed integer signal subtract constant and scale.
- **NppStatus nppsSubC_32sc_ISfs** (**Npp32sc** nValue, **Npp32sc** *pSrcDst, int nLength, int nScaleFactor)
32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract constant and scale.
- **NppStatus nppsSubC_32sc_Sfs** (const **Npp32sc** *pSrc, **Npp32sc** nValue, **Npp32sc** *pDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract constant and scale.

- **NppStatus nppsSubC_32f_I** (**Npp32f** nValue, **Npp32f** *pSrcDst, int nLength)
32-bit floating point in place signal subtract constant.
- **NppStatus nppsSubC_32f** (const **Npp32f** *pSrc, **Npp32f** nValue, **Npp32f** *pDst, int nLength)
32-bit floating point signal subtract constant.
- **NppStatus nppsSubC_32fc_I** (**Npp32fc** nValue, **Npp32fc** *pSrcDst, int nLength)
32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract constant.
- **NppStatus nppsSubC_32fc** (const **Npp32fc** *pSrc, **Npp32fc** nValue, **Npp32fc** *pDst, int nLength)
32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract constant.
- **NppStatus nppsSubC_64f_I** (**Npp64f** nValue, **Npp64f** *pSrcDst, int nLength)
64-bit floating point, in place signal subtract constant.
- **NppStatus nppsSubC_64f** (const **Npp64f** *pSrc, **Npp64f** nValue, **Npp64f** *pDst, int nLength)
64-bit floating point signal subtract constant.
- **NppStatus nppsSubC_64fc_I** (**Npp64fc** nValue, **Npp64fc** *pSrcDst, int nLength)
64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract constant.
- **NppStatus nppsSubC_64fc** (const **Npp64fc** *pSrc, **Npp64fc** nValue, **Npp64fc** *pDst, int nLength)
64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract constant.

7.147.1 Detailed Description

Subtracts a constant from each sample of a signal.

7.147.2 Function Documentation

7.147.2.1 **NppStatus nppsSubC_16s_ISfs** (**Npp16s** nValue, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal subtract constant, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be subtracted from each vector element

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.147.2.2 NppStatus nppsSubC_16s_Sfs (const Npp16s * *pSrc*, Npp16s *nValue*, Npp16s * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.3 NppStatus nppsSubC_16sc_ISfs (Npp16sc *nValue*, Npp16sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.4 NppStatus nppsSubC_16sc_Sfs (const Npp16sc * *pSrc*, Npp16sc *nValue*, Npp16sc * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.5 NppStatus nppsSubC_16u_ISfs (Npp16u *nValue*, Npp16u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal subtract constant, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be subtracted from each vector element

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.147.2.6 NppStatus nppsSubC_16u_Sfs (const Npp16u * *pSrc*, Npp16u *nValue*, Npp16u * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be subtracted from each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.147.2.7 NppStatus nppsSubC_32f (const Npp32f * *pSrc*, Npp32f *nValue*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal subtract constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be subtracted from each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.147.2.8 NppStatus nppsSubC_32f_I (Npp32f *nValue*, Npp32f * *pSrcDst*, int *nLength*)

32-bit floating point in place signal subtract constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.9 NppStatus nppsSubC_32fc (const Npp32fc * *pSrc*, Npp32fc *nValue*, Npp32fc * *pDst*, int *nLength*)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.10 NppStatus nppsSubC_32fc_I (Npp32fc *nValue*, Npp32fc * *pSrcDst*, int *nLength*)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.11 NppStatus nppsSubC_32s_ISfs (Npp32s *nValue*, Npp32s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer in place signal subtract constant and scale.

Parameters:

pSrcDst In-Place Signal Pointer.
nValue Constant value to be subtracted from each vector element
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.147.2.12 `NppStatus nppsSubC_32s_Sfs (const Npp32s * pSrc, Npp32s nValue, Npp32s * pDst, int nLength, int nScaleFactor)`

32-bit signed integer signal subtract constant and scale.

Parameters:

pSrc Source Signal Pointer.
nValue Constant value to be subtracted from each vector element
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.147.2.13 `NppStatus nppsSubC_32sc_ISfs (Npp32sc nValue, Npp32sc * pSrcDst, int nLength, int nScaleFactor)`

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract constant and scale.

Parameters:

pSrcDst In-Place Signal Pointer.
nValue Constant value to be subtracted from each vector element
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.147.2.14 NppStatus nppsSubC_32sc_Sfs (const Npp32sc * *pSrc*, Npp32sc *nValue*, Npp32sc * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract constant and scale.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.15 NppStatus nppsSubC_64f (const Npp64f * *pSrc*, Npp64f *nValue*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal subtract constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.16 NppStatus nppsSubC_64f_I (Npp64f *nValue*, Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point, in place signal subtract constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

nLength Length of the vectors, number of items.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.17 NppStatus nppsSubC_64fc (const Npp64fc * *pSrc*, Npp64fc *nValue*, Npp64fc * *pDst*, int *nLength*)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.18 NppStatus nppsSubC_64fc_I (Npp64fc *nValue*, Npp64fc * *pSrcDst*, int *nLength*)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.19 NppStatus nppsSubC_8u_ISfs (Npp8u *nValue*, Npp8u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal subtract constant, scale, then clamp to saturated value

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.147.2.20 NppStatus nppsSubC_8u_Sfs (const Npp8u * *pSrc*, Npp8u *nValue*, Npp8u * *pDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char signal subtract constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be subtracted from each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.148 SubCRev

Subtracts each sample of a signal from a constant.

Functions

- **NppStatus** **nppsSubCRev_8u_ISfs** (**Npp8u** nValue, **Npp8u** *pSrcDst, int nLength, int nScaleFactor)
8-bit unsigned char in place signal subtract from constant, scale, then clamp to saturated value
- **NppStatus** **nppsSubCRev_8u_Sfs** (const **Npp8u** *pSrc, **Npp8u** nValue, **Npp8u** *pDst, int nLength, int nScaleFactor)
8-bit unsigned char signal subtract from constant, scale, then clamp to saturated value.
- **NppStatus** **nppsSubCRev_16u_ISfs** (**Npp16u** nValue, **Npp16u** *pSrcDst, int nLength, int nScaleFactor)
16-bit unsigned short in place signal subtract from constant, scale, then clamp to saturated value.
- **NppStatus** **nppsSubCRev_16u_Sfs** (const **Npp16u** *pSrc, **Npp16u** nValue, **Npp16u** *pDst, int nLength, int nScaleFactor)
16-bit unsigned short signal subtract from constant, scale, then clamp to saturated value.
- **NppStatus** **nppsSubCRev_16s_ISfs** (**Npp16s** nValue, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)
16-bit signed short in place signal subtract from constant, scale, then clamp to saturated value.
- **NppStatus** **nppsSubCRev_16s_Sfs** (const **Npp16s** *pSrc, **Npp16s** nValue, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal subtract from constant, scale, then clamp to saturated value.
- **NppStatus** **nppsSubCRev_16sc_ISfs** (**Npp16sc** nValue, **Npp16sc** *pSrcDst, int nLength, int nScaleFactor)
16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.
- **NppStatus** **nppsSubCRev_16sc_Sfs** (const **Npp16sc** *pSrc, **Npp16sc** nValue, **Npp16sc** *pDst, int nLength, int nScaleFactor)
16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.
- **NppStatus** **nppsSubCRev_32s_ISfs** (**Npp32s** nValue, **Npp32s** *pSrcDst, int nLength, int nScaleFactor)
32-bit signed integer in place signal subtract from constant and scale.
- **NppStatus** **nppsSubCRev_32s_Sfs** (const **Npp32s** *pSrc, **Npp32s** nValue, **Npp32s** *pDst, int nLength, int nScaleFactor)
32-bit signed integersignal subtract from constant and scale.
- **NppStatus** **nppsSubCRev_32sc_ISfs** (**Npp32sc** nValue, **Npp32sc** *pSrcDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant and scale.

- **NppStatus nppsSubCRev_32sc_Sfs** (const **Npp32sc** *pSrc, **Npp32sc** nValue, **Npp32sc** *pDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract from constant and scale.

- **NppStatus nppsSubCRev_32f_I** (**Npp32f** nValue, **Npp32f** *pSrcDst, int nLength)

32-bit floating point in place signal subtract from constant.

- **NppStatus nppsSubCRev_32f** (const **Npp32f** *pSrc, **Npp32f** nValue, **Npp32f** *pDst, int nLength)

32-bit floating point signal subtract from constant.

- **NppStatus nppsSubCRev_32fc_I** (**Npp32fc** nValue, **Npp32fc** *pSrcDst, int nLength)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant.

- **NppStatus nppsSubCRev_32fc** (const **Npp32fc** *pSrc, **Npp32fc** nValue, **Npp32fc** *pDst, int nLength)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract from constant.

- **NppStatus nppsSubCRev_64f_I** (**Npp64f** nValue, **Npp64f** *pSrcDst, int nLength)

64-bit floating point, in place signal subtract from constant.

- **NppStatus nppsSubCRev_64f** (const **Npp64f** *pSrc, **Npp64f** nValue, **Npp64f** *pDst, int nLength)

64-bit floating point signal subtract from constant.

- **NppStatus nppsSubCRev_64fc_I** (**Npp64fc** nValue, **Npp64fc** *pSrcDst, int nLength)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract from constant.

- **NppStatus nppsSubCRev_64fc** (const **Npp64fc** *pSrc, **Npp64fc** nValue, **Npp64fc** *pDst, int nLength)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract from constant.

7.148.1 Detailed Description

Subtracts each sample of a signal from a constant.

7.148.2 Function Documentation

7.148.2.1 **NppStatus nppsSubCRev_16s_ISfs** (**Npp16s** nValue, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal subtract from constant, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value each vector element is to be subtracted from

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.2 `NppStatus nppsSubCRev_16s_Sfs (const Npp16s * pSrc, Npp16s nValue, Npp16s * pDst, int nLength, int nScaleFactor)`

16-bit signed short signal subtract from constant, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value each vector element is to be subtracted from

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.3 `NppStatus nppsSubCRev_16sc_ISfs (Npp16sc nValue, Npp16sc * pSrcDst, int nLength, int nScaleFactor)`

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value each vector element is to be subtracted from

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.4 `NppStatus nppsSubCRev_16sc_Sfs (const Npp16sc * pSrc, Npp16sc nValue, Npp16sc * pDst, int nLength, int nScaleFactor)`

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value each vector element is to be subtracted from

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.5 NppStatus nppsSubCRev_16u_ISfs (Npp16u *nValue*, Npp16u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal subtract from constant, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value each vector element is to be subtracted from

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.6 NppStatus nppsSubCRev_16u_Sfs (const Npp16u * *pSrc*, Npp16u *nValue*, Npp16u * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal subtract from constant, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value each vector element is to be subtracted from

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.7 NppStatus nppsSubCRev_32f (const Npp32f * *pSrc*, Npp32f *nValue*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal subtract from constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value each vector element is to be subtracted from
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.8 NppStatus nppsSubCRev_32f_I (Npp32f nValue, Npp32f * pSrcDst, int nLength)

32-bit floating point in place signal subtract from constant.

Parameters:

pSrcDst In-Place Signal Pointer.
nValue Constant value each vector element is to be subtracted from
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.9 NppStatus nppsSubCRev_32fc (const Npp32fc * pSrc, Npp32fc nValue, Npp32fc * pDst, int nLength)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract from constant.

Parameters:

pSrc Source Signal Pointer.
nValue Constant value each vector element is to be subtracted from
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.10 NppStatus nppsSubCRev_32fc_I (Npp32fc nValue, Npp32fc * pSrcDst, int nLength)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant.

Parameters:

pSrcDst In-Place Signal Pointer.
nValue Constant value each vector element is to be subtracted from
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.11 NppStatus nppsSubCRev_32s_ISfs (Npp32s *nValue*, Npp32s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer in place signal subtract from constant and scale.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value each vector element is to be subtracted from

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.12 NppStatus nppsSubCRev_32s_Sfs (const Npp32s * *pSrc*, Npp32s *nValue*, Npp32s * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integersignal subtract from constant and scale.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value each vector element is to be subtracted from

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.13 NppStatus nppsSubCRev_32sc_ISfs (Npp32sc *nValue*, Npp32sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant and scale.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value each vector element is to be subtracted from

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.14 NppStatus nppsSubCRev_32sc_Sfs (const Npp32sc * *pSrc*, Npp32sc *nValue*, Npp32sc * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract from constant and scale.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value each vector element is to be subtracted from

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.148.2.15 NppStatus nppsSubCRev_64f (const Npp64f * *pSrc*, Npp64f *nValue*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal subtract from constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value each vector element is to be subtracted from

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.148.2.16 NppStatus nppsSubCRev_64f_I (Npp64f *nValue*, Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point, in place signal subtract from constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value each vector element is to be subtracted from

nLength Length of the vectors, number of items.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.148.2.17 NppStatus nppsSubCRev_64fc (const Npp64fc * *pSrc*, Npp64fc *nValue*, Npp64fc * *pDst*, int *nLength*)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract from constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value each vector element is to be subtracted from

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.18 NppStatus nppsSubCRev_64fc_I (Npp64fc *nValue*, Npp64fc * *pSrcDst*, int *nLength*)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract from constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value each vector element is to be subtracted from

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.19 NppStatus nppsSubCRev_8u_ISfs (Npp8u *nValue*, Npp8u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal subtract from constant, scale, then clamp to saturated value

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value each vector element is to be subtracted from

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.148.2.20 NppStatus nppsSubCRev_8u_Sfs (const Npp8u * *pSrc*, Npp8u *nValue*, Npp8u * *pDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char signal subtract from constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value each vector element is to be subtracted from

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149 DivC

Divides each sample of a signal by a constant.

Functions

- **NppStatus nppsDivC_8u_ISfs** (**Npp8u** nValue, **Npp8u** *pSrcDst, int nLength, int nScaleFactor)
8-bit unsigned char in place signal divided by constant, scale, then clamp to saturated value
- **NppStatus nppsDivC_8u_Sfs** (const **Npp8u** *pSrc, **Npp8u** nValue, **Npp8u** *pDst, int nLength, int nScaleFactor)
8-bit unsigned char signal divided by constant, scale, then clamp to saturated value.
- **NppStatus nppsDivC_16u_ISfs** (**Npp16u** nValue, **Npp16u** *pSrcDst, int nLength, int nScaleFactor)
16-bit unsigned short in place signal divided by constant, scale, then clamp to saturated value.
- **NppStatus nppsDivC_16u_Sfs** (const **Npp16u** *pSrc, **Npp16u** nValue, **Npp16u** *pDst, int nLength, int nScaleFactor)
16-bit unsigned short signal divided by constant, scale, then clamp to saturated value.
- **NppStatus nppsDivC_16s_ISfs** (**Npp16s** nValue, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)
16-bit signed short in place signal divided by constant, scale, then clamp to saturated value.
- **NppStatus nppsDivC_16s_Sfs** (const **Npp16s** *pSrc, **Npp16s** nValue, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal divided by constant, scale, then clamp to saturated value.
- **NppStatus nppsDivC_16sc_ISfs** (**Npp16sc** nValue, **Npp16sc** *pSrcDst, int nLength, int nScaleFactor)
16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.
- **NppStatus nppsDivC_16sc_Sfs** (const **Npp16sc** *pSrc, **Npp16sc** nValue, **Npp16sc** *pDst, int nLength, int nScaleFactor)
16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.
- **NppStatus nppsDivC_32f_I** (**Npp32f** nValue, **Npp32f** *pSrcDst, int nLength)
32-bit floating point in place signal divided by constant.
- **NppStatus nppsDivC_32f** (const **Npp32f** *pSrc, **Npp32f** nValue, **Npp32f** *pDst, int nLength)
32-bit floating point signal divided by constant.
- **NppStatus nppsDivC_32fc_I** (**Npp32fc** nValue, **Npp32fc** *pSrcDst, int nLength)
32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal divided by constant.
- **NppStatus nppsDivC_32fc** (const **Npp32fc** *pSrc, **Npp32fc** nValue, **Npp32fc** *pDst, int nLength)
32-bit floating point complex number (32 bit real, 32 bit imaginary) signal divided by constant.
- **NppStatus nppsDivC_64f_I** (**Npp64f** nValue, **Npp64f** *pSrcDst, int nLength)

64-bit floating point in place signal divided by constant.

- **NppStatus nppsDivC_64f** (const **Npp64f** *pSrc, **Npp64f** nValue, **Npp64f** *pDst, int nLength)
64-bit floating point signal divided by constant.
- **NppStatus nppsDivC_64fc_I** (**Npp64fc** nValue, **Npp64fc** *pSrcDst, int nLength)
64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal divided by constant.
- **NppStatus nppsDivC_64fc** (const **Npp64fc** *pSrc, **Npp64fc** nValue, **Npp64fc** *pDst, int nLength)
64-bit floating point complex number (64 bit real, 64 bit imaginary) signal divided by constant.

7.149.1 Detailed Description

Divides each sample of a signal by a constant.

7.149.2 Function Documentation

7.149.2.1 **NppStatus nppsDivC_16s_ISfs** (**Npp16s** nValue, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal divided by constant, scale, then clamp to saturated value.

Parameters:

pSrcDst [In-Place Signal Pointer](#).
nValue Constant value to be divided into each vector element
nLength [Signal Length](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.2 **NppStatus nppsDivC_16s_Sfs** (const **Npp16s** *pSrc, **Npp16s** nValue, **Npp16s** *pDst, int nLength, int nScaleFactor)

16-bit signed short signal divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).
nValue Constant value to be divided into each vector element
pDst [Destination Signal Pointer](#).
nLength [Signal Length](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.3 NppStatus nppsDivC_16sc_ISfs (Npp16sc *nValue*, Npp16sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be divided into each vector element

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.4 NppStatus nppsDivC_16sc_Sfs (const Npp16sc * *pSrc*, Npp16sc *nValue*, Npp16sc * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be divided into each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.5 NppStatus nppsDivC_16u_ISfs (Npp16u *nValue*, Npp16u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal divided by constant, scale, then clamp to saturated value.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be divided into each vector element

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.6 NppStatus nppsDivC_16u_Sfs (const Npp16u * *pSrc*, Npp16u *nValue*, Npp16u * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be divided into each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.7 NppStatus nppsDivC_32f (const Npp32f * *pSrc*, Npp32f *nValue*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal divided by constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be divided into each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.8 NppStatus nppsDivC_32f_I (Npp32f *nValue*, Npp32f * *pSrcDst*, int *nLength*)

32-bit floating point in place signal divided by constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be divided into each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.9 NppStatus nppsDivC_32fc (const Npp32fc * *pSrc*, Npp32fc *nValue*, Npp32fc * *pDst*, int *nLength*)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal divided by constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be divided into each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.149.2.10 NppStatus nppsDivC_32fc_I (Npp32fc *nValue*, Npp32fc * *pSrcDst*, int *nLength*)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal divided by constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be divided into each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.149.2.11 NppStatus nppsDivC_64f (const Npp64f * *pSrc*, Npp64f *nValue*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal divided by constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be divided into each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.149.2.12 NppStatus nppsDivC_64f_I (Npp64f *nValue*, Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point in place signal divided by constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be divided into each vector element

nLength Length of the vectors, number of items.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.13 NppStatus nppsDivC_64fc (const Npp64fc * *pSrc*, Npp64fc *nValue*, Npp64fc * *pDst*, int *nLength*)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal divided by constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be divided into each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.14 NppStatus nppsDivC_64fc_I (Npp64fc *nValue*, Npp64fc * *pSrcDst*, int *nLength*)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal divided by constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be divided into each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.15 NppStatus nppsDivC_8u_ISfs (Npp8u *nValue*, Npp8u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal divided by constant, scale, then clamp to saturated value

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be divided into each vector element

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.149.2.16 NppStatus nppsDivC_8u_Sfs (const Npp8u * pSrc, Npp8u nValue, Npp8u * pDst, int nLength, int nScaleFactor)

8-bit unsigned char signal divided by constant, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be divided into each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.150 DivCRev

Divides a constant by each sample of a signal.

Functions

- **NppStatus nppsDivCRev_16u_I** (**Npp16u** nValue, **Npp16u** *pSrcDst, int nLength)
16-bit unsigned short in place constant divided by signal, then clamp to saturated value.
- **NppStatus nppsDivCRev_16u** (const **Npp16u** *pSrc, **Npp16u** nValue, **Npp16u** *pDst, int nLength)
16-bit unsigned short signal divided by constant, then clamp to saturated value.
- **NppStatus nppsDivCRev_32f_I** (**Npp32f** nValue, **Npp32f** *pSrcDst, int nLength)
32-bit floating point in place constant divided by signal.
- **NppStatus nppsDivCRev_32f** (const **Npp32f** *pSrc, **Npp32f** nValue, **Npp32f** *pDst, int nLength)
32-bit floating point constant divided by signal.

7.150.1 Detailed Description

Divides a constant by each sample of a signal.

7.150.2 Function Documentation

7.150.2.1 NppStatus nppsDivCRev_16u (const Npp16u * pSrc, Npp16u nValue, Npp16u * pDst, int nLength)

16-bit unsigned short signal divided by constant, then clamp to saturated value.

Parameters:

- pSrc* **Source Signal Pointer.**
nValue **Constant value to be divided by each vector element**
pDst **Destination Signal Pointer.**
nLength **Signal Length.**

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.150.2.2 NppStatus nppsDivCRev_16u_I (Npp16u nValue, Npp16u * pSrcDst, int nLength)

16-bit unsigned short in place constant divided by signal, then clamp to saturated value.

Parameters:

- pSrcDst* **In-Place Signal Pointer.**

nValue Constant value to be divided by each vector element
nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.150.2.3 NppStatus nppsDivCRev_32f (const Npp32f * pSrc, Npp32f nValue, Npp32f * pDst, int nLength)

32-bit floating point constant divided by signal.

Parameters:

pSrc [Source Signal Pointer](#).
nValue Constant value to be divided by each vector element
pDst [Destination Signal Pointer](#).
nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.150.2.4 NppStatus nppsDivCRev_32f_I (Npp32f nValue, Npp32f * pSrcDst, int nLength)

32-bit floating point in place constant divided by signal.

Parameters:

pSrcDst [In-Place Signal Pointer](#).
nValue Constant value to be divided by each vector element
nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.151 Add

Sample by sample addition of two signals.

Functions

- **NppStatus** **nppsAdd_16s** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength)
16-bit signed short signal add signal, then clamp to saturated value.
- **NppStatus** **nppsAdd_16u** (const **Npp16u** *pSrc1, const **Npp16u** *pSrc2, **Npp16u** *pDst, int nLength)
16-bit unsigned short signal add signal, then clamp to saturated value.
- **NppStatus** **nppsAdd_32u** (const **Npp32u** *pSrc1, const **Npp32u** *pSrc2, **Npp32u** *pDst, int nLength)
32-bit unsigned int signal add signal, then clamp to saturated value.
- **NppStatus** **nppsAdd_32f** (const **Npp32f** *pSrc1, const **Npp32f** *pSrc2, **Npp32f** *pDst, int nLength)
32-bit floating point signal add signal, then clamp to saturated value.
- **NppStatus** **nppsAdd_64f** (const **Npp64f** *pSrc1, const **Npp64f** *pSrc2, **Npp64f** *pDst, int nLength)
64-bit floating point signal add signal, then clamp to saturated value.
- **NppStatus** **nppsAdd_32fc** (const **Npp32fc** *pSrc1, const **Npp32fc** *pSrc2, **Npp32fc** *pDst, int nLength)
32-bit complex floating point signal add signal, then clamp to saturated value.
- **NppStatus** **nppsAdd_64fc** (const **Npp64fc** *pSrc1, const **Npp64fc** *pSrc2, **Npp64fc** *pDst, int nLength)
64-bit complex floating point signal add signal, then clamp to saturated value.
- **NppStatus** **nppsAdd_8u16u** (const **Npp8u** *pSrc1, const **Npp8u** *pSrc2, **Npp16u** *pDst, int nLength)
8-bit unsigned char signal add signal with 16-bit unsigned result, then clamp to saturated value.
- **NppStatus** **nppsAdd_16s32f** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp32f** *pDst, int nLength)
16-bit signed short signal add signal with 32-bit floating point result, then clamp to saturated value.
- **NppStatus** **nppsAdd_8u_Sfs** (const **Npp8u** *pSrc1, const **Npp8u** *pSrc2, **Npp8u** *pDst, int nLength, int nScaleFactor)
8-bit unsigned char add signal, scale, then clamp to saturated value.
- **NppStatus** **nppsAdd_16u_Sfs** (const **Npp16u** *pSrc1, const **Npp16u** *pSrc2, **Npp16u** *pDst, int nLength, int nScaleFactor)
16-bit unsigned short add signal, scale, then clamp to saturated value.
- **NppStatus** **nppsAdd_16s_Sfs** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength, int nScaleFactor)

16-bit signed short add signal, scale, then clamp to saturated value.

- `NppStatus nppsAdd_32s_Sfs` (const `Npp32s` *pSrc1, const `Npp32s` *pSrc2, `Npp32s` *pDst, int nLength, int nScaleFactor)

32-bit signed integer add signal, scale, then clamp to saturated value.

- `NppStatus nppsAdd_64s_Sfs` (const `Npp64s` *pSrc1, const `Npp64s` *pSrc2, `Npp64s` *pDst, int nLength, int nScaleFactor)

64-bit signed integer add signal, scale, then clamp to saturated value.

- `NppStatus nppsAdd_16sc_Sfs` (const `Npp16sc` *pSrc1, const `Npp16sc` *pSrc2, `Npp16sc` *pDst, int nLength, int nScaleFactor)

16-bit signed complex short add signal, scale, then clamp to saturated value.

- `NppStatus nppsAdd_32sc_Sfs` (const `Npp32sc` *pSrc1, const `Npp32sc` *pSrc2, `Npp32sc` *pDst, int nLength, int nScaleFactor)

32-bit signed complex integer add signal, scale, then clamp to saturated value.

- `NppStatus nppsAdd_16s_I` (const `Npp16s` *pSrc, `Npp16s` *pSrcDst, int nLength)

16-bit signed short in place signal add signal, then clamp to saturated value.

- `NppStatus nppsAdd_32f_I` (const `Npp32f` *pSrc, `Npp32f` *pSrcDst, int nLength)

32-bit floating point in place signal add signal, then clamp to saturated value.

- `NppStatus nppsAdd_64f_I` (const `Npp64f` *pSrc, `Npp64f` *pSrcDst, int nLength)

64-bit floating point in place signal add signal, then clamp to saturated value.

- `NppStatus nppsAdd_32fc_I` (const `Npp32fc` *pSrc, `Npp32fc` *pSrcDst, int nLength)

32-bit complex floating point in place signal add signal, then clamp to saturated value.

- `NppStatus nppsAdd_64fc_I` (const `Npp64fc` *pSrc, `Npp64fc` *pSrcDst, int nLength)

64-bit complex floating point in place signal add signal, then clamp to saturated value.

- `NppStatus nppsAdd_16s32s_I` (const `Npp16s` *pSrc, `Npp32s` *pSrcDst, int nLength)

16/32-bit signed short in place signal add signal with 32-bit signed integer results, then clamp to saturated value.

- `NppStatus nppsAdd_8u_ISfs` (const `Npp8u` *pSrc, `Npp8u` *pSrcDst, int nLength, int nScaleFactor)

8-bit unsigned char in place signal add signal, with scaling, then clamp to saturated value.

- `NppStatus nppsAdd_16u_ISfs` (const `Npp16u` *pSrc, `Npp16u` *pSrcDst, int nLength, int nScaleFactor)

16-bit unsigned short in place signal add signal, with scaling, then clamp to saturated value.

- `NppStatus nppsAdd_16s_ISfs` (const `Npp16s` *pSrc, `Npp16s` *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal add signal, with scaling, then clamp to saturated value.

- `NppStatus nppsAdd_32s_ISfs` (const `Npp32s` *pSrc, `Npp32s` *pSrcDst, int nLength, int nScaleFactor)

32-bit signed integer in place signal add signal, with scaling, then clamp to saturated value.

- **NppStatus nppsAdd_16sc_ISfs** (const [Npp16sc](#) *pSrc, [Npp16sc](#) *pSrcDst, int nLength, int nScaleFactor)
16-bit complex signed short in place signal add signal, with scaling, then clamp to saturated value.
- **NppStatus nppsAdd_32sc_ISfs** (const [Npp32sc](#) *pSrc, [Npp32sc](#) *pSrcDst, int nLength, int nScaleFactor)
32-bit complex signed integer in place signal add signal, with scaling, then clamp to saturated value.

7.151.1 Detailed Description

Sample by sample addition of two signals.

7.151.2 Function Documentation

7.151.2.1 **NppStatus nppsAdd_16s** (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, [Npp16s](#) *pDst, int nLength)

16-bit signed short signal add signal, then clamp to saturated value.

Parameters:

[pSrc1](#) [Source Signal Pointer](#).
[pSrc2](#) [Source Signal Pointer](#). signal2 elements to be added to signal1 elements
[pDst](#) [Destination Signal Pointer](#).
[nLength](#) [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.151.2.2 **NppStatus nppsAdd_16s32f** (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, [Npp32f](#) *pDst, int nLength)

16-bit signed short signal add signal with 32-bit floating point result, then clamp to saturated value.

Parameters:

[pSrc1](#) [Source Signal Pointer](#).
[pSrc2](#) [Source Signal Pointer](#). signal2 elements to be added to signal1 elements
[pDst](#) [Destination Signal Pointer](#).
[nLength](#) [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.151.2.3 NppStatus nppsAdd_16s32s_I (const Npp16s * pSrc, Npp32s * pSrcDst, int nLength)

16/32-bit signed short in place signal add signal with 32-bit signed integer results, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be added to signal1 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.4 NppStatus nppsAdd_16s_I (const Npp16s * pSrc, Npp16s * pSrcDst, int nLength)

16-bit signed short in place signal add signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be added to signal1 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.5 NppStatus nppsAdd_16s_ISfs (const Npp16s * pSrc, Npp16s * pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal add signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be added to signal1 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.6 NppStatus nppsAdd_16s_Sfs (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, Npp16s * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short add signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal2 elements to be added to signal1 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.7 NppStatus nppsAdd_16sc_ISfs (const Npp16sc * *pSrc*, Npp16sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit complex signed short in place signal add signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be added to signal1 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.8 NppStatus nppsAdd_16sc_Sfs (const Npp16sc * *pSrc1*, const Npp16sc * *pSrc2*, Npp16sc * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed complex short add signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal2 elements to be added to signal1 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.9 NppStatus nppsAdd_16u (const Npp16u * *pSrc1*, const Npp16u * *pSrc2*, Npp16u * *pDst*, int *nLength*)

16-bit unsigned short signal add signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be added to signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.10 NppStatus nppsAdd_16u_ISfs (const Npp16u * *pSrc*, Npp16u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal add signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be added to signal1 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.11 NppStatus nppsAdd_16u_Sfs (const Npp16u * *pSrc1*, const Npp16u * *pSrc2*, Npp16u * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short add signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal2 elements to be added to signal1 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.12 NppStatus nppsAdd_32f (const Npp32f * *pSrc1*, const Npp32f * *pSrc2*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal add signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be added to signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.13 NppStatus nppsAdd_32f_I (const Npp32f * *pSrc*, Npp32f * *pSrcDst*, int *nLength*)

32-bit floating point in place signal add signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be added to signal1 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.14 NppStatus nppsAdd_32fc (const Npp32fc * *pSrc1*, const Npp32fc * *pSrc2*, Npp32fc * *pDst*, int *nLength*)

32-bit complex floating point signal add signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be added to signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.15 NppStatus nppsAdd_32fc_I (const Npp32fc * *pSrc*, Npp32fc * *pSrcDst*, int *nLength*)

32-bit complex floating point in place signal add signal, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be added to signal1 elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.151.2.16 NppStatus nppsAdd_32s_ISfs (const Npp32s * *pSrc*, Npp32s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer in place signal add signal, with scaling, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be added to signal1 elements

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.151.2.17 NppStatus nppsAdd_32s_Sfs (const Npp32s * *pSrc1*, const Npp32s * *pSrc2*, Npp32s * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer add signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal2 elements to be added to signal1 elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.151.2.18 NppStatus nppsAdd_32sc_ISfs (const Npp32sc * *pSrc*, Npp32sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit complex signed integer in place signal add signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be added to signal1 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.19 NppStatus nppsAdd_32sc_Sfs (const Npp32sc * *pSrc1*, const Npp32sc * *pSrc2*, Npp32sc * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed complex integer add signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be added to signal1 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.20 NppStatus nppsAdd_32u (const Npp32u * *pSrc1*, const Npp32u * *pSrc2*, Npp32u * *pDst*, int *nLength*)

32-bit unsigned int signal add signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be added to signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.21 NppStatus nppsAdd_64f (const Npp64f * *pSrc1*, const Npp64f * *pSrc2*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal add signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be added to signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.22 NppStatus nppsAdd_64f_I (const Npp64f * *pSrc*, Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point in place signal add signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be added to signal1 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.23 NppStatus nppsAdd_64fc (const Npp64fc * *pSrc1*, const Npp64fc * *pSrc2*, Npp64fc * *pDst*, int *nLength*)

64-bit complex floating point signal add signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be added to signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.151.2.24 NppStatus nppsAdd_64fc_I (const Npp64fc * *pSrc*, Npp64fc * *pSrcDst*, int *nLength*)

64-bit complex floating point in place signal add signal, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be added to signal1 elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.151.2.25 NppStatus nppsAdd_64s_Sfs (const Npp64s * *pSrc1*, const Npp64s * *pSrc2*, Npp64s * *pDst*, int *nLength*, int *nScaleFactor*)

64-bit signed integer add signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal2 elements to be added to signal1 elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.151.2.26 NppStatus nppsAdd_8u16u (const Npp8u * *pSrc1*, const Npp8u * *pSrc2*, Npp16u * *pDst*, int *nLength*)

8-bit unsigned char signal add signal with 16-bit unsigned result, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#). signal2 elements to be added to signal1 elements

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.151.2.27 NppStatus nppsAdd_8u_ISfs (const Npp8u * *pSrc*, Npp8u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal add signal, with scaling, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be added to signal1 elements

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.151.2.28 NppStatus nppsAdd_8u_Sfs (const Npp8u * *pSrc1*, const Npp8u * *pSrc2*, Npp8u * *pDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char add signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal2 elements to be added to signal1 elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.152 AddProduct

Adds sample by sample product of two signals to the destination signal.

Functions

- **NppStatus** **nppsAddProduct_32f** (const **Npp32f** *pSrc1, const **Npp32f** *pSrc2, **Npp32f** *pDst, int nLength)
32-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.
- **NppStatus** **nppsAddProduct_64f** (const **Npp64f** *pSrc1, const **Npp64f** *pSrc2, **Npp64f** *pDst, int nLength)
64-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.
- **NppStatus** **nppsAddProduct_32fc** (const **Npp32fc** *pSrc1, const **Npp32fc** *pSrc2, **Npp32fc** *pDst, int nLength)
32-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.
- **NppStatus** **nppsAddProduct_64fc** (const **Npp64fc** *pSrc1, const **Npp64fc** *pSrc2, **Npp64fc** *pDst, int nLength)
64-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.
- **NppStatus** **nppsAddProduct_16s_Sfs** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.
- **NppStatus** **nppsAddProduct_32s_Sfs** (const **Npp32s** *pSrc1, const **Npp32s** *pSrc2, **Npp32s** *pDst, int nLength, int nScaleFactor)
32-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.
- **NppStatus** **nppsAddProduct_16s32s_Sfs** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp32s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal add product of source signal1 times source signal2 to 32-bit signed integer destination signal, with scaling, then clamp to saturated value.

7.152.1 Detailed Description

Adds sample by sample product of two signals to the destination signal.

7.152.2 Function Documentation

7.152.2.1 NppStatus nppsAddProduct_16s32s_Sfs (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, Npp32s * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal add product of source signal1 times source signal2 to 32-bit signed integer destination signal, with scaling, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

pDst Destination Signal Pointer. product of source1 and source2 signal elements to be added to destination elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.152.2.2 NppStatus nppsAddProduct_16s_Sfs (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, Npp16s * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

pDst Destination Signal Pointer. product of source1 and source2 signal elements to be added to destination elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.152.2.3 NppStatus nppsAddProduct_32f (const Npp32f * *pSrc1*, const Npp32f * *pSrc2*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

pDst [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.152.2.4 NppStatus nppsAddProduct_32fc (const Npp32fc * pSrc1, const Npp32fc * pSrc2, Npp32fc * pDst, int nLength)

32-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.152.2.5 NppStatus nppsAddProduct_32s_Sfs (const Npp32s * pSrc1, const Npp32s * pSrc2, Npp32s * pDst, int nLength, int nScaleFactor)

32-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.152.2.6 NppStatus nppsAddProduct_64f (const Npp64f * *pSrc1*, const Npp64f * *pSrc2*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

pDst Destination Signal Pointer. product of source1 and source2 signal elements to be added to destination elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.152.2.7 NppStatus nppsAddProduct_64fc (const Npp64fc * *pSrc1*, const Npp64fc * *pSrc2*, Npp64fc * *pDst*, int *nLength*)

64-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

pDst Destination Signal Pointer. product of source1 and source2 signal elements to be added to destination elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153 Mul

Sample by sample multiplication the samples of two signals.

Functions

- **NppStatus nppsMul_16s** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength)
16-bit signed short signal times signal, then clamp to saturated value.
- **NppStatus nppsMul_32f** (const **Npp32f** *pSrc1, const **Npp32f** *pSrc2, **Npp32f** *pDst, int nLength)
32-bit floating point signal times signal, then clamp to saturated value.
- **NppStatus nppsMul_64f** (const **Npp64f** *pSrc1, const **Npp64f** *pSrc2, **Npp64f** *pDst, int nLength)
64-bit floating point signal times signal, then clamp to saturated value.
- **NppStatus nppsMul_32fc** (const **Npp32fc** *pSrc1, const **Npp32fc** *pSrc2, **Npp32fc** *pDst, int nLength)
32-bit complex floating point signal times signal, then clamp to saturated value.
- **NppStatus nppsMul_64fc** (const **Npp64fc** *pSrc1, const **Npp64fc** *pSrc2, **Npp64fc** *pDst, int nLength)
64-bit complex floating point signal times signal, then clamp to saturated value.
- **NppStatus nppsMul_8u16u** (const **Npp8u** *pSrc1, const **Npp8u** *pSrc2, **Npp16u** *pDst, int nLength)
8-bit unsigned char signal times signal with 16-bit unsigned result, then clamp to saturated value.
- **NppStatus nppsMul_16s32f** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp32f** *pDst, int nLength)
16-bit signed short signal times signal with 32-bit floating point result, then clamp to saturated value.
- **NppStatus nppsMul_32f32fc** (const **Npp32f** *pSrc1, const **Npp32fc** *pSrc2, **Npp32fc** *pDst, int nLength)
32-bit floating point signal times 32-bit complex floating point signal with complex 32-bit floating point result, then clamp to saturated value.
- **NppStatus nppsMul_8u_Sfs** (const **Npp8u** *pSrc1, const **Npp8u** *pSrc2, **Npp8u** *pDst, int nLength, int nScaleFactor)
8-bit unsigned char signal times signal, scale, then clamp to saturated value.
- **NppStatus nppsMul_16u_Sfs** (const **Npp16u** *pSrc1, const **Npp16u** *pSrc2, **Npp16u** *pDst, int nLength, int nScaleFactor)
16-bit unsigned short signal time signal, scale, then clamp to saturated value.
- **NppStatus nppsMul_16s_Sfs** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal times signal, scale, then clamp to saturated value.
- **NppStatus nppsMul_32s_Sfs** (const **Npp32s** *pSrc1, const **Npp32s** *pSrc2, **Npp32s** *pDst, int nLength, int nScaleFactor)

32-bit signed integer signal times signal, scale, then clamp to saturated value.

- **NppStatus nppsMul_16sc_Sfs** (const **Npp16sc** *pSrc1, const **Npp16sc** *pSrc2, **Npp16sc** *pDst, int nLength, int nScaleFactor)

16-bit signed complex short signal times signal, scale, then clamp to saturated value.

- **NppStatus nppsMul_32sc_Sfs** (const **Npp32sc** *pSrc1, const **Npp32sc** *pSrc2, **Npp32sc** *pDst, int nLength, int nScaleFactor)

32-bit signed complex integer signal times signal, scale, then clamp to saturated value.

- **NppStatus nppsMul_16u16s_Sfs** (const **Npp16u** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength, int nScaleFactor)

16-bit unsigned short signal times 16-bit signed short signal, scale, then clamp to 16-bit signed saturated value.

- **NppStatus nppsMul_16s32s_Sfs** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp32s** *pDst, int nLength, int nScaleFactor)

16-bit signed short signal times signal, scale, then clamp to 32-bit signed saturated value.

- **NppStatus nppsMul_32s32sc_Sfs** (const **Npp32s** *pSrc1, const **Npp32sc** *pSrc2, **Npp32sc** *pDst, int nLength, int nScaleFactor)

32-bit signed integer signal times 32-bit complex signed integer signal, scale, then clamp to 32-bit complex integer saturated value.

- **NppStatus nppsMul_Low_32s_Sfs** (const **Npp32s** *pSrc1, const **Npp32s** *pSrc2, **Npp32s** *pDst, int nLength, int nScaleFactor)

32-bit signed integer signal times signal, scale, then clamp to saturated value.

- **NppStatus nppsMul_16s_I** (const **Npp16s** *pSrc, **Npp16s** *pSrcDst, int nLength)

16-bit signed short in place signal times signal, then clamp to saturated value.

- **NppStatus nppsMul_32f_I** (const **Npp32f** *pSrc, **Npp32f** *pSrcDst, int nLength)

32-bit floating point in place signal times signal, then clamp to saturated value.

- **NppStatus nppsMul_64f_I** (const **Npp64f** *pSrc, **Npp64f** *pSrcDst, int nLength)

64-bit floating point in place signal times signal, then clamp to saturated value.

- **NppStatus nppsMul_32fc_I** (const **Npp32fc** *pSrc, **Npp32fc** *pSrcDst, int nLength)

32-bit complex floating point in place signal times signal, then clamp to saturated value.

- **NppStatus nppsMul_64fc_I** (const **Npp64fc** *pSrc, **Npp64fc** *pSrcDst, int nLength)

64-bit complex floating point in place signal times signal, then clamp to saturated value.

- **NppStatus nppsMul_32f32fc_I** (const **Npp32f** *pSrc, **Npp32fc** *pSrcDst, int nLength)

32-bit complex floating point in place signal times 32-bit floating point signal, then clamp to 32-bit complex floating point saturated value.

- **NppStatus nppsMul_8u_ISfs** (const **Npp8u** *pSrc, **Npp8u** *pSrcDst, int nLength, int nScaleFactor)

8-bit unsigned char in place signal times signal, with scaling, then clamp to saturated value.

- **NppStatus nppsMul_16u_ISfs** (const **Npp16u** *pSrc, **Npp16u** *pSrcDst, int nLength, int nScaleFactor)

16-bit unsigned short in place signal times signal, with scaling, then clamp to saturated value.

- **NppStatus nppsMul_16s_ISfs** (const **Npp16s** *pSrc, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal times signal, with scaling, then clamp to saturated value.

- **NppStatus nppsMul_32s_ISfs** (const **Npp32s** *pSrc, **Npp32s** *pSrcDst, int nLength, int nScaleFactor)

32-bit signed integer in place signal times signal, with scaling, then clamp to saturated value.

- **NppStatus nppsMul_16sc_ISfs** (const **Npp16sc** *pSrc, **Npp16sc** *pSrcDst, int nLength, int nScaleFactor)

16-bit complex signed short in place signal times signal, with scaling, then clamp to saturated value.

- **NppStatus nppsMul_32sc_ISfs** (const **Npp32sc** *pSrc, **Npp32sc** *pSrcDst, int nLength, int nScaleFactor)

32-bit complex signed integer in place signal times signal, with scaling, then clamp to saturated value.

- **NppStatus nppsMul_32s32sc_ISfs** (const **Npp32s** *pSrc, **Npp32sc** *pSrcDst, int nLength, int nScaleFactor)

32-bit complex signed integer in place signal times 32-bit signed integer signal, with scaling, then clamp to saturated value.

7.153.1 Detailed Description

Sample by sample multiplication the samples of two signals.

7.153.2 Function Documentation

7.153.2.1 **NppStatus nppsMul_16s** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength)

16-bit signed short signal times signal, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.153.2.2 NppStatus nppsMul_16s32f (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, Npp32f * *pDst*, int *nLength*)

16-bit signed short signal times signal with 32-bit floating point result, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.3 NppStatus nppsMul_16s32s_Sfs (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, Npp32s * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal times signal, scale, then clamp to 32-bit signed saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.4 NppStatus nppsMul_16s_I (const Npp16s * *pSrc*, Npp16s * *pSrcDst*, int *nLength*)

16-bit signed short in place signal times signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.5 NppStatus nppsMul_16s_ISfs (const Npp16s * *pSrc*, Npp16s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short in place signal times signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.6 NppStatus nppsMul_16s_Sfs (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, Npp16s * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal times signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.7 NppStatus nppsMul_16sc_ISfs (const Npp16sc * *pSrc*, Npp16sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit complex signed short in place signal times signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.8 NppStatus nppsMul_16sc_Sfs (const Npp16sc * *pSrc1*, const Npp16sc * *pSrc2*, Npp16sc * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed complex short signal times signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.9 NppStatus nppsMul_16u16s_Sfs (const Npp16u * *pSrc1*, const Npp16s * *pSrc2*, Npp16s * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal times 16-bit signed short signal, scale, then clamp to 16-bit signed saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.10 NppStatus nppsMul_16u_ISfs (const Npp16u * *pSrc*, Npp16u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal times signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.11 **NppStatus nppsMul_16u_Sfs** (const Npp16u * *pSrc1*, const Npp16u * *pSrc2*, Npp16u * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal time signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.153.2.12 **NppStatus nppsMul_32f** (const Npp32f * *pSrc1*, const Npp32f * *pSrc2*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal times signal, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.153.2.13 **NppStatus nppsMul_32f32fc** (const Npp32f * *pSrc1*, const Npp32fc * *pSrc2*, Npp32fc * *pDst*, int *nLength*)

32-bit floating point signal times 32-bit complex floating point signal with complex 32-bit floating point result, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.153.2.14 NppStatus nppsMul_32f32fc_I (const Npp32f * *pSrc*, Npp32fc * *pSrcDst*, int *nLength*)

32-bit complex floating point in place signal times 32-bit floating point signal, then clamp to 32-bit complex floating point saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.15 NppStatus nppsMul_32f_I (const Npp32f * *pSrc*, Npp32f * *pSrcDst*, int *nLength*)

32-bit floating point in place signal times signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.16 NppStatus nppsMul_32fc (const Npp32fc * *pSrc1*, const Npp32fc * *pSrc2*, Npp32fc * *pDst*, int *nLength*)

32-bit complex floating point signal times signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.17 NppStatus nppsMul_32fc_I (const Npp32fc * *pSrc*, Npp32fc * *pSrcDst*, int *nLength*)

32-bit complex floating point in place signal times signal, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.153.2.18 NppStatus nppsMul_32s32sc_ISfs (const Npp32s * *pSrc*, Npp32sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit complex signed integer in place signal times 32-bit signed integer signal, with scaling, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.153.2.19 NppStatus nppsMul_32s32sc_Sfs (const Npp32s * *pSrc1*, const Npp32sc * *pSrc2*, Npp32sc * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal times 32-bit complex signed integer signal, scale, then clamp to 32-bit complex integer saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.153.2.20 NppStatus nppsMul_32s_ISfs (const Npp32s * *pSrc*, Npp32s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer in place signal times signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.21 NppStatus nppsMul_32s_Sfs (const Npp32s * *pSrc1*, const Npp32s * *pSrc2*, Npp32s * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal times signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.22 NppStatus nppsMul_32sc_ISfs (const Npp32sc * *pSrc*, Npp32sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit complex signed integer in place signal times signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.23 **NppStatus nppsMul_32sc_Sfs** (const Npp32sc * *pSrc1*, const Npp32sc * *pSrc2*, Npp32sc * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed complex integer signal times signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.24 **NppStatus nppsMul_64f** (const Npp64f * *pSrc1*, const Npp64f * *pSrc2*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal times signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal2 elements to be multiplied by signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.25 **NppStatus nppsMul_64f_I** (const Npp64f * *pSrc*, Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point in place signal times signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer, signal2 elements to be multiplied by signal1 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.26 NppStatus nppsMul_64fc (const Npp64fc * *pSrc1*, const Npp64fc * *pSrc2*, Npp64fc * *pDst*, int *nLength*)

64-bit complex floating point signal times signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.27 NppStatus nppsMul_64fc_I (const Npp64fc * *pSrc*, Npp64fc * *pSrcDst*, int *nLength*)

64-bit complex floating point in place signal times signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.28 NppStatus nppsMul_8u16u (const Npp8u * *pSrc1*, const Npp8u * *pSrc2*, Npp16u * *pDst*, int *nLength*)

8-bit unsigned char signal times signal with 16-bit unsigned result, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.153.2.29 **NppStatus nppsMul_8u_ISfs** (const Npp8u * *pSrc*, Npp8u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal times signal, with scaling, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#), signal2 elements to be multiplied by signal1 elements

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.153.2.30 **NppStatus nppsMul_8u_Sfs** (const Npp8u * *pSrc1*, const Npp8u * *pSrc2*, Npp8u * *pDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char signal times signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.153.2.31 **NppStatus nppsMul_Low_32s_Sfs** (const Npp32s * *pSrc1*, const Npp32s * *pSrc2*, Npp32s * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal times signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.154 Sub

Sample by sample subtraction of the samples of two signals.

Functions

- **NppStatus nppsSub_16s** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength)
16-bit signed short signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub_32f** (const **Npp32f** *pSrc1, const **Npp32f** *pSrc2, **Npp32f** *pDst, int nLength)
32-bit floating point signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub_64f** (const **Npp64f** *pSrc1, const **Npp64f** *pSrc2, **Npp64f** *pDst, int nLength)
64-bit floating point signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub_32fc** (const **Npp32fc** *pSrc1, const **Npp32fc** *pSrc2, **Npp32fc** *pDst, int nLength)
32-bit complex floating point signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub_64fc** (const **Npp64fc** *pSrc1, const **Npp64fc** *pSrc2, **Npp64fc** *pDst, int nLength)
64-bit complex floating point signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub_16s32f** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp32f** *pDst, int nLength)
16-bit signed short signal subtract 16-bit signed short signal, then clamp and convert to 32-bit floating point saturated value.
- **NppStatus nppsSub_8u_Sfs** (const **Npp8u** *pSrc1, const **Npp8u** *pSrc2, **Npp8u** *pDst, int nLength, int nScaleFactor)
8-bit unsigned char signal subtract signal, scale, then clamp to saturated value.
- **NppStatus nppsSub_16u_Sfs** (const **Npp16u** *pSrc1, const **Npp16u** *pSrc2, **Npp16u** *pDst, int nLength, int nScaleFactor)
16-bit unsigned short signal subtract signal, scale, then clamp to saturated value.
- **NppStatus nppsSub_16s_Sfs** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal subtract signal, scale, then clamp to saturated value.
- **NppStatus nppsSub_32s_Sfs** (const **Npp32s** *pSrc1, const **Npp32s** *pSrc2, **Npp32s** *pDst, int nLength, int nScaleFactor)
32-bit signed integer signal subtract signal, scale, then clamp to saturated value.
- **NppStatus nppsSub_16sc_Sfs** (const **Npp16sc** *pSrc1, const **Npp16sc** *pSrc2, **Npp16sc** *pDst, int nLength, int nScaleFactor)
16-bit signed complex short signal subtract signal, scale, then clamp to saturated value.
- **NppStatus nppsSub_32sc_Sfs** (const **Npp32sc** *pSrc1, const **Npp32sc** *pSrc2, **Npp32sc** *pDst, int nLength, int nScaleFactor)

32-bit signed complex integer signal subtract signal, scale, then clamp to saturated value.

- **NppStatus nppsSub_16s_I** (const **Npp16s** *pSrc, **Npp16s** *pSrcDst, int nLength)
16-bit signed short in place signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub_32f_I** (const **Npp32f** *pSrc, **Npp32f** *pSrcDst, int nLength)
32-bit floating point in place signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub_64f_I** (const **Npp64f** *pSrc, **Npp64f** *pSrcDst, int nLength)
64-bit floating point in place signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub_32fc_I** (const **Npp32fc** *pSrc, **Npp32fc** *pSrcDst, int nLength)
32-bit complex floating point in place signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub_64fc_I** (const **Npp64fc** *pSrc, **Npp64fc** *pSrcDst, int nLength)
64-bit complex floating point in place signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub_8u_ISfs** (const **Npp8u** *pSrc, **Npp8u** *pSrcDst, int nLength, int nScaleFactor)
8-bit unsigned char in place signal subtract signal, with scaling, then clamp to saturated value.
- **NppStatus nppsSub_16u_ISfs** (const **Npp16u** *pSrc, **Npp16u** *pSrcDst, int nLength, int nScaleFactor)
16-bit unsigned short in place signal subtract signal, with scaling, then clamp to saturated value.
- **NppStatus nppsSub_16s_ISfs** (const **Npp16s** *pSrc, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)
16-bit signed short in place signal subtract signal, with scaling, then clamp to saturated value.
- **NppStatus nppsSub_32s_ISfs** (const **Npp32s** *pSrc, **Npp32s** *pSrcDst, int nLength, int nScaleFactor)
32-bit signed integer in place signal subtract signal, with scaling, then clamp to saturated value.
- **NppStatus nppsSub_16sc_ISfs** (const **Npp16sc** *pSrc, **Npp16sc** *pSrcDst, int nLength, int nScaleFactor)
16-bit complex signed short in place signal subtract signal, with scaling, then clamp to saturated value.
- **NppStatus nppsSub_32sc_ISfs** (const **Npp32sc** *pSrc, **Npp32sc** *pSrcDst, int nLength, int nScaleFactor)
32-bit complex signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

7.154.1 Detailed Description

Sample by sample subtraction of the samples of two signals.

7.154.2 Function Documentation

7.154.2.1 **NppStatus nppsSub_16s** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength)

16-bit signed short signal subtract signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.
pSrc2 Source Signal Pointer. signal1 elements to be subtracted from signal2 elements
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.2 NppStatus nppsSub_16s32f (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, Npp32f * *pDst*, int *nLength*)

16-bit signed short signal subtract 16-bit signed short signal, then clamp and convert to 32-bit floating point saturated value.

Parameters:

pSrc1 Source Signal Pointer.
pSrc2 Source Signal Pointer. signal1 elements to be subtracted from signal2 elements
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.3 NppStatus nppsSub_16s_I (const Npp16s * *pSrc*, Npp16s * *pSrcDst*, int *nLength*)

16-bit signed short in place signal subtract signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.
pSrcDst In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.4 NppStatus nppsSub_16s_ISfs (const Npp16s * *pSrc*, Npp16s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements
nLength [Signal Length](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.154.2.5 **NppStatus nppsSub_16s_Sfs** (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, Npp16s * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal subtract signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).
pSrc2 [Source Signal Pointer](#), signal1 elements to be subtracted from signal2 elements.
pDst [Destination Signal Pointer](#).
nLength [Signal Length](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.154.2.6 **NppStatus nppsSub_16sc_ISfs** (const Npp16sc * *pSrc*, Npp16sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit complex signed short in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).
pSrcDst [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements
nLength [Signal Length](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.154.2.7 **NppStatus nppsSub_16sc_Sfs** (const Npp16sc * *pSrc1*, const Npp16sc * *pSrc2*, Npp16sc * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed complex short signal subtract signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal1 elements to be subtracted from signal2 elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.154.2.8 NppStatus nppsSub_16u_ISfs (const Npp16u * pSrc, Npp16u * pSrcDst, int nLength, int nScaleFactor)

16-bit unsigned short in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#), signal1 elements to be subtracted from signal2 elements

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.154.2.9 NppStatus nppsSub_16u_Sfs (const Npp16u * pSrc1, const Npp16u * pSrc2, Npp16u * pDst, int nLength, int nScaleFactor)

16-bit unsigned short signal subtract signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal1 elements to be subtracted from signal2 elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.154.2.10 NppStatus nppsSub_32f (const Npp32f * pSrc1, const Npp32f * pSrc2, Npp32f * pDst, int nLength)

32-bit floating point signal subtract signal, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#). signal1 elements to be subtracted from signal2 elements
pDst [Destination Signal Pointer](#).
nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.154.2.11 NppStatus nppsSub_32f_I (const Npp32f * pSrc, Npp32f * pSrcDst, int nLength)

32-bit floating point in place signal subtract signal, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).
pSrcDst [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements
nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.154.2.12 NppStatus nppsSub_32fc (const Npp32fc * pSrc1, const Npp32fc * pSrc2, Npp32fc * pDst, int nLength)

32-bit complex floating point signal subtract signal, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).
pSrc2 [Source Signal Pointer](#). signal1 elements to be subtracted from signal2 elements
pDst [Destination Signal Pointer](#).
nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.154.2.13 NppStatus nppsSub_32fc_I (const Npp32fc * pSrc, Npp32fc * pSrcDst, int nLength)

32-bit complex floating point in place signal subtract signal, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).
pSrcDst [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements
nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.154.2.14 NppStatus nppsSub_32s_ISfs (const Npp32s * *pSrc*, Npp32s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.15 NppStatus nppsSub_32s_Sfs (const Npp32s * *pSrc1*, const Npp32s * *pSrc2*, Npp32s * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal subtract signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.16 NppStatus nppsSub_32sc_ISfs (const Npp32sc * *pSrc*, Npp32sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit complex signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.17 **NppStatus nppsSub_32sc_Sfs** (const Npp32sc * *pSrc1*, const Npp32sc * *pSrc2*, Npp32sc * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed complex integer signal subtract signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.18 **NppStatus nppsSub_64f** (const Npp64f * *pSrc1*, const Npp64f * *pSrc2*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal subtract signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal1 elements to be subtracted from signal2 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.19 **NppStatus nppsSub_64f_I** (const Npp64f * *pSrc*, Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point in place signal subtract signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer, signal1 elements to be subtracted from signal2 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.20 NppStatus nppsSub_64fc (const Npp64fc * *pSrc1*, const Npp64fc * *pSrc2*, Npp64fc * *pDst*, int *nLength*)

64-bit complex floating point signal subtract signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal1 elements to be subtracted from signal2 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.21 NppStatus nppsSub_64fc_I (const Npp64fc * *pSrc*, Npp64fc * *pSrcDst*, int *nLength*)

64-bit complex floating point in place signal subtract signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.22 NppStatus nppsSub_8u_ISfs (const Npp8u * *pSrc*, Npp8u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.154.2.23 `NppStatus nppsSub_8u_Sfs (const Npp8u * pSrc1, const Npp8u * pSrc2, Npp8u * pDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal subtract signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal1 elements to be subtracted from signal2 elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.155 Div

Sample by sample division of the samples of two signals.

Functions

- **NppStatus nppsDiv_8u_Sfs** (const **Npp8u** *pSrc1, const **Npp8u** *pSrc2, **Npp8u** *pDst, int nLength, int nScaleFactor)
8-bit unsigned char signal divide signal, scale, then clamp to saturated value.
- **NppStatus nppsDiv_16u_Sfs** (const **Npp16u** *pSrc1, const **Npp16u** *pSrc2, **Npp16u** *pDst, int nLength, int nScaleFactor)
16-bit unsigned short signal divide signal, scale, then clamp to saturated value.
- **NppStatus nppsDiv_16s_Sfs** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal divide signal, scale, then clamp to saturated value.
- **NppStatus nppsDiv_32s_Sfs** (const **Npp32s** *pSrc1, const **Npp32s** *pSrc2, **Npp32s** *pDst, int nLength, int nScaleFactor)
32-bit signed integer signal divide signal, scale, then clamp to saturated value.
- **NppStatus nppsDiv_16sc_Sfs** (const **Npp16sc** *pSrc1, const **Npp16sc** *pSrc2, **Npp16sc** *pDst, int nLength, int nScaleFactor)
16-bit signed complex short signal divide signal, scale, then clamp to saturated value.
- **NppStatus nppsDiv_32s16s_Sfs** (const **Npp16s** *pSrc1, const **Npp32s** *pSrc2, **Npp16s** *pDst, int nLength, int nScaleFactor)
32-bit signed integer signal divided by 16-bit signed short signal, scale, then clamp to 16-bit signed short saturated value.
- **NppStatus nppsDiv_32f** (const **Npp32f** *pSrc1, const **Npp32f** *pSrc2, **Npp32f** *pDst, int nLength)
32-bit floating point signal divide signal, then clamp to saturated value.
- **NppStatus nppsDiv_64f** (const **Npp64f** *pSrc1, const **Npp64f** *pSrc2, **Npp64f** *pDst, int nLength)
64-bit floating point signal divide signal, then clamp to saturated value.
- **NppStatus nppsDiv_32fc** (const **Npp32fc** *pSrc1, const **Npp32fc** *pSrc2, **Npp32fc** *pDst, int nLength)
32-bit complex floating point signal divide signal, then clamp to saturated value.
- **NppStatus nppsDiv_64fc** (const **Npp64fc** *pSrc1, const **Npp64fc** *pSrc2, **Npp64fc** *pDst, int nLength)
64-bit complex floating point signal divide signal, then clamp to saturated value.
- **NppStatus nppsDiv_8u_ISfs** (const **Npp8u** *pSrc, **Npp8u** *pSrcDst, int nLength, int nScaleFactor)
8-bit unsigned char in place signal divide signal, with scaling, then clamp to saturated value.
- **NppStatus nppsDiv_16u_ISfs** (const **Npp16u** *pSrc, **Npp16u** *pSrcDst, int nLength, int nScaleFactor)
16-bit unsigned short in place signal divide signal, with scaling, then clamp to saturated value.

16-bit unsigned short in place signal divide signal, with scaling, then clamp to saturated value.

- **NppStatus nppsDiv_16s_ISfs** (const **Npp16s** *pSrc, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal divide signal, with scaling, then clamp to saturated value.

- **NppStatus nppsDiv_16sc_ISfs** (const **Npp16sc** *pSrc, **Npp16sc** *pSrcDst, int nLength, int nScaleFactor)

16-bit complex signed short in place signal divide signal, with scaling, then clamp to saturated value.

- **NppStatus nppsDiv_32s_ISfs** (const **Npp32s** *pSrc, **Npp32s** *pSrcDst, int nLength, int nScaleFactor)

32-bit signed integer in place signal divide signal, with scaling, then clamp to saturated value.

- **NppStatus nppsDiv_32f_I** (const **Npp32f** *pSrc, **Npp32f** *pSrcDst, int nLength)

32-bit floating point in place signal divide signal, then clamp to saturated value.

- **NppStatus nppsDiv_64f_I** (const **Npp64f** *pSrc, **Npp64f** *pSrcDst, int nLength)

64-bit floating point in place signal divide signal, then clamp to saturated value.

- **NppStatus nppsDiv_32fc_I** (const **Npp32fc** *pSrc, **Npp32fc** *pSrcDst, int nLength)

32-bit complex floating point in place signal divide signal, then clamp to saturated value.

- **NppStatus nppsDiv_64fc_I** (const **Npp64fc** *pSrc, **Npp64fc** *pSrcDst, int nLength)

64-bit complex floating point in place signal divide signal, then clamp to saturated value.

7.155.1 Detailed Description

Sample by sample division of the samples of two signals.

7.155.2 Function Documentation

7.155.2.1 **NppStatus nppsDiv_16s_ISfs** (const **Npp16s** *pSrc, **Npp16s** *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal divide signal, with scaling, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.155.2.2 NppStatus nppsDiv_16s_Sfs (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, Npp16s * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal divide signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.3 NppStatus nppsDiv_16sc_ISfs (const Npp16sc * *pSrc*, Npp16sc * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit complex signed short in place signal divide signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.4 NppStatus nppsDiv_16sc_Sfs (const Npp16sc * *pSrc1*, const Npp16sc * *pSrc2*, Npp16sc * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed complex short signal divide signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.5 NppStatus nppsDiv_16u_ISfs (const Npp16u * *pSrc*, Npp16u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal divide signal, with scaling, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.155.2.6 NppStatus nppsDiv_16u_Sfs (const Npp16u * *pSrc1*, const Npp16u * *pSrc2*, Npp16u * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal divide signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.155.2.7 NppStatus nppsDiv_32f (const Npp32f * *pSrc1*, const Npp32f * *pSrc2*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal divide signal, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.155.2.8 NppStatus nppsDiv_32f_I (const Npp32f * pSrc, Npp32f * pSrcDst, int nLength)

32-bit floating point in place signal divide signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.9 NppStatus nppsDiv_32fc (const Npp32fc * pSrc1, const Npp32fc * pSrc2, Npp32fc * pDst, int nLength)

32-bit complex floating point signal divide signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.10 NppStatus nppsDiv_32fc_I (const Npp32fc * pSrc, Npp32fc * pSrcDst, int nLength)

32-bit complex floating point in place signal divide signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.11 NppStatus nppsDiv_32s16s_Sfs (const Npp16s * pSrc1, const Npp32s * pSrc2, Npp16s * pDst, int nLength, int nScaleFactor)

32-bit signed integer signal divided by 16-bit signed short signal, scale, then clamp to 16-bit signed short saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.12 `NppStatus nppsDiv_32s_ISfs (const Npp32s * pSrc, Npp32s * pSrcDst, int nLength, int nScaleFactor)`

32-bit signed integer in place signal divide signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.13 `NppStatus nppsDiv_32s_Sfs (const Npp32s * pSrc1, const Npp32s * pSrc2, Npp32s * pDst, int nLength, int nScaleFactor)`

32-bit signed integer signal divide signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.14 NppStatus nppsDiv_64f (const Npp64f * *pSrc1*, const Npp64f * *pSrc2*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal divide signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.15 NppStatus nppsDiv_64f_I (const Npp64f * *pSrc*, Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point in place signal divide signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.16 NppStatus nppsDiv_64fc (const Npp64fc * *pSrc1*, const Npp64fc * *pSrc2*, Npp64fc * *pDst*, int *nLength*)

64-bit complex floating point signal divide signal, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.17 NppStatus nppsDiv_64fc_I (const Npp64fc * *pSrc*, Npp64fc * *pSrcDst*, int *nLength*)

64-bit complex floating point in place signal divide signal, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.18 NppStatus nppsDiv_8u_ISfs (const Npp8u * *pSrc*, Npp8u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal divide signal, with scaling, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.155.2.19 NppStatus nppsDiv_8u_Sfs (const Npp8u * *pSrc1*, const Npp8u * *pSrc2*, Npp8u * *pDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char signal divide signal, scale, then clamp to saturated value.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.156 Div_Round

Sample by sample division of the samples of two signals with rounding.

Functions

- **NppStatus nppsDiv_Round_8u_Sfs** (const **Npp8u** *pSrc1, const **Npp8u** *pSrc2, **Npp8u** *pDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)
8-bit unsigned char signal divide signal, scale, then clamp to saturated value.
- **NppStatus nppsDiv_Round_16u_Sfs** (const **Npp16u** *pSrc1, const **Npp16u** *pSrc2, **Npp16u** *pDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)
16-bit unsigned short signal divide signal, scale, round, then clamp to saturated value.
- **NppStatus nppsDiv_Round_16s_Sfs** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, **Npp16s** *pDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)
16-bit signed short signal divide signal, scale, round, then clamp to saturated value.
- **NppStatus nppsDiv_Round_8u_ISfs** (const **Npp8u** *pSrc, **Npp8u** *pSrcDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)
8-bit unsigned char in place signal divide signal, with scaling, rounding then clamp to saturated value.
- **NppStatus nppsDiv_Round_16u_ISfs** (const **Npp16u** *pSrc, **Npp16u** *pSrcDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)
16-bit unsigned short in place signal divide signal, with scaling, rounding then clamp to saturated value.
- **NppStatus nppsDiv_Round_16s_ISfs** (const **Npp16s** *pSrc, **Npp16s** *pSrcDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)
16-bit signed short in place signal divide signal, with scaling, rounding then clamp to saturated value.

7.156.1 Detailed Description

Sample by sample division of the samples of two signals with rounding.

7.156.2 Function Documentation

7.156.2.1 **NppStatus nppsDiv_Round_16s_ISfs** (const **Npp16s** *pSrc, **Npp16s** *pSrcDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)

16-bit signed short in place signal divide signal, with scaling, rounding then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements
nLength [Signal Length](#).

nRndMode various rounding modes.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.156.2.2 NppStatus nppsDiv_Round_16s_Sfs (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, Npp16s * *pDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)

16-bit signed short signal divide signal, scale, round, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nRndMode various rounding modes.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.156.2.3 NppStatus nppsDiv_Round_16u_ISfs (const Npp16u * *pSrc*, Npp16u * *pSrcDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)

16-bit unsigned short in place signal divide signal, with scaling, rounding then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

nLength [Signal Length](#).

nRndMode various rounding modes.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.156.2.4 NppStatus nppsDiv_Round_16u_Sfs (const Npp16u * *pSrc1*, const Npp16u * *pSrc2*, Npp16u * *pDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)

16-bit unsigned short signal divide signal, scale, round, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nRndMode various rounding modes.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.156.2.5 NppStatus nppsDiv_Round_8u_ISfs (const Npp8u * *pSrc*, Npp8u * *pSrcDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)

8-bit unsigned char in place signal divide signal, with scaling, rounding then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

nLength [Signal Length](#).

nRndMode various rounding modes.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.156.2.6 NppStatus nppsDiv_Round_8u_Sfs (const Npp8u * *pSrc1*, const Npp8u * *pSrc2*, Npp8u * *pDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)

8-bit unsigned char signal divide signal, scale, then clamp to saturated value.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nRndMode various rounding modes.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.157 Abs

Absolute value of each sample of a signal.

Functions

- **NppStatus nppsAbs_16s** (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength)
16-bit signed short signal absolute value.
- **NppStatus nppsAbs_32s** (const **Npp32s** *pSrc, **Npp32s** *pDst, int nLength)
32-bit signed integer signal absolute value.
- **NppStatus nppsAbs_32f** (const **Npp32f** *pSrc, **Npp32f** *pDst, int nLength)
32-bit floating point signal absolute value.
- **NppStatus nppsAbs_64f** (const **Npp64f** *pSrc, **Npp64f** *pDst, int nLength)
64-bit floating point signal absolute value.
- **NppStatus nppsAbs_16s_I** (**Npp16s** *pSrcDst, int nLength)
16-bit signed short signal absolute value.
- **NppStatus nppsAbs_32s_I** (**Npp32s** *pSrcDst, int nLength)
32-bit signed integer signal absolute value.
- **NppStatus nppsAbs_32f_I** (**Npp32f** *pSrcDst, int nLength)
32-bit floating point signal absolute value.
- **NppStatus nppsAbs_64f_I** (**Npp64f** *pSrcDst, int nLength)
64-bit floating point signal absolute value.

7.157.1 Detailed Description

Absolute value of each sample of a signal.

7.157.2 Function Documentation

7.157.2.1 NppStatus nppsAbs_16s (const Npp16s *pSrc, Npp16s *pDst, int nLength)

16-bit signed short signal absolute value.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.157.2.2 NppStatus nppsAbs_16s_I (Npp16s * *pSrcDst*, int *nLength*)

16-bit signed short signal absolute value.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.157.2.3 NppStatus nppsAbs_32f (const Npp32f * *pSrc*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal absolute value.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.157.2.4 NppStatus nppsAbs_32f_I (Npp32f * *pSrcDst*, int *nLength*)

32-bit floating point signal absolute value.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.157.2.5 NppStatus nppsAbs_32s (const Npp32s * *pSrc*, Npp32s * *pDst*, int *nLength*)

32-bit signed integer signal absolute value.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.157.2.6 NppStatus nppsAbs_32s_I (Npp32s * *pSrcDst*, int *nLength*)

32-bit signed integer signal absolute value.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.157.2.7 NppStatus nppsAbs_64f (const Npp64f * *pSrc*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal absolute value.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.157.2.8 NppStatus nppsAbs_64f_I (Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point signal absolute value.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158 Sqr

Squares each sample of a signal.

Functions

- **NppStatus nppsSqr_32f** (const **Npp32f** *pSrc, **Npp32f** *pDst, int nLength)
32-bit floating point signal squared.
- **NppStatus nppsSqr_64f** (const **Npp64f** *pSrc, **Npp64f** *pDst, int nLength)
64-bit floating point signal squared.
- **NppStatus nppsSqr_32fc** (const **Npp32fc** *pSrc, **Npp32fc** *pDst, int nLength)
32-bit complex floating point signal squared.
- **NppStatus nppsSqr_64fc** (const **Npp64fc** *pSrc, **Npp64fc** *pDst, int nLength)
64-bit complex floating point signal squared.
- **NppStatus nppsSqr_32f_I** (**Npp32f** *pSrcDst, int nLength)
32-bit floating point signal squared.
- **NppStatus nppsSqr_64f_I** (**Npp64f** *pSrcDst, int nLength)
64-bit floating point signal squared.
- **NppStatus nppsSqr_32fc_I** (**Npp32fc** *pSrcDst, int nLength)
32-bit complex floating point signal squared.
- **NppStatus nppsSqr_64fc_I** (**Npp64fc** *pSrcDst, int nLength)
64-bit complex floating point signal squared.
- **NppStatus nppsSqr_8u_Sfs** (const **Npp8u** *pSrc, **Npp8u** *pDst, int nLength, int nScaleFactor)
8-bit unsigned char signal squared, scale, then clamp to saturated value.
- **NppStatus nppsSqr_16u_Sfs** (const **Npp16u** *pSrc, **Npp16u** *pDst, int nLength, int nScaleFactor)
16-bit unsigned short signal squared, scale, then clamp to saturated value.
- **NppStatus nppsSqr_16s_Sfs** (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal squared, scale, then clamp to saturated value.
- **NppStatus nppsSqr_16sc_Sfs** (const **Npp16sc** *pSrc, **Npp16sc** *pDst, int nLength, int nScaleFactor)
16-bit complex signed short signal squared, scale, then clamp to saturated value.
- **NppStatus nppsSqr_8u_ISfs** (**Npp8u** *pSrcDst, int nLength, int nScaleFactor)
8-bit unsigned char signal squared, scale, then clamp to saturated value.
- **NppStatus nppsSqr_16u_ISfs** (**Npp16u** *pSrcDst, int nLength, int nScaleFactor)
16-bit unsigned short signal squared, scale, then clamp to saturated value.

- **NppStatus nppsSqr_16s_ISfs** (**Npp16s** *pSrcDst, int nLength, int nScaleFactor)
16-bit signed short signal squared, scale, then clamp to saturated value.
- **NppStatus nppsSqr_16sc_ISfs** (**Npp16sc** *pSrcDst, int nLength, int nScaleFactor)
16-bit complex signed short signal squared, scale, then clamp to saturated value.

7.158.1 Detailed Description

Squares each sample of a signal.

7.158.2 Function Documentation

7.158.2.1 NppStatus nppsSqr_16s_ISfs (Npp16s *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short signal squared, scale, then clamp to saturated value.

Parameters:

pSrcDst [In-Place Signal Pointer](#).
nLength [Signal Length](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.158.2.2 NppStatus nppsSqr_16s_Sfs (const Npp16s *pSrc, Npp16s *pDst, int nLength, int nScaleFactor)

16-bit signed short signal squared, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).
pDst [Destination Signal Pointer](#).
nLength [Signal Length](#).
nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.158.2.3 NppStatus nppsSqr_16sc_ISfs (Npp16sc *pSrcDst, int nLength, int nScaleFactor)

16-bit complex signed short signal squared, scale, then clamp to saturated value.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.4 NppStatus nppsSqr_16sc_Sfs (const Npp16sc * *pSrc*, Npp16sc * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit complex signed short signal squared, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.5 NppStatus nppsSqr_16u_ISfs (Npp16u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal squared, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.6 NppStatus nppsSqr_16u_Sfs (const Npp16u * *pSrc*, Npp16u * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal squared, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.7 NppStatus nppsSqr_32f (const Npp32f * *pSrc*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal squared.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.8 NppStatus nppsSqr_32f_I (Npp32f * *pSrcDst*, int *nLength*)

32-bit floating point signal squared.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.9 NppStatus nppsSqr_32fc (const Npp32fc * *pSrc*, Npp32fc * *pDst*, int *nLength*)

32-bit complex floating point signal squared.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.10 NppStatus nppsSqr_32fc_I (Npp32fc * *pSrcDst*, int *nLength*)

32-bit complex floating point signal squared.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.11 NppStatus nppsSqr_64f (const Npp64f * *pSrc*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal squared.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.12 NppStatus nppsSqr_64f_I (Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point signal squared.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.13 NppStatus nppsSqr_64fc (const Npp64fc * *pSrc*, Npp64fc * *pDst*, int *nLength*)

64-bit complex floating point signal squared.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.14 NppStatus nppsSqr_64fc_I (Npp64fc * *pSrcDst*, int *nLength*)

64-bit complex floating point signal squared.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.158.2.15 `NppStatus nppsSqr_8u_ISfs (Npp8u * pSrcDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal squared, scale, then clamp to saturated value.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.158.2.16 `NppStatus nppsSqr_8u_Sfs (const Npp8u * pSrc, Npp8u * pDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal squared, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.159 Sqrt

Square root of each sample of a signal.

Functions

- **NppStatus nppsSqrt_32f** (const **Npp32f** *pSrc, **Npp32f** *pDst, int nLength)
32-bit floating point signal square root.
- **NppStatus nppsSqrt_64f** (const **Npp64f** *pSrc, **Npp64f** *pDst, int nLength)
64-bit floating point signal square root.
- **NppStatus nppsSqrt_32fc** (const **Npp32fc** *pSrc, **Npp32fc** *pDst, int nLength)
32-bit complex floating point signal square root.
- **NppStatus nppsSqrt_64fc** (const **Npp64fc** *pSrc, **Npp64fc** *pDst, int nLength)
64-bit complex floating point signal square root.
- **NppStatus nppsSqrt_32f_I** (**Npp32f** *pSrcDst, int nLength)
32-bit floating point signal square root.
- **NppStatus nppsSqrt_64f_I** (**Npp64f** *pSrcDst, int nLength)
64-bit floating point signal square root.
- **NppStatus nppsSqrt_32fc_I** (**Npp32fc** *pSrcDst, int nLength)
32-bit complex floating point signal square root.
- **NppStatus nppsSqrt_64fc_I** (**Npp64fc** *pSrcDst, int nLength)
64-bit complex floating point signal square root.
- **NppStatus nppsSqrt_8u_Sfs** (const **Npp8u** *pSrc, **Npp8u** *pDst, int nLength, int nScaleFactor)
8-bit unsigned char signal square root, scale, then clamp to saturated value.
- **NppStatus nppsSqrt_16u_Sfs** (const **Npp16u** *pSrc, **Npp16u** *pDst, int nLength, int nScaleFactor)
16-bit unsigned short signal square root, scale, then clamp to saturated value.
- **NppStatus nppsSqrt_16s_Sfs** (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal square root, scale, then clamp to saturated value.
- **NppStatus nppsSqrt_16sc_Sfs** (const **Npp16sc** *pSrc, **Npp16sc** *pDst, int nLength, int nScaleFactor)
16-bit complex signed short signal square root, scale, then clamp to saturated value.
- **NppStatus nppsSqrt_64s_Sfs** (const **Npp64s** *pSrc, **Npp64s** *pDst, int nLength, int nScaleFactor)
64-bit signed integer signal square root, scale, then clamp to saturated value.
- **NppStatus nppsSqrt_32s16s_Sfs** (const **Npp32s** *pSrc, **Npp16s** *pDst, int nLength, int nScaleFactor)
32-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.

- **NppStatus nppsSqrt_64s16s_Sfs** (const **Npp64s** *pSrc, **Npp16s** *pDst, int nLength, int nScaleFactor)
64-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.
- **NppStatus nppsSqrt_8u_ISfs** (**Npp8u** *pSrcDst, int nLength, int nScaleFactor)
8-bit unsigned char signal square root, scale, then clamp to saturated value.
- **NppStatus nppsSqrt_16u_ISfs** (**Npp16u** *pSrcDst, int nLength, int nScaleFactor)
16-bit unsigned short signal square root, scale, then clamp to saturated value.
- **NppStatus nppsSqrt_16s_ISfs** (**Npp16s** *pSrcDst, int nLength, int nScaleFactor)
16-bit signed short signal square root, scale, then clamp to saturated value.
- **NppStatus nppsSqrt_16sc_ISfs** (**Npp16sc** *pSrcDst, int nLength, int nScaleFactor)
16-bit complex signed short signal square root, scale, then clamp to saturated value.
- **NppStatus nppsSqrt_64s_ISfs** (**Npp64s** *pSrcDst, int nLength, int nScaleFactor)
64-bit signed integer signal square root, scale, then clamp to saturated value.

7.159.1 Detailed Description

Square root of each sample of a signal.

7.159.2 Function Documentation

7.159.2.1 NppStatus nppsSqrt_16s_ISfs (**Npp16s** *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short signal square root, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.2 NppStatus nppsSqrt_16s_Sfs (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength, int nScaleFactor)

16-bit signed short signal square root, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.3 NppStatus nppsSqrt_16sc_ISfs (Npp16sc * pSrcDst, int nLength, int nScaleFactor)

16-bit complex signed short signal square root, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.4 NppStatus nppsSqrt_16sc_Sfs (const Npp16sc * pSrc, Npp16sc * pDst, int nLength, int nScaleFactor)

16-bit complex signed short signal square root, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.5 NppStatus nppsSqrt_16u_ISfs (Npp16u * pSrcDst, int nLength, int nScaleFactor)

16-bit unsigned short signal square root, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.6 NppStatus nppsSqrt_16u_Sfs (const Npp16u * *pSrc*, Npp16u * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal square root, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.7 NppStatus nppsSqrt_32f (const Npp32f * *pSrc*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal square root.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.8 NppStatus nppsSqrt_32f_I (Npp32f * *pSrcDst*, int *nLength*)

32-bit floating point signal square root.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.9 NppStatus nppsSqrt_32fc (const Npp32fc * *pSrc*, Npp32fc * *pDst*, int *nLength*)

32-bit complex floating point signal square root.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.10 NppStatus nppsSqrt_32fc_I (Npp32fc * pSrcDst, int nLength)

32-bit complex floating point signal square root.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.11 NppStatus nppsSqrt_32s16s_Sfs (const Npp32s * pSrc, Npp16s * pDst, int nLength, int nScaleFactor)

32-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.12 NppStatus nppsSqrt_64f (const Npp64f * pSrc, Npp64f * pDst, int nLength)

64-bit floating point signal square root.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.13 NppStatus nppsSqrt_64f_I (Npp64f * pSrcDst, int nLength)

64-bit floating point signal square root.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.14 NppStatus nppsSqrt_64fc (const Npp64fc * pSrc, Npp64fc * pDst, int nLength)

64-bit complex floating point signal square root.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.15 NppStatus nppsSqrt_64fc_I (Npp64fc * pSrcDst, int nLength)

64-bit complex floating point signal square root.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.16 NppStatus nppsSqrt_64s16s_Sfs (const Npp64s * pSrc, Npp16s * pDst, int nLength, int nScaleFactor)

64-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.17 NppStatus nppsSqrt_64s_ISfs (Npp64s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

64-bit signed integer signal square root, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.18 NppStatus nppsSqrt_64s_Sfs (const Npp64s * *pSrc*, Npp64s * *pDst*, int *nLength*, int *nScaleFactor*)

64-bit signed integer signal square root, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.19 NppStatus nppsSqrt_8u_ISfs (Npp8u * *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char signal square root, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.159.2.20 NppStatus nppsSqrt_8u_Sfs (const Npp8u * *pSrc*, Npp8u * *pDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char signal square root, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.160 Cubrt

Cube root of each sample of a signal.

Functions

- **NppStatus nppsCubrt_32f** (const **Npp32f** *pSrc, **Npp32f** *pDst, int nLength)
32-bit floating point signal cube root.
- **NppStatus nppsCubrt_32s16s_Sfs** (const **Npp32s** *pSrc, **Npp16s** *pDst, int nLength, int nScaleFactor)
32-bit signed integer signal cube root, scale, then clamp to 16-bit signed integer saturated value.

7.160.1 Detailed Description

Cube root of each sample of a signal.

7.160.2 Function Documentation

7.160.2.1 NppStatus nppsCubrt_32f (const Npp32f *pSrc, Npp32f *pDst, int nLength)

32-bit floating point signal cube root.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.160.2.2 NppStatus nppsCubrt_32s16s_Sfs (const Npp32s *pSrc, Npp16s *pDst, int nLength, int nScaleFactor)

32-bit signed integer signal cube root, scale, then clamp to 16-bit signed integer saturated value.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.161 Exp

E raised to the power of each sample of a signal.

Functions

- **NppStatus nppsExp_32f** (const **Npp32f** *pSrc, **Npp32f** *pDst, int nLength)
32-bit floating point signal exponent.
- **NppStatus nppsExp_64f** (const **Npp64f** *pSrc, **Npp64f** *pDst, int nLength)
64-bit floating point signal exponent.
- **NppStatus nppsExp_32f64f** (const **Npp32f** *pSrc, **Npp64f** *pDst, int nLength)
32-bit floating point signal exponent with 64-bit floating point result.
- **NppStatus nppsExp_32f_I** (**Npp32f** *pSrcDst, int nLength)
32-bit floating point signal exponent.
- **NppStatus nppsExp_64f_I** (**Npp64f** *pSrcDst, int nLength)
64-bit floating point signal exponent.
- **NppStatus nppsExp_16s_Sfs** (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal exponent, scale, then clamp to saturated value.
- **NppStatus nppsExp_32s_Sfs** (const **Npp32s** *pSrc, **Npp32s** *pDst, int nLength, int nScaleFactor)
32-bit signed integer signal exponent, scale, then clamp to saturated value.
- **NppStatus nppsExp_64s_Sfs** (const **Npp64s** *pSrc, **Npp64s** *pDst, int nLength, int nScaleFactor)
64-bit signed integer signal exponent, scale, then clamp to saturated value.
- **NppStatus nppsExp_16s_ISfs** (**Npp16s** *pSrcDst, int nLength, int nScaleFactor)
16-bit signed short signal exponent, scale, then clamp to saturated value.
- **NppStatus nppsExp_32s_ISfs** (**Npp32s** *pSrcDst, int nLength, int nScaleFactor)
32-bit signed integer signal exponent, scale, then clamp to saturated value.
- **NppStatus nppsExp_64s_ISfs** (**Npp64s** *pSrcDst, int nLength, int nScaleFactor)
64-bit signed integer signal exponent, scale, then clamp to saturated value.

7.161.1 Detailed Description

E raised to the power of each sample of a signal.

7.161.2 Function Documentation

7.161.2.1 NppStatus nppsExp_16s_ISfs (Npp16s *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short signal exponent, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.161.2.2 NppStatus nppsExp_16s_Sfs (const Npp16s * pSrc, Npp16s * pDst, int nLength, int nScaleFactor)

16-bit signed short signal exponent, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.161.2.3 NppStatus nppsExp_32f (const Npp32f * pSrc, Npp32f * pDst, int nLength)

32-bit floating point signal exponent.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.161.2.4 NppStatus nppsExp_32f64f (const Npp32f * pSrc, Npp64f * pDst, int nLength)

32-bit floating point signal exponent with 64-bit floating point result.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.161.2.5 NppStatus nppsExp_32f_I (Npp32f * *pSrcDst*, int *nLength*)

32-bit floating point signal exponent.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.161.2.6 NppStatus nppsExp_32s_ISfs (Npp32s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal exponent, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.161.2.7 NppStatus nppsExp_32s_Sfs (const Npp32s * *pSrc*, Npp32s * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal exponent, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.161.2.8 NppStatus nppsExp_64f (const Npp64f * *pSrc*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal exponent.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.161.2.9 NppStatus nppsExp_64f_I (Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point signal exponent.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.161.2.10 NppStatus nppsExp_64s_ISfs (Npp64s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

64-bit signed integer signal exponent, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.161.2.11 NppStatus nppsExp_64s_Sfs (const Npp64s * *pSrc*, Npp64s * *pDst*, int *nLength*, int *nScaleFactor*)

64-bit signed integer signal exponent, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.162 Ln

Natural logarithm of each sample of a signal.

Functions

- **NppStatus nppsLn_32f** (const **Npp32f** *pSrc, **Npp32f** *pDst, int nLength)
32-bit floating point signal natural logarithm.
- **NppStatus nppsLn_64f** (const **Npp64f** *pSrc, **Npp64f** *pDst, int nLength)
64-bit floating point signal natural logarithm.
- **NppStatus nppsLn_64f32f** (const **Npp64f** *pSrc, **Npp32f** *pDst, int nLength)
64-bit floating point signal natural logarithm with 32-bit floating point result.
- **NppStatus nppsLn_32f_I** (**Npp32f** *pSrcDst, int nLength)
32-bit floating point signal natural logarithm.
- **NppStatus nppsLn_64f_I** (**Npp64f** *pSrcDst, int nLength)
64-bit floating point signal natural logarithm.
- **NppStatus nppsLn_16s_Sfs** (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength, int nScaleFactor)
16-bit signed short signal natural logarithm, scale, then clamp to saturated value.
- **NppStatus nppsLn_32s_Sfs** (const **Npp32s** *pSrc, **Npp32s** *pDst, int nLength, int nScaleFactor)
32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.
- **NppStatus nppsLn_32s16s_Sfs** (const **Npp32s** *pSrc, **Npp16s** *pDst, int nLength, int nScaleFactor)
32-bit signed integer signal natural logarithm, scale, then clamp to 16-bit signed short saturated value.
- **NppStatus nppsLn_16s_ISfs** (**Npp16s** *pSrcDst, int nLength, int nScaleFactor)
16-bit signed short signal natural logarithm, scale, then clamp to saturated value.
- **NppStatus nppsLn_32s_ISfs** (**Npp32s** *pSrcDst, int nLength, int nScaleFactor)
32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.

7.162.1 Detailed Description

Natural logarithm of each sample of a signal.

7.162.2 Function Documentation

7.162.2.1 NppStatus nppsLn_16s_ISfs (Npp16s *pSrcDst, int nLength, int nScaleFactor)

16-bit signed short signal natural logarithm, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.162.2.2 NppStatus nppsLn_16s_Sfs (const Npp16s * *pSrc*, Npp16s * *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal natural logarithm, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.162.2.3 NppStatus nppsLn_32f (const Npp32f * *pSrc*, Npp32f * *pDst*, int *nLength*)

32-bit floating point signal natural logarithm.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.162.2.4 NppStatus nppsLn_32f_I (Npp32f * *pSrcDst*, int *nLength*)

32-bit floating point signal natural logarithm.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.162.2.5 NppStatus nppsLn_32s16s_Sfs (const Npp32s * *pSrc*, Npp16s * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal natural logarithm, scale, then clamp to 16-bit signed short saturated value.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.162.2.6 NppStatus nppsLn_32s_ISfs (Npp32s * *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.162.2.7 NppStatus nppsLn_32s_Sfs (const Npp32s * *pSrc*, Npp32s * *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.162.2.8 NppStatus nppsLn_64f (const Npp64f * *pSrc*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal natural logarithm.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.162.2.9 NppStatus nppsLn_64f32f (const Npp64f * *pSrc*, Npp32f * *pDst*, int *nLength*)

64-bit floating point signal natural logarithm with 32-bit floating point result.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.162.2.10 NppStatus nppsLn_64f_I (Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point signal natural logarithm.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.163 10Log10

Ten times the decimal logarithm of each sample of a signal.

Functions

- **NppStatus npps10Log10_32s_Sfs** (const **Npp32s** *pSrc, **Npp32s** *pDst, int nLength, int nScaleFactor)
32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.
- **NppStatus npps10Log10_32s_ISfs** (**Npp32s** *pSrcDst, int nLength, int nScaleFactor)
32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.

7.163.1 Detailed Description

Ten times the decimal logarithm of each sample of a signal.

7.163.2 Function Documentation

7.163.2.1 NppStatus npps10Log10_32s_ISfs (Npp32s *pSrcDst, int nLength, int nScaleFactor)

32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.163.2.2 NppStatus npps10Log10_32s_Sfs (const Npp32s *pSrc, Npp32s *pDst, int nLength, int nScaleFactor)

32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.
nScaleFactor Integer Result Scaling.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.164 SumLn

Sums up the natural logarithm of each sample of a signal.

Functions

- **NppStatus nppsSumLnGetBufferSize_32f** (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for 32f SumLn.
- **NppStatus nppsSumLn_32f** (const Npp32f *pSrc, int nLength, Npp32f *pDst, Npp8u *pDeviceBuffer)
32-bit floating point signal sum natural logarithm.
- **NppStatus nppsSumLnGetBufferSize_64f** (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for 64f SumLn.
- **NppStatus nppsSumLn_64f** (const Npp64f *pSrc, int nLength, Npp64f *pDst, Npp8u *pDeviceBuffer)
64-bit floating point signal sum natural logarithm.
- **NppStatus nppsSumLnGetBufferSize_32f64f** (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for 32f64f SumLn.
- **NppStatus nppsSumLn_32f64f** (const Npp32f *pSrc, int nLength, Npp64f *pDst, Npp8u *pDeviceBuffer)
32-bit floating point input, 64-bit floating point output signal sum natural logarithm.
- **NppStatus nppsSumLnGetBufferSize_16s32f** (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for 16s32f SumLn.
- **NppStatus nppsSumLn_16s32f** (const Npp16s *pSrc, int nLength, Npp32f *pDst, Npp8u *pDeviceBuffer)
16-bit signed short integer input, 32-bit floating point output signal sum natural logarithm.

7.164.1 Detailed Description

Sums up the natural logarithm of each sample of a signal.

7.164.2 Function Documentation

7.164.2.1 NppStatus nppsSumLn_16s32f (const Npp16s *pSrc, int nLength, Npp32f *pDst, Npp8u *pDeviceBuffer)

16-bit signed short integer input, 32-bit floating point output signal sum natural logarithm.

Parameters:

pSrc Source Signal Pointer.

nLength [Signal Length](#).

pDst Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.164.2.2 NppStatus nppsSumLn_32f (const Npp32f * pSrc, int nLength, Npp32f * pDst, Npp8u * pDeviceBuffer)

32-bit floating point signal sum natural logarithm.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.164.2.3 NppStatus nppsSumLn_32f64f (const Npp32f * pSrc, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

32-bit floating point input, 64-bit floating point output signal sum natural logarithm.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.164.2.4 NppStatus nppsSumLn_64f (const Npp64f * pSrc, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

64-bit floating point signal sum natural logarithm.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.164.2.5 NppStatus nppsSumLnGetBufferSize_16s32f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for 16s32f SumLn.

This primitive provides the correct buffer size for nppsSumLn_16s32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.164.2.6 NppStatus nppsSumLnGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for 32f SumLn.

This primitive provides the correct buffer size for nppsSumLn_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.164.2.7 NppStatus nppsSumLnGetBufferSize_32f64f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for 32f64f SumLn.

This primitive provides the correct buffer size for nppsSumLn_32f64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.164.2.8 NppStatus nppsSumLnGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for 64f SumLn.

This primitive provides the correct buffer size for nppsSumLn_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.165 Arctan

Inverse tangent of each sample of a signal.

Functions

- `NppStatus nppsArctan_32f` (const `Npp32f` *pSrc, `Npp32f` *pDst, int nLength)
32-bit floating point signal inverse tangent.
- `NppStatus nppsArctan_64f` (const `Npp64f` *pSrc, `Npp64f` *pDst, int nLength)
64-bit floating point signal inverse tangent.
- `NppStatus nppsArctan_32f_I` (`Npp32f` *pSrcDst, int nLength)
32-bit floating point signal inverse tangent.
- `NppStatus nppsArctan_64f_I` (`Npp64f` *pSrcDst, int nLength)
64-bit floating point signal inverse tangent.

7.165.1 Detailed Description

Inverse tangent of each sample of a signal.

7.165.2 Function Documentation

7.165.2.1 `NppStatus nppsArctan_32f` (const `Npp32f` *pSrc, `Npp32f` *pDst, int nLength)

32-bit floating point signal inverse tangent.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.165.2.2 `NppStatus nppsArctan_32f_I` (`Npp32f` *pSrcDst, int nLength)

32-bit floating point signal inverse tangent.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.165.2.3 NppStatus nppsArctan_64f (const Npp64f * *pSrc*, Npp64f * *pDst*, int *nLength*)

64-bit floating point signal inverse tangent.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.165.2.4 NppStatus nppsArctan_64f_I (Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point signal inverse tangent.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.166 Normalize

Normalize each sample of a real or complex signal using offset and division operations.

Functions

- **NppStatus nppsNormalize_32f** (const **Npp32f** *pSrc, **Npp32f** *pDst, int nLength, **Npp32f** vSub, **Npp32f** vDiv)
32-bit floating point signal normalize.
- **NppStatus nppsNormalize_32fc** (const **Npp32fc** *pSrc, **Npp32fc** *pDst, int nLength, **Npp32fc** vSub, **Npp32f** vDiv)
32-bit complex floating point signal normalize.
- **NppStatus nppsNormalize_64f** (const **Npp64f** *pSrc, **Npp64f** *pDst, int nLength, **Npp64f** vSub, **Npp64f** vDiv)
64-bit floating point signal normalize.
- **NppStatus nppsNormalize_64fc** (const **Npp64fc** *pSrc, **Npp64fc** *pDst, int nLength, **Npp64fc** vSub, **Npp64f** vDiv)
64-bit complex floating point signal normalize.
- **NppStatus nppsNormalize_16s_Sfs** (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength, **Npp16s** vSub, int vDiv, int nScaleFactor)
16-bit signed short signal normalize, scale, then clamp to saturated value.
- **NppStatus nppsNormalize_16sc_Sfs** (const **Npp16sc** *pSrc, **Npp16sc** *pDst, int nLength, **Npp16sc** vSub, int vDiv, int nScaleFactor)
16-bit complex signed short signal normalize, scale, then clamp to saturated value.

7.166.1 Detailed Description

Normalize each sample of a real or complex signal using offset and division operations.

7.166.2 Function Documentation

7.166.2.1 NppStatus nppsNormalize_16s_Sfs (const Npp16s * pSrc, Npp16s * pDst, int nLength, Npp16s vSub, int vDiv, int nScaleFactor)

16-bit signed short signal normalize, scale, then clamp to saturated value.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

vSub value subtracted from each signal element before division

vDiv divisor of post-subtracted signal element dividend

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.166.2.2 `NppStatus nppsNormalize_16sc_Sfs (const Npp16sc * pSrc, Npp16sc * pDst, int nLength, Npp16sc vSub, int vDiv, int nScaleFactor)`

16-bit complex signed short signal normalize, scale, then clamp to saturated value.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

vSub value subtracted from each signal element before division

vDiv divisor of post-subtracted signal element dividend

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.166.2.3 `NppStatus nppsNormalize_32f (const Npp32f * pSrc, Npp32f * pDst, int nLength, Npp32f vSub, Npp32f vDiv)`

32-bit floating point signal normalize.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

vSub value subtracted from each signal element before division

vDiv divisor of post-subtracted signal element dividend

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.166.2.4 `NppStatus nppsNormalize_32fc (const Npp32fc * pSrc, Npp32fc * pDst, int nLength, Npp32fc vSub, Npp32fc vDiv)`

32-bit complex floating point signal normalize.

Parameters:

pSrc [Source Signal Pointer](#).

pDst Destination Signal Pointer.

nLength Signal Length.

vSub value subtracted from each signal element before division

vDiv divisor of post-subtracted signal element dividend

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.166.2.5 NppStatus nppsNormalize_64f (const Npp64f * pSrc, Npp64f * pDst, int nLength, Npp64f vSub, Npp64f vDiv)

64-bit floating point signal normalize.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

vSub value subtracted from each signal element before division

vDiv divisor of post-subtracted signal element dividend

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.166.2.6 NppStatus nppsNormalize_64fc (const Npp64fc * pSrc, Npp64fc * pDst, int nLength, Npp64fc vSub, Npp64fc vDiv)

64-bit complex floating point signal normalize.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

vSub value subtracted from each signal element before division

vDiv divisor of post-subtracted signal element dividend

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.167 Cauchy, CauchyD, and CauchyDD2

Determine Cauchy robust error function and its first and second derivatives for each sample of a signal.

Functions

- **NppStatus nppsCauchy_32f_I** (**Npp32f** *pSrcDst, int nLength, **Npp32f** nParam)
32-bit floating point signal Cauchy error calculation.
- **NppStatus nppsCauchyD_32f_I** (**Npp32f** *pSrcDst, int nLength, **Npp32f** nParam)
32-bit floating point signal Cauchy first derivative.
- **NppStatus nppsCauchyDD2_32f_I** (**Npp32f** *pSrcDst, **Npp32f** *pD2FVal, int nLength, **Npp32f** nParam)
32-bit floating point signal Cauchy first and second derivatives.

7.167.1 Detailed Description

Determine Cauchy robust error function and its first and second derivatives for each sample of a signal.

7.167.2 Function Documentation

7.167.2.1 NppStatus nppsCauchy_32f_I (Npp32f * pSrcDst, int nLength, Npp32f nParam)

32-bit floating point signal Cauchy error calculation.

Parameters:

- pSrcDst** [In-Place Signal Pointer](#).
nLength [Signal Length](#).
nParam constant used in Cauchy formula

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.167.2.2 NppStatus nppsCauchyD_32f_I (Npp32f * pSrcDst, int nLength, Npp32f nParam)

32-bit floating point signal Cauchy first derivative.

Parameters:

- pSrcDst** [In-Place Signal Pointer](#).
nLength [Signal Length](#).
nParam constant used in Cauchy formula

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.167.2.3 NppStatus nppsCauchyDD2_32f_I (Npp32f * *pSrcDst*, Npp32f * *pD2FVal*, int *nLength*, Npp32f *nParam*)

32-bit floating point signal Cauchy first and second derivatives.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

pD2FVal [Source Signal Pointer](#). This signal contains the second derivative of the source signal.

nLength [Signal Length](#).

nParam constant used in Cauchy formula

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.168 Logical And Shift Operations

Modules

- [AndC](#)

Bitwise AND of a constant and each sample of a signal.

- [And](#)

Sample by sample bitwise AND of samples from two signals.

- [OrC](#)

Bitwise OR of a constant and each sample of a signal.

- [Or](#)

Sample by sample bitwise OR of the samples from two signals.

- [XorC](#)

Bitwise XOR of a constant and each sample of a signal.

- [Xor](#)

Sample by sample bitwise XOR of the samples from two signals.

- [Not](#)

Bitwise NOT of each sample of a signal.

- [LShiftC](#)

Left shifts the bits of each sample of a signal by a constant amount.

- [RShiftC](#)

Right shifts the bits of each sample of a signal by a constant amount.

7.169 AndC

Bitwise AND of a constant and each sample of a signal.

Functions

- **NppStatus nppsAndC_8u** (const **Npp8u** *pSrc, **Npp8u** nValue, **Npp8u** *pDst, int nLength)
8-bit unsigned char signal and with constant.
- **NppStatus nppsAndC_16u** (const **Npp16u** *pSrc, **Npp16u** nValue, **Npp16u** *pDst, int nLength)
16-bit unsigned short signal and with constant.
- **NppStatus nppsAndC_32u** (const **Npp32u** *pSrc, **Npp32u** nValue, **Npp32u** *pDst, int nLength)
32-bit unsigned integer signal and with constant.
- **NppStatus nppsAndC_8u_I** (**Npp8u** nValue, **Npp8u** *pSrcDst, int nLength)
8-bit unsigned char in place signal and with constant.
- **NppStatus nppsAndC_16u_I** (**Npp16u** nValue, **Npp16u** *pSrcDst, int nLength)
16-bit unsigned short in place signal and with constant.
- **NppStatus nppsAndC_32u_I** (**Npp32u** nValue, **Npp32u** *pSrcDst, int nLength)
32-bit unsigned signed integer in place signal and with constant.

7.169.1 Detailed Description

Bitwise AND of a constant and each sample of a signal.

7.169.2 Function Documentation

7.169.2.1 NppStatus nppsAndC_16u (const Npp16u *pSrc, Npp16u nValue, Npp16u *pDst, int nLength)

16-bit unsigned short signal and with constant.

Parameters:

- pSrc** [Source Signal Pointer](#).
nValue Constant value to be anded with each vector element
pDst [Destination Signal Pointer](#).
nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.169.2.2 NppStatus nppsAndC_16u_I (Npp16u *nValue*, Npp16u * *pSrcDst*, int *nLength*)

16-bit unsigned short in place signal and with constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be added with each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.169.2.3 NppStatus nppsAndC_32u (const Npp32u * *pSrc*, Npp32u *nValue*, Npp32u * *pDst*, int *nLength*)

32-bit unsigned integer signal and with constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be added with each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.169.2.4 NppStatus nppsAndC_32u_I (Npp32u *nValue*, Npp32u * *pSrcDst*, int *nLength*)

32-bit unsigned signed integer in place signal and with constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be added with each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.169.2.5 NppStatus nppsAndC_8u (const Npp8u * *pSrc*, Npp8u *nValue*, Npp8u * *pDst*, int *nLength*)

8-bit unsigned char signal and with constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be anded with each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.169.2.6 NppStatus nppsAndC_8u_I (Npp8u nValue, Npp8u * pSrcDst, int nLength)

8-bit unsigned char in place signal and with constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be anded with each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.170 And

Sample by sample bitwise AND of samples from two signals.

Functions

- **NppStatus nppsAnd_8u** (const **Npp8u** *pSrc1, const **Npp8u** *pSrc2, **Npp8u** *pDst, int nLength)
8-bit unsigned char signal and with signal.
- **NppStatus nppsAnd_16u** (const **Npp16u** *pSrc1, const **Npp16u** *pSrc2, **Npp16u** *pDst, int nLength)
16-bit unsigned short signal and with signal.
- **NppStatus nppsAnd_32u** (const **Npp32u** *pSrc1, const **Npp32u** *pSrc2, **Npp32u** *pDst, int nLength)
32-bit unsigned integer signal and with signal.
- **NppStatus nppsAnd_8u_I** (const **Npp8u** *pSrc, **Npp8u** *pSrcDst, int nLength)
8-bit unsigned char in place signal and with signal.
- **NppStatus nppsAnd_16u_I** (const **Npp16u** *pSrc, **Npp16u** *pSrcDst, int nLength)
16-bit unsigned short in place signal and with signal.
- **NppStatus nppsAnd_32u_I** (const **Npp32u** *pSrc, **Npp32u** *pSrcDst, int nLength)
32-bit unsigned integer in place signal and with signal.

7.170.1 Detailed Description

Sample by sample bitwise AND of samples from two signals.

7.170.2 Function Documentation

7.170.2.1 **NppStatus nppsAnd_16u** (const **Npp16u** *pSrc1, const **Npp16u** *pSrc2, **Npp16u** *pDst, int nLength)

16-bit unsigned short signal and with signal.

Parameters:

- pSrc1** Source Signal Pointer.
pSrc2 Source Signal Pointer. signal2 elements to be anded with signal1 elements
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.170.2.2 NppStatus nppsAnd_16u_I (const Npp16u * *pSrc*, Npp16u * *pSrcDst*, int *nLength*)

16-bit unsigned short in place signal and with signal.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be anded with signal1 elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.170.2.3 NppStatus nppsAnd_32u (const Npp32u * *pSrc1*, const Npp32u * *pSrc2*, Npp32u * *pDst*, int *nLength*)

32-bit unsigned integer signal and with signal.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#). signal2 elements to be anded with signal1 elements

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.170.2.4 NppStatus nppsAnd_32u_I (const Npp32u * *pSrc*, Npp32u * *pSrcDst*, int *nLength*)

32-bit unsigned integer in place signal and with signal.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be anded with signal1 elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.170.2.5 NppStatus nppsAnd_8u (const Npp8u * *pSrc1*, const Npp8u * *pSrc2*, Npp8u * *pDst*, int *nLength*)

8-bit unsigned char signal and with signal.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#). signal2 elements to be anded with signal1 elements

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.170.2.6 NppStatus nppsAnd_8u_I (const Npp8u * *pSrc*, Npp8u * *pSrcDst*, int *nLength*)

8-bit unsigned char in place signal and with signal.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be anded with signal1 elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.171 OrC

Bitwise OR of a constant and each sample of a signal.

Functions

- **NppStatus nppsOrC_8u** (const **Npp8u** *pSrc, **Npp8u** nValue, **Npp8u** *pDst, int nLength)
8-bit unsigned char signal or with constant.
- **NppStatus nppsOrC_16u** (const **Npp16u** *pSrc, **Npp16u** nValue, **Npp16u** *pDst, int nLength)
16-bit unsigned short signal or with constant.
- **NppStatus nppsOrC_32u** (const **Npp32u** *pSrc, **Npp32u** nValue, **Npp32u** *pDst, int nLength)
32-bit unsigned integer signal or with constant.
- **NppStatus nppsOrC_8u_I** (**Npp8u** nValue, **Npp8u** *pSrcDst, int nLength)
8-bit unsigned char in place signal or with constant.
- **NppStatus nppsOrC_16u_I** (**Npp16u** nValue, **Npp16u** *pSrcDst, int nLength)
16-bit unsigned short in place signal or with constant.
- **NppStatus nppsOrC_32u_I** (**Npp32u** nValue, **Npp32u** *pSrcDst, int nLength)
32-bit unsigned signed integer in place signal or with constant.

7.171.1 Detailed Description

Bitwise OR of a constant and each sample of a signal.

7.171.2 Function Documentation

7.171.2.1 NppStatus nppsOrC_16u (const Npp16u *pSrc, Npp16u nValue, Npp16u *pDst, int nLength)

16-bit unsigned short signal or with constant.

Parameters:

- pSrc** Source Signal Pointer.
nValue Constant value to be ored with each vector element
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.171.2.2 NppStatus nppsOrC_16u_I (Npp16u nValue, Npp16u * pSrcDst, int nLength)

16-bit unsigned short in place signal or with constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be ored with each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.171.2.3 NppStatus nppsOrC_32u (const Npp32u * pSrc, Npp32u nValue, Npp32u * pDst, int nLength)

32-bit unsigned integer signal or with constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be ored with each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.171.2.4 NppStatus nppsOrC_32u_I (Npp32u nValue, Npp32u * pSrcDst, int nLength)

32-bit unsigned signed integer in place signal or with constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be ored with each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.171.2.5 NppStatus nppsOrC_8u (const Npp8u * pSrc, Npp8u nValue, Npp8u * pDst, int nLength)

8-bit unsigned char signal or with constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be ored with each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.171.2.6 NppStatus nppsOrC_8u_I (Npp8u *nValue*, Npp8u * *pSrcDst*, int *nLength*)

8-bit unsigned char in place signal or with constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be ored with each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.172 Or

Sample by sample bitwise OR of the samples from two signals.

Functions

- **NppStatus nppsOr_8u** (const **Npp8u** *pSrc1, const **Npp8u** *pSrc2, **Npp8u** *pDst, int nLength)
8-bit unsigned char signal or with signal.
- **NppStatus nppsOr_16u** (const **Npp16u** *pSrc1, const **Npp16u** *pSrc2, **Npp16u** *pDst, int nLength)
16-bit unsigned short signal or with signal.
- **NppStatus nppsOr_32u** (const **Npp32u** *pSrc1, const **Npp32u** *pSrc2, **Npp32u** *pDst, int nLength)
32-bit unsigned integer signal or with signal.
- **NppStatus nppsOr_8u_I** (const **Npp8u** *pSrc, **Npp8u** *pSrcDst, int nLength)
8-bit unsigned char in place signal or with signal.
- **NppStatus nppsOr_16u_I** (const **Npp16u** *pSrc, **Npp16u** *pSrcDst, int nLength)
16-bit unsigned short in place signal or with signal.
- **NppStatus nppsOr_32u_I** (const **Npp32u** *pSrc, **Npp32u** *pSrcDst, int nLength)
32-bit unsigned integer in place signal or with signal.

7.172.1 Detailed Description

Sample by sample bitwise OR of the samples from two signals.

7.172.2 Function Documentation

7.172.2.1 NppStatus nppsOr_16u (const Npp16u *pSrc1, const Npp16u *pSrc2, Npp16u *pDst, int nLength)

16-bit unsigned short signal or with signal.

Parameters:

- pSrc1** [Source Signal Pointer](#).
pSrc2 [Source Signal Pointer](#). signal2 elements to be ored with signal1 elements
pDst [Destination Signal Pointer](#).
nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.172.2.2 NppStatus nppsOr_16u_I (const Npp16u * *pSrc*, Npp16u * *pSrcDst*, int *nLength*)

16-bit unsigned short in place signal or with signal.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be ored with signal1 elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.172.2.3 NppStatus nppsOr_32u (const Npp32u * *pSrc1*, const Npp32u * *pSrc2*, Npp32u * *pDst*, int *nLength*)

32-bit unsigned integer signal or with signal.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#). signal2 elements to be ored with signal1 elements

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.172.2.4 NppStatus nppsOr_32u_I (const Npp32u * *pSrc*, Npp32u * *pSrcDst*, int *nLength*)

32-bit unsigned integer in place signal or with signal.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be ored with signal1 elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.172.2.5 NppStatus nppsOr_8u (const Npp8u * *pSrc1*, const Npp8u * *pSrc2*, Npp8u * *pDst*, int *nLength*)

8-bit unsigned char signal or with signal.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer. signal2 elements to be ored with signal1 elements

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.172.2.6 NppStatus nppsOr_8u_I (const Npp8u * *pSrc*, Npp8u * *pSrcDst*, int *nLength*)

8-bit unsigned char in place signal or with signal.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer. signal2 elements to be ored with signal1 elements

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.173 XorC

Bitwise XOR of a constant and each sample of a signal.

Functions

- **NppStatus nppsXorC_8u** (const **Npp8u** *pSrc, **Npp8u** nValue, **Npp8u** *pDst, int nLength)
8-bit unsigned char signal exclusive or with constant.
- **NppStatus nppsXorC_16u** (const **Npp16u** *pSrc, **Npp16u** nValue, **Npp16u** *pDst, int nLength)
16-bit unsigned short signal exclusive or with constant.
- **NppStatus nppsXorC_32u** (const **Npp32u** *pSrc, **Npp32u** nValue, **Npp32u** *pDst, int nLength)
32-bit unsigned integer signal exclusive or with constant.
- **NppStatus nppsXorC_8u_I** (**Npp8u** nValue, **Npp8u** *pSrcDst, int nLength)
8-bit unsigned char in place signal exclusive or with constant.
- **NppStatus nppsXorC_16u_I** (**Npp16u** nValue, **Npp16u** *pSrcDst, int nLength)
16-bit unsigned short in place signal exclusive or with constant.
- **NppStatus nppsXorC_32u_I** (**Npp32u** nValue, **Npp32u** *pSrcDst, int nLength)
32-bit unsigned signed integer in place signal exclusive or with constant.

7.173.1 Detailed Description

Bitwise XOR of a constant and each sample of a signal.

7.173.2 Function Documentation

7.173.2.1 NppStatus nppsXorC_16u (const Npp16u *pSrc, Npp16u nValue, Npp16u *pDst, int nLength)

16-bit unsigned short signal exclusive or with constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be exclusive ored with each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.173.2.2 NppStatus nppsXorC_16u_I (Npp16u nValue, Npp16u * pSrcDst, int nLength)

16-bit unsigned short in place signal exclusive or with constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be exclusive ored with each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.173.2.3 NppStatus nppsXorC_32u (const Npp32u * pSrc, Npp32u nValue, Npp32u * pDst, int nLength)

32-bit unsigned integer signal exclusive or with constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be exclusive ored with each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.173.2.4 NppStatus nppsXorC_32u_I (Npp32u nValue, Npp32u * pSrcDst, int nLength)

32-bit unsigned signed integer in place signal exclusive or with constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be exclusive ored with each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.173.2.5 NppStatus nppsXorC_8u (const Npp8u * pSrc, Npp8u nValue, Npp8u * pDst, int nLength)

8-bit unsigned char signal exclusive or with constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be exclusive ored with each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.173.2.6 NppStatus nppsXorC_8u_I (Npp8u nValue, Npp8u * pSrcDst, int nLength)

8-bit unsigned char in place signal exclusive or with constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be exclusive ored with each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.174 Xor

Sample by sample bitwise XOR of the samples from two signals.

Functions

- **NppStatus nppsXor_8u** (const **Npp8u** *pSrc1, const **Npp8u** *pSrc2, **Npp8u** *pDst, int nLength)
8-bit unsigned char signal exclusive or with signal.
- **NppStatus nppsXor_16u** (const **Npp16u** *pSrc1, const **Npp16u** *pSrc2, **Npp16u** *pDst, int nLength)
16-bit unsigned short signal exclusive or with signal.
- **NppStatus nppsXor_32u** (const **Npp32u** *pSrc1, const **Npp32u** *pSrc2, **Npp32u** *pDst, int nLength)
32-bit unsigned integer signal exclusive or with signal.
- **NppStatus nppsXor_8u_I** (const **Npp8u** *pSrc, **Npp8u** *pSrcDst, int nLength)
8-bit unsigned char in place signal exclusive or with signal.
- **NppStatus nppsXor_16u_I** (const **Npp16u** *pSrc, **Npp16u** *pSrcDst, int nLength)
16-bit unsigned short in place signal exclusive or with signal.
- **NppStatus nppsXor_32u_I** (const **Npp32u** *pSrc, **Npp32u** *pSrcDst, int nLength)
32-bit unsigned integer in place signal exclusive or with signal.

7.174.1 Detailed Description

Sample by sample bitwise XOR of the samples from two signals.

7.174.2 Function Documentation

7.174.2.1 NppStatus nppsXor_16u (const Npp16u *pSrc1, const Npp16u *pSrc2, Npp16u *pDst, int nLength)

16-bit unsigned short signal exclusive or with signal.

Parameters:

- pSrc1** [Source Signal Pointer](#).
pSrc2 [Source Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements
pDst [Destination Signal Pointer](#).
nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.174.2.2 NppStatus nppsXor_16u_I (const Npp16u * pSrc, Npp16u * pSrcDst, int nLength)

16-bit unsigned short in place signal exclusive or with signal.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.174.2.3 NppStatus nppsXor_32u (const Npp32u * pSrc1, const Npp32u * pSrc2, Npp32u * pDst, int nLength)

32-bit unsigned integer signal exclusive or with signal.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.174.2.4 NppStatus nppsXor_32u_I (const Npp32u * pSrc, Npp32u * pSrcDst, int nLength)

32-bit unsigned integer in place signal exclusive or with signal.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.174.2.5 NppStatus nppsXor_8u (const Npp8u * pSrc1, const Npp8u * pSrc2, Npp8u * pDst, int nLength)

8-bit unsigned char signal exclusive or with signal.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.174.2.6 NppStatus nppsXor_8u_I (const Npp8u * pSrc, Npp8u * pSrcDst, int nLength)

8-bit unsigned char in place signal exclusive or with signal.

Parameters:

pSrc [Source Signal Pointer](#).

pSrcDst [In-Place Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.175 Not

Bitwise NOT of each sample of a signal.

Functions

- **NppStatus nppsNot_8u** (const **Npp8u** *pSrc, **Npp8u** *pDst, int nLength)
8-bit unsigned char not signal.
- **NppStatus nppsNot_16u** (const **Npp16u** *pSrc, **Npp16u** *pDst, int nLength)
16-bit unsigned short not signal.
- **NppStatus nppsNot_32u** (const **Npp32u** *pSrc, **Npp32u** *pDst, int nLength)
32-bit unsigned integer not signal.
- **NppStatus nppsNot_8u_I** (**Npp8u** *pSrcDst, int nLength)
8-bit unsigned char in place not signal.
- **NppStatus nppsNot_16u_I** (**Npp16u** *pSrcDst, int nLength)
16-bit unsigned short in place not signal.
- **NppStatus nppsNot_32u_I** (**Npp32u** *pSrcDst, int nLength)
32-bit unsigned signed integer in place not signal.

7.175.1 Detailed Description

Bitwise NOT of each sample of a signal.

7.175.2 Function Documentation

7.175.2.1 NppStatus nppsNot_16u (const Npp16u *pSrc, Npp16u *pDst, int nLength)

16-bit unsigned short not signal.

Parameters:

- pSrc** Source Signal Pointer.
- pDst** Destination Signal Pointer.
- nLength** Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.175.2.2 NppStatus nppsNot_16u_I (Npp16u * *pSrcDst*, int *nLength*)

16-bit unsigned short in place not signal.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.175.2.3 NppStatus nppsNot_32u (const Npp32u * *pSrc*, Npp32u * *pDst*, int *nLength*)

32-bit unsigned integer not signal.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.175.2.4 NppStatus nppsNot_32u_I (Npp32u * *pSrcDst*, int *nLength*)

32-bit unsigned signed integer in place not signal.

Parameters:

pSrcDst In-Place Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.175.2.5 NppStatus nppsNot_8u (const Npp8u * *pSrc*, Npp8u * *pDst*, int *nLength*)

8-bit unsigned char not signal.

Parameters:

pSrc Source Signal Pointer.
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.175.2.6 NppStatus nppsNot_8u_I (Npp8u * *pSrcDst*, int *nLength*)

8-bit unsigned char in place not signal.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.176 LShiftC

Left shifts the bits of each sample of a signal by a constant amount.

Functions

- **NppStatus nppsLShiftC_8u** (const **Npp8u** *pSrc, int nValue, **Npp8u** *pDst, int nLength)
8-bit unsigned char signal left shift with constant.
- **NppStatus nppsLShiftC_16u** (const **Npp16u** *pSrc, int nValue, **Npp16u** *pDst, int nLength)
16-bit unsigned short signal left shift with constant.
- **NppStatus nppsLShiftC_16s** (const **Npp16s** *pSrc, int nValue, **Npp16s** *pDst, int nLength)
16-bit signed short signal left shift with constant.
- **NppStatus nppsLShiftC_32u** (const **Npp32u** *pSrc, int nValue, **Npp32u** *pDst, int nLength)
32-bit unsigned integer signal left shift with constant.
- **NppStatus nppsLShiftC_32s** (const **Npp32s** *pSrc, int nValue, **Npp32s** *pDst, int nLength)
32-bit signed integer signal left shift with constant.
- **NppStatus nppsLShiftC_8u_I** (int nValue, **Npp8u** *pSrcDst, int nLength)
8-bit unsigned char in place signal left shift with constant.
- **NppStatus nppsLShiftC_16u_I** (int nValue, **Npp16u** *pSrcDst, int nLength)
16-bit unsigned short in place signal left shift with constant.
- **NppStatus nppsLShiftC_16s_I** (int nValue, **Npp16s** *pSrcDst, int nLength)
16-bit signed short in place signal left shift with constant.
- **NppStatus nppsLShiftC_32u_I** (int nValue, **Npp32u** *pSrcDst, int nLength)
32-bit unsigned signed integer in place signal left shift with constant.
- **NppStatus nppsLShiftC_32s_I** (int nValue, **Npp32s** *pSrcDst, int nLength)
32-bit signed signed integer in place signal left shift with constant.

7.176.1 Detailed Description

Left shifts the bits of each sample of a signal by a constant amount.

7.176.2 Function Documentation

7.176.2.1 **NppStatus nppsLShiftC_16s** (const **Npp16s** *pSrc, int nValue, **Npp16s** *pDst, int nLength)

16-bit signed short signal left shift with constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be used to left shift each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.176.2.2 NppStatus nppsLShiftC_16s_I (int *nValue*, Npp16s * *pSrcDst*, int *nLength*)

16-bit signed short in place signal left shift with constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be used to left shift each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.176.2.3 NppStatus nppsLShiftC_16u (const Npp16u * *pSrc*, int *nValue*, Npp16u * *pDst*, int *nLength*)

16-bit unsigned short signal left shift with constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be used to left shift each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.176.2.4 NppStatus nppsLShiftC_16u_I (int *nValue*, Npp16u * *pSrcDst*, int *nLength*)

16-bit unsigned short in place signal left shift with constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be used to left shift each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.176.2.5 NppStatus nppsLShiftC_32s (const Npp32s * *pSrc*, int *nValue*, Npp32s * *pDst*, int *nLength*)

32-bit signed integer signal left shift with constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be used to left shift each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.176.2.6 NppStatus nppsLShiftC_32s_I (int *nValue*, Npp32s * *pSrcDst*, int *nLength*)

32-bit signed integer in place signal left shift with constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be used to left shift each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.176.2.7 NppStatus nppsLShiftC_32u (const Npp32u * *pSrc*, int *nValue*, Npp32u * *pDst*, int *nLength*)

32-bit unsigned integer signal left shift with constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be used to left shift each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.176.2.8 NppStatus nppsLShiftC_32u_I (int *nValue*, Npp32u * *pSrcDst*, int *nLength*)

32-bit unsigned signed integer in place signal left shift with constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be used to left shift each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.176.2.9 NppStatus nppsLShiftC_8u (const Npp8u * *pSrc*, int *nValue*, Npp8u * *pDst*, int *nLength*)

8-bit unsigned char signal left shift with constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be used to left shift each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.176.2.10 NppStatus nppsLShiftC_8u_I (int *nValue*, Npp8u * *pSrcDst*, int *nLength*)

8-bit unsigned char in place signal left shift with constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be used to left shift each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.177 RShiftC

Right shifts the bits of each sample of a signal by a constant amount.

Functions

- **NppStatus nppsRShiftC_8u** (const **Npp8u** *pSrc, int nValue, **Npp8u** *pDst, int nLength)
8-bit unsigned char signal right shift with constant.
- **NppStatus nppsRShiftC_16u** (const **Npp16u** *pSrc, int nValue, **Npp16u** *pDst, int nLength)
16-bit unsigned short signal right shift with constant.
- **NppStatus nppsRShiftC_16s** (const **Npp16s** *pSrc, int nValue, **Npp16s** *pDst, int nLength)
16-bit signed short signal right shift with constant.
- **NppStatus nppsRShiftC_32u** (const **Npp32u** *pSrc, int nValue, **Npp32u** *pDst, int nLength)
32-bit unsigned integer signal right shift with constant.
- **NppStatus nppsRShiftC_32s** (const **Npp32s** *pSrc, int nValue, **Npp32s** *pDst, int nLength)
32-bit signed integer signal right shift with constant.
- **NppStatus nppsRShiftC_8u_I** (int nValue, **Npp8u** *pSrcDst, int nLength)
8-bit unsigned char in place signal right shift with constant.
- **NppStatus nppsRShiftC_16u_I** (int nValue, **Npp16u** *pSrcDst, int nLength)
16-bit unsigned short in place signal right shift with constant.
- **NppStatus nppsRShiftC_16s_I** (int nValue, **Npp16s** *pSrcDst, int nLength)
16-bit signed short in place signal right shift with constant.
- **NppStatus nppsRShiftC_32u_I** (int nValue, **Npp32u** *pSrcDst, int nLength)
32-bit unsigned signed integer in place signal right shift with constant.
- **NppStatus nppsRShiftC_32s_I** (int nValue, **Npp32s** *pSrcDst, int nLength)
32-bit signed signed integer in place signal right shift with constant.

7.177.1 Detailed Description

Right shifts the bits of each sample of a signal by a constant amount.

7.177.2 Function Documentation

7.177.2.1 **NppStatus nppsRShiftC_16s** (const **Npp16s** *pSrc, int nValue, **Npp16s** *pDst, int nLength)

16-bit signed short signal right shift with constant.

Parameters:

pSrc Source Signal Pointer.
nValue Constant value to be used to right shift each vector element
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.177.2.2 NppStatus nppsRShiftC_16s_I (int *nValue*, Npp16s * *pSrcDst*, int *nLength*)

16-bit signed short in place signal right shift with constant.

Parameters:

pSrcDst In-Place Signal Pointer.
nValue Constant value to be used to right shift each vector element
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.177.2.3 NppStatus nppsRShiftC_16u (const Npp16u * *pSrc*, int *nValue*, Npp16u * *pDst*, int *nLength*)

16-bit unsigned short signal right shift with constant.

Parameters:

pSrc Source Signal Pointer.
nValue Constant value to be used to right shift each vector element
pDst Destination Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.177.2.4 NppStatus nppsRShiftC_16u_I (int *nValue*, Npp16u * *pSrcDst*, int *nLength*)

16-bit unsigned short in place signal right shift with constant.

Parameters:

pSrcDst In-Place Signal Pointer.
nValue Constant value to be used to right shift each vector element
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.177.2.5 NppStatus nppsRShiftC_32s (const Npp32s * *pSrc*, int *nValue*, Npp32s * *pDst*, int *nLength*)

32-bit signed integer signal right shift with constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be used to right shift each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.177.2.6 NppStatus nppsRShiftC_32s_I (int *nValue*, Npp32s * *pSrcDst*, int *nLength*)

32-bit signed integer in place signal right shift with constant.

Parameters:

pSrcDst In-Place Signal Pointer.

nValue Constant value to be used to right shift each vector element

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.177.2.7 NppStatus nppsRShiftC_32u (const Npp32u * *pSrc*, int *nValue*, Npp32u * *pDst*, int *nLength*)

32-bit unsigned integer signal right shift with constant.

Parameters:

pSrc Source Signal Pointer.

nValue Constant value to be used to right shift each vector element

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.177.2.8 NppStatus nppsRShiftC_32u_I (int *nValue*, Npp32u * *pSrcDst*, int *nLength*)

32-bit unsigned signed integer in place signal right shift with constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be used to right shift each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.177.2.9 NppStatus nppsRShiftC_8u (const Npp8u * *pSrc*, int *nValue*, Npp8u * *pDst*, int *nLength*)

8-bit unsigned char signal right shift with constant.

Parameters:

pSrc [Source Signal Pointer](#).

nValue Constant value to be used to right shift each vector element

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.177.2.10 NppStatus nppsRShiftC_8u_I (int *nValue*, Npp8u * *pSrcDst*, int *nLength*)

8-bit unsigned char in place signal right shift with constant.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nValue Constant value to be used to right shift each vector element

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.178 Conversion Functions

Modules

- [Convert](#)
- [Threshold](#)

7.179 Convert

Convert

Routines for converting the sample-data type of signals.

- `NppStatus nppsConvert_8s16s` (const `Npp8s` *pSrc, `Npp16s` *pDst, int nLength)
- `NppStatus nppsConvert_8s32f` (const `Npp8s` *pSrc, `Npp32f` *pDst, int nLength)
- `NppStatus nppsConvert_8u32f` (const `Npp8u` *pSrc, `Npp32f` *pDst, int nLength)
- `NppStatus nppsConvert_16s8s_Sfs` (const `Npp16s` *pSrc, `Npp8s` *pDst, `Npp32u` nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_16s32s` (const `Npp16s` *pSrc, `Npp32s` *pDst, int nLength)
- `NppStatus nppsConvert_16s32f` (const `Npp16s` *pSrc, `Npp32f` *pDst, int nLength)
- `NppStatus nppsConvert_16u32f` (const `Npp16u` *pSrc, `Npp32f` *pDst, int nLength)
- `NppStatus nppsConvert_32s16s` (const `Npp32s` *pSrc, `Npp16s` *pDst, int nLength)
- `NppStatus nppsConvert_32s32f` (const `Npp32s` *pSrc, `Npp32f` *pDst, int nLength)
- `NppStatus nppsConvert_32s64f` (const `Npp32s` *pSrc, `Npp64f` *pDst, int nLength)
- `NppStatus nppsConvert_32f64f` (const `Npp32f` *pSrc, `Npp64f` *pDst, int nLength)
- `NppStatus nppsConvert_64s64f` (const `Npp64s` *pSrc, `Npp64f` *pDst, int nLength)
- `NppStatus nppsConvert_64f32f` (const `Npp64f` *pSrc, `Npp32f` *pDst, int nLength)
- `NppStatus nppsConvert_16s32f_Sfs` (const `Npp16s` *pSrc, `Npp32f` *pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_16s64f_Sfs` (const `Npp16s` *pSrc, `Npp64f` *pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32s16s_Sfs` (const `Npp32s` *pSrc, `Npp16s` *pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32s32f_Sfs` (const `Npp32s` *pSrc, `Npp32f` *pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32s64f_Sfs` (const `Npp32s` *pSrc, `Npp64f` *pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32f8s_Sfs` (const `Npp32f` *pSrc, `Npp8s` *pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f8u_Sfs` (const `Npp32f` *pSrc, `Npp8u` *pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f16s_Sfs` (const `Npp32f` *pSrc, `Npp16s` *pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f16u_Sfs` (const `Npp32f` *pSrc, `Npp16u` *pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f32s_Sfs` (const `Npp32f` *pSrc, `Npp32s` *pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64s32s_Sfs` (const `Npp64s` *pSrc, `Npp32s` *pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64f16s_Sfs` (const `Npp64f` *pSrc, `Npp16s` *pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64f32s_Sfs` (const `Npp64f` *pSrc, `Npp32s` *pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64f64s_Sfs` (const `Npp64f` *pSrc, `Npp64s` *pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)

7.179.1 Function Documentation

- 7.179.1.1** `NppStatus nppsConvert_16s32f (const Npp16s * pSrc, Npp32f * pDst, int nLength)`
- 7.179.1.2** `NppStatus nppsConvert_16s32f_Sfs (const Npp16s * pSrc, Npp32f * pDst, int nLength, int nScaleFactor)`
- 7.179.1.3** `NppStatus nppsConvert_16s32s (const Npp16s * pSrc, Npp32s * pDst, int nLength)`
- 7.179.1.4** `NppStatus nppsConvert_16s64f_Sfs (const Npp16s * pSrc, Npp64f * pDst, int nLength, int nScaleFactor)`
- 7.179.1.5** `NppStatus nppsConvert_16s8s_Sfs (const Npp16s * pSrc, Npp8s * pDst, Npp32u nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.179.1.6** `NppStatus nppsConvert_16u32f (const Npp16u * pSrc, Npp32f * pDst, int nLength)`
- 7.179.1.7** `NppStatus nppsConvert_32f16s_Sfs (const Npp32f * pSrc, Npp16s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.179.1.8** `NppStatus nppsConvert_32f16u_Sfs (const Npp32f * pSrc, Npp16u * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.179.1.9** `NppStatus nppsConvert_32f32s_Sfs (const Npp32f * pSrc, Npp32s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.179.1.10** `NppStatus nppsConvert_32f64f (const Npp32f * pSrc, Npp64f * pDst, int nLength)`
- 7.179.1.11** `NppStatus nppsConvert_32f8s_Sfs (const Npp32f * pSrc, Npp8s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.179.1.12** `NppStatus nppsConvert_32f8u_Sfs (const Npp32f * pSrc, Npp8u * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.179.1.13** `NppStatus nppsConvert_32s16s (const Npp32s * pSrc, Npp16s * pDst, int nLength)`
- 7.179.1.14** `NppStatus nppsConvert_32s16s_Sfs (const Npp32s * pSrc, Npp16s * pDst, int nLength, int nScaleFactor)`
- 7.179.1.15** `NppStatus nppsConvert_32s32f (const Npp32s * pSrc, Npp32f * pDst, int nLength)`
- 7.179.1.16** `NppStatus nppsConvert_32s32f_Sfs (const Npp32s * pSrc, Npp32f * pDst, int nLength, int nScaleFactor)`
- 7.179.1.17** `NppStatus nppsConvert_32s64f (const Npp32s * pSrc, Npp64f * pDst, int nLength)`
- 7.179.1.18** `NppStatus nppsConvert_32s64f_Sfs (const Npp32s * pSrc, Npp64f * pDst, int nLength, int nScaleFactor)`
- 7.179.1.19** `NppStatus nppsConvert_64f16s_Sfs (const Npp64f * pSrc, Npp16s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.179.1.20** `NppStatus nppsConvert_64f32f (const Npp64f * pSrc, Npp32f * pDst, int nLength)`
- 7.179.1.21** `NppStatus nppsConvert_64f32s_Sfs (const Npp64f * pSrc, Npp32s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.179.1.22** `NppStatus nppsConvert_64f64s_Sfs (const Npp64f * pSrc, Npp64s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.179.1.23** `NppStatus nppsConvert_64s32s_Sfs (const Npp64s * pSrc, Npp32s * pDst, int nLength,`

7.180 Threshold

Threshold Functions

Performs the threshold operation on the samples of a signal by limiting the sample values by a specified constant value.

- [NppStatus nppsThreshold_16s](#) (const [Npp16s](#) *pSrc, [Npp16s](#) *pDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)
16-bit signed short signal threshold with constant level.
- [NppStatus nppsThreshold_16s_I](#) ([Npp16s](#) *pSrcDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)
16-bit in place signed short signal threshold with constant level.
- [NppStatus nppsThreshold_16sc](#) (const [Npp16sc](#) *pSrc, [Npp16sc](#) *pDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)
16-bit signed short complex number signal threshold with constant level.
- [NppStatus nppsThreshold_16sc_I](#) ([Npp16sc](#) *pSrcDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)
16-bit in place signed short complex number signal threshold with constant level.
- [NppStatus nppsThreshold_32f](#) (const [Npp32f](#) *pSrc, [Npp32f](#) *pDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)
32-bit floating point signal threshold with constant level.
- [NppStatus nppsThreshold_32f_I](#) ([Npp32f](#) *pSrcDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)
32-bit in place floating point signal threshold with constant level.
- [NppStatus nppsThreshold_32fc](#) (const [Npp32fc](#) *pSrc, [Npp32fc](#) *pDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)
32-bit floating point complex number signal threshold with constant level.
- [NppStatus nppsThreshold_32fc_I](#) ([Npp32fc](#) *pSrcDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)
32-bit in place floating point complex number signal threshold with constant level.
- [NppStatus nppsThreshold_64f](#) (const [Npp64f](#) *pSrc, [Npp64f](#) *pDst, int nLength, [Npp64f](#) nLevel, [NppCmpOp](#) nRelOp)
64-bit floating point signal threshold with constant level.
- [NppStatus nppsThreshold_64f_I](#) ([Npp64f](#) *pSrcDst, int nLength, [Npp64f](#) nLevel, [NppCmpOp](#) nRelOp)
64-bit in place floating point signal threshold with constant level.
- [NppStatus nppsThreshold_64fc](#) (const [Npp64fc](#) *pSrc, [Npp64fc](#) *pDst, int nLength, [Npp64f](#) nLevel, [NppCmpOp](#) nRelOp)
64-bit floating point complex number signal threshold with constant level.

- **NppStatus nppsThreshold_64fc_I** (**Npp64fc** *pSrcDst, int nLength, **Npp64f** nLevel, **NppCmpOp** nRelOp)
64-bit in place floating point complex number signal threshold with constant level.
- **NppStatus nppsThreshold_LT_16s** (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength, **Npp16s** nLevel)
16-bit signed short signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LT_16s_I** (**Npp16s** *pSrcDst, int nLength, **Npp16s** nLevel)
16-bit in place signed short signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LT_16sc** (const **Npp16sc** *pSrc, **Npp16sc** *pDst, int nLength, **Npp16s** nLevel)
16-bit signed short complex number signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LT_16sc_I** (**Npp16sc** *pSrcDst, int nLength, **Npp16s** nLevel)
16-bit in place signed short complex number signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LT_32f** (const **Npp32f** *pSrc, **Npp32f** *pDst, int nLength, **Npp32f** nLevel)
32-bit floating point signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LT_32f_I** (**Npp32f** *pSrcDst, int nLength, **Npp32f** nLevel)
32-bit in place floating point signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LT_32fc** (const **Npp32fc** *pSrc, **Npp32fc** *pDst, int nLength, **Npp32f** nLevel)
32-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LT_32fc_I** (**Npp32fc** *pSrcDst, int nLength, **Npp32f** nLevel)
32-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LT_64f** (const **Npp64f** *pSrc, **Npp64f** *pDst, int nLength, **Npp64f** nLevel)
64-bit floating point signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LT_64f_I** (**Npp64f** *pSrcDst, int nLength, **Npp64f** nLevel)
64-bit in place floating point signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LT_64fc** (const **Npp64fc** *pSrc, **Npp64fc** *pDst, int nLength, **Npp64f** nLevel)
64-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LT_64fc_I** (**Npp64fc** *pSrcDst, int nLength, **Npp64f** nLevel)
64-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_GT_16s** (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength, **Npp16s** nLevel)
16-bit signed short signal NPP_CMP_GREATER threshold with constant level.

- [NppStatus nppsThreshold_GT_16s_I](#) ([Npp16s](#) *pSrcDst, int nLength, [Npp16s](#) nLevel)
16-bit in place signed short signal NPP_CMP_GREATER threshold with constant level.
- [NppStatus nppsThreshold_GT_16sc](#) (const [Npp16sc](#) *pSrc, [Npp16sc](#) *pDst, int nLength, [Npp16s](#) nLevel)
16-bit signed short complex number signal NPP_CMP_GREATER threshold with constant level.
- [NppStatus nppsThreshold_GT_16sc_I](#) ([Npp16sc](#) *pSrcDst, int nLength, [Npp16s](#) nLevel)
16-bit in place signed short complex number signal NPP_CMP_GREATER threshold with constant level.
- [NppStatus nppsThreshold_GT_32f](#) (const [Npp32f](#) *pSrc, [Npp32f](#) *pDst, int nLength, [Npp32f](#) nLevel)
32-bit floating point signal NPP_CMP_GREATER threshold with constant level.
- [NppStatus nppsThreshold_GT_32f_I](#) ([Npp32f](#) *pSrcDst, int nLength, [Npp32f](#) nLevel)
32-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.
- [NppStatus nppsThreshold_GT_32fc](#) (const [Npp32fc](#) *pSrc, [Npp32fc](#) *pDst, int nLength, [Npp32f](#) nLevel)
32-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.
- [NppStatus nppsThreshold_GT_32fc_I](#) ([Npp32fc](#) *pSrcDst, int nLength, [Npp32f](#) nLevel)
32-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.
- [NppStatus nppsThreshold_GT_64f](#) (const [Npp64f](#) *pSrc, [Npp64f](#) *pDst, int nLength, [Npp64f](#) nLevel)
64-bit floating point signal NPP_CMP_GREATER threshold with constant level.
- [NppStatus nppsThreshold_GT_64f_I](#) ([Npp64f](#) *pSrcDst, int nLength, [Npp64f](#) nLevel)
64-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.
- [NppStatus nppsThreshold_GT_64fc](#) (const [Npp64fc](#) *pSrc, [Npp64fc](#) *pDst, int nLength, [Npp64f](#) nLevel)
64-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.
- [NppStatus nppsThreshold_GT_64fc_I](#) ([Npp64fc](#) *pSrcDst, int nLength, [Npp64f](#) nLevel)
64-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.
- [NppStatus nppsThreshold_LTV_16s](#) (const [Npp16s](#) *pSrc, [Npp16s](#) *pDst, int nLength, [Npp16s](#) nLevel, [Npp16s](#) nValue)
16-bit signed short signal NPP_CMP_LESS threshold with constant level.
- [NppStatus nppsThreshold_LTV_16s_I](#) ([Npp16s](#) *pSrcDst, int nLength, [Npp16s](#) nLevel, [Npp16s](#) nValue)
16-bit in place signed short signal NPP_CMP_LESS threshold with constant level.
- [NppStatus nppsThreshold_LTV_16sc](#) (const [Npp16sc](#) *pSrc, [Npp16sc](#) *pDst, int nLength, [Npp16s](#) nLevel, [Npp16sc](#) nValue)
16-bit signed short complex number signal NPP_CMP_LESS threshold with constant level.

- **NppStatus nppsThreshold_LTVal_16sc_I** (**Npp16sc** *pSrcDst, int nLength, **Npp16s** nLevel, **Npp16sc** nValue)
16-bit in place signed short complex number signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LTVal_32f** (const **Npp32f** *pSrc, **Npp32f** *pDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)
32-bit floating point signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LTVal_32f_I** (**Npp32f** *pSrcDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)
32-bit in place floating point signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LTVal_32fc** (const **Npp32fc** *pSrc, **Npp32fc** *pDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)
32-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LTVal_32fc_I** (**Npp32fc** *pSrcDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)
32-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LTVal_64f** (const **Npp64f** *pSrc, **Npp64f** *pDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)
64-bit floating point signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LTVal_64f_I** (**Npp64f** *pSrcDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)
64-bit in place floating point signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LTVal_64fc** (const **Npp64fc** *pSrc, **Npp64fc** *pDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)
64-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_LTVal_64fc_I** (**Npp64fc** *pSrcDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)
64-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.
- **NppStatus nppsThreshold_GTVal_16s** (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength, **Npp16s** nLevel, **Npp16s** nValue)
16-bit signed short signal NPP_CMP_GREATER threshold with constant level.
- **NppStatus nppsThreshold_GTVal_16s_I** (**Npp16s** *pSrcDst, int nLength, **Npp16s** nLevel, **Npp16s** nValue)
16-bit in place signed short signal NPP_CMP_GREATER threshold with constant level.
- **NppStatus nppsThreshold_GTVal_16sc** (const **Npp16sc** *pSrc, **Npp16sc** *pDst, int nLength, **Npp16s** nLevel, **Npp16sc** nValue)
16-bit signed short complex number signal NPP_CMP_GREATER threshold with constant level.
- **NppStatus nppsThreshold_GTVal_16sc_I** (**Npp16sc** *pSrcDst, int nLength, **Npp16s** nLevel, **Npp16sc** nValue)
16-bit in place signed short complex number signal NPP_CMP_GREATER threshold with constant level.

- **NppStatus nppsThreshold_GTVVal_32f** (const **Npp32f** *pSrc, **Npp32f** *pDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)
32-bit floating point signal NPP_CMP_GREATER threshold with constant level.
- **NppStatus nppsThreshold_GTVVal_32f_I** (**Npp32f** *pSrcDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)
32-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.
- **NppStatus nppsThreshold_GTVVal_32fc** (const **Npp32fc** *pSrc, **Npp32fc** *pDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)
32-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.
- **NppStatus nppsThreshold_GTVVal_32fc_I** (**Npp32fc** *pSrcDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)
32-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.
- **NppStatus nppsThreshold_GTVVal_64f** (const **Npp64f** *pSrc, **Npp64f** *pDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)
64-bit floating point signal NPP_CMP_GREATER threshold with constant level.
- **NppStatus nppsThreshold_GTVVal_64f_I** (**Npp64f** *pSrcDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)
64-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.
- **NppStatus nppsThreshold_GTVVal_64fc** (const **Npp64fc** *pSrc, **Npp64fc** *pDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)
64-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.
- **NppStatus nppsThreshold_GTVVal_64fc_I** (**Npp64fc** *pSrcDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)
64-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.

7.180.1 Function Documentation

7.180.1.1 **NppStatus nppsThreshold_16s** (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength, **Npp16s** nLevel, **NppCmpOp** nRelOp)

16-bit signed short signal threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

nRelOp NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.2 NppStatus nppsThreshold_16s_I (Npp16s * *pSrcDst*, int *nLength*, Npp16s *nLevel*, NppCmpOp *nRelOp*)

16-bit in place signed short signal threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nRelOp NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.3 NppStatus nppsThreshold_16sc (const Npp16sc * *pSrc*, Npp16sc * *pDst*, int *nLength*, Npp16s *nLevel*, NppCmpOp *nRelOp*)

16-bit signed short complex number signal threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nRelOp NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.4 NppStatus nppsThreshold_16sc_I (Npp16sc * *pSrcDst*, int *nLength*, Npp16s *nLevel*, NppCmpOp *nRelOp*)

16-bit in place signed short complex number signal threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nRelOp NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.5 NppStatus nppsThreshold_32f (const Npp32f * *pSrc*, Npp32f * *pDst*, int *nLength*, Npp32f *nLevel*, NppCmpOp *nRelOp*)

32-bit floating point signal threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nRelOp NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.6 NppStatus nppsThreshold_32f_I (Npp32f * *pSrcDst*, int *nLength*, Npp32f *nLevel*, NppCmpOp *nRelOp*)

32-bit in place floating point signal threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nRelOp NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.7 NppStatus nppsThreshold_32fc (const Npp32fc * *pSrc*, Npp32fc * *pDst*, int *nLength*, Npp32f *nLevel*, NppCmpOp *nRelOp*)

32-bit floating point complex number signal threshold with constant level.

Parameters:*pSrc* [Source Signal Pointer](#).*pDst* [Destination Signal Pointer](#).*nLength* [Signal Length](#).*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample*nRelOp* NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).**Returns:**[Signal Data Related Error Codes](#), [Length Related Error Codes](#).**7.180.1.8 NppStatus nppsThreshold_32fc_I (Npp32fc * *pSrcDst*, int *nLength*, Npp32f *nLevel*, NppCmpOp *nRelOp*)**

32-bit in place floating point complex number signal threshold with constant level.

Parameters:*pSrcDst* [In-Place Signal Pointer](#).*nLength* [Signal Length](#).*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample*nRelOp* NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).**Returns:**[Signal Data Related Error Codes](#), [Length Related Error Codes](#).**7.180.1.9 NppStatus nppsThreshold_64f (const Npp64f * *pSrc*, Npp64f * *pDst*, int *nLength*, Npp64f *nLevel*, NppCmpOp *nRelOp*)**

64-bit floating point signal threshold with constant level.

Parameters:*pSrc* [Source Signal Pointer](#).*pDst* [Destination Signal Pointer](#).*nLength* [Signal Length](#).*nLevel* Constant threshold value to be used to limit each signal sample*nRelOp* NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).**Returns:**[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.10 **NppStatus nppsThreshold_64f_I** (Npp64f * *pSrcDst*, int *nLength*, Npp64f *nLevel*, NppCmpOp *nRelOp*)

64-bit in place floating point signal threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nRelOp NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.11 **NppStatus nppsThreshold_64fc** (const Npp64fc * *pSrc*, Npp64fc * *pDst*, int *nLength*, Npp64f *nLevel*, NppCmpOp *nRelOp*)

64-bit floating point complex number signal threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nRelOp NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.12 **NppStatus nppsThreshold_64fc_I** (Npp64fc * *pSrcDst*, int *nLength*, Npp64f *nLevel*, NppCmpOp *nRelOp*)

64-bit in place floating point complex number signal threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nRelOp NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.13 NppStatus nppsThreshold_GT_16s (const Npp16s * *pSrc*, Npp16s * *pDst*, int *nLength*, Npp16s *nLevel*)

16-bit signed short signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.14 NppStatus nppsThreshold_GT_16s_I (Npp16s * *pSrcDst*, int *nLength*, Npp16s *nLevel*)

16-bit in place signed short signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.15 NppStatus nppsThreshold_GT_16sc (const Npp16sc * *pSrc*, Npp16sc * *pDst*, int *nLength*, Npp16s *nLevel*)

16-bit signed short complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.16 NppStatus nppsThreshold_GT_16sc_I (Npp16sc * *pSrcDst*, int *nLength*, Npp16s *nLevel*)

16-bit in place signed short complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.17 NppStatus nppsThreshold_GT_32f (const Npp32f * *pSrc*, Npp32f * *pDst*, int *nLength*, Npp32f *nLevel*)

32-bit floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.18 NppStatus nppsThreshold_GT_32f_I (Npp32f * *pSrcDst*, int *nLength*, Npp32f *nLevel*)

32-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.19 **NppStatus nppsThreshold_GT_32fc** (const Npp32fc * *pSrc*, Npp32fc * *pDst*, int *nLength*, Npp32f *nLevel*)

32-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.20 **NppStatus nppsThreshold_GT_32fc_I** (Npp32fc * *pSrcDst*, int *nLength*, Npp32f *nLevel*)

32-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.21 **NppStatus nppsThreshold_GT_64f** (const Npp64f * *pSrc*, Npp64f * *pDst*, int *nLength*, Npp64f *nLevel*)

64-bit floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.22 NppStatus nppsThreshold_GT_64f_I (Npp64f * *pSrcDst*, int *nLength*, Npp64f *nLevel*)

64-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.23 NppStatus nppsThreshold_GT_64fc (const Npp64fc * *pSrc*, Npp64fc * *pDst*, int *nLength*, Npp64f *nLevel*)

64-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.24 NppStatus nppsThreshold_GT_64fc_I (Npp64fc * *pSrcDst*, int *nLength*, Npp64f *nLevel*)

64-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.25 NppStatus nppsThreshold_GTVal_16s (const Npp16s * *pSrc*, Npp16s * *pDst*, int *nLength*, Npp16s *nLevel*, Npp16s *nValue*)

16-bit signed short signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.26 NppStatus nppsThreshold_GTVal_16s_I (Npp16s * *pSrcDst*, int *nLength*, Npp16s *nLevel*, Npp16s *nValue*)

16-bit in place signed short signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.27 NppStatus nppsThreshold_GTVal_16sc (const Npp16sc * *pSrc*, Npp16sc * *pDst*, int *nLength*, Npp16s *nLevel*, Npp16sc *nValue*)

16-bit signed short complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.28 **NppStatus nppsThreshold_GTVal_16sc_I** (**Npp16sc** * *pSrcDst*, **int** *nLength*, **Npp16s** *nLevel*, **Npp16sc** *nValue*)

16-bit in place signed short complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.29 **NppStatus nppsThreshold_GTVal_32f** (**const Npp32f** * *pSrc*, **Npp32f** * *pDst*, **int** *nLength*, **Npp32f** *nLevel*, **Npp32f** *nValue*)

32-bit floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.30 **NppStatus nppsThreshold_GTVal_32f_I** (**Npp32f** * *pSrcDst*, **int** *nLength*, **Npp32f** *nLevel*, **Npp32f** *nValue*)

32-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.31 **NppStatus nppsThreshold_GTVal_32fc** (const Npp32fc * *pSrc*, Npp32fc * *pDst*, int *nLength*, Npp32f *nLevel*, Npp32fc *nValue*)

32-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.32 **NppStatus nppsThreshold_GTVal_32fc_I** (Npp32fc * *pSrcDst*, int *nLength*, Npp32f *nLevel*, Npp32fc *nValue*)

32-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.33 **NppStatus nppsThreshold_GTVal_64f** (const Npp64f * *pSrc*, Npp64f * *pDst*, int *nLength*, Npp64f *nLevel*, Npp64f *nValue*)

64-bit floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.34 **NppStatus nppsThreshold_GTVal_64f_I (Npp64f * pSrcDst, int nLength, Npp64f nLevel, Npp64f nValue)**

64-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.35 **NppStatus nppsThreshold_GTVal_64fc (const Npp64fc * pSrc, Npp64fc * pDst, int nLength, Npp64f nLevel, Npp64fc nValue)**

64-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.36 **NppStatus nppsThreshold_GTVal_64fc_I (Npp64fc * pSrcDst, int nLength, Npp64f nLevel, Npp64fc nValue)**

64-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.37 NppStatus nppsThreshold_LT_16s (const Npp16s * *pSrc*, Npp16s * *pDst*, int *nLength*, Npp16s *nLevel*)

16-bit signed short signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.38 NppStatus nppsThreshold_LT_16s_I (Npp16s * *pSrcDst*, int *nLength*, Npp16s *nLevel*)

16-bit in place signed short signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.39 NppStatus nppsThreshold_LT_16sc (const Npp16sc * *pSrc*, Npp16sc * *pDst*, int *nLength*, Npp16s *nLevel*)

16-bit signed short complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.40 NppStatus nppsThreshold_LT_16sc_I (Npp16sc * *pSrcDst*, int *nLength*, Npp16s *nLevel*)

16-bit in place signed short complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.41 NppStatus nppsThreshold_LT_32f (const Npp32f * *pSrc*, Npp32f * *pDst*, int *nLength*, Npp32f *nLevel*)

32-bit floating point signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.42 NppStatus nppsThreshold_LT_32f_I (Npp32f * *pSrcDst*, int *nLength*, Npp32f *nLevel*)

32-bit in place floating point signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.43 NppStatus nppsThreshold_LT_32fc (const Npp32fc * *pSrc*, Npp32fc * *pDst*, int *nLength*, Npp32f *nLevel*)

32-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.44 NppStatus nppsThreshold_LT_32fc_I (Npp32fc * *pSrcDst*, int *nLength*, Npp32f *nLevel*)

32-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.45 NppStatus nppsThreshold_LT_64f (const Npp64f * *pSrc*, Npp64f * *pDst*, int *nLength*, Npp64f *nLevel*)

64-bit floating point signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.46 NppStatus nppsThreshold_LT_64f_I (Npp64f * *pSrcDst*, int *nLength*, Npp64f *nLevel*)

64-bit in place floating point signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.47 NppStatus nppsThreshold_LT_64fc (const Npp64fc * *pSrc*, Npp64fc * *pDst*, int *nLength*, Npp64f *nLevel*)

64-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.48 NppStatus nppsThreshold_LT_64fc_I (Npp64fc * *pSrcDst*, int *nLength*, Npp64f *nLevel*)

64-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.180.1.49 NppStatus nppsThreshold_LTVal_16s (const Npp16s * *pSrc*, Npp16s * *pDst*, int *nLength*, Npp16s *nLevel*, Npp16s *nValue*)

16-bit signed short signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.50 NppStatus nppsThreshold_LTVal_16s_I (Npp16s * *pSrcDst*, int *nLength*, Npp16s *nLevel*, Npp16s *nValue*)

16-bit in place signed short signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.51 NppStatus nppsThreshold_LTVal_16sc (const Npp16sc * *pSrc*, Npp16sc * *pDst*, int *nLength*, Npp16s *nLevel*, Npp16sc *nValue*)

16-bit signed short complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.52 **NppStatus nppsThreshold_LTVVal_16sc_I** (Npp16sc * *pSrcDst*, int *nLength*, Npp16s *nLevel*, Npp16sc *nValue*)

16-bit in place signed short complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.53 **NppStatus nppsThreshold_LTVVal_32f** (const Npp32f * *pSrc*, Npp32f * *pDst*, int *nLength*, Npp32f *nLevel*, Npp32f *nValue*)

32-bit floating point signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.54 **NppStatus nppsThreshold_LTVVal_32f_I** (Npp32f * *pSrcDst*, int *nLength*, Npp32f *nLevel*, Npp32f *nValue*)

32-bit in place floating point signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.55 **NppStatus nppsThreshold_LTVal_32fc** (const Npp32fc * *pSrc*, Npp32fc * *pDst*, int *nLength*, Npp32f *nLevel*, Npp32fc *nValue*)

32-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.56 **NppStatus nppsThreshold_LTVal_32fc_I** (Npp32fc * *pSrcDst*, int *nLength*, Npp32f *nLevel*, Npp32fc *nValue*)

32-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.57 **NppStatus nppsThreshold_LTVal_64f** (const Npp64f * *pSrc*, Npp64f * *pDst*, int *nLength*, Npp64f *nLevel*, Npp64f *nValue*)

64-bit floating point signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.58 NppStatus nppsThreshold_LTVaI_64f_I (Npp64f * pSrcDst, int nLength, Npp64f nLevel, Npp64f nValue)

64-bit in place floating point signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.59 NppStatus nppsThreshold_LTVaC_64fc (const Npp64fc * pSrc, Npp64fc * pDst, int nLength, Npp64f nLevel, Npp64fc nValue)

64-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrc [Source Signal Pointer](#).

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.180.1.60 NppStatus nppsThreshold_LTVaC_64fc_I (Npp64fc * pSrcDst, int nLength, Npp64f nLevel, Npp64fc nValue)

64-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters:

pSrcDst [In-Place Signal Pointer](#).

nLength [Signal Length](#).

nLevel Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

nValue Constant value to replace source value when threshold test is true.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.181 Filtering Functions

Functions that provide functionality of generating output signal based on the input signal like signal integral, etc.

Modules

- [Integral](#)

Compute the indefinite interal of a given signal.

7.181.1 Detailed Description

Functions that provide functionality of generating output signal based on the input signal like signal integral, etc.

7.182 Integral

Compute the indefinite integral of a given signal.

Functions

- `NppStatus nppsIntegralGetBufferSize_32s` (int *nLength*, int **hpBufferSize*)
- `NppStatus nppsIntegral_32s` (const `Npp32s` **pSrc*, `Npp32s` **pDst*, int *nLength*, `Npp8u` **pDeviceBuffer*)

7.182.1 Detailed Description

Compute the indefinite integral of a given signal.

The i-th element is computed to be

$$s'_i = \sum_0^i s_j$$

7.182.2 Function Documentation

7.182.2.1 `NppStatus nppsIntegral_32s` (const `Npp32s` **pSrc*, `Npp32s` **pDst*, int *nLength*, `Npp8u` **pDeviceBuffer*)

7.182.2.2 `NppStatus nppsIntegralGetBufferSize_32s` (int *nLength*, int **hpBufferSize*)

7.183 Initialization

Modules

- [Set](#)
- [Zero](#)
- [Copy](#)

7.184 Set

Set

Set methods for 1D vectors of various types.

The copy methods operate on vector data given as a pointer to the underlying data-type (e.g. 8-bit vectors would be passed as pointers to Npp8u type) and length of the vectors, i.e. the number of items.

- [NppStatus nppsSet_8u](#) ([Npp8u](#) nValue, [Npp8u](#) *pDst, int nLength)
8-bit unsigned char, vector set method.
- [NppStatus nppsSet_8s](#) ([Npp8s](#) nValue, [Npp8s](#) *pDst, int nLength)
8-bit signed char, vector set method.
- [NppStatus nppsSet_16u](#) ([Npp16u](#) nValue, [Npp16u](#) *pDst, int nLength)
16-bit unsigned integer, vector set method.
- [NppStatus nppsSet_16s](#) ([Npp16s](#) nValue, [Npp16s](#) *pDst, int nLength)
16-bit signed integer, vector set method.
- [NppStatus nppsSet_16sc](#) ([Npp16sc](#) nValue, [Npp16sc](#) *pDst, int nLength)
16-bit integer complex, vector set method.
- [NppStatus nppsSet_32u](#) ([Npp32u](#) nValue, [Npp32u](#) *pDst, int nLength)
32-bit unsigned integer, vector set method.
- [NppStatus nppsSet_32s](#) ([Npp32s](#) nValue, [Npp32s](#) *pDst, int nLength)
32-bit signed integer, vector set method.
- [NppStatus nppsSet_32sc](#) ([Npp32sc](#) nValue, [Npp32sc](#) *pDst, int nLength)
32-bit integer complex, vector set method.
- [NppStatus nppsSet_32f](#) ([Npp32f](#) nValue, [Npp32f](#) *pDst, int nLength)
32-bit float, vector set method.
- [NppStatus nppsSet_32fc](#) ([Npp32fc](#) nValue, [Npp32fc](#) *pDst, int nLength)
32-bit float complex, vector set method.
- [NppStatus nppsSet_64s](#) ([Npp64s](#) nValue, [Npp64s](#) *pDst, int nLength)
64-bit long long integer, vector set method.
- [NppStatus nppsSet_64sc](#) ([Npp64sc](#) nValue, [Npp64sc](#) *pDst, int nLength)
64-bit long long integer complex, vector set method.
- [NppStatus nppsSet_64f](#) ([Npp64f](#) nValue, [Npp64f](#) *pDst, int nLength)
64-bit double, vector set method.
- [NppStatus nppsSet_64fc](#) ([Npp64fc](#) nValue, [Npp64fc](#) *pDst, int nLength)
64-bit double complex, vector set method.

7.184.1 Function Documentation

7.184.1.1 NppStatus nppsSet_16s (Npp16s *nValue*, Npp16s * *pDst*, int *nLength*)

16-bit signed integer, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.2 NppStatus nppsSet_16sc (Npp16sc *nValue*, Npp16sc * *pDst*, int *nLength*)

16-bit integer complex, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.3 NppStatus nppsSet_16u (Npp16u *nValue*, Npp16u * *pDst*, int *nLength*)

16-bit unsigned integer, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.4 NppStatus nppsSet_32f (Npp32f *nValue*, Npp32f * *pDst*, int *nLength*)

32-bit float, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.5 NppStatus nppsSet_32fc (Npp32fc *nValue*, Npp32fc * *pDst*, int *nLength*)

32-bit float complex, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.6 NppStatus nppsSet_32s (Npp32s *nValue*, Npp32s * *pDst*, int *nLength*)

32-bit signed integer, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.7 NppStatus nppsSet_32sc (Npp32sc *nValue*, Npp32sc * *pDst*, int *nLength*)

32-bit integer complex, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.8 NppStatus nppsSet_32u (Npp32u *nValue*, Npp32u * *pDst*, int *nLength*)

32-bit unsigned integer, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.9 NppStatus nppsSet_64f (Npp64f *nValue*, Npp64f * *pDst*, int *nLength*)

64-bit double, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.10 NppStatus nppsSet_64fc (Npp64fc *nValue*, Npp64fc * *pDst*, int *nLength*)

64-bit double complex, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.11 NppStatus nppsSet_64s (Npp64s *nValue*, Npp64s * *pDst*, int *nLength*)

64-bit long long integer, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.12 NppStatus nppsSet_64sc (Npp64sc nValue, Npp64sc * pDst, int nLength)

64-bit long long integer complex, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.13 NppStatus nppsSet_8s (Npp8s nValue, Npp8s * pDst, int nLength)

8-bit signed char, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.184.1.14 NppStatus nppsSet_8u (Npp8u nValue, Npp8u * pDst, int nLength)

8-bit unsigned char, vector set method.

Parameters:

nValue Value used to initialize the vector pDst.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.185 Zero

Zero

Set signals to zero.

- [NppStatus nppsZero_8u](#) ([Npp8u](#) *pDst, int nLength)
8-bit unsigned char, vector zero method.
- [NppStatus nppsZero_16s](#) ([Npp16s](#) *pDst, int nLength)
16-bit integer, vector zero method.
- [NppStatus nppsZero_16sc](#) ([Npp16sc](#) *pDst, int nLength)
16-bit integer complex, vector zero method.
- [NppStatus nppsZero_32s](#) ([Npp32s](#) *pDst, int nLength)
32-bit integer, vector zero method.
- [NppStatus nppsZero_32sc](#) ([Npp32sc](#) *pDst, int nLength)
32-bit integer complex, vector zero method.
- [NppStatus nppsZero_32f](#) ([Npp32f](#) *pDst, int nLength)
32-bit float, vector zero method.
- [NppStatus nppsZero_32fc](#) ([Npp32fc](#) *pDst, int nLength)
32-bit float complex, vector zero method.
- [NppStatus nppsZero_64s](#) ([Npp64s](#) *pDst, int nLength)
64-bit long long integer, vector zero method.
- [NppStatus nppsZero_64sc](#) ([Npp64sc](#) *pDst, int nLength)
64-bit long long integer complex, vector zero method.
- [NppStatus nppsZero_64f](#) ([Npp64f](#) *pDst, int nLength)
64-bit double, vector zero method.
- [NppStatus nppsZero_64fc](#) ([Npp64fc](#) *pDst, int nLength)
64-bit double complex, vector zero method.

7.185.1 Function Documentation

7.185.1.1 NppStatus nppsZero_16s (Npp16s * pDst, int nLength)

16-bit integer, vector zero method.

Parameters:

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.185.1.2 NppStatus nppsZero_16sc (Npp16sc * *pDst*, int *nLength*)

16-bit integer complex, vector zero method.

Parameters:

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.185.1.3 NppStatus nppsZero_32f (Npp32f * *pDst*, int *nLength*)

32-bit float, vector zero method.

Parameters:

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.185.1.4 NppStatus nppsZero_32fc (Npp32fc * *pDst*, int *nLength*)

32-bit float complex, vector zero method.

Parameters:

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.185.1.5 NppStatus nppsZero_32s (Npp32s * *pDst*, int *nLength*)

32-bit integer, vector zero method.

Parameters:

pDst [Destination Signal Pointer](#).

nLength [Signal Length](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.185.1.6 NppStatus nppsZero_32sc (Npp32sc * *pDst*, int *nLength*)

32-bit integer complex, vector zero method.

Parameters:

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.185.1.7 NppStatus nppsZero_64f (Npp64f * *pDst*, int *nLength*)

64-bit double, vector zero method.

Parameters:

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.185.1.8 NppStatus nppsZero_64fc (Npp64fc * *pDst*, int *nLength*)

64-bit double complex, vector zero method.

Parameters:

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.185.1.9 NppStatus nppsZero_64s (Npp64s * *pDst*, int *nLength*)

64-bit long long integer, vector zero method.

Parameters:

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.185.1.10 NppStatus nppsZero_64sc (Npp64sc * *pDst*, int *nLength*)

64-bit long long integer complex, vector zero method.

Parameters:

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.185.1.11 NppStatus nppsZero_8u (Npp8u * *pDst*, int *nLength*)

8-bit unsigned char, vector zero method.

Parameters:

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.186 Copy

Copy

Copy methods for various type signals.

Copy methods operate on signal data given as a pointer to the underlying data-type (e.g. 8-bit vectors would be passed as pointers to Npp8u type) and length of the vectors, i.e. the number of items.

- **NppStatus nppsCopy_8u** (const **Npp8u** *pSrc, **Npp8u** *pDst, int nLength)
8-bit unsigned char, vector copy method
- **NppStatus nppsCopy_16s** (const **Npp16s** *pSrc, **Npp16s** *pDst, int nLength)
16-bit signed short, vector copy method.
- **NppStatus nppsCopy_32s** (const **Npp32s** *pSrc, **Npp32s** *pDst, int nLength)
32-bit signed integer, vector copy method.
- **NppStatus nppsCopy_32f** (const **Npp32f** *pSrc, **Npp32f** *pDst, int nLength)
32-bit float, vector copy method.
- **NppStatus nppsCopy_64s** (const **Npp64s** *pSrc, **Npp64s** *pDst, int nLength)
64-bit signed integer, vector copy method.
- **NppStatus nppsCopy_16sc** (const **Npp16sc** *pSrc, **Npp16sc** *pDst, int nLength)
16-bit complex short, vector copy method.
- **NppStatus nppsCopy_32sc** (const **Npp32sc** *pSrc, **Npp32sc** *pDst, int nLength)
32-bit complex signed integer, vector copy method.
- **NppStatus nppsCopy_32fc** (const **Npp32fc** *pSrc, **Npp32fc** *pDst, int nLength)
32-bit complex float, vector copy method.
- **NppStatus nppsCopy_64sc** (const **Npp64sc** *pSrc, **Npp64sc** *pDst, int nLength)
64-bit complex signed integer, vector copy method.
- **NppStatus nppsCopy_64fc** (const **Npp64fc** *pSrc, **Npp64fc** *pDst, int nLength)
64-bit complex double, vector copy method.

7.186.1 Function Documentation

7.186.1.1 NppStatus nppsCopy_16s (const Npp16s *pSrc, Npp16s *pDst, int nLength)

16-bit signed short, vector copy method.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.186.1.2 NppStatus nppsCopy_16sc (const Npp16sc * *pSrc*, Npp16sc * *pDst*, int *nLength*)

16-bit complex short, vector copy method.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.186.1.3 NppStatus nppsCopy_32f (const Npp32f * *pSrc*, Npp32f * *pDst*, int *nLength*)

32-bit float, vector copy method.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.186.1.4 NppStatus nppsCopy_32fc (const Npp32fc * *pSrc*, Npp32fc * *pDst*, int *nLength*)

32-bit complex float, vector copy method.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.186.1.5 NppStatus nppsCopy_32s (const Npp32s * *pSrc*, Npp32s * *pDst*, int *nLength*)

32-bit signed integer, vector copy method.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.186.1.6 NppStatus nppsCopy_32sc (const Npp32sc * *pSrc*, Npp32sc * *pDst*, int *nLength*)

32-bit complex signed integer, vector copy method.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.186.1.7 NppStatus nppsCopy_64fc (const Npp64fc * *pSrc*, Npp64fc * *pDst*, int *nLength*)

64-bit complex double, vector copy method.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.186.1.8 NppStatus nppsCopy_64s (const Npp64s * *pSrc*, Npp64s * *pDst*, int *nLength*)

64-bit signed integer, vector copy method.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.186.1.9 NppStatus nppsCopy_64sc (const Npp64sc * *pSrc*, Npp64sc * *pDst*, int *nLength*)

64-bit complex signed integer, vector copy method.

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.186.1.10 NppStatus nppsCopy_8u (const Npp8u * *pSrc*, Npp8u * *pDst*, int *nLength*)

8-bit unsigned char, vector copy method

Parameters:

pSrc Source Signal Pointer.

pDst Destination Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.187 Statistical Functions

Functions that provide global signal statistics like: sum, mean, standard deviation, min, max, etc.

Modules

- [MinEvery And MaxEvery Functions](#)

Performs the min or max operation on the samples of a signal.

- [Sum](#)

signal_min_every_or_max_every

- [Maximum](#)
- [Minimum](#)
- [Mean](#)
- [Standard Deviation](#)
- [Mean And Standard Deviation](#)
- [Minimum_Maximum](#)
- [Infinity Norm](#)
- [L1 Norm](#)
- [L2 Norm](#)
- [Infinity Norm Diff](#)
- [L1 Norm Diff](#)
- [L2 Norm Diff](#)
- [Dot Product](#)
- [Count In Range](#)
- [Count Zero Crossings](#)
- [MaximumError](#)

Primitives for computing the maximum error between two signals.

- [AverageError](#)

Primitives for computing the Average error between two signals.

- [MaximumRelativeError](#)

Primitives for computing the MaximumRelative error between two signals.

- [AverageRelativeError](#)

Primitives for computing the AverageRelative error between two signals.

7.187.1 Detailed Description

Functions that provide global signal statistics like: sum, mean, standard deviation, min, max, etc.

7.188 MinEvery And MaxEvery Functions

Performs the min or max operation on the samples of a signal.

Functions

- **NppStatus nppsMinEvery_8u_I** (const **Npp8u** *pSrc, **Npp8u** *pSrcDst, int nLength)
8-bit in place min value for each pair of elements.
- **NppStatus nppsMinEvery_16u_I** (const **Npp16u** *pSrc, **Npp16u** *pSrcDst, int nLength)
16-bit unsigned short integer in place min value for each pair of elements.
- **NppStatus nppsMinEvery_16s_I** (const **Npp16s** *pSrc, **Npp16s** *pSrcDst, int nLength)
16-bit signed short integer in place min value for each pair of elements.
- **NppStatus nppsMinEvery_32s_I** (const **Npp32s** *pSrc, **Npp32s** *pSrcDst, int nLength)
32-bit signed integer in place min value for each pair of elements.
- **NppStatus nppsMinEvery_32f_I** (const **Npp32f** *pSrc, **Npp32f** *pSrcDst, int nLength)
32-bit floating point in place min value for each pair of elements.
- **NppStatus nppsMinEvery_64f_I** (const **Npp64f** *pSrc, **Npp64f** *pSrcDst, int nLength)
64-bit floating point in place min value for each pair of elements.
- **NppStatus nppsMaxEvery_8u_I** (const **Npp8u** *pSrc, **Npp8u** *pSrcDst, int nLength)
8-bit in place max value for each pair of elements.
- **NppStatus nppsMaxEvery_16u_I** (const **Npp16u** *pSrc, **Npp16u** *pSrcDst, int nLength)
16-bit unsigned short integer in place max value for each pair of elements.
- **NppStatus nppsMaxEvery_16s_I** (const **Npp16s** *pSrc, **Npp16s** *pSrcDst, int nLength)
16-bit signed short integer in place max value for each pair of elements.
- **NppStatus nppsMaxEvery_32s_I** (const **Npp32s** *pSrc, **Npp32s** *pSrcDst, int nLength)
32-bit signed integer in place max value for each pair of elements.
- **NppStatus nppsMaxEvery_32f_I** (const **Npp32f** *pSrc, **Npp32f** *pSrcDst, int nLength)
32-bit floating point in place max value for each pair of elements.

7.188.1 Detailed Description

Performs the min or max operation on the samples of a signal.

7.188.2 Function Documentation

7.188.2.1 NppStatus nppsMaxEvery_16s_I (const Npp16s *pSrc, Npp16s *pSrcDst, int nLength)

16-bit signed short integer in place max value for each pair of elements.

Parameters:

pSrc Source Signal Pointer.
pSrcDst In-Place Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.188.2.2 NppStatus nppsMaxEvery_16u_I (const Npp16u * pSrc, Npp16u * pSrcDst, int nLength)

16-bit unsigned short integer in place max value for each pair of elements.

Parameters:

pSrc Source Signal Pointer.
pSrcDst In-Place Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.188.2.3 NppStatus nppsMaxEvery_32f_I (const Npp32f * pSrc, Npp32f * pSrcDst, int nLength)

32-bit floating point in place max value for each pair of elements.

Parameters:

pSrc Source Signal Pointer.
pSrcDst In-Place Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.188.2.4 NppStatus nppsMaxEvery_32s_I (const Npp32s * pSrc, Npp32s * pSrcDst, int nLength)

32-bit signed integer in place max value for each pair of elements.

Parameters:

pSrc Source Signal Pointer.
pSrcDst In-Place Signal Pointer.
nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.188.2.5 NppStatus nppsMaxEvery_8u_I (const Npp8u * *pSrc*, Npp8u * *pSrcDst*, int *nLength*)

8-bit in place max value for each pair of elements.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.188.2.6 NppStatus nppsMinEvery_16s_I (const Npp16s * *pSrc*, Npp16s * *pSrcDst*, int *nLength*)

16-bit signed short integer in place min value for each pair of elements.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.188.2.7 NppStatus nppsMinEvery_16u_I (const Npp16u * *pSrc*, Npp16u * *pSrcDst*, int *nLength*)

16-bit unsigned short integer in place min value for each pair of elements.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.188.2.8 NppStatus nppsMinEvery_32f_I (const Npp32f * *pSrc*, Npp32f * *pSrcDst*, int *nLength*)

32-bit floating point in place min value for each pair of elements.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.188.2.9 NppStatus nppsMinEvery_32s_I (const Npp32s * *pSrc*, Npp32s * *pSrcDst*, int *nLength*)

32-bit signed integer in place min value for each pair of elements.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.188.2.10 NppStatus nppsMinEvery_64f_I (const Npp64f * *pSrc*, Npp64f * *pSrcDst*, int *nLength*)

64-bit floating point in place min value for each pair of elements.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.188.2.11 NppStatus nppsMinEvery_8u_I (const Npp8u * *pSrc*, Npp8u * *pSrcDst*, int *nLength*)

8-bit in place min value for each pair of elements.

Parameters:

pSrc Source Signal Pointer.

pSrcDst In-Place Signal Pointer.

nLength Signal Length.

Returns:

Signal Data Related Error Codes, Length Related Error Codes.

7.189 Sum

signal_min_every_or_max_every

Functions

- [NppStatus nppsSumGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsSum_32f.
- [NppStatus nppsSumGetBufferSize_32fc](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsSum_32fc.
- [NppStatus nppsSumGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsSum_64f.
- [NppStatus nppsSumGetBufferSize_64fc](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsSum_64fc.
- [NppStatus nppsSumGetBufferSize_16s_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsSum_16s_Sfs.
- [NppStatus nppsSumGetBufferSize_16sc_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsSum_16sc_Sfs.
- [NppStatus nppsSumGetBufferSize_16sc32sc_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsSum_16sc32sc_Sfs.
- [NppStatus nppsSumGetBufferSize_32s_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsSum_32s_Sfs.
- [NppStatus nppsSumGetBufferSize_16s32s_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsSum_16s32s_Sfs.
- [NppStatus nppsSum_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pSum, [Npp8u](#) *pDeviceBuffer)
32-bit float vector sum method
- [NppStatus nppsSum_32fc](#) (const [Npp32fc](#) *pSrc, int nLength, [Npp32fc](#) *pSum, [Npp8u](#) *pDeviceBuffer)
32-bit float complex vector sum method
- [NppStatus nppsSum_64f](#) (const [Npp64f](#) *pSrc, int nLength, [Npp64f](#) *pSum, [Npp8u](#) *pDeviceBuffer)
64-bit double vector sum method
- [NppStatus nppsSum_64fc](#) (const [Npp64fc](#) *pSrc, int nLength, [Npp64fc](#) *pSum, [Npp8u](#) *pDeviceBuffer)
64-bit double complex vector sum method

- **NppStatus nppsSum_16s_Sfs** (const **Npp16s** *pSrc, int nLength, **Npp16s** *pSum, int nScaleFactor, **Npp8u** *pDeviceBuffer)
16-bit short vector sum with integer scaling method
- **NppStatus nppsSum_32s_Sfs** (const **Npp32s** *pSrc, int nLength, **Npp32s** *pSum, int nScaleFactor, **Npp8u** *pDeviceBuffer)
32-bit integer vector sum with integer scaling method
- **NppStatus nppsSum_16sc_Sfs** (const **Npp16sc** *pSrc, int nLength, **Npp16sc** *pSum, int nScaleFactor, **Npp8u** *pDeviceBuffer)
16-bit short complex vector sum with integer scaling method
- **NppStatus nppsSum_16sc32sc_Sfs** (const **Npp16sc** *pSrc, int nLength, **Npp32sc** *pSum, int nScaleFactor, **Npp8u** *pDeviceBuffer)
16-bit short complex vector sum (32bit int complex) with integer scaling method
- **NppStatus nppsSum_16s32s_Sfs** (const **Npp16s** *pSrc, int nLength, **Npp32s** *pSum, int nScaleFactor, **Npp8u** *pDeviceBuffer)
16-bit integer vector sum (32bit) with integer scaling method

7.189.1 Detailed Description

signal_min_every_or_max_every

7.189.2 Function Documentation

7.189.2.1 NppStatus nppsSum_16s32s_Sfs (const Npp16s *pSrc, int nLength, Npp32s *pSum, int nScaleFactor, Npp8u *pDeviceBuffer)

16-bit integer vector sum (32bit) with integer scaling method

Parameters:

pSrc Source Signal Pointer.

nLength Signal Length.

pSum Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsSumGetBufferSize_16s32s_Sfs](#) to determine the minium number of bytes required.

nScaleFactor Integer Result Scaling.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.189.2.2 NppStatus nppsSum_16s_Sfs (const Npp16s *pSrc, int nLength, Npp16s *pSum, int nScaleFactor, Npp8u *pDeviceBuffer)

16-bit short vector sum with integer scaling method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pSum Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsSumGetBufferSize_16s_Sfs](#) to determine the minium number of bytes required.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.189.2.3 `NppStatus nppsSum_16sc32sc_Sfs (const Npp16sc * pSrc, int nLength, Npp32sc * pSum, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit short complex vector sum (32bit int complex) with integer scaling method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pSum Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsSumGetBufferSize_16sc32sc_Sfs](#) to determine the minium number of bytes required.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.189.2.4 `NppStatus nppsSum_16sc_Sfs (const Npp16sc * pSrc, int nLength, Npp16sc * pSum, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit short complex vector sum with integer scaling method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pSum Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsSumGetBufferSize_16sc_Sfs](#) to determine the minium number of bytes required.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.189.2.5 NppStatus nppsSum_32f (const Npp32f * *pSrc*, int *nLength*, Npp32f * *pSum*, Npp8u * *pDeviceBuffer*)

32-bit float vector sum method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pSum Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsSumGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.189.2.6 NppStatus nppsSum_32fc (const Npp32fc * *pSrc*, int *nLength*, Npp32fc * *pSum*, Npp8u * *pDeviceBuffer*)

32-bit float complex vector sum method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pSum Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsSumGetBufferSize_32fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.189.2.7 NppStatus nppsSum_32s_Sfs (const Npp32s * *pSrc*, int *nLength*, Npp32s * *pSum*, int *nScaleFactor*, Npp8u * *pDeviceBuffer*)

32-bit integer vector sum with integer scaling method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pSum Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsSumGetBufferSize_32s_Sfs](#) to determine the minium number of bytes required.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.189.2.8 NppStatus nppsSum_64f (const Npp64f * *pSrc*, int *nLength*, Npp64f * *pSum*, Npp8u * *pDeviceBuffer*)

64-bit double vector sum method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pSum Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsSumGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.189.2.9 NppStatus nppsSum_64fc (const Npp64fc * *pSrc*, int *nLength*, Npp64fc * *pSum*, Npp8u * *pDeviceBuffer*)

64-bit double complex vector sum method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pSum Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsSumGetBufferSize_64fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.189.2.10 NppStatus nppsSumGetBufferSize_16s32s_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsSum_16s32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.189.2.11 NppStatus nppsSumGetBufferSize_16s_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsSum_16s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.189.2.12 NppStatus nppsSumGetBufferSize_16sc32sc_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsSum_16sc32sc_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.189.2.13 NppStatus nppsSumGetBufferSize_16sc_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsSum_16sc_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.189.2.14 NppStatus nppsSumGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsSum_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.189.2.15 NppStatus nppsSumGetBufferSize_32fc (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsSum_32fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.189.2.16 NppStatus nppsSumGetBufferSize_32s_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsSum_32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.189.2.17 NppStatus nppsSumGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsSum_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.189.2.18 NppStatus nppsSumGetBufferSize_64fc (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsSum_64fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190 Maximum

Functions

- [NppStatus nppsMaxGetBufferSize_16s](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMax_16s.
- [NppStatus nppsMaxGetBufferSize_32s](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMax_32s.
- [NppStatus nppsMaxGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMax_32f.
- [NppStatus nppsMaxGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMax_64f.
- [NppStatus nppsMax_16s](#) (const [Npp16s](#) *pSrc, int nLength, [Npp16s](#) *pMax, [Npp8u](#) *pDeviceBuffer)
16-bit integer vector max method
- [NppStatus nppsMax_32s](#) (const [Npp32s](#) *pSrc, int nLength, [Npp32s](#) *pMax, [Npp8u](#) *pDeviceBuffer)
32-bit integer vector max method
- [NppStatus nppsMax_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pMax, [Npp8u](#) *pDeviceBuffer)
32-bit float vector max method
- [NppStatus nppsMax_64f](#) (const [Npp64f](#) *pSrc, int nLength, [Npp64f](#) *pMax, [Npp8u](#) *pDeviceBuffer)
64-bit float vector max method
- [NppStatus nppsMaxIndxGetBufferSize_16s](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMaxIndx_16s.
- [NppStatus nppsMaxIndxGetBufferSize_32s](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMaxIndx_32s.
- [NppStatus nppsMaxIndxGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMaxIndx_32f.
- [NppStatus nppsMaxIndxGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMaxIndx_64f.
- [NppStatus nppsMaxIndx_16s](#) (const [Npp16s](#) *pSrc, int nLength, [Npp16s](#) *pMax, int *pIndx, [Npp8u](#) *pDeviceBuffer)
16-bit integer vector max index method
- [NppStatus nppsMaxIndx_32s](#) (const [Npp32s](#) *pSrc, int nLength, [Npp32s](#) *pMax, int *pIndx, [Npp8u](#) *pDeviceBuffer)

32-bit integer vector max index method

- **NppStatus nppsMaxIndx_32f** (const **Npp32f** *pSrc, int nLength, **Npp32f** *pMax, int *pIndx, **Npp8u** *pDeviceBuffer)

32-bit float vector max index method

- **NppStatus nppsMaxIndx_64f** (const **Npp64f** *pSrc, int nLength, **Npp64f** *pMax, int *pIndx, **Npp8u** *pDeviceBuffer)

64-bit float vector max index method

- **NppStatus nppsMaxAbsGetBufferSize_16s** (int nLength, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppsMaxAbs_16s.

- **NppStatus nppsMaxAbsGetBufferSize_32s** (int nLength, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppsMaxAbs_32s.

- **NppStatus nppsMaxAbs_16s** (const **Npp16s** *pSrc, int nLength, **Npp16s** *pMaxAbs, **Npp8u** *pDeviceBuffer)

16-bit integer vector max absolute method

- **NppStatus nppsMaxAbs_32s** (const **Npp32s** *pSrc, int nLength, **Npp32s** *pMaxAbs, **Npp8u** *pDeviceBuffer)

32-bit integer vector max absolute method

- **NppStatus nppsMaxAbsIndxGetBufferSize_16s** (int nLength, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppsMaxAbsIndx_16s.

- **NppStatus nppsMaxAbsIndxGetBufferSize_32s** (int nLength, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppsMaxAbsIndx_32s.

- **NppStatus nppsMaxAbsIndx_16s** (const **Npp16s** *pSrc, int nLength, **Npp16s** *pMaxAbs, int *pIndx, **Npp8u** *pDeviceBuffer)

16-bit integer vector max absolute index method

- **NppStatus nppsMaxAbsIndx_32s** (const **Npp32s** *pSrc, int nLength, **Npp32s** *pMaxAbs, int *pIndx, **Npp8u** *pDeviceBuffer)

32-bit integer vector max absolute index method

7.190.1 Function Documentation

7.190.1.1 **NppStatus nppsMax_16s** (const **Npp16s** *pSrc, int nLength, **Npp16s** *pMax, **Npp8u** *pDeviceBuffer)

16-bit integer vector max method

Parameters:

pSrc Source Signal Pointer.

nLength Signal Length.

pMax Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.2 NppStatus nppsMax_32f (const Npp32f * pSrc, int nLength, Npp32f * pMax, Npp8u * pDeviceBuffer)

32-bit float vector max method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMax Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.3 NppStatus nppsMax_32s (const Npp32s * pSrc, int nLength, Npp32s * pMax, Npp8u * pDeviceBuffer)

32-bit integer vector max method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMax Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.4 NppStatus nppsMax_64f (const Npp64f * pSrc, int nLength, Npp64f * pMax, Npp8u * pDeviceBuffer)

64-bit float vector max method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMax Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.5 NppStatus nppsMaxAbs_16s (const Npp16s * pSrc, int nLength, Npp16s * pMaxAbs, Npp8u * pDeviceBuffer)

16-bit integer vector max absolute method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMaxAbs Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxAbsGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.6 NppStatus nppsMaxAbs_32s (const Npp32s * pSrc, int nLength, Npp32s * pMaxAbs, Npp8u * pDeviceBuffer)

32-bit integer vector max absolute method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMaxAbs Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxAbsGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.7 NppStatus nppsMaxAbsGetBufferSize_16s (int nLength, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppsMaxAbs_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190.1.8 NppStatus nppsMaxAbsGetBufferSize_32s (int nLength, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppsMaxAbs_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190.1.9 NppStatus nppsMaxAbsIndx_16s (const Npp16s * pSrc, int nLength, Npp16s * pMaxAbs, int * pIndx, Npp8u * pDeviceBuffer)

16-bit integer vector max absolute index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMaxAbs Pointer to the output result.

pIndx Pointer to the index value of the first maximum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxAbsIndxGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.10 NppStatus nppsMaxAbsIndx_32s (const Npp32s * pSrc, int nLength, Npp32s * pMaxAbs, int * pIndx, Npp8u * pDeviceBuffer)

32-bit integer vector max absolute index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMaxAbs Pointer to the output result.

pIndx Pointer to the index value of the first maximum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxAbsIndxGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.11 NppStatus nppsMaxAbsIndxGetBufferSize_16s (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMaxAbsIndx_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190.1.12 NppStatus nppsMaxAbsIndxGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMaxAbsIndx_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190.1.13 NppStatus nppsMaxGetBufferSize_16s (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMax_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190.1.14 NppStatus nppsMaxGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMax_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190.1.15 NppStatus nppsMaxGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMax_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190.1.16 NppStatus nppsMaxGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMax_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190.1.17 NppStatus nppsMaxIndx_16s (const Npp16s * *pSrc*, int *nLength*, Npp16s * *pMax*, int * *pIndx*, Npp8u * *pDeviceBuffer*)

16-bit integer vector max index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMax Pointer to the output result.

pIndx Pointer to the index value of the first maximum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxIndxGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.18 NppStatus nppsMaxIndx_32f (const Npp32f * pSrc, int nLength, Npp32f * pMax, int * pIndx, Npp8u * pDeviceBuffer)

32-bit float vector max index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMax Pointer to the output result.

pIndx Pointer to the index value of the first maximum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxIndxGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.19 NppStatus nppsMaxIndx_32s (const Npp32s * pSrc, int nLength, Npp32s * pMax, int * pIndx, Npp8u * pDeviceBuffer)

32-bit integer vector max index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMax Pointer to the output result.

pIndx Pointer to the index value of the first maximum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxIndxGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.20 **NppStatus nppsMaxIndx_64f** (const Npp64f * *pSrc*, int *nLength*, Npp64f * *pMax*, int * *pIndx*, Npp8u * *pDeviceBuffer*)

64-bit float vector max index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMax Pointer to the output result.

pIndx Pointer to the index value of the first maximum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaxIndxGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.190.1.21 **NppStatus nppsMaxIndxGetBufferSize_16s** (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMaxIndx_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190.1.22 **NppStatus nppsMaxIndxGetBufferSize_32f** (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMaxIndx_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190.1.23 NppStatus nppsMaxIndxGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMaxIndx_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.190.1.24 NppStatus nppsMaxIndxGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMaxIndx_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191 Minimum

Functions

- [NppStatus nppsMinGetBufferSize_16s](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMin_16s.
- [NppStatus nppsMinGetBufferSize_32s](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMin_32s.
- [NppStatus nppsMinGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMin_32f.
- [NppStatus nppsMinGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMin_64f.
- [NppStatus nppsMin_16s](#) (const [Npp16s](#) *pSrc, int nLength, [Npp16s](#) *pMin, [Npp8u](#) *pDeviceBuffer)
16-bit integer vector min method
- [NppStatus nppsMin_32s](#) (const [Npp32s](#) *pSrc, int nLength, [Npp32s](#) *pMin, [Npp8u](#) *pDeviceBuffer)
32-bit integer vector min method
- [NppStatus nppsMin_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pMin, [Npp8u](#) *pDeviceBuffer)
32-bit integer vector min method
- [NppStatus nppsMin_64f](#) (const [Npp64f](#) *pSrc, int nLength, [Npp64f](#) *pMin, [Npp8u](#) *pDeviceBuffer)
64-bit integer vector min method
- [NppStatus nppsMinIndxGetBufferSize_16s](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMinIndx_16s.
- [NppStatus nppsMinIndxGetBufferSize_32s](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMinIndx_32s.
- [NppStatus nppsMinIndxGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMinIndx_32f.
- [NppStatus nppsMinIndxGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMinIndx_64f.
- [NppStatus nppsMinIndx_16s](#) (const [Npp16s](#) *pSrc, int nLength, [Npp16s](#) *pMin, int *pIndx, [Npp8u](#) *pDeviceBuffer)
16-bit integer vector min index method
- [NppStatus nppsMinIndx_32s](#) (const [Npp32s](#) *pSrc, int nLength, [Npp32s](#) *pMin, int *pIndx, [Npp8u](#) *pDeviceBuffer)

32-bit integer vector min index method

- **NppStatus nppsMinIndx_32f** (const **Npp32f** *pSrc, int nLength, **Npp32f** *pMin, int *pIndx, **Npp8u** *pDeviceBuffer)

32-bit float vector min index method

- **NppStatus nppsMinIndx_64f** (const **Npp64f** *pSrc, int nLength, **Npp64f** *pMin, int *pIndx, **Npp8u** *pDeviceBuffer)

64-bit float vector min index method

- **NppStatus nppsMinAbsGetBufferSize_16s** (int nLength, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppsMinAbs_16s.

- **NppStatus nppsMinAbsGetBufferSize_32s** (int nLength, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppsMinAbs_32s.

- **NppStatus nppsMinAbs_16s** (const **Npp16s** *pSrc, int nLength, **Npp16s** *pMinAbs, **Npp8u** *pDeviceBuffer)

16-bit integer vector min absolute method

- **NppStatus nppsMinAbs_32s** (const **Npp32s** *pSrc, int nLength, **Npp32s** *pMinAbs, **Npp8u** *pDeviceBuffer)

32-bit integer vector min absolute method

- **NppStatus nppsMinAbsIndxGetBufferSize_16s** (int nLength, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppsMinAbsIndx_16s.

- **NppStatus nppsMinAbsIndxGetBufferSize_32s** (int nLength, int *hpBufferSize)

Device scratch buffer size (in bytes) for nppsMinAbsIndx_32s.

- **NppStatus nppsMinAbsIndx_16s** (const **Npp16s** *pSrc, int nLength, **Npp16s** *pMinAbs, int *pIndx, **Npp8u** *pDeviceBuffer)

16-bit integer vector min absolute index method

- **NppStatus nppsMinAbsIndx_32s** (const **Npp32s** *pSrc, int nLength, **Npp32s** *pMinAbs, int *pIndx, **Npp8u** *pDeviceBuffer)

32-bit integer vector min absolute index method

7.191.1 Function Documentation

7.191.1.1 **NppStatus nppsMin_16s** (const **Npp16s** *pSrc, int nLength, **Npp16s** *pMin, **Npp8u** *pDeviceBuffer)

16-bit integer vector min method

Parameters:

pSrc Source Signal Pointer.

nLength Signal Length.

pMin Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.2 NppStatus nppsMin_32f (const Npp32f * pSrc, int nLength, Npp32f * pMin, Npp8u * pDeviceBuffer)

32-bit integer vector min method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.3 NppStatus nppsMin_32s (const Npp32s * pSrc, int nLength, Npp32s * pMin, Npp8u * pDeviceBuffer)

32-bit integer vector min method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.4 NppStatus nppsMin_64f (const Npp64f * pSrc, int nLength, Npp64f * pMin, Npp8u * pDeviceBuffer)

64-bit integer vector min method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.5 NppStatus nppsMinAbs_16s (const Npp16s * pSrc, int nLength, Npp16s * pMinAbs, Npp8u * pDeviceBuffer)

16-bit integer vector min absolute method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMinAbs Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinAbsGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.6 NppStatus nppsMinAbs_32s (const Npp32s * pSrc, int nLength, Npp32s * pMinAbs, Npp8u * pDeviceBuffer)

32-bit integer vector min absolute method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMinAbs Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinAbsGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.7 NppStatus nppsMinAbsGetBufferSize_16s (int nLength, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppsMinAbs_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191.1.8 NppStatus nppsMinAbsGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for *nppsMinAbs_32s*.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191.1.9 NppStatus nppsMinAbsIndx_16s (const Npp16s * *pSrc*, int *nLength*, Npp16s * *pMinAbs*, int * *pIndx*, Npp8u * *pDeviceBuffer*)

16-bit integer vector min absolute index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMinAbs Pointer to the output result.

pIndx Pointer to the index value of the first minimum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinAbsIndxGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.10 NppStatus nppsMinAbsIndx_32s (const Npp32s * *pSrc*, int *nLength*, Npp32s * *pMinAbs*, int * *pIndx*, Npp8u * *pDeviceBuffer*)

32-bit integer vector min absolute index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMinAbs Pointer to the output result.

pIdx Pointer to the index value of the first minimum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinAbsIndxGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.11 NppStatus nppsMinAbsIndxGetBufferSize_16s (int nLength, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppsMinAbsIndx_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191.1.12 NppStatus nppsMinAbsIndxGetBufferSize_32s (int nLength, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppsMinAbsIndx_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191.1.13 NppStatus nppsMinGetBufferSize_16s (int nLength, int * hpBufferSize)

Device scratch buffer size (in bytes) for nppsMin_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191.1.14 NppStatus nppsMinGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMin_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191.1.15 NppStatus nppsMinGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMin_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191.1.16 NppStatus nppsMinGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMin_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191.1.17 NppStatus nppsMinIndx_16s (const Npp16s * *pSrc*, int *nLength*, Npp16s * *pMin*, int * *pIndx*, Npp8u * *pDeviceBuffer*)

16-bit integer vector min index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the output result.

pIndx Pointer to the index value of the first minimum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinIndxGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.18 NppStatus nppsMinIndx_32f (const Npp32f * pSrc, int nLength, Npp32f * pMin, int * pIndx, Npp8u * pDeviceBuffer)

32-bit float vector min index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the output result.

pIndx Pointer to the index value of the first minimum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinIndxGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.19 NppStatus nppsMinIndx_32s (const Npp32s * pSrc, int nLength, Npp32s * pMin, int * pIndx, Npp8u * pDeviceBuffer)

32-bit integer vector min index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the output result.

pIndx Pointer to the index value of the first minimum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinIndxGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.20 **NppStatus nppsMinIndx_64f** (const Npp64f * *pSrc*, int *nLength*, Npp64f * *pMin*, int * *pIndx*, Npp8u * *pDeviceBuffer*)

64-bit float vector min index method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the output result.

pIndx Pointer to the index value of the first minimum element.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinIndxGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.191.1.21 **NppStatus nppsMinIndxGetBufferSize_16s** (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMinIndx_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191.1.22 **NppStatus nppsMinIndxGetBufferSize_32f** (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMinIndx_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191.1.23 NppStatus nppsMinIndxGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMinIndx_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.191.1.24 NppStatus nppsMinIndxGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMinIndx_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.192 Mean

Functions

- [NppStatus nppsMeanGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMean_32f.
- [NppStatus nppsMeanGetBufferSize_32fc](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMean_32fc.
- [NppStatus nppsMeanGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMean_64f.
- [NppStatus nppsMeanGetBufferSize_64fc](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMean_64fc.
- [NppStatus nppsMeanGetBufferSize_16s_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMean_16s_Sfs.
- [NppStatus nppsMeanGetBufferSize_32s_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMean_32s_Sfs.
- [NppStatus nppsMeanGetBufferSize_16sc_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMean_16sc_Sfs.
- [NppStatus nppsMean_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pMean, [Npp8u](#) *pDeviceBuffer)
32-bit float vector mean method
- [NppStatus nppsMean_32fc](#) (const [Npp32fc](#) *pSrc, int nLength, [Npp32fc](#) *pMean, [Npp8u](#) *pDeviceBuffer)
32-bit float complex vector mean method
- [NppStatus nppsMean_64f](#) (const [Npp64f](#) *pSrc, int nLength, [Npp64f](#) *pMean, [Npp8u](#) *pDeviceBuffer)
64-bit double vector mean method
- [NppStatus nppsMean_64fc](#) (const [Npp64fc](#) *pSrc, int nLength, [Npp64fc](#) *pMean, [Npp8u](#) *pDeviceBuffer)
64-bit double complex vector mean method
- [NppStatus nppsMean_16s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp16s](#) *pMean, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit short vector mean with integer scaling method
- [NppStatus nppsMean_32s_Sfs](#) (const [Npp32s](#) *pSrc, int nLength, [Npp32s](#) *pMean, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
32-bit integer vector mean with integer scaling method
- [NppStatus nppsMean_16sc_Sfs](#) (const [Npp16sc](#) *pSrc, int nLength, [Npp16sc](#) *pMean, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)

16-bit short complex vector mean with integer scaling method

7.192.1 Function Documentation

7.192.1.1 `NppStatus nppsMean_16s_Sfs (const Npp16s * pSrc, int nLength, Npp16s * pMean, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit short vector mean with integer scaling method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMean Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMeanGetBufferSize_16s_Sfs](#) to determine the minium number of bytes required.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.192.1.2 `NppStatus nppsMean_16sc_Sfs (const Npp16sc * pSrc, int nLength, Npp16sc * pMean, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit short complex vector mean with integer scaling method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMean Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMeanGetBufferSize_16sc_Sfs](#) to determine the minium number of bytes required.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.192.1.3 `NppStatus nppsMean_32f (const Npp32f * pSrc, int nLength, Npp32f * pMean, Npp8u * pDeviceBuffer)`

32-bit float vector mean method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMean Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMeanGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.192.1.4 NppStatus nppsMean_32fc (const Npp32fc * pSrc, int nLength, Npp32fc * pMean, Npp8u * pDeviceBuffer)

32-bit float complex vector mean method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMean Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMeanGetBufferSize_32fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.192.1.5 NppStatus nppsMean_32s_Sfs (const Npp32s * pSrc, int nLength, Npp32s * pMean, int nScaleFactor, Npp8u * pDeviceBuffer)

32-bit integer vector mean with integer scaling method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMean Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMeanGetBufferSize_32s_Sfs](#) to determine the minium number of bytes required.

nScaleFactor [Integer Result Scaling](#).

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.192.1.6 NppStatus nppsMean_64f (const Npp64f * *pSrc*, int *nLength*, Npp64f * *pMean*, Npp8u * *pDeviceBuffer*)

64-bit double vector mean method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMean Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMeanGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.192.1.7 NppStatus nppsMean_64fc (const Npp64fc * *pSrc*, int *nLength*, Npp64fc * *pMean*, Npp8u * *pDeviceBuffer*)

64-bit double complex vector mean method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMean Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMeanGetBufferSize_64fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.192.1.8 NppStatus nppsMeanGetBufferSize_16s_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMean_16s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.192.1.9 NppStatus nppsMeanGetBufferSize_16sc_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMean_16sc_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.192.1.10 NppStatus nppsMeanGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMean_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.192.1.11 NppStatus nppsMeanGetBufferSize_32fc (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMean_32fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.192.1.12 NppStatus nppsMeanGetBufferSize_32s_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMean_32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.192.1.13 NppStatus nppsMeanGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMean_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.192.1.14 NppStatus nppsMeanGetBufferSize_64fc (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMean_64fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.193 Standard Deviation

Functions

- [NppStatus nppsStdDevGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsStdDev_32f.
- [NppStatus nppsStdDevGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsStdDev_64f.
- [NppStatus nppsStdDevGetBufferSize_16s32s_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsStdDev_16s32s_Sfs.
- [NppStatus nppsStdDevGetBufferSize_16s_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsStdDev_16s_Sfs.
- [NppStatus nppsStdDev_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pStdDev, [Npp8u](#) *pDeviceBuffer)
32-bit float vector standard deviation method
- [NppStatus nppsStdDev_64f](#) (const [Npp64f](#) *pSrc, int nLength, [Npp64f](#) *pStdDev, [Npp8u](#) *pDeviceBuffer)
64-bit float vector standard deviation method
- [NppStatus nppsStdDev_16s32s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32s](#) *pStdDev, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit float vector standard deviation method (return value is 32-bit)
- [NppStatus nppsStdDev_16s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp16s](#) *pStdDev, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit float vector standard deviation method (return value is also 16-bit)

7.193.1 Function Documentation

7.193.1.1 [NppStatus nppsStdDev_16s32s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32s](#) *pStdDev, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)

16-bit float vector standard deviation method (return value is 32-bit)

Parameters:

[pSrc](#) [Source Signal Pointer](#).

[nLength](#) [Signal Length](#).

[pStdDev](#) Pointer to the output result.

[nScaleFactor](#) [Integer Result Scaling](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsStdDevGetBufferSize_16s32s_Sfs](#) to determine the minimum number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.193.1.2 NppStatus nppsStdDev_16s_Sfs (const Npp16s * *pSrc*, int *nLength*, Npp16s * *pStdDev*, int *nScaleFactor*, Npp8u * *pDeviceBuffer*)

16-bit float vector standard deviation method (return value is also 16-bit)

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pStdDev Pointer to the output result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsStdDevGetBufferSize_16s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.193.1.3 NppStatus nppsStdDev_32f (const Npp32f * *pSrc*, int *nLength*, Npp32f * *pStdDev*, Npp8u * *pDeviceBuffer*)

32-bit float vector standard deviation method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pStdDev Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsStdDevGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.193.1.4 NppStatus nppsStdDev_64f (const Npp64f * *pSrc*, int *nLength*, Npp64f * *pStdDev*, Npp8u * *pDeviceBuffer*)

64-bit float vector standard deviation method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pStdDev Pointer to the output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsStdDevGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.193.1.5 NppStatus nppsStdDevGetBufferSize_16s32s_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsStdDev_16s32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.193.1.6 NppStatus nppsStdDevGetBufferSize_16s_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsStdDev_16s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.193.1.7 NppStatus nppsStdDevGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsStdDev_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.193.1.8 NppStatus nppsStdDevGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsStdDev_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.194 Mean And Standard Deviation

Functions

- [NppStatus nppsMeanStdDevGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMeanStdDev_32f.
- [NppStatus nppsMeanStdDevGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMeanStdDev_64f.
- [NppStatus nppsMeanStdDevGetBufferSize_16s32s_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMeanStdDev_16s32s_Sfs.
- [NppStatus nppsMeanStdDevGetBufferSize_16s_Sfs](#) (int nLength, int *hpBufferSize)
Device scratch buffer size (in bytes) for nppsMeanStdDev_16s_Sfs.
- [NppStatus nppsMeanStdDev_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pMean, [Npp32f](#) *pStdDev, [Npp8u](#) *pDeviceBuffer)
32-bit float vector mean and standard deviation method
- [NppStatus nppsMeanStdDev_64f](#) (const [Npp64f](#) *pSrc, int nLength, [Npp64f](#) *pMean, [Npp64f](#) *pStdDev, [Npp8u](#) *pDeviceBuffer)
64-bit float vector mean and standard deviation method
- [NppStatus nppsMeanStdDev_16s32s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32s](#) *pMean, [Npp32s](#) *pStdDev, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit float vector mean and standard deviation method (return values are 32-bit)
- [NppStatus nppsMeanStdDev_16s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp16s](#) *pMean, [Npp16s](#) *pStdDev, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit float vector mean and standard deviation method (return values are also 16-bit)

7.194.1 Function Documentation

7.194.1.1 [NppStatus nppsMeanStdDev_16s32s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32s](#) *pMean, [Npp32s](#) *pStdDev, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)

16-bit float vector mean and standard deviation method (return values are 32-bit)

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMean Pointer to the output mean value.

pStdDev Pointer to the output standard deviation value.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMeanStdDevGetBufferSize_16s32s_Sfs](#) to determine the minimum number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.194.1.2 NppStatus nppsMeanStdDev_16s_Sfs (const Npp16s * *pSrc*, int *nLength*, Npp16s * *pMean*, Npp16s * *pStdDev*, int *nScaleFactor*, Npp8u * *pDeviceBuffer*)

16-bit float vector mean and standard deviation method (return values are also 16-bit)

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMean Pointer to the output mean value.

pStdDev Pointer to the output standard deviation value.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMeanStdDevGetBufferSize_16s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.194.1.3 NppStatus nppsMeanStdDev_32f (const Npp32f * *pSrc*, int *nLength*, Npp32f * *pMean*, Npp32f * *pStdDev*, Npp8u * *pDeviceBuffer*)

32-bit float vector mean and standard deviation method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMean Pointer to the output mean value.

pStdDev Pointer to the output standard deviation value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMeanStdDevGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.194.1.4 NppStatus nppsMeanStdDev_64f (const Npp64f * *pSrc*, int *nLength*, Npp64f * *pMean*, Npp64f * *pStdDev*, Npp8u * *pDeviceBuffer*)

64-bit float vector mean and standard deviation method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMean Pointer to the output mean value.

pStdDev Pointer to the output standard deviation value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMeanStdDevGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.194.1.5 NppStatus nppsMeanStdDevGetBufferSize_16s32s_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMeanStdDev_16s32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.194.1.6 NppStatus nppsMeanStdDevGetBufferSize_16s_Sfs (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMeanStdDev_16s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.194.1.7 NppStatus nppsMeanStdDevGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMeanStdDev_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.194.1.8 NppStatus nppsMeanStdDevGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMeanStdDev_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

Returns:

NPP_SUCCESS

7.195 Minimum_Maximum

Functions

- [NppStatus nppsMinMaxGetBufferSize_8u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMax_8u.
- [NppStatus nppsMinMaxGetBufferSize_16s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMax_16s.
- [NppStatus nppsMinMaxGetBufferSize_16u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMax_16u.
- [NppStatus nppsMinMaxGetBufferSize_32s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMax_32s.
- [NppStatus nppsMinMaxGetBufferSize_32u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMax_32u.
- [NppStatus nppsMinMaxGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMax_32f.
- [NppStatus nppsMinMaxGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMax_64f.
- [NppStatus nppsMinMax_8u](#) (const [Npp8u](#) *pSrc, int nLength, [Npp8u](#) *pMin, [Npp8u](#) *pMax, [Npp8u](#) *pDeviceBuffer)
8-bit char vector min and max method
- [NppStatus nppsMinMax_16s](#) (const [Npp16s](#) *pSrc, int nLength, [Npp16s](#) *pMin, [Npp16s](#) *pMax, [Npp8u](#) *pDeviceBuffer)
16-bit signed short vector min and max method
- [NppStatus nppsMinMax_16u](#) (const [Npp16u](#) *pSrc, int nLength, [Npp16u](#) *pMin, [Npp16u](#) *pMax, [Npp8u](#) *pDeviceBuffer)
16-bit unsigned short vector min and max method
- [NppStatus nppsMinMax_32u](#) (const [Npp32u](#) *pSrc, int nLength, [Npp32u](#) *pMin, [Npp32u](#) *pMax, [Npp8u](#) *pDeviceBuffer)
32-bit unsigned int vector min and max method
- [NppStatus nppsMinMax_32s](#) (const [Npp32s](#) *pSrc, int nLength, [Npp32s](#) *pMin, [Npp32s](#) *pMax, [Npp8u](#) *pDeviceBuffer)
32-bit signed int vector min and max method
- [NppStatus nppsMinMax_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pMin, [Npp32f](#) *pMax, [Npp8u](#) *pDeviceBuffer)
32-bit float vector min and max method
- [NppStatus nppsMinMax_64f](#) (const [Npp64f](#) *pSrc, int nLength, [Npp64f](#) *pMin, [Npp64f](#) *pMax, [Npp8u](#) *pDeviceBuffer)

64-bit double vector min and max method

- [NppStatus nppsMinMaxIdxGetBufferSize_8u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMaxIdx_8u.
- [NppStatus nppsMinMaxIdxGetBufferSize_16s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMaxIdx_16s.
- [NppStatus nppsMinMaxIdxGetBufferSize_16u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMaxIdx_16u.
- [NppStatus nppsMinMaxIdxGetBufferSize_32s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMaxIdx_32s.
- [NppStatus nppsMinMaxIdxGetBufferSize_32u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMaxIdx_32u.
- [NppStatus nppsMinMaxIdxGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMaxIdx_32f.
- [NppStatus nppsMinMaxIdxGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMinMaxIdx_64f.
- [NppStatus nppsMinMaxIdx_8u](#) (const [Npp8u](#) *pSrc, int nLength, [Npp8u](#) *pMin, int *pMinIdx, [Npp8u](#) *pMax, int *pMaxIdx, [Npp8u](#) *pDeviceBuffer)
8-bit char vector min and max with indices method
- [NppStatus nppsMinMaxIdx_16s](#) (const [Npp16s](#) *pSrc, int nLength, [Npp16s](#) *pMin, int *pMinIdx, [Npp16s](#) *pMax, int *pMaxIdx, [Npp8u](#) *pDeviceBuffer)
16-bit signed short vector min and max with indices method
- [NppStatus nppsMinMaxIdx_16u](#) (const [Npp16u](#) *pSrc, int nLength, [Npp16u](#) *pMin, int *pMinIdx, [Npp16u](#) *pMax, int *pMaxIdx, [Npp8u](#) *pDeviceBuffer)
16-bit unsigned short vector min and max with indices method
- [NppStatus nppsMinMaxIdx_32s](#) (const [Npp32s](#) *pSrc, int nLength, [Npp32s](#) *pMin, int *pMinIdx, [Npp32s](#) *pMax, int *pMaxIdx, [Npp8u](#) *pDeviceBuffer)
32-bit signed short vector min and max with indices method
- [NppStatus nppsMinMaxIdx_32u](#) (const [Npp32u](#) *pSrc, int nLength, [Npp32u](#) *pMin, int *pMinIdx, [Npp32u](#) *pMax, int *pMaxIdx, [Npp8u](#) *pDeviceBuffer)
32-bit unsigned short vector min and max with indices method
- [NppStatus nppsMinMaxIdx_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pMin, int *pMinIdx, [Npp32f](#) *pMax, int *pMaxIdx, [Npp8u](#) *pDeviceBuffer)
32-bit float vector min and max with indices method
- [NppStatus nppsMinMaxIdx_64f](#) (const [Npp64f](#) *pSrc, int nLength, [Npp64f](#) *pMin, int *pMinIdx, [Npp64f](#) *pMax, int *pMaxIdx, [Npp8u](#) *pDeviceBuffer)
64-bit float vector min and max with indices method

7.195.1 Function Documentation

7.195.1.1 `NppStatus nppsMinMax_16s (const Npp16s * pSrc, int nLength, Npp16s * pMin, Npp16s * pMax, Npp8u * pDeviceBuffer)`

16-bit signed short vector min and max method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMax Pointer to the max output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.2 `NppStatus nppsMinMax_16u (const Npp16u * pSrc, int nLength, Npp16u * pMin, Npp16u * pMax, Npp8u * pDeviceBuffer)`

16-bit unsigned short vector min and max method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMax Pointer to the max output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxGetBufferSize_16u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.3 `NppStatus nppsMinMax_32f (const Npp32f * pSrc, int nLength, Npp32f * pMin, Npp32f * pMax, Npp8u * pDeviceBuffer)`

32-bit float vector min and max method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMax Pointer to the max output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.4 NppStatus nppsMinMax_32s (const Npp32s * pSrc, int nLength, Npp32s * pMin, Npp32s * pMax, Npp8u * pDeviceBuffer)

32-bit signed int vector min and max method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMax Pointer to the max output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.5 NppStatus nppsMinMax_32u (const Npp32u * pSrc, int nLength, Npp32u * pMin, Npp32u * pMax, Npp8u * pDeviceBuffer)

32-bit unsigned int vector min and max method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMax Pointer to the max output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxGetBufferSize_32u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.6 NppStatus nppsMinMax_64f (const Npp64f * pSrc, int nLength, Npp64f * pMin, Npp64f * pMax, Npp8u * pDeviceBuffer)

64-bit double vector min and max method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMax Pointer to the max output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.7 `NppStatus nppsMinMax_8u (const Npp8u * pSrc, int nLength, Npp8u * pMin, Npp8u * pMax, Npp8u * pDeviceBuffer)`

8-bit char vector min and max method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMax Pointer to the max output result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxGetBufferSize_8u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.8 `NppStatus nppsMinMaxGetBufferSize_16s (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsMinMax_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.9 `NppStatus nppsMinMaxGetBufferSize_16u (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsMinMax_16u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.10 NppStatus nppsMinMaxGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for *nppsMinMax_32f*.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.11 NppStatus nppsMinMaxGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for *nppsMinMax_32s*.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.12 NppStatus nppsMinMaxGetBufferSize_32u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for *nppsMinMax_32u*.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.13 NppStatus nppsMinMaxGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMinMax_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.14 NppStatus nppsMinMaxGetBufferSize_8u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMinMax_8u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.15 NppStatus nppsMinMaxIndx_16s (const Npp16s * *pSrc*, int *nLength*, Npp16s * *pMin*, int * *pMinIndx*, Npp16s * *pMax*, int * *pMaxIndx*, Npp8u * *pDeviceBuffer*)

16-bit signed short vector min and max with indices method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMinIndx Pointer to the index of the first min value.

pMax Pointer to the max output result.

pMaxIndx Pointer to the index of the first max value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxIndxGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.16 **NppStatus nppsMinMaxIndx_16u** (const Npp16u * *pSrc*, int *nLength*, Npp16u * *pMin*, int * *pMinIndx*, Npp16u * *pMax*, int * *pMaxIndx*, Npp8u * *pDeviceBuffer*)

16-bit unsigned short vector min and max with indices method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMinIndx Pointer to the index of the first min value.

pMax Pointer to the max output result.

pMaxIndx Pointer to the index of the first max value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxIndxGetBufferSize_16u](#) to determine the minimum number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.17 **NppStatus nppsMinMaxIndx_32f** (const Npp32f * *pSrc*, int *nLength*, Npp32f * *pMin*, int * *pMinIndx*, Npp32f * *pMax*, int * *pMaxIndx*, Npp8u * *pDeviceBuffer*)

32-bit float vector min and max with indices method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMinIndx Pointer to the index of the first min value.

pMax Pointer to the max output result.

pMaxIndx Pointer to the index of the first max value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxIndxGetBufferSize_32f](#) to determine the minimum number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.18 **NppStatus nppsMinMaxIndx_32s** (const Npp32s * *pSrc*, int *nLength*, Npp32s * *pMin*, int * *pMinIndx*, Npp32s * *pMax*, int * *pMaxIndx*, Npp8u * *pDeviceBuffer*)

32-bit signed short vector min and max with indices method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMinIndx Pointer to the index of the first min value.

pMax Pointer to the max output result.

pMaxIndx Pointer to the index of the first max value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxIndxGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.19 `NppStatus nppsMinMaxIndx_32u (const Npp32u * pSrc, int nLength, Npp32u * pMin, int * pMinIndx, Npp32u * pMax, int * pMaxIndx, Npp8u * pDeviceBuffer)`

32-bit unsigned short vector min and max with indices method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMinIndx Pointer to the index of the first min value.

pMax Pointer to the max output result.

pMaxIndx Pointer to the index of the first max value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxIndxGetBufferSize_32u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.20 `NppStatus nppsMinMaxIndx_64f (const Npp64f * pSrc, int nLength, Npp64f * pMin, int * pMinIndx, Npp64f * pMax, int * pMaxIndx, Npp8u * pDeviceBuffer)`

64-bit float vector min and max with indices method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMinIndx Pointer to the index of the first min value.

pMax Pointer to the max output result.

pMaxIndx Pointer to the index of the first max value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxIndxGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.21 NppStatus nppsMinMaxIndx_8u (const Npp8u * *pSrc*, int *nLength*, Npp8u * *pMin*, int * *pMinIndx*, Npp8u * *pMax*, int * *pMaxIndx*, Npp8u * *pDeviceBuffer*)

8-bit char vector min and max with indices method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pMin Pointer to the min output result.

pMinIndx Pointer to the index of the first min value.

pMax Pointer to the max output result.

pMaxIndx Pointer to the index of the first max value.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMinMaxIndxGetBufferSize_8u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.195.1.22 NppStatus nppsMinMaxIndxGetBufferSize_16s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMinMaxIndx_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.23 NppStatus nppsMinMaxIndxGetBufferSize_16u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMinMaxIndx_16u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.24 NppStatus nppsMinMaxIdxGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMinMaxIdx_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.25 NppStatus nppsMinMaxIdxGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMinMaxIdx_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.26 NppStatus nppsMinMaxIdxGetBufferSize_32u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMinMaxIdx_32u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.27 NppStatus nppsMinMaxIdxGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMinMaxIdx_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.195.1.28 NppStatus nppsMinMaxIndxGetBufferSize_8u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMinMaxIndx_8u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.196 Infinity Norm

Functions

- [NppStatus nppsNormInfGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_Inf_32f.
- [NppStatus nppsNorm_Inf_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float vector C norm method
- [NppStatus nppsNormInfGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_Inf_64f.
- [NppStatus nppsNorm_Inf_64f](#) (const [Npp64f](#) *pSrc, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float vector C norm method
- [NppStatus nppsNormInfGetBufferSize_16s32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_Inf_16s32f.
- [NppStatus nppsNorm_Inf_16s32f](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer vector C norm method, return value is 32-bit float.
- [NppStatus nppsNormInfGetBufferSize_32fc32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_Inf_32fc32f.
- [NppStatus nppsNorm_Inf_32fc32f](#) (const [Npp32fc](#) *pSrc, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float complex vector C norm method, return value is 32-bit float.
- [NppStatus nppsNormInfGetBufferSize_64fc64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_Inf_64fc64f.
- [NppStatus nppsNorm_Inf_64fc64f](#) (const [Npp64fc](#) *pSrc, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float complex vector C norm method, return value is 64-bit float.
- [NppStatus nppsNormInfGetBufferSize_16s32s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_Inf_16s32s_Sfs.
- [NppStatus nppsNorm_Inf_16s32s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32s](#) *pNorm, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer vector C norm method, return value is 32-bit signed integer.

7.196.1 Function Documentation

7.196.1.1 `NppStatus nppsNorm_Inf_16s32f (const Npp16s * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector C norm method, return value is 32-bit float.

Parameters:

pSrc Source Signal Pointer.

nLength Signal Length.

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormInfGetBufferSize_16s32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.196.1.2 `NppStatus nppsNorm_Inf_16s32s_Sfs (const Npp16s * pSrc, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector C norm method, return value is 32-bit signed integer.

Parameters:

pSrc Source Signal Pointer.

nLength Signal Length.

pNorm Pointer to the norm result.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormInfGetBufferSize_16s32s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.196.1.3 `NppStatus nppsNorm_Inf_32f (const Npp32f * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float vector C norm method

Parameters:

pSrc Source Signal Pointer.

nLength Signal Length.

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormInfGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.196.1.4 **NppStatus nppsNorm_Inf_32fc32f** (const Npp32fc * *pSrc*, int *nLength*, Npp32f * *pNorm*, Npp8u * *pDeviceBuffer*)

32-bit float complex vector C norm method, return value is 32-bit float.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormInfGetBufferSize_32fc32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.196.1.5 **NppStatus nppsNorm_Inf_64f** (const Npp64f * *pSrc*, int *nLength*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

64-bit float vector C norm method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormInfGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.196.1.6 **NppStatus nppsNorm_Inf_64fc64f** (const Npp64fc * *pSrc*, int *nLength*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

64-bit float complex vector C norm method, return value is 64-bit float.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormInfGetBufferSize_64fc64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.196.1.7 NppStatus nppsNormInfGetBufferSize_16s32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_Inf_16s32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.196.1.8 NppStatus nppsNormInfGetBufferSize_16s32s_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_Inf_16s32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.196.1.9 NppStatus nppsNormInfGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_Inf_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.196.1.10 NppStatus nppsNormInfGetBufferSize_32fc32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_Inf_32fc32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.196.1.11 NppStatus nppsNormInfGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_Inf_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.196.1.12 NppStatus nppsNormInfGetBufferSize_64fc64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_Inf_64fc64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.197 L1 Norm

Functions

- [NppStatus nppsNormL1GetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L1_32f.
- [NppStatus nppsNorm_L1_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float vector L1 norm method
- [NppStatus nppsNormL1GetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L1_64f.
- [NppStatus nppsNorm_L1_64f](#) (const [Npp64f](#) *pSrc, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float vector L1 norm method
- [NppStatus nppsNormL1GetBufferSize_16s32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L1_16s32f.
- [NppStatus nppsNorm_L1_16s32f](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer vector L1 norm method, return value is 32-bit float.
- [NppStatus nppsNormL1GetBufferSize_32fc64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L1_32fc64f.
- [NppStatus nppsNorm_L1_32fc64f](#) (const [Npp32fc](#) *pSrc, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float complex vector L1 norm method, return value is 64-bit float.
- [NppStatus nppsNormL1GetBufferSize_64fc64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L1_64fc64f.
- [NppStatus nppsNorm_L1_64fc64f](#) (const [Npp64fc](#) *pSrc, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float complex vector L1 norm method, return value is 64-bit float.
- [NppStatus nppsNormL1GetBufferSize_16s32s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L1_16s32s_Sfs.
- [NppStatus nppsNorm_L1_16s32s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32s](#) *pNorm, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer vector L1 norm method, return value is 32-bit signed integer.
- [NppStatus nppsNormL1GetBufferSize_16s64s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L1_16s64s_Sfs.
- [NppStatus nppsNorm_L1_16s64s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp64s](#) *pNorm, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)

16-bit signed short integer vector L1 norm method, return value is 64-bit signed integer.

7.197.1 Function Documentation

7.197.1.1 `NppStatus nppsNorm_L1_16s32f (const Npp16s * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector L1 norm method, return value is 32-bit float.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the L1 norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormL1GetBufferSize_16s32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.197.1.2 `NppStatus nppsNorm_L1_16s32s_Sfs (const Npp16s * pSrc, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector L1 norm method, return value is 32-bit signed integer.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormL1GetBufferSize_16s32s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.197.1.3 `NppStatus nppsNorm_L1_16s64s_Sfs (const Npp16s * pSrc, int nLength, Npp64s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector L1 norm method, return value is 64-bit signed integer.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormL1GetBufferSize_16s64s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.197.1.4 `NppStatus nppsNorm_L1_32f (const Npp32f * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float vector L1 norm method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormL1GetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.197.1.5 `NppStatus nppsNorm_L1_32fc64f (const Npp32fc * pSrc, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float complex vector L1 norm method, return value is 64-bit float.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormL1GetBufferSize_32fc64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.197.1.6 `NppStatus nppsNorm_L1_64f (const Npp64f * pSrc, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float vector L1 norm method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormL1GetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.197.1.7 `NppStatus nppsNorm_L1_64fc64f (const Npp64fc * pSrc, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float complex vector L1 norm method, return value is 64-bit float.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormL1GetBufferSize_64fc64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.197.1.8 `NppStatus nppsNormL1GetBufferSize_16s32f (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsNorm_L1_16s32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*.

Returns:

NPP_SUCCESS

7.197.1.9 `NppStatus nppsNormL1GetBufferSize_16s32s_Sfs (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsNorm_L1_16s32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*.

Returns:

NPP_SUCCESS

7.197.1.10 NppStatus nppsNormL1GetBufferSize_16s64s_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_L1_16s64s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.197.1.11 NppStatus nppsNormL1GetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_L1_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.197.1.12 NppStatus nppsNormL1GetBufferSize_32fc64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_L1_32fc64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.197.1.13 NppStatus nppsNormL1GetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_L1_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.197.1.14 NppStatus nppsNormL1GetBufferSize_64fc64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_L1_64fc64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.198 L2 Norm

Functions

- [NppStatus nppsNormL2GetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L2_32f.
- [NppStatus nppsNorm_L2_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float vector L2 norm method
- [NppStatus nppsNormL2GetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L2_64f.
- [NppStatus nppsNorm_L2_64f](#) (const [Npp64f](#) *pSrc, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float vector L2 norm method
- [NppStatus nppsNormL2GetBufferSize_16s32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L2_16s32f.
- [NppStatus nppsNorm_L2_16s32f](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer vector L2 norm method, return value is 32-bit float.
- [NppStatus nppsNormL2GetBufferSize_32fc64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L2_32fc64f.
- [NppStatus nppsNorm_L2_32fc64f](#) (const [Npp32fc](#) *pSrc, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float complex vector L2 norm method, return value is 64-bit float.
- [NppStatus nppsNormL2GetBufferSize_64fc64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L2_64fc64f.
- [NppStatus nppsNorm_L2_64fc64f](#) (const [Npp64fc](#) *pSrc, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float complex vector L2 norm method, return value is 64-bit float.
- [NppStatus nppsNormL2GetBufferSize_16s32s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L2_16s32s_Sfs.
- [NppStatus nppsNorm_L2_16s32s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32s](#) *pNorm, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer vector L2 norm method, return value is 32-bit signed integer.
- [NppStatus nppsNormL2SqrGetBufferSize_16s64s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNorm_L2Sqr_16s64s_Sfs.
- [NppStatus nppsNorm_L2Sqr_16s64s_Sfs](#) (const [Npp16s](#) *pSrc, int nLength, [Npp64s](#) *pNorm, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)

16-bit signed short integer vector L2 Square norm method, return value is 64-bit signed integer.

7.198.1 Function Documentation

7.198.1.1 `NppStatus nppsNorm_L2_16s32f (const Npp16s * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector L2 norm method, return value is 32-bit float.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormL2GetBufferSize_16s32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.198.1.2 `NppStatus nppsNorm_L2_16s32s_Sfs (const Npp16s * pSrc, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector L2 norm method, return value is 32-bit signed integer.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormL2GetBufferSize_16s32s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.198.1.3 `NppStatus nppsNorm_L2_32f (const Npp32f * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float vector L2 norm method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL2GetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.198.1.4 `NppStatus nppsNorm_L2_32fc64f (const Npp32fc * pSrc, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float complex vector L2 norm method, return value is 64-bit float.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL2GetBufferSize_32fc64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.198.1.5 `NppStatus nppsNorm_L2_64f (const Npp64f * pSrc, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float vector L2 norm method

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL2GetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.198.1.6 `NppStatus nppsNorm_L2_64fc64f (const Npp64fc * pSrc, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float complex vector L2 norm method, return value is 64-bit float.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormL2GetBufferSize_64fc64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.198.1.7 `NppStatus nppsNorm_L2Sqr_16s64s_Sfs (const Npp16s * pSrc, int nLength, Npp64s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector L2 Square norm method, return value is 64-bit signed integer.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormL2SqrGetBufferSize_16s64s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.198.1.8 `NppStatus nppsNormL2GetBufferSize_16s32f (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsNorm_L2_16s32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*.

Returns:

NPP_SUCCESS

7.198.1.9 `NppStatus nppsNormL2GetBufferSize_16s32s_Sfs (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsNorm_L2_16s32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*.

Returns:

NPP_SUCCESS

7.198.1.10 NppStatus nppsNormL2GetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_L2_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.198.1.11 NppStatus nppsNormL2GetBufferSize_32fc64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_L2_32fc64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.198.1.12 NppStatus nppsNormL2GetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_L2_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.198.1.13 NppStatus nppsNormL2GetBufferSize_64fc64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_L2_64fc64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.198.1.14 NppStatus nppsNormL2SqrGetBufferSize_16s64s_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm_L2Sqr_16s64s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.199 Infinity Norm Diff

Functions

- [NppStatus nppsNormDiffInfGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_Inf_32f.
- [NppStatus nppsNormDiff_Inf_32f](#) (const [Npp32f](#) *pSrc1, const [Npp32f](#) *pSrc2, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float C norm method on two vectors' difference
- [NppStatus nppsNormDiffInfGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_Inf_64f.
- [NppStatus nppsNormDiff_Inf_64f](#) (const [Npp64f](#) *pSrc1, const [Npp64f](#) *pSrc2, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float C norm method on two vectors' difference
- [NppStatus nppsNormDiffInfGetBufferSize_16s32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_Inf_16s32f.
- [NppStatus nppsNormDiff_Inf_16s32f](#) (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit float.
- [NppStatus nppsNormDiffInfGetBufferSize_32fc32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_Inf_32fc32f.
- [NppStatus nppsNormDiff_Inf_32fc32f](#) (const [Npp32fc](#) *pSrc1, const [Npp32fc](#) *pSrc2, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float complex C norm method on two vectors' difference, return value is 32-bit float.
- [NppStatus nppsNormDiffInfGetBufferSize_64fc64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_Inf_64fc64f.
- [NppStatus nppsNormDiff_Inf_64fc64f](#) (const [Npp64fc](#) *pSrc1, const [Npp64fc](#) *pSrc2, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float complex C norm method on two vectors' difference, return value is 64-bit float.
- [NppStatus nppsNormDiffInfGetBufferSize_16s32s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_Inf_16s32s_Sfs.
- [NppStatus nppsNormDiff_Inf_16s32s_Sfs](#) (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, int nLength, [Npp32s](#) *pNorm, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit signed integer.

7.199.1 Function Documentation

7.199.1.1 `NppStatus nppsNormDiff_Inf_16s32f (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit float.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffInfGetBufferSize_16s32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.199.1.2 `NppStatus nppsNormDiff_Inf_16s32s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit signed integer.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffInfGetBufferSize_16s32s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.199.1.3 `NppStatus nppsNormDiff_Inf_32f (const Npp32f * pSrc1, const Npp32f * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float C norm method on two vectors' difference

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffInfGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.199.1.4 NppStatus nppsNormDiff_Inf_32fc32f (const Npp32fc * pSrc1, const Npp32fc * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)

32-bit float complex C norm method on two vectors' difference, return value is 32-bit float.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffInfGetBufferSize_32fc32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.199.1.5 NppStatus nppsNormDiff_Inf_64f (const Npp64f * pSrc1, const Npp64f * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)

64-bit float C norm method on two vectors' difference

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffInfGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.199.1.6 **NppStatus nppsNormDiff_Inf_64fc64f** (const Npp64fc * *pSrc1*, const Npp64fc * *pSrc2*, int *nLength*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

64-bit float complex C norm method on two vectors' difference, return value is 64-bit float.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffInfGetBufferSize_64fc64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.199.1.7 **NppStatus nppsNormDiffInfGetBufferSize_16s32f** (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_Inf_16s32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.199.1.8 **NppStatus nppsNormDiffInfGetBufferSize_16s32s_Sfs** (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_Inf_16s32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.199.1.9 **NppStatus nppsNormDiffInfGetBufferSize_32f** (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_Inf_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.199.1.10 NppStatus nppsNormDiffInfGetBufferSize_32fc32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for `nppsNormDiff_Inf_32fc32f`.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.199.1.11 NppStatus nppsNormDiffInfGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for `nppsNormDiff_Inf_64f`.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.199.1.12 NppStatus nppsNormDiffInfGetBufferSize_64fc64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for `nppsNormDiff_Inf_64fc64f`.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.200 L1 Norm Diff

Functions

- [NppStatus nppsNormDiffL1GetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L1_32f.
- [NppStatus nppsNormDiff_L1_32f](#) (const [Npp32f](#) *pSrc1, const [Npp32f](#) *pSrc2, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float L1 norm method on two vectors' difference
- [NppStatus nppsNormDiffL1GetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L1_64f.
- [NppStatus nppsNormDiff_L1_64f](#) (const [Npp64f](#) *pSrc1, const [Npp64f](#) *pSrc2, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float L1 norm method on two vectors' difference
- [NppStatus nppsNormDiffL1GetBufferSize_16s32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L1_16s32f.
- [NppStatus nppsNormDiff_L1_16s32f](#) (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit float.
- [NppStatus nppsNormDiffL1GetBufferSize_32fc64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L1_32fc64f.
- [NppStatus nppsNormDiff_L1_32fc64f](#) (const [Npp32fc](#) *pSrc1, const [Npp32fc](#) *pSrc2, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.
- [NppStatus nppsNormDiffL1GetBufferSize_64fc64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L1_64fc64f.
- [NppStatus nppsNormDiff_L1_64fc64f](#) (const [Npp64fc](#) *pSrc1, const [Npp64fc](#) *pSrc2, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.
- [NppStatus nppsNormDiffL1GetBufferSize_16s32s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L1_16s32s_Sfs.
- [NppStatus nppsNormDiff_L1_16s32s_Sfs](#) (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, int nLength, [Npp32s](#) *pNorm, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit signed integer.
- [NppStatus nppsNormDiffL1GetBufferSize_16s64s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L1_16s64s_Sfs.
- [NppStatus nppsNormDiff_L1_16s64s_Sfs](#) (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, int nLength, [Npp64s](#) *pNorm, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)

16-bit signed short integer L1 norm method on two vectors' difference, return value is 64-bit signed integer.

7.200.1 Function Documentation

7.200.1.1 `NppStatus nppsNormDiff_L1_16s32f (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit float.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the L1 norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL1GetBufferSize_16s32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.200.1.2 `NppStatus nppsNormDiff_L1_16s32s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit signed integer.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL1GetBufferSize_16s32s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.200.1.3 `NppStatus nppsNormDiff_L1_16s64s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp64s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer L1 norm method on two vectors' difference, return value is 64-bit signed integer.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL1GetBufferSize_16s64s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.200.1.4 NppStatus nppsNormDiff_L1_32f (const Npp32f * pSrc1, const Npp32f * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)

32-bit float L1 norm method on two vectors' difference

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL1GetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.200.1.5 NppStatus nppsNormDiff_L1_32fc64f (const Npp32fc * pSrc1, const Npp32fc * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)

32-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL1GetBufferSize_32fc64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.200.1.6 NppStatus nppsNormDiff_L1_64f (const Npp64f * *pSrc1*, const Npp64f * *pSrc2*, int *nLength*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

64-bit float L1 norm method on two vectors' difference

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL1GetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.200.1.7 NppStatus nppsNormDiff_L1_64fc64f (const Npp64fc * *pSrc1*, const Npp64fc * *pSrc2*, int *nLength*, Npp64f * *pNorm*, Npp8u * *pDeviceBuffer*)

64-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL1GetBufferSize_64fc64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.200.1.8 NppStatus nppsNormDiffL1GetBufferSize_16s32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L1_16s32f.

Parameters:

nLength Signal Length.

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.200.1.9 NppStatus nppsNormDiffL1GetBufferSize_16s32s_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L1_16s32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.200.1.10 NppStatus nppsNormDiffL1GetBufferSize_16s64s_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L1_16s64s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.200.1.11 NppStatus nppsNormDiffL1GetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L1_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.200.1.12 NppStatus nppsNormDiffL1GetBufferSize_32fc64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L1_32fc64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.200.1.13 NppStatus nppsNormDiffL1GetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L1_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.200.1.14 NppStatus nppsNormDiffL1GetBufferSize_64fc64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L1_64fc64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.201 L2 Norm Diff

Functions

- [NppStatus nppsNormDiffL2GetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L2_32f.
- [NppStatus nppsNormDiff_L2_32f](#) (const [Npp32f](#) *pSrc1, const [Npp32f](#) *pSrc2, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float L2 norm method on two vectors' difference
- [NppStatus nppsNormDiffL2GetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L2_64f.
- [NppStatus nppsNormDiff_L2_64f](#) (const [Npp64f](#) *pSrc1, const [Npp64f](#) *pSrc2, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float L2 norm method on two vectors' difference
- [NppStatus nppsNormDiffL2GetBufferSize_16s32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L2_16s32f.
- [NppStatus nppsNormDiff_L2_16s32f](#) (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, int nLength, [Npp32f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit float.
- [NppStatus nppsNormDiffL2GetBufferSize_32fc64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L2_32fc64f.
- [NppStatus nppsNormDiff_L2_32fc64f](#) (const [Npp32fc](#) *pSrc1, const [Npp32fc](#) *pSrc2, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
32-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.
- [NppStatus nppsNormDiffL2GetBufferSize_64fc64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L2_64fc64f.
- [NppStatus nppsNormDiff_L2_64fc64f](#) (const [Npp64fc](#) *pSrc1, const [Npp64fc](#) *pSrc2, int nLength, [Npp64f](#) *pNorm, [Npp8u](#) *pDeviceBuffer)
64-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.
- [NppStatus nppsNormDiffL2GetBufferSize_16s32s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L2_16s32s_Sfs.
- [NppStatus nppsNormDiff_L2_16s32s_Sfs](#) (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, int nLength, [Npp32s](#) *pNorm, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit signed integer.
- [NppStatus nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsNormDiff_L2Sqr_16s64s_Sfs.
- [NppStatus nppsNormDiff_L2Sqr_16s64s_Sfs](#) (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, int nLength, [Npp64s](#) *pNorm, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)

16-bit signed short integer L2 Square norm method on two vectors' difference, return value is 64-bit signed integer.

7.201.1 Function Documentation

7.201.1.1 NppStatus nppsNormDiff_L2_16s32f (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)

16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit float.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL2GetBufferSize_16s32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.201.1.2 NppStatus nppsNormDiff_L2_16s32s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)

16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit signed integer.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL2GetBufferSize_16s32s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.201.1.3 NppStatus nppsNormDiff_L2_32f (const Npp32f * pSrc1, const Npp32f * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)

32-bit float L2 norm method on two vectors' difference

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL2GetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.201.1.4 `NppStatus nppsNormDiff_L2_32fc64f (const Npp32fc * pSrc1, const Npp32fc * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL2GetBufferSize_32fc64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.201.1.5 `NppStatus nppsNormDiff_L2_64f (const Npp64f * pSrc1, const Npp64f * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float L2 norm method on two vectors' difference

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL2GetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.201.1.6 `NppStatus nppsNormDiff_L2_64fc64f (const Npp64fc * pSrc1, const Npp64fc * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL2GetBufferSize_64fc64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.201.1.7 `NppStatus nppsNormDiff_L2Sqr_16s64s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp64s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer L2 Square norm method on two vectors' difference, return value is 64-bit signed integer.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pNorm Pointer to the norm result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.201.1.8 `NppStatus nppsNormDiffL2GetBufferSize_16s32f (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for `nppsNormDiff_L2_16s32f`.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.201.1.9 NppStatus nppsNormDiffL2GetBufferSize_16s32s_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L2_16s32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.201.1.10 NppStatus nppsNormDiffL2GetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L2_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.201.1.11 NppStatus nppsNormDiffL2GetBufferSize_32fc64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L2_32fc64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.201.1.12 NppStatus nppsNormDiffL2GetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L2_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.201.1.13 NppStatus nppsNormDiffL2GetBufferSize_64fc64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L2_64fc64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.201.1.14 NppStatus nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff_L2Sqr_16s64s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202 Dot Product

Functions

- [NppStatus nppsDotProdGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_32f.
- [NppStatus nppsDotProd_32f](#) (const [Npp32f](#) *pSrc1, const [Npp32f](#) *pSrc2, int nLength, [Npp32f](#) *pDp, [Npp8u](#) *pDeviceBuffer)
32-bit float dot product method, return value is 32-bit float.
- [NppStatus nppsDotProdGetBufferSize_32fc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_32fc.
- [NppStatus nppsDotProd_32fc](#) (const [Npp32fc](#) *pSrc1, const [Npp32fc](#) *pSrc2, int nLength, [Npp32fc](#) *pDp, [Npp8u](#) *pDeviceBuffer)
32-bit float complex dot product method, return value is 32-bit float complex.
- [NppStatus nppsDotProdGetBufferSize_32f32fc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_32f32fc.
- [NppStatus nppsDotProd_32f32fc](#) (const [Npp32f](#) *pSrc1, const [Npp32fc](#) *pSrc2, int nLength, [Npp32fc](#) *pDp, [Npp8u](#) *pDeviceBuffer)
32-bit float and 32-bit float complex dot product method, return value is 32-bit float complex.
- [NppStatus nppsDotProdGetBufferSize_32f64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_32f64f.
- [NppStatus nppsDotProd_32f64f](#) (const [Npp32f](#) *pSrc1, const [Npp32f](#) *pSrc2, int nLength, [Npp64f](#) *pDp, [Npp8u](#) *pDeviceBuffer)
32-bit float dot product method, return value is 64-bit float.
- [NppStatus nppsDotProdGetBufferSize_32fc64fc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_32fc64fc.
- [NppStatus nppsDotProd_32fc64fc](#) (const [Npp32fc](#) *pSrc1, const [Npp32fc](#) *pSrc2, int nLength, [Npp64fc](#) *pDp, [Npp8u](#) *pDeviceBuffer)
32-bit float complex dot product method, return value is 64-bit float complex.
- [NppStatus nppsDotProdGetBufferSize_32f32fc64fc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_32f32fc64fc.
- [NppStatus nppsDotProd_32f32fc64fc](#) (const [Npp32f](#) *pSrc1, const [Npp32fc](#) *pSrc2, int nLength, [Npp64fc](#) *pDp, [Npp8u](#) *pDeviceBuffer)
32-bit float and 32-bit float complex dot product method, return value is 64-bit float complex.
- [NppStatus nppsDotProdGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_64f.
- [NppStatus nppsDotProd_64f](#) (const [Npp64f](#) *pSrc1, const [Npp64f](#) *pSrc2, int nLength, [Npp64f](#) *pDp, [Npp8u](#) *pDeviceBuffer)

64-bit float dot product method, return value is 64-bit float.

- **NppStatus nppsDotProdGetBufferSize_64fc** (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_64fc.
- **NppStatus nppsDotProd_64fc** (const **Npp64fc** *pSrc1, const **Npp64fc** *pSrc2, int nLength, **Npp64fc** *pDp, **Npp8u** *pDeviceBuffer)
64-bit float complex dot product method, return value is 64-bit float complex.
- **NppStatus nppsDotProdGetBufferSize_64f64fc** (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_64f64fc.
- **NppStatus nppsDotProd_64f64fc** (const **Npp64f** *pSrc1, const **Npp64fc** *pSrc2, int nLength, **Npp64fc** *pDp, **Npp8u** *pDeviceBuffer)
64-bit float and 64-bit float complex dot product method, return value is 64-bit float complex.
- **NppStatus nppsDotProdGetBufferSize_16s64s** (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16s64s.
- **NppStatus nppsDotProd_16s64s** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, int nLength, **Npp64s** *pDp, **Npp8u** *pDeviceBuffer)
16-bit signed short integer dot product method, return value is 64-bit signed integer.
- **NppStatus nppsDotProdGetBufferSize_16sc64sc** (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16sc64sc.
- **NppStatus nppsDotProd_16sc64sc** (const **Npp16sc** *pSrc1, const **Npp16sc** *pSrc2, int nLength, **Npp64sc** *pDp, **Npp8u** *pDeviceBuffer)
16-bit signed short integer complex dot product method, return value is 64-bit signed integer complex.
- **NppStatus nppsDotProdGetBufferSize_16s16sc64sc** (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16s16sc64sc.
- **NppStatus nppsDotProd_16s16sc64sc** (const **Npp16s** *pSrc1, const **Npp16sc** *pSrc2, int nLength, **Npp64sc** *pDp, **Npp8u** *pDeviceBuffer)
16-bit signed short integer and 16-bit signed short integer short dot product method, return value is 64-bit signed integer complex.
- **NppStatus nppsDotProdGetBufferSize_16s32f** (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16s32f.
- **NppStatus nppsDotProd_16s32f** (const **Npp16s** *pSrc1, const **Npp16s** *pSrc2, int nLength, **Npp32f** *pDp, **Npp8u** *pDeviceBuffer)
16-bit signed short integer dot product method, return value is 32-bit float.
- **NppStatus nppsDotProdGetBufferSize_16sc32fc** (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16sc32fc.
- **NppStatus nppsDotProd_16sc32fc** (const **Npp16sc** *pSrc1, const **Npp16sc** *pSrc2, int nLength, **Npp32fc** *pDp, **Npp8u** *pDeviceBuffer)
16-bit signed short integer complex dot product method, return value is 32-bit float complex.

- [NppStatus nppsDotProdGetBufferSize_16s16sc32fc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16s16sc32fc.
- [NppStatus nppsDotProd_16s16sc32fc](#) (const [Npp16s](#) *pSrc1, const [Npp16sc](#) *pSrc2, int nLength, [Npp32fc](#) *pDp, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit float complex.
- [NppStatus nppsDotProdGetBufferSize_16s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16s_Sfs.
- [NppStatus nppsDotProd_16s_Sfs](#) (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, int nLength, [Npp16s](#) *pDp, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer dot product method, return value is 16-bit signed short integer.
- [NppStatus nppsDotProdGetBufferSize_16sc_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16sc_Sfs.
- [NppStatus nppsDotProd_16sc_Sfs](#) (const [Npp16sc](#) *pSrc1, const [Npp16sc](#) *pSrc2, int nLength, [Npp16sc](#) *pDp, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.
- [NppStatus nppsDotProdGetBufferSize_32s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_32s_Sfs.
- [NppStatus nppsDotProd_32s_Sfs](#) (const [Npp32s](#) *pSrc1, const [Npp32s](#) *pSrc2, int nLength, [Npp32s](#) *pDp, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
32-bit signed integer dot product method, return value is 32-bit signed integer.
- [NppStatus nppsDotProdGetBufferSize_32sc_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_32sc_Sfs.
- [NppStatus nppsDotProd_32sc_Sfs](#) (const [Npp32sc](#) *pSrc1, const [Npp32sc](#) *pSrc2, int nLength, [Npp32sc](#) *pDp, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
32-bit signed integer complex dot product method, return value is 32-bit signed integer complex.
- [NppStatus nppsDotProdGetBufferSize_16s32s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16s32s_Sfs.
- [NppStatus nppsDotProd_16s32s_Sfs](#) (const [Npp16s](#) *pSrc1, const [Npp16s](#) *pSrc2, int nLength, [Npp32s](#) *pDp, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer dot product method, return value is 32-bit signed integer.
- [NppStatus nppsDotProdGetBufferSize_16s16sc32sc_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16s16sc32sc_Sfs.
- [NppStatus nppsDotProd_16s16sc32sc_Sfs](#) (const [Npp16s](#) *pSrc1, const [Npp16sc](#) *pSrc2, int nLength, [Npp32sc](#) *pDp, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

- [NppStatus nppsDotProdGetBufferSize_16s32s32s_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16s32s32s_Sfs.
- [NppStatus nppsDotProd_16s32s32s_Sfs](#) (const [Npp16s](#) *pSrc1, const [Npp32s](#) *pSrc2, int nLength, [Npp32s](#) *pDp, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer and 32-bit signed integer dot product method, return value is 32-bit signed integer.
- [NppStatus nppsDotProdGetBufferSize_16s16sc_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16s16sc_Sfs.
- [NppStatus nppsDotProd_16s16sc_Sfs](#) (const [Npp16s](#) *pSrc1, const [Npp16sc](#) *pSrc2, int nLength, [Npp16sc](#) *pDp, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.
- [NppStatus nppsDotProdGetBufferSize_16sc32sc_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_16sc32sc_Sfs.
- [NppStatus nppsDotProd_16sc32sc_Sfs](#) (const [Npp16sc](#) *pSrc1, const [Npp16sc](#) *pSrc2, int nLength, [Npp32sc](#) *pDp, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.
- [NppStatus nppsDotProdGetBufferSize_32s32sc_Sfs](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsDotProd_32s32sc_Sfs.
- [NppStatus nppsDotProd_32s32sc_Sfs](#) (const [Npp32s](#) *pSrc1, const [Npp32sc](#) *pSrc2, int nLength, [Npp32sc](#) *pDp, int nScaleFactor, [Npp8u](#) *pDeviceBuffer)
32-bit signed short integer and 32-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

7.202.1 Function Documentation

7.202.1.1 [NppStatus nppsDotProd_16s16sc32fc](#) (const [Npp16s](#) *pSrc1, const [Npp16sc](#) *pSrc2, int nLength, [Npp32fc](#) *pDp, [Npp8u](#) *pDeviceBuffer)

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit float complex.

Parameters:

[pSrc1](#) [Source Signal Pointer](#).

[pSrc2](#) [Source Signal Pointer](#).

[nLength](#) [Signal Length](#).

[pDp](#) Pointer to the dot product result.

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16s16sc32fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.2 `NppStatus nppsDotProd_16s16sc32sc_Sfs (const Npp16s * pSrc1, const Npp16sc * pSrc2, int nLength, Npp32sc * pDp, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp [Pointer to the dot product result](#).

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16s16sc32sc_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.3 `NppStatus nppsDotProd_16s16sc64sc (const Npp16s * pSrc1, const Npp16sc * pSrc2, int nLength, Npp64sc * pDp, Npp8u * pDeviceBuffer)`

16-bit signed short integer and 16-bit signed short integer short dot product method, return value is 64-bit signed integer complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp [Pointer to the dot product result](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16s16sc64sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.4 NppStatus nppsDotProd_16s16sc_Sfs (const Npp16s * *pSrc1*, const Npp16sc * *pSrc2*, int *nLength*, Npp16sc * *pDp*, int *nScaleFactor*, Npp8u * *pDeviceBuffer*)

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp [Pointer to the dot product result](#).

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16s16sc_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.5 NppStatus nppsDotProd_16s32f (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, int *nLength*, Npp32f * *pDp*, Npp8u * *pDeviceBuffer*)

16-bit signed short integer dot product method, return value is 32-bit float.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp [Pointer to the dot product result](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16s32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.6 NppStatus nppsDotProd_16s32s32s_Sfs (const Npp16s * *pSrc1*, const Npp32s * *pSrc2*, int *nLength*, Npp32s * *pDp*, int *nScaleFactor*, Npp8u * *pDeviceBuffer*)

16-bit signed short integer and 32-bit signed integer dot product method, return value is 32-bit signed integer.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp Pointer to the dot product result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16s32s32s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.7 `NppStatus nppsDotProd_16s32s_Sfs (const Npp16s *pSrc1, const Npp16s *pSrc2, int nLength, Npp32s *pDp, int nScaleFactor, Npp8u *pDeviceBuffer)`

16-bit signed short integer dot product method, return value is 32-bit signed integer.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp Pointer to the dot product result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16s32s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.8 `NppStatus nppsDotProd_16s64s (const Npp16s *pSrc1, const Npp16s *pSrc2, int nLength, Npp64s *pDp, Npp8u *pDeviceBuffer)`

16-bit signed short integer dot product method, return value is 64-bit signed integer.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp Pointer to the dot product result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16s64s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.9 NppStatus nppsDotProd_16s_Sfs (const Npp16s * *pSrc1*, const Npp16s * *pSrc2*, int *nLength*, Npp16s * *pDp*, int *nScaleFactor*, Npp8u * *pDeviceBuffer*)

16-bit signed short integer dot product method, return value is 16-bit signed short integer.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp [Pointer to the dot product result](#).

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.10 NppStatus nppsDotProd_16sc32fc (const Npp16sc * *pSrc1*, const Npp16sc * *pSrc2*, int *nLength*, Npp32fc * *pDp*, Npp8u * *pDeviceBuffer*)

16-bit signed short integer complex dot product method, return value is 32-bit float complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp [Pointer to the dot product result](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16sc32fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.11 NppStatus nppsDotProd_16sc32sc_Sfs (const Npp16sc * *pSrc1*, const Npp16sc * *pSrc2*, int *nLength*, Npp32sc * *pDp*, int *nScaleFactor*, Npp8u * *pDeviceBuffer*)

16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp [Pointer to the dot product result](#).

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16sc32sc_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.12 NppStatus nppsDotProd_16sc64sc (const Npp16sc * pSrc1, const Npp16sc * pSrc2, int nLength, Npp64sc * pDp, Npp8u * pDeviceBuffer)

16-bit signed short integer complex dot product method, return value is 64-bit signed integer complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp Pointer to the dot product result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16sc64sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.13 NppStatus nppsDotProd_16sc_Sfs (const Npp16sc * pSrc1, const Npp16sc * pSrc2, int nLength, Npp16sc * pDp, int nScaleFactor, Npp8u * pDeviceBuffer)

16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp Pointer to the dot product result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_16sc_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.14 **NppStatus nppsDotProd_32f** (const Npp32f * *pSrc1*, const Npp32f * *pSrc2*, int *nLength*, Npp32f * *pDp*, Npp8u * *pDeviceBuffer*)

32-bit float dot product method, return value is 32-bit float.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp Pointer to the dot product result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.15 **NppStatus nppsDotProd_32f32fc** (const Npp32f * *pSrc1*, const Npp32fc * *pSrc2*, int *nLength*, Npp32fc * *pDp*, Npp8u * *pDeviceBuffer*)

32-bit float and 32-bit float complex dot product method, return value is 32-bit float complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp Pointer to the dot product result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_32f32fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.16 **NppStatus nppsDotProd_32f32fc64fc** (const Npp32f * *pSrc1*, const Npp32fc * *pSrc2*, int *nLength*, Npp64fc * *pDp*, Npp8u * *pDeviceBuffer*)

32-bit float and 32-bit float complex dot product method, return value is 64-bit float complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp Pointer to the dot product result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_32f32fc64fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.17 **NppStatus nppsDotProd_32f64f** (const Npp32f * *pSrc1*, const Npp32f * *pSrc2*, int *nLength*, Npp64f * *pDp*, Npp8u * *pDeviceBuffer*)

32-bit float dot product method, return value is 64-bit float.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp [Pointer to the dot product result](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_32f64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.18 **NppStatus nppsDotProd_32fc** (const Npp32fc * *pSrc1*, const Npp32fc * *pSrc2*, int *nLength*, Npp32fc * *pDp*, Npp8u * *pDeviceBuffer*)

32-bit float complex dot product method, return value is 32-bit float complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp [Pointer to the dot product result](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_32fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.19 **NppStatus nppsDotProd_32fc64fc** (const Npp32fc * *pSrc1*, const Npp32fc * *pSrc2*, int *nLength*, Npp64fc * *pDp*, Npp8u * *pDeviceBuffer*)

32-bit float complex dot product method, return value is 64-bit float complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp [Pointer to the dot product result](#).

pDeviceBuffer [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_32fc64fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.20 NppStatus nppsDotProd_32s32sc_Sfs (const Npp32s * *pSrc1*, const Npp32sc * *pSrc2*, int *nLength*, Npp32sc * *pDp*, int *nScaleFactor*, Npp8u * *pDeviceBuffer*)

32-bit signed short integer and 32-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDp Pointer to the dot product result.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_32s32sc_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.21 NppStatus nppsDotProd_32s_Sfs (const Npp32s * *pSrc1*, const Npp32s * *pSrc2*, int *nLength*, Npp32s * *pDp*, int *nScaleFactor*, Npp8u * *pDeviceBuffer*)

32-bit signed integer dot product method, return value is 32-bit signed integer.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDp Pointer to the dot product result.

nScaleFactor Integer Result Scaling.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_32s_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.22 NppStatus nppsDotProd_32sc_Sfs (const Npp32sc * *pSrc1*, const Npp32sc * *pSrc2*, int *nLength*, Npp32sc * *pDp*, int *nScaleFactor*, Npp8u * *pDeviceBuffer*)

32-bit signed integer complex dot product method, return value is 32-bit signed integer complex.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength [Signal Length](#).

pDp Pointer to the dot product result.

nScaleFactor [Integer Result Scaling](#).

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_32sc_Sfs](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.23 NppStatus nppsDotProd_64f (const Npp64f * pSrc1, const Npp64f * pSrc2, int nLength, Npp64f * pDp, Npp8u * pDeviceBuffer)

64-bit float dot product method, return value is 64-bit float.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp Pointer to the dot product result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.24 NppStatus nppsDotProd_64f64fc (const Npp64f * pSrc1, const Npp64fc * pSrc2, int nLength, Npp64fc * pDp, Npp8u * pDeviceBuffer)

64-bit float and 64-bit float complex dot product method, return value is 64-bit float complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp Pointer to the dot product result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_64f64fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.25 NppStatus nppsDotProd_64fc (const Npp64fc * *pSrc1*, const Npp64fc * *pSrc2*, int *nLength*, Npp64fc * *pDp*, Npp8u * *pDeviceBuffer*)

64-bit float complex dot product method, return value is 64-bit float complex.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDp Pointer to the dot product result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsDotProdGetBufferSize_64fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.202.1.26 NppStatus nppsDotProdGetBufferSize_16s16sc32fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_16s16sc32fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.27 NppStatus nppsDotProdGetBufferSize_16s16sc32sc_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_16s16sc32sc_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.28 NppStatus nppsDotProdGetBufferSize_16s16sc64sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_16s16sc64sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.29 NppStatus nppsDotProdGetBufferSize_16s16sc_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for *nppsDotProd_16s16sc_Sfs*.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.30 NppStatus nppsDotProdGetBufferSize_16s32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for *nppsDotProd_16s32f*.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.31 NppStatus nppsDotProdGetBufferSize_16s32s32s_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for *nppsDotProd_16s32s32s_Sfs*.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.32 NppStatus nppsDotProdGetBufferSize_16s32s_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_16s32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.33 NppStatus nppsDotProdGetBufferSize_16s64s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_16s64s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.34 NppStatus nppsDotProdGetBufferSize_16s_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_16s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.35 NppStatus nppsDotProdGetBufferSize_16sc32fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_16sc32fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.36 NppStatus nppsDotProdGetBufferSize_16sc32sc_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_16sc32sc_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.37 NppStatus nppsDotProdGetBufferSize_16sc64sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_16sc64sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.38 NppStatus nppsDotProdGetBufferSize_16sc_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_16sc_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.39 NppStatus nppsDotProdGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.40 NppStatus nppsDotProdGetBufferSize_32f32fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_32f32fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.41 NppStatus nppsDotProdGetBufferSize_32f32fc64fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_32f32fc64fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.42 NppStatus nppsDotProdGetBufferSize_32f64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_32f64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.43 NppStatus nppsDotProdGetBufferSize_32fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_32fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.44 NppStatus nppsDotProdGetBufferSize_32fc64fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_32fc64fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.45 NppStatus nppsDotProdGetBufferSize_32s32sc_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_32s32sc_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.46 NppStatus nppsDotProdGetBufferSize_32s_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_32s_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.47 NppStatus nppsDotProdGetBufferSize_32sc_Sfs (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_32sc_Sfs.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.48 NppStatus nppsDotProdGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.49 NppStatus nppsDotProdGetBufferSize_64f64fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_64f64fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.202.1.50 NppStatus nppsDotProdGetBufferSize_64fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd_64fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.203 Count In Range

Functions

- [NppStatus nppsCountInRangeGetBufferSize_32s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsCountInRange_32s.
- [NppStatus nppsCountInRange_32s](#) (const [Npp32s](#) *pSrc, int nLength, int *pCounts, [Npp32s](#) nLowerBound, [Npp32s](#) nUpperBound, [Npp8u](#) *pDeviceBuffer)
Computes the number of elements whose values fall into the specified range on a 32-bit signed integer array.

7.203.1 Function Documentation

7.203.1.1 [NppStatus nppsCountInRange_32s](#) (const [Npp32s](#) *pSrc, int nLength, int *pCounts, [Npp32s](#) nLowerBound, [Npp32s](#) nUpperBound, [Npp8u](#) *pDeviceBuffer)

Computes the number of elements whose values fall into the specified range on a 32-bit signed integer array.

Parameters:

pSrc [Source Signal Pointer](#).

nLength [Signal Length](#).

pCounts Pointer to the number of elements.

nLowerBound Lower bound of the specified range.

nUpperBound Upper bound of the specified range.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsCountInRangeGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.203.1.2 [NppStatus nppsCountInRangeGetBufferSize_32s](#) (int nLength, int *hpBufferSize)

Device-buffer size (in bytes) for nppsCountInRange_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*.

Returns:

NPP_SUCCESS

7.204 Count Zero Crossings

Functions

- [NppStatus nppsZeroCrossingGetBufferSize_16s32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsZeroCrossing_16s32f.
- [NppStatus nppsZeroCrossing_16s32f](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32f](#) *pValZC, [NppsZC-Type](#) tZCType, [Npp8u](#) *pDeviceBuffer)
16-bit signed short integer zero crossing method, return value is 32-bit floating point.
- [NppStatus nppsZeroCrossingGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsZeroCrossing_32f.
- [NppStatus nppsZeroCrossing_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pValZC, [NppsZC-Type](#) tZCType, [Npp8u](#) *pDeviceBuffer)
32-bit floating-point zero crossing method, return value is 32-bit floating point.

7.204.1 Function Documentation

7.204.1.1 [NppStatus nppsZeroCrossing_16s32f](#) (const [Npp16s](#) *pSrc, int nLength, [Npp32f](#) *pValZC, [NppsZCType](#) tZCType, [Npp8u](#) *pDeviceBuffer)

16-bit signed short integer zero crossing method, return value is 32-bit floating point.

Parameters:

[pSrc](#) [Source Signal Pointer](#).

[nLength](#) [Signal Length](#).

[pValZC](#) Pointer to the output result.

[tZCType](#) Type of the zero crossing measure: [nppZCR](#), [nppZCXor](#) or [nppZCC](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsZeroCrossingGetBufferSize_16s32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.204.1.2 [NppStatus nppsZeroCrossing_32f](#) (const [Npp32f](#) *pSrc, int nLength, [Npp32f](#) *pValZC, [NppsZCType](#) tZCType, [Npp8u](#) *pDeviceBuffer)

32-bit floating-point zero crossing method, return value is 32-bit floating point.

Parameters:

[pSrc](#) [Source Signal Pointer](#).

[nLength](#) [Signal Length](#).

[pValZC](#) Pointer to the output result.

tZCType Type of the zero crossing measure: nppZCR, nppZCXor or nppZCC.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsZeroCrossingGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.204.1.3 NppStatus nppsZeroCrossingGetBufferSize_16s32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsZeroCrossing_16s32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*.

Returns:

NPP_SUCCESS

7.204.1.4 NppStatus nppsZeroCrossingGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsZeroCrossing_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: hpBufferSize is a *host pointer*.

Returns:

NPP_SUCCESS

7.205 MaximumError

Primitives for computing the maximum error between two signals.

Functions

- `NppStatus nppsMaximumError_8u` (const `Npp8u` *pSrc1, const `Npp8u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
8-bit unsigned char maximum method.
- `NppStatus nppsMaximumError_8s` (const `Npp8s` *pSrc1, const `Npp8s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
8-bit signed char maximum method.
- `NppStatus nppsMaximumError_16u` (const `Npp16u` *pSrc1, const `Npp16u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit unsigned short integer maximum method.
- `NppStatus nppsMaximumError_16s` (const `Npp16s` *pSrc1, const `Npp16s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit signed short integer maximum method.
- `NppStatus nppsMaximumError_16sc` (const `Npp16sc` *pSrc1, const `Npp16sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit unsigned short complex integer maximum method.
- `NppStatus nppsMaximumError_32u` (const `Npp32u` *pSrc1, const `Npp32u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit unsigned short integer maximum method.
- `NppStatus nppsMaximumError_32s` (const `Npp32s` *pSrc1, const `Npp32s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit signed short integer maximum method.
- `NppStatus nppsMaximumError_32sc` (const `Npp32sc` *pSrc1, const `Npp32sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit unsigned short complex integer maximum method.
- `NppStatus nppsMaximumError_64s` (const `Npp64s` *pSrc1, const `Npp64s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
64-bit signed short integer maximum method.
- `NppStatus nppsMaximumError_64sc` (const `Npp64sc` *pSrc1, const `Npp64sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
64-bit unsigned short complex integer maximum method.
- `NppStatus nppsMaximumError_32f` (const `Npp32f` *pSrc1, const `Npp32f` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit floating point maximum method.

- [NppStatus nppsMaximumError_32fc](#) (const [Npp32fc](#) *pSrc1, const [Npp32fc](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
32-bit floating point complex maximum method.
- [NppStatus nppsMaximumError_64f](#) (const [Npp64f](#) *pSrc1, const [Npp64f](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
64-bit floating point maximum method.
- [NppStatus nppsMaximumError_64fc](#) (const [Npp64fc](#) *pSrc1, const [Npp64fc](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
64-bit floating point complex maximum method.
- [NppStatus nppsMaximumErrorGetBufferSize_8u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_8u.
- [NppStatus nppsMaximumErrorGetBufferSize_8s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_8s.
- [NppStatus nppsMaximumErrorGetBufferSize_16u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_16u.
- [NppStatus nppsMaximumErrorGetBufferSize_16s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_16s.
- [NppStatus nppsMaximumErrorGetBufferSize_16sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_16sc.
- [NppStatus nppsMaximumErrorGetBufferSize_32u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_32u.
- [NppStatus nppsMaximumErrorGetBufferSize_32s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_32s.
- [NppStatus nppsMaximumErrorGetBufferSize_32sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_32sc.
- [NppStatus nppsMaximumErrorGetBufferSize_64s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_64s.
- [NppStatus nppsMaximumErrorGetBufferSize_64sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_64sc.
- [NppStatus nppsMaximumErrorGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_32f.
- [NppStatus nppsMaximumErrorGetBufferSize_32fc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_32fc.
- [NppStatus nppsMaximumErrorGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumError_64f.

- [NppStatus nppsMaximumErrorGetBufferSize_64fc](#) (int nLength, int *hpBufferSize)

Device-buffer size (in bytes) for nppsMaximumError_64fc.

7.205.1 Detailed Description

Primitives for computing the maximum error between two signals.

Given two signals *pSrc1* and *pSrc2* both with length *N*, the maximum error is defined as the largest absolute difference between the corresponding elements of two signals.

If the signal is in complex format, the absolute value of the complex number is used.

7.205.2 Function Documentation

7.205.2.1 NppStatus nppsMaximumError_16s (const Npp16s *pSrc1, const Npp16s *pSrc2, int nLength, Npp64f *pDst, Npp8u *pDeviceBuffer)

16-bit signed short integer maximum method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.2 NppStatus nppsMaximumError_16sc (const Npp16sc *pSrc1, const Npp16sc *pSrc2, int nLength, Npp64f *pDst, Npp8u *pDeviceBuffer)

16-bit unsigned short complex integer maximum method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_16sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.3 **NppStatus nppsMaximumError_16u** (const Npp16u * *pSrc1*, const Npp16u * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

16-bit unsigned short integer maximum method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_16u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.4 **NppStatus nppsMaximumError_32f** (const Npp32f * *pSrc1*, const Npp32f * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit floating point maximum method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.5 **NppStatus nppsMaximumError_32fc** (const Npp32fc * *pSrc1*, const Npp32fc * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit floating point complex maximum method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_32fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.6 NppStatus nppsMaximumError_32s (const Npp32s * *pSrc1*, const Npp32s * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit signed short integer maximum method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.7 NppStatus nppsMaximumError_32sc (const Npp32sc * *pSrc1*, const Npp32sc * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit unsigned short complex integer maximum method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_32sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.8 NppStatus nppsMaximumError_32u (const Npp32u * *pSrc1*, const Npp32u * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit unsigned short integer maximum method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_32u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.9 **NppStatus nppsMaximumError_64f** (const Npp64f * *pSrc1*, const Npp64f * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

64-bit floating point maximum method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.10 **NppStatus nppsMaximumError_64fc** (const Npp64fc * *pSrc1*, const Npp64fc * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

64-bit floating point complex maximum method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_64fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.11 **NppStatus nppsMaximumError_64s** (const Npp64s * *pSrc1*, const Npp64s * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

64-bit signed short integer maximum method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_64s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.12 **NppStatus nppsMaximumError_64sc** (const Npp64sc * *pSrc1*, const Npp64sc * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

64-bit unsigned short complex integer maximum method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_64sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.13 **NppStatus nppsMaximumError_8s** (const Npp8s * *pSrc1*, const Npp8s * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

8-bit signed char maximum method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_8s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.14 **NppStatus nppsMaximumError_8u** (const Npp8u * *pSrc1*, const Npp8u * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

8-bit unsigned char maximum method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumErrorGetBufferSize_8u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.205.2.15 NppStatus nppsMaximumErrorGetBufferSize_16s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.16 NppStatus nppsMaximumErrorGetBufferSize_16sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_16sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.17 NppStatus nppsMaximumErrorGetBufferSize_16u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_16u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.18 NppStatus nppsMaximumErrorGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.19 NppStatus nppsMaximumErrorGetBufferSize_32fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_32fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.20 NppStatus nppsMaximumErrorGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.21 NppStatus nppsMaximumErrorGetBufferSize_32sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_32sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.22 NppStatus nppsMaximumErrorGetBufferSize_32u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_32u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.23 NppStatus nppsMaximumErrorGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.24 NppStatus nppsMaximumErrorGetBufferSize_64fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_64fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.25 NppStatus nppsMaximumErrorGetBufferSize_64s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_64s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.26 NppStatus nppsMaximumErrorGetBufferSize_64sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_64sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.27 NppStatus nppsMaximumErrorGetBufferSize_8s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_8s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.205.2.28 NppStatus nppsMaximumErrorGetBufferSize_8u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumError_8u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206 AverageError

Primitives for computing the Average error between two signals.

Functions

- `NppStatus nppsAverageError_8u` (const `Npp8u` *pSrc1, const `Npp8u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
8-bit unsigned char Average method.
- `NppStatus nppsAverageError_8s` (const `Npp8s` *pSrc1, const `Npp8s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
8-bit signed char Average method.
- `NppStatus nppsAverageError_16u` (const `Npp16u` *pSrc1, const `Npp16u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit unsigned short integer Average method.
- `NppStatus nppsAverageError_16s` (const `Npp16s` *pSrc1, const `Npp16s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit signed short integer Average method.
- `NppStatus nppsAverageError_16sc` (const `Npp16sc` *pSrc1, const `Npp16sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit unsigned short complex integer Average method.
- `NppStatus nppsAverageError_32u` (const `Npp32u` *pSrc1, const `Npp32u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit unsigned short integer Average method.
- `NppStatus nppsAverageError_32s` (const `Npp32s` *pSrc1, const `Npp32s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit signed short integer Average method.
- `NppStatus nppsAverageError_32sc` (const `Npp32sc` *pSrc1, const `Npp32sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit unsigned short complex integer Average method.
- `NppStatus nppsAverageError_64s` (const `Npp64s` *pSrc1, const `Npp64s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
64-bit signed short integer Average method.
- `NppStatus nppsAverageError_64sc` (const `Npp64sc` *pSrc1, const `Npp64sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
64-bit unsigned short complex integer Average method.
- `NppStatus nppsAverageError_32f` (const `Npp32f` *pSrc1, const `Npp32f` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit floating point Average method.

- [NppStatus nppsAverageError_32fc](#) (const [Npp32fc](#) *pSrc1, const [Npp32fc](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
32-bit floating point complex Average method.
- [NppStatus nppsAverageError_64f](#) (const [Npp64f](#) *pSrc1, const [Npp64f](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
64-bit floating point Average method.
- [NppStatus nppsAverageError_64fc](#) (const [Npp64fc](#) *pSrc1, const [Npp64fc](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
64-bit floating point complex Average method.
- [NppStatus nppsAverageErrorGetBufferSize_8u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_8u.
- [NppStatus nppsAverageErrorGetBufferSize_8s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_8s.
- [NppStatus nppsAverageErrorGetBufferSize_16u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_16u.
- [NppStatus nppsAverageErrorGetBufferSize_16s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_16s.
- [NppStatus nppsAverageErrorGetBufferSize_16sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_16sc.
- [NppStatus nppsAverageErrorGetBufferSize_32u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_32u.
- [NppStatus nppsAverageErrorGetBufferSize_32s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_32s.
- [NppStatus nppsAverageErrorGetBufferSize_32sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_32sc.
- [NppStatus nppsAverageErrorGetBufferSize_64s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_64s.
- [NppStatus nppsAverageErrorGetBufferSize_64sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_64sc.
- [NppStatus nppsAverageErrorGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_32f.
- [NppStatus nppsAverageErrorGetBufferSize_32fc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_32fc.
- [NppStatus nppsAverageErrorGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageError_64f.

- [NppStatus nppsAverageErrorGetBufferSize_64fc](#) (int nLength, int *hpBufferSize)

Device-buffer size (in bytes) for nppsAverageError_64fc.

7.206.1 Detailed Description

Primitives for computing the Average error between two signals.

Given two signals $pSrc1$ and $pSrc2$ both with length N , the average error is defined as

$$AverageError = \frac{1}{N} \sum_{n=0}^{N-1} |pSrc1(n) - pSrc2(n)|$$

If the signal is in complex format, the absolute value of the complex number is used.

7.206.2 Function Documentation

7.206.2.1 NppStatus nppsAverageError_16s (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

16-bit signed short integer Average method.

Parameters:

[pSrc1](#) Source Signal Pointer.

[pSrc2](#) Source Signal Pointer.

[nLength](#) Signal Length.

[pDst](#) Pointer to the error result.

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsAverageErrorGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.2 NppStatus nppsAverageError_16sc (const Npp16sc * pSrc1, const Npp16sc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

16-bit unsigned short complex integer Average method.

Parameters:

[pSrc1](#) Source Signal Pointer.

[pSrc2](#) Source Signal Pointer.

[nLength](#) Signal Length.

[pDst](#) Pointer to the error result.

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsAverageErrorGetBufferSize_16sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.3 `NppStatus nppsAverageError_16u (const Npp16u * pSrc1, const Npp16u * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

16-bit unsigned short integer Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_16u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.4 `NppStatus nppsAverageError_32f (const Npp32f * pSrc1, const Npp32f * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

32-bit floating point Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.5 `NppStatus nppsAverageError_32fc (const Npp32fc * pSrc1, const Npp32fc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

32-bit floating point complex Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_32fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.6 **NppStatus nppsAverageError_32s** (const Npp32s * *pSrc1*, const Npp32s * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit signed short integer Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.7 **NppStatus nppsAverageError_32sc** (const Npp32sc * *pSrc1*, const Npp32sc * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit unsigned short complex integer Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_32sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.8 **NppStatus nppsAverageError_32u** (const Npp32u * *pSrc1*, const Npp32u * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit unsigned short integer Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_32u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.9 **NppStatus nppsAverageError_64f** (const Npp64f * *pSrc1*, const Npp64f * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

64-bit floating point Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.10 **NppStatus nppsAverageError_64fc** (const Npp64fc * *pSrc1*, const Npp64fc * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

64-bit floating point complex Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_64fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.11 **NppStatus nppsAverageError_64s** (const Npp64s * *pSrc1*, const Npp64s * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

64-bit signed short integer Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_64s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.12 **NppStatus nppsAverageError_64sc** (const Npp64sc * *pSrc1*, const Npp64sc * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

64-bit unsigned short complex integer Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_64sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.13 **NppStatus nppsAverageError_8s** (const Npp8s * *pSrc1*, const Npp8s * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

8-bit signed char Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_8s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.14 **NppStatus nppsAverageError_8u** (const Npp8u * *pSrc1*, const Npp8u * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

8-bit unsigned char Average method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageErrorGetBufferSize_8u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.206.2.15 NppStatus nppsAverageErrorGetBufferSize_16s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.16 NppStatus nppsAverageErrorGetBufferSize_16sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_16sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.17 NppStatus nppsAverageErrorGetBufferSize_16u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_16u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.18 NppStatus nppsAverageErrorGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.19 NppStatus nppsAverageErrorGetBufferSize_32fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_32fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.20 NppStatus nppsAverageErrorGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.21 NppStatus nppsAverageErrorGetBufferSize_32sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_32sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.22 NppStatus nppsAverageErrorGetBufferSize_32u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_32u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.23 NppStatus nppsAverageErrorGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.24 NppStatus nppsAverageErrorGetBufferSize_64fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_64fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.25 NppStatus nppsAverageErrorGetBufferSize_64s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_64s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.26 NppStatus nppsAverageErrorGetBufferSize_64sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_64sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.27 NppStatus nppsAverageErrorGetBufferSize_8s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_8s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.206.2.28 NppStatus nppsAverageErrorGetBufferSize_8u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageError_8u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207 MaximumRelativeError

Primitives for computing the MaximumRelative error between two signals.

Functions

- `NppStatus nppsMaximumRelativeError_8u` (const `Npp8u` *pSrc1, const `Npp8u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
8-bit unsigned char MaximumRelative method.
- `NppStatus nppsMaximumRelativeError_8s` (const `Npp8s` *pSrc1, const `Npp8s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
8-bit signed char MaximumRelative method.
- `NppStatus nppsMaximumRelativeError_16u` (const `Npp16u` *pSrc1, const `Npp16u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit unsigned short integer MaximumRelative method.
- `NppStatus nppsMaximumRelativeError_16s` (const `Npp16s` *pSrc1, const `Npp16s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit signed short integer MaximumRelative method.
- `NppStatus nppsMaximumRelativeError_16sc` (const `Npp16sc` *pSrc1, const `Npp16sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit unsigned short complex integer MaximumRelative method.
- `NppStatus nppsMaximumRelativeError_32u` (const `Npp32u` *pSrc1, const `Npp32u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit unsigned short integer MaximumRelative method.
- `NppStatus nppsMaximumRelativeError_32s` (const `Npp32s` *pSrc1, const `Npp32s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit signed short integer MaximumRelative method.
- `NppStatus nppsMaximumRelativeError_32sc` (const `Npp32sc` *pSrc1, const `Npp32sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit unsigned short complex integer MaximumRelative method.
- `NppStatus nppsMaximumRelativeError_64s` (const `Npp64s` *pSrc1, const `Npp64s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
64-bit signed short integer MaximumRelative method.
- `NppStatus nppsMaximumRelativeError_64sc` (const `Npp64sc` *pSrc1, const `Npp64sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
64-bit unsigned short complex integer MaximumRelative method.
- `NppStatus nppsMaximumRelativeError_32f` (const `Npp32f` *pSrc1, const `Npp32f` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit floating point MaximumRelative method.

- [NppStatus nppsMaximumRelativeError_32fc](#) (const [Npp32fc](#) *pSrc1, const [Npp32fc](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
32-bit floating point complex MaximumRelative method.
- [NppStatus nppsMaximumRelativeError_64f](#) (const [Npp64f](#) *pSrc1, const [Npp64f](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
64-bit floating point MaximumRelative method.
- [NppStatus nppsMaximumRelativeError_64fc](#) (const [Npp64fc](#) *pSrc1, const [Npp64fc](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
64-bit floating point complex MaximumRelative method.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_8u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_8u.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_8s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_8s.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_16u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_16u.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_16s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_16s.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_16sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_16sc.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_32u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_32u.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_32s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_32s.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_32sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_32sc.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_64s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_64s.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_64sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_64sc.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_32f.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_32fc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_32fc.
- [NppStatus nppsMaximumRelativeErrorGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsMaximumRelativeError_64f.

- [NppStatus nppsMaximumRelativeErrorGetBufferSize_64fc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for *nppsMaximumRelativeError_64fc*.

7.207.1 Detailed Description

Primitives for computing the MaximumRelative error between two signals.

Given two signals *pSrc1* and *pSrc2* both with length *N*, the maximum relative error is defined as

$$\text{MaximumRelativeError} = \max \frac{|pSrc1(n) - pSrc2(n)|}{\max(|pSrc1(n)|, |pSrc2(n)|)}$$

If the signal is in complex format, the absolute value of the complex number is used.

7.207.2 Function Documentation

7.207.2.1 NppStatus nppsMaximumRelativeError_16s (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

16-bit signed short integer MaximumRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.2 NppStatus nppsMaximumRelativeError_16sc (const Npp16sc * pSrc1, const Npp16sc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

16-bit unsigned short complex integer MaximumRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_16sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.3 NppStatus nppsMaximumRelativeError_16u (const Npp16u * pSrc1, const Npp16u * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

16-bit unsigned short integer MaximumRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_16u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.4 NppStatus nppsMaximumRelativeError_32f (const Npp32f * pSrc1, const Npp32f * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

32-bit floating point MaximumRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.5 NppStatus nppsMaximumRelativeError_32fc (const Npp32fc * *pSrc1*, const Npp32fc * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit floating point complex MaximumRelative method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_32fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.6 NppStatus nppsMaximumRelativeError_32s (const Npp32s * *pSrc1*, const Npp32s * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit signed short integer MaximumRelative method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.7 NppStatus nppsMaximumRelativeError_32sc (const Npp32sc * *pSrc1*, const Npp32sc * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit unsigned short complex integer MaximumRelative method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_32sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.8 `NppStatus nppsMaximumRelativeError_32u (const Npp32u * pSrc1, const Npp32u * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

32-bit unsigned short integer MaximumRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_32u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.9 `NppStatus nppsMaximumRelativeError_64f (const Npp64f * pSrc1, const Npp64f * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit floating point MaximumRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.10 `NppStatus nppsMaximumRelativeError_64fc (const Npp64fc * pSrc1, const Npp64fc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit floating point complex MaximumRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_64fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.11 `NppStatus nppsMaximumRelativeError_64s (const Npp64s * pSrc1, const Npp64s * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit signed short integer MaximumRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_64s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.12 `NppStatus nppsMaximumRelativeError_64sc (const Npp64sc * pSrc1, const Npp64sc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit unsigned short complex integer MaximumRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_64sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.13 `NppStatus nppsMaximumRelativeError_8s (const Npp8s * pSrc1, const Npp8s * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

8-bit signed char MaximumRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_8s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.14 `NppStatus nppsMaximumRelativeError_8u (const Npp8u * pSrc1, const Npp8u * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

8-bit unsigned char MaximumRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsMaximumRelativeErrorGetBufferSize_8u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.207.2.15 NppStatus nppsMaximumRelativeErrorGetBufferSize_16s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.16 NppStatus nppsMaximumRelativeErrorGetBufferSize_16sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_16sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.17 NppStatus nppsMaximumRelativeErrorGetBufferSize_16u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_16u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.18 NppStatus nppsMaximumRelativeErrorGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.19 NppStatus nppsMaximumRelativeErrorGetBufferSize_32fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_32fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.20 NppStatus nppsMaximumRelativeErrorGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.21 NppStatus nppsMaximumRelativeErrorGetBufferSize_32sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_32sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.22 NppStatus nppsMaximumRelativeErrorGetBufferSize_32u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_32u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.23 NppStatus nppsMaximumRelativeErrorGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.24 NppStatus nppsMaximumRelativeErrorGetBufferSize_64fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_64fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.25 NppStatus nppsMaximumRelativeErrorGetBufferSize_64s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_64s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.26 NppStatus nppsMaximumRelativeErrorGetBufferSize_64sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_64sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.27 NppStatus nppsMaximumRelativeErrorGetBufferSize_8s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_8s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.207.2.28 NppStatus nppsMaximumRelativeErrorGetBufferSize_8u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsMaximumRelativeError_8u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208 AverageRelativeError

Primitives for computing the AverageRelative error between two signals.

Functions

- `NppStatus nppsAverageRelativeError_8u` (const `Npp8u` *pSrc1, const `Npp8u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
8-bit unsigned char AverageRelative method.
- `NppStatus nppsAverageRelativeError_8s` (const `Npp8s` *pSrc1, const `Npp8s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
8-bit signed char AverageRelative method.
- `NppStatus nppsAverageRelativeError_16u` (const `Npp16u` *pSrc1, const `Npp16u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit unsigned short integer AverageRelative method.
- `NppStatus nppsAverageRelativeError_16s` (const `Npp16s` *pSrc1, const `Npp16s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit signed short integer AverageRelative method.
- `NppStatus nppsAverageRelativeError_16sc` (const `Npp16sc` *pSrc1, const `Npp16sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
16-bit unsigned short complex integer AverageRelative method.
- `NppStatus nppsAverageRelativeError_32u` (const `Npp32u` *pSrc1, const `Npp32u` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit unsigned short integer AverageRelative method.
- `NppStatus nppsAverageRelativeError_32s` (const `Npp32s` *pSrc1, const `Npp32s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit signed short integer AverageRelative method.
- `NppStatus nppsAverageRelativeError_32sc` (const `Npp32sc` *pSrc1, const `Npp32sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit unsigned short complex integer AverageRelative method.
- `NppStatus nppsAverageRelativeError_64s` (const `Npp64s` *pSrc1, const `Npp64s` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
64-bit signed short integer AverageRelative method.
- `NppStatus nppsAverageRelativeError_64sc` (const `Npp64sc` *pSrc1, const `Npp64sc` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
64-bit unsigned short complex integer AverageRelative method.
- `NppStatus nppsAverageRelativeError_32f` (const `Npp32f` *pSrc1, const `Npp32f` *pSrc2, int nLength, `Npp64f` *pDst, `Npp8u` *pDeviceBuffer)
32-bit floating point AverageRelative method.

- [NppStatus nppsAverageRelativeError_32fc](#) (const [Npp32fc](#) *pSrc1, const [Npp32fc](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
32-bit floating point complex AverageRelative method.
- [NppStatus nppsAverageRelativeError_64f](#) (const [Npp64f](#) *pSrc1, const [Npp64f](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
64-bit floating point AverageRelative method.
- [NppStatus nppsAverageRelativeError_64fc](#) (const [Npp64fc](#) *pSrc1, const [Npp64fc](#) *pSrc2, int nLength, [Npp64f](#) *pDst, [Npp8u](#) *pDeviceBuffer)
64-bit floating point complex AverageRelative method.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_8u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_8u.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_8s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_8s.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_16u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_16u.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_16s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_16s.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_16sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_16sc.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_32u](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_32u.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_32s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_32s.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_32sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_32sc.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_64s](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_64s.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_64sc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_64sc.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_32f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_32f.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_32fc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_32fc.
- [NppStatus nppsAverageRelativeErrorGetBufferSize_64f](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for nppsAverageRelativeError_64f.

- [NppStatus nppsAverageRelativeErrorGetBufferSize_64fc](#) (int nLength, int *hpBufferSize)
Device-buffer size (in bytes) for [nppsAverageRelativeError_64fc](#).

7.208.1 Detailed Description

Primitives for computing the AverageRelative error between two signals.

Given two signals $pSrc1$ and $pSrc2$ both with length N , the average relative error is defined as

$$AverageRelativeError = \frac{1}{N} \sum_{n=0}^{N-1} \frac{|pSrc1(n) - pSrc2(n)|}{\max(|pSrc1(n)|, |pSrc2(n)|)}$$

If the signal is in complex format, the absolute value of the complex number is used.

7.208.2 Function Documentation

7.208.2.1 NppStatus nppsAverageRelativeError_16s (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

16-bit signed short integer AverageRelative method.

Parameters:

[pSrc1](#) Source Signal Pointer.

[pSrc2](#) Source Signal Pointer.

[nLength](#) Signal Length.

[pDst](#) Pointer to the error result.

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_16s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.2 NppStatus nppsAverageRelativeError_16sc (const Npp16sc * pSrc1, const Npp16sc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

16-bit unsigned short complex integer AverageRelative method.

Parameters:

[pSrc1](#) Source Signal Pointer.

[pSrc2](#) Source Signal Pointer.

[nLength](#) Signal Length.

[pDst](#) Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_16sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.3 NppStatus nppsAverageRelativeError_16u (const Npp16u * pSrc1, const Npp16u * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

16-bit unsigned short integer AverageRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_16u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.4 NppStatus nppsAverageRelativeError_32f (const Npp32f * pSrc1, const Npp32f * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)

32-bit floating point AverageRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_32f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.5 NppStatus nppsAverageRelativeError_32fc (const Npp32fc * *pSrc1*, const Npp32fc * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit floating point complex AverageRelative method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_32fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.6 NppStatus nppsAverageRelativeError_32s (const Npp32s * *pSrc1*, const Npp32s * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit signed short integer AverageRelative method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_32s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.7 NppStatus nppsAverageRelativeError_32sc (const Npp32sc * *pSrc1*, const Npp32sc * *pSrc2*, int *nLength*, Npp64f * *pDst*, Npp8u * *pDeviceBuffer*)

32-bit unsigned short complex integer AverageRelative method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_32sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.8 `NppStatus nppsAverageRelativeError_32u (const Npp32u * pSrc1, const Npp32u * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

32-bit unsigned short integer AverageRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_32u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.9 `NppStatus nppsAverageRelativeError_64f (const Npp64f * pSrc1, const Npp64f * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit floating point AverageRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_64f](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.10 `NppStatus nppsAverageRelativeError_64fc (const Npp64fc * pSrc1, const Npp64fc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit floating point complex AverageRelative method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_64fc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.11 `NppStatus nppsAverageRelativeError_64s (const Npp64s * pSrc1, const Npp64s * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit signed short integer AverageRelative method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_64s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.12 `NppStatus nppsAverageRelativeError_64sc (const Npp64sc * pSrc1, const Npp64sc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit unsigned short complex integer AverageRelative method.

Parameters:

pSrc1 Source Signal Pointer.

pSrc2 Source Signal Pointer.

nLength Signal Length.

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_64sc](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.13 `NppStatus nppsAverageRelativeError_8s (const Npp8s * pSrc1, const Npp8s * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

8-bit signed char AverageRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_8s](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.14 `NppStatus nppsAverageRelativeError_8u (const Npp8u * pSrc1, const Npp8u * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

8-bit unsigned char AverageRelative method.

Parameters:

pSrc1 [Source Signal Pointer](#).

pSrc2 [Source Signal Pointer](#).

nLength [Signal Length](#).

pDst Pointer to the error result.

pDeviceBuffer Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).
Use [nppsAverageRelativeErrorGetBufferSize_8u](#) to determine the minium number of bytes required.

Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

7.208.2.15 NppStatus nppsAverageRelativeErrorGetBufferSize_16s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_16s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.16 NppStatus nppsAverageRelativeErrorGetBufferSize_16sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_16sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.17 NppStatus nppsAverageRelativeErrorGetBufferSize_16u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_16u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.18 NppStatus nppsAverageRelativeErrorGetBufferSize_32f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_32f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.19 NppStatus nppsAverageRelativeErrorGetBufferSize_32fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_32fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.20 NppStatus nppsAverageRelativeErrorGetBufferSize_32s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_32s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.21 NppStatus nppsAverageRelativeErrorGetBufferSize_32sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_32sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.22 NppStatus nppsAverageRelativeErrorGetBufferSize_32u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_32u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.23 NppStatus nppsAverageRelativeErrorGetBufferSize_64f (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_64f.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.24 NppStatus nppsAverageRelativeErrorGetBufferSize_64fc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_64fc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.25 NppStatus nppsAverageRelativeErrorGetBufferSize_64s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_64s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.26 NppStatus nppsAverageRelativeErrorGetBufferSize_64sc (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_64sc.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.27 NppStatus nppsAverageRelativeErrorGetBufferSize_8s (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_8s.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.208.2.28 NppStatus nppsAverageRelativeErrorGetBufferSize_8u (int *nLength*, int * *hpBufferSize*)

Device-buffer size (in bytes) for nppsAverageRelativeError_8u.

Parameters:

nLength [Signal Length](#).

hpBufferSize Required buffer size. Important: *hpBufferSize* is a *host pointer*.

Returns:

NPP_SUCCESS

7.209 Memory Management

Modules

- [Malloc](#)

Signal-allocator methods for allocating 1D arrays of data in device memory.

- [Free](#)

Free signal memory.

7.210 Malloc

Signal-allocator methods for allocating 1D arrays of data in device memory.

Functions

- [Npp8u](#) * [nppsMalloc_8u](#) (int nSize)
8-bit unsigned signal allocator.
- [Npp8s](#) * [nppsMalloc_8s](#) (int nSize)
8-bit signed signal allocator.
- [Npp16u](#) * [nppsMalloc_16u](#) (int nSize)
16-bit unsigned signal allocator.
- [Npp16s](#) * [nppsMalloc_16s](#) (int nSize)
16-bit signal allocator.
- [Npp16sc](#) * [nppsMalloc_16sc](#) (int nSize)
16-bit complex-value signal allocator.
- [Npp32u](#) * [nppsMalloc_32u](#) (int nSize)
32-bit unsigned signal allocator.
- [Npp32s](#) * [nppsMalloc_32s](#) (int nSize)
32-bit integer signal allocator.
- [Npp32sc](#) * [nppsMalloc_32sc](#) (int nSize)
32-bit complex integer signal allocator.
- [Npp32f](#) * [nppsMalloc_32f](#) (int nSize)
32-bit float signal allocator.
- [Npp32fc](#) * [nppsMalloc_32fc](#) (int nSize)
32-bit complex float signal allocator.
- [Npp64s](#) * [nppsMalloc_64s](#) (int nSize)
64-bit long integer signal allocator.
- [Npp64sc](#) * [nppsMalloc_64sc](#) (int nSize)
64-bit complex long integer signal allocator.
- [Npp64f](#) * [nppsMalloc_64f](#) (int nSize)
64-bit float (double) signal allocator.
- [Npp64fc](#) * [nppsMalloc_64fc](#) (int nSize)
64-bit complex complex signal allocator.

7.210.1 Detailed Description

Signal-allocator methods for allocating 1D arrays of data in device memory.

All allocators have size parameters to specify the size of the signal (1D array) being allocated.

The allocator methods return a pointer to the newly allocated memory of appropriate type. If device-memory allocation is not possible due to resource constraints the allocators return 0 (i.e. NULL pointer).

All signal allocators allocate memory aligned such that it is beneficial to the performance of the majority of the signal-processing primitives. It is no mandatory however to use these allocators. Any valid CUDA device-memory pointers can be passed to NPP primitives.

7.210.2 Function Documentation

7.210.2.1 Npp16s* nppsMalloc_16s (int *nSize*)

16-bit signal allocator.

Parameters:

nSize Number of shorts in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.2 Npp16sc* nppsMalloc_16sc (int *nSize*)

16-bit complex-value signal allocator.

Parameters:

nSize Number of 16-bit complex numbers in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.3 Npp16u* nppsMalloc_16u (int *nSize*)

16-bit unsigned signal allocator.

Parameters:

nSize Number of unsigned shorts in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.4 Npp32f* nppsMalloc_32f (int *nSize*)

32-bit float signal allocator.

Parameters:

nSize Number of floats in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.5 Npp32fc* nppsMalloc_32fc (int *nSize*)

32-bit complex float signal allocator.

Parameters:

nSize Number of complex float values in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.6 Npp32s* nppsMalloc_32s (int *nSize*)

32-bit integer signal allocator.

Parameters:

nSize Number of ints in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.7 Npp32sc* nppsMalloc_32sc (int *nSize*)

32-bit complex integer signal allocator.

Parameters:

nSize Number of complex integner values in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.8 Npp32u* nppsMalloc_32u (int nSize)

32-bit unsigned signal allocator.

Parameters:

nSize Number of unsigned ints in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.9 Npp64f* nppsMalloc_64f (int nSize)

64-bit float (double) signal allocator.

Parameters:

nSize Number of doubles in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.10 Npp64fc* nppsMalloc_64fc (int nSize)

64-bit complex complex signal allocator.

Parameters:

nSize Number of complex double values in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.11 Npp64s* nppsMalloc_64s (int nSize)

64-bit long integer signal allocator.

Parameters:

nSize Number of long ints in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.12 Npp64sc* nppsMalloc_64sc (int *nSize*)

64-bit complex long integer signal allocator.

Parameters:

nSize Number of complex long int values in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.13 Npp8s* nppsMalloc_8s (int *nSize*)

8-bit signed signal allocator.

Parameters:

nSize Number of (signed) chars in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.210.2.14 Npp8u* nppsMalloc_8u (int *nSize*)

8-bit unsigned signal allocator.

Parameters:

nSize Number of unsigned chars in the new signal.

Returns:

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

7.211 Free

Free signal memory.

Functions

- void [nppsFree](#) (void *pValues)
Free method for any signal memory.

7.211.1 Detailed Description

Free signal memory.

7.211.2 Function Documentation

7.211.2.1 void nppsFree (void * pValues)

Free method for any signal memory.

Parameters:

pValues A pointer to memory allocated using `nppiMalloc_<modifier>`.

Chapter 8

Data Structure Documentation

8.1 NPP_ALIGN_16 Struct Reference

Complex Number This struct represents a long long complex number.

```
#include <nppdefs.h>
```

Data Fields

- [Npp64s re](#)
Real part.
- [Npp64s im](#)
Imaginary part.
- [Npp64f re](#)
Real part.
- [Npp64f im](#)
Imaginary part.

8.1.1 Detailed Description

Complex Number This struct represents a long long complex number.

Complex Number This struct represents a double floating-point complex number.

8.1.2 Field Documentation

8.1.2.1 Npp64f NPP_ALIGN_16::im

Imaginary part.

8.1.2.2 Npp64s NPP_ALIGN_16::im

Imaginary part.

8.1.2.3 Npp64f NPP_ALIGN_16::re

Real part.

8.1.2.4 Npp64s NPP_ALIGN_16::re

Real part.

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r7.5/NPP/npp/include/nppdefs.h

8.2 NPP_ALIGN_8 Struct Reference

Complex Number This struct represents an unsigned int complex number.

```
#include <nppdefs.h>
```

Data Fields

- [Npp32u re](#)
Real part.
- [Npp32u im](#)
Imaginary part.
- [Npp32s re](#)
Real part.
- [Npp32s im](#)
Imaginary part.
- [Npp32f re](#)
Real part.
- [Npp32f im](#)
Imaginary part.

8.2.1 Detailed Description

Complex Number This struct represents an unsigned int complex number.

Complex Number This struct represents a single floating-point complex number.

Complex Number This struct represents a signed int complex number.

8.2.2 Field Documentation

8.2.2.1 Npp32f NPP_ALIGN_8::im

Imaginary part.

8.2.2.2 Npp32s NPP_ALIGN_8::im

Imaginary part.

8.2.2.3 Npp32u NPP_ALIGN_8::im

Imaginary part.

8.2.2.4 Npp32f NPP_ALIGN_8::re

Real part.

8.2.2.5 Npp32s NPP_ALIGN_8::re

Real part.

8.2.2.6 Npp32u NPP_ALIGN_8::re

Real part.

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r7.5/NPP/npp/include/nppdefs.h

8.3 NppiHaarBuffer Struct Reference

```
#include <nppdefs.h>
```

Data Fields

- `int haarBufferSize`
size of the buffer
- `Npp32s * haarBuffer`
buffer

8.3.1 Field Documentation

8.3.1.1 `Npp32s* NppiHaarBuffer::haarBuffer`

buffer

8.3.1.2 `int NppiHaarBuffer::haarBufferSize`

size of the buffer

The documentation for this struct was generated from the following file:

- `C:/src/sw/rel/gpgpu/toolkit/r7.5/NPP/npp/include/nppdefs.h`

8.4 NppiHaarClassifier_32f Struct Reference

```
#include <nppdefs.h>
```

Data Fields

- int [numClassifiers](#)
number of classifiers
- [Npp32s](#) * [classifiers](#)
packed classifier data 40 bytes each
- size_t [classifierStep](#)
- [NppiSize](#) [classifierSize](#)
- [Npp32s](#) * [counterDevice](#)

8.4.1 Field Documentation

8.4.1.1 [Npp32s](#)* [NppiHaarClassifier_32f::classifiers](#)

packed classifier data 40 bytes each

8.4.1.2 [NppiSize](#) [NppiHaarClassifier_32f::classifierSize](#)

8.4.1.3 [size_t](#) [NppiHaarClassifier_32f::classifierStep](#)

8.4.1.4 [Npp32s](#)* [NppiHaarClassifier_32f::counterDevice](#)

8.4.1.5 [int](#) [NppiHaarClassifier_32f::numClassifiers](#)

number of classifiers

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r7.5/NPP/npp/include/nppdefs.h

8.5 NppiPoint Struct Reference

2D Point

```
#include <nppdefs.h>
```

Data Fields

- `int x`
x-coordinate.
- `int y`
y-coordinate.

8.5.1 Detailed Description

2D Point

8.5.2 Field Documentation

8.5.2.1 `int NppiPoint::x`

x-coordinate.

8.5.2.2 `int NppiPoint::y`

y-coordinate.

The documentation for this struct was generated from the following file:

- `C:/src/sw/rel/gpgpu/toolkit/r7.5/NPP/npp/include/nppdefs.h`

8.6 NppiRect Struct Reference

2D Rectangle This struct contains position and size information of a rectangle in two space.

```
#include <nppdefs.h>
```

Data Fields

- `int x`
x-coordinate of upper left corner.
- `int y`
y-coordinate of upper left corner.
- `int width`
Rectangle width.
- `int height`
Rectangle height.

8.6.1 Detailed Description

2D Rectangle This struct contains position and size information of a rectangle in two space.

The rectangle's position is usually signified by the coordinate of its upper-left corner.

8.6.2 Field Documentation

8.6.2.1 `int NppiRect::height`

Rectangle height.

8.6.2.2 `int NppiRect::width`

Rectangle width.

8.6.2.3 `int NppiRect::x`

x-coordinate of upper left corner.

8.6.2.4 `int NppiRect::y`

y-coordinate of upper left corner.

The documentation for this struct was generated from the following file:

- `C:/src/sw/rel/gpgpu/toolkit/r7.5/NPP/npp/include/nppdefs.h`

8.7 NppiSize Struct Reference

2D Size This struct typically represents the size of a rectangular region in two space.

```
#include <nppdefs.h>
```

Data Fields

- int `width`
Rectangle width.
- int `height`
Rectangle height.

8.7.1 Detailed Description

2D Size This struct typically represents the size of a rectangular region in two space.

8.7.2 Field Documentation

8.7.2.1 int NppiSize::height

Rectangle height.

8.7.2.2 int NppiSize::width

Rectangle width.

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r7.5/NPP/npp/include/nppdefs.h

8.8 NppLibraryVersion Struct Reference

```
#include <nppdefs.h>
```

Data Fields

- int [major](#)
Major version number.
- int [minor](#)
Minor version number.
- int [build](#)
Build number.

8.8.1 Field Documentation

8.8.1.1 int NppLibraryVersion::build

Build number.

This reflects the nightly build this release was made from.

8.8.1.2 int NppLibraryVersion::major

Major version number.

8.8.1.3 int NppLibraryVersion::minor

Minor version number.

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r7.5/NPP/npp/include/nppdefs.h

Index

- [__align__](#)
 - [npp_basic_types](#), [50](#), [51](#)
- [10Log10](#), [2612](#)
- [1D Linear Filter](#), [1103](#)
- [1D Window Sum](#), [1197](#)
- [1D Window Sum with Border Control](#), [1208](#)
- [2D Fixed Linear Filters](#), [1280](#)
-
- [Abs](#), [321](#), [2586](#)
- [AbsDiff](#), [328](#)
- [AbsDiffC](#), [167](#)
- [Add](#), [169](#), [2536](#)
- [AddC](#), [56](#), [2489](#)
- [AddProduct](#), [201](#), [2548](#)
- [AddProductC](#), [2498](#)
- [AddSquare](#), [198](#)
- [AddWeighted](#), [205](#)
- [Affine Transforms](#), [1479](#)
- [Alpha Composition](#), [473](#)
- [AlphaComp](#), [489](#)
- [AlphaCompC](#), [474](#)
- [AlphaPremul](#), [496](#)
- [AlphaPremulC](#), [482](#)
- [And](#), [433](#), [2628](#)
- [AndC](#), [372](#), [2625](#)
- [Arctan](#), [2617](#)
- [Arithmetic and Logical Operations](#), [53](#), [2486](#)
- [Arithmetic Operations](#), [54](#), [2487](#)
- [AverageError](#), [2289](#), [2824](#)
- [AverageRelativeError](#), [2336](#), [2847](#)
-
- [Basic NPP Data Types](#), [48](#)
- [build](#)
 - [NppLibraryVersion](#), [2876](#)
-
- [Cauchy, CauchyD, and CauchyDD2](#), [2622](#)
- [classifiers](#)
 - [NppiHaarClassifier_32f](#), [2872](#)
- [classifierSize](#)
 - [NppiHaarClassifier_32f](#), [2872](#)
- [classifierStep](#)
 - [NppiHaarClassifier_32f](#), [2872](#)
- [Color and Sampling Conversion](#), [498](#)
- [Color Gamma Correction](#), [614](#)
- [Color Model Conversion](#), [499](#)
-
- [Color Processing](#), [623](#)
- [Color Sampling Format Conversion](#), [586](#)
- [Compare Operations](#), [2462](#)
- [Complement Color Key](#), [620](#)
- [Compression](#), [720](#)
- [Conversion Functions](#), [2654](#)
- [Convert](#), [820](#), [2655](#)
- [Convolution](#), [1221](#)
- [Copy](#), [773](#), [2695](#)
- [Copy Constant Border](#), [879](#)
- [Copy Replicate Border](#), [892](#)
- [Copy Sub-Pixel](#), [917](#)
- [Copy Wrap Border](#), [904](#)
- [core_npp](#)
 - [nppGetGpuComputeCapability](#), [32](#)
 - [nppGetGpuName](#), [32](#)
 - [nppGetGpuNumSMs](#), [32](#)
 - [nppGetLibVersion](#), [32](#)
 - [nppGetMaxThreadsPerBlock](#), [32](#)
 - [nppGetMaxThreadsPerSM](#), [32](#)
 - [nppGetStream](#), [33](#)
 - [nppSetStream](#), [33](#)
- [Count In Range](#), [2810](#)
- [Count Zero Crossings](#), [2811](#)
- [counterDevice](#)
 - [NppiHaarClassifier_32f](#), [2872](#)
- [CountInRange.](#), [2068](#)
- [CrossCorrFull_Norm](#), [2161](#)
- [CrossCorrFull_NormLevel](#), [2197](#)
- [crosscorrfullnorm](#)
 - [nppiCrossCorrFull_Norm_16u32f_AC4R](#), [2163](#)
 - [nppiCrossCorrFull_Norm_16u32f_C1R](#), [2163](#)
 - [nppiCrossCorrFull_Norm_16u32f_C3R](#), [2163](#)
 - [nppiCrossCorrFull_Norm_16u32f_C4R](#), [2164](#)
 - [nppiCrossCorrFull_Norm_32f_AC4R](#), [2164](#)
 - [nppiCrossCorrFull_Norm_32f_C1R](#), [2165](#)
 - [nppiCrossCorrFull_Norm_32f_C3R](#), [2165](#)
 - [nppiCrossCorrFull_Norm_32f_C4R](#), [2166](#)
 - [nppiCrossCorrFull_Norm_8s32f_AC4R](#), [2166](#)
 - [nppiCrossCorrFull_Norm_8s32f_C1R](#), [2166](#)
 - [nppiCrossCorrFull_Norm_8s32f_C3R](#), [2167](#)
 - [nppiCrossCorrFull_Norm_8s32f_C4R](#), [2167](#)
 - [nppiCrossCorrFull_Norm_8u32f_AC4R](#), [2168](#)
 - [nppiCrossCorrFull_Norm_8u32f_C1R](#), [2168](#)

- nppiCrossCorrFull_Norm_8u32f_C3R, [2169](#)
- nppiCrossCorrFull_Norm_8u32f_C4R, [2169](#)
- nppiCrossCorrFull_Norm_8u_AC4RSfs, [2169](#)
- nppiCrossCorrFull_Norm_8u_C1RSfs, [2170](#)
- nppiCrossCorrFull_Norm_8u_C3RSfs, [2170](#)
- nppiCrossCorrFull_Norm_8u_C4RSfs, [2171](#)
- crosscorrfullnormlevel
 - nppiCrossCorrFull_NormLevel_16u32f_-AC4R, [2201](#)
 - nppiCrossCorrFull_NormLevel_16u32f_C1R, [2201](#)
 - nppiCrossCorrFull_NormLevel_16u32f_C3R, [2201](#)
 - nppiCrossCorrFull_NormLevel_16u32f_C4R, [2202](#)
 - nppiCrossCorrFull_NormLevel_32f_AC4R, [2202](#)
 - nppiCrossCorrFull_NormLevel_32f_C1R, [2203](#)
 - nppiCrossCorrFull_NormLevel_32f_C3R, [2203](#)
 - nppiCrossCorrFull_NormLevel_32f_C4R, [2204](#)
 - nppiCrossCorrFull_NormLevel_8s32f_AC4R, [2204](#)
 - nppiCrossCorrFull_NormLevel_8s32f_C1R, [2205](#)
 - nppiCrossCorrFull_NormLevel_8s32f_C3R, [2205](#)
 - nppiCrossCorrFull_NormLevel_8s32f_C4R, [2206](#)
 - nppiCrossCorrFull_NormLevel_8u32f_AC4R, [2206](#)
 - nppiCrossCorrFull_NormLevel_8u32f_C1R, [2207](#)
 - nppiCrossCorrFull_NormLevel_8u32f_C3R, [2207](#)
 - nppiCrossCorrFull_NormLevel_8u32f_C4R, [2208](#)
 - nppiCrossCorrFull_NormLevel_8u_AC4RSfs, [2208](#)
 - nppiCrossCorrFull_NormLevel_8u_C1RSfs, [2209](#)
 - nppiCrossCorrFull_NormLevel_8u_C3RSfs, [2209](#)
 - nppiCrossCorrFull_NormLevel_8u_C4RSfs, [2210](#)
 - nppiFullNormLevelGetBufferHostSize_-16u32f_AC4R, [2210](#)
 - nppiFullNormLevelGetBufferHostSize_-16u32f_C1R, [2211](#)
 - nppiFullNormLevelGetBufferHostSize_-16u32f_C3R, [2211](#)
 - nppiFullNormLevelGetBufferHostSize_-16u32f_C4R, [2211](#)
 - nppiFullNormLevelGetBufferHostSize_32f_-AC4R, [2212](#)
 - nppiFullNormLevelGetBufferHostSize_32f_-C1R, [2212](#)
 - nppiFullNormLevelGetBufferHostSize_32f_-C3R, [2212](#)
 - nppiFullNormLevelGetBufferHostSize_32f_-C4R, [2212](#)
 - nppiFullNormLevelGetBufferHostSize_-8s32f_AC4R, [2213](#)
 - nppiFullNormLevelGetBufferHostSize_-8s32f_C1R, [2213](#)
 - nppiFullNormLevelGetBufferHostSize_-8s32f_C3R, [2213](#)
 - nppiFullNormLevelGetBufferHostSize_-8s32f_C4R, [2214](#)
 - nppiFullNormLevelGetBufferHostSize_-8u32f_AC4R, [2214](#)
 - nppiFullNormLevelGetBufferHostSize_-8u32f_C1R, [2214](#)
 - nppiFullNormLevelGetBufferHostSize_-8u32f_C3R, [2214](#)
 - nppiFullNormLevelGetBufferHostSize_-8u32f_C4R, [2215](#)
 - nppiFullNormLevelGetBufferHostSize_8u_-AC4RSfs, [2215](#)
 - nppiFullNormLevelGetBufferHostSize_8u_-C1RSfs, [2215](#)
 - nppiFullNormLevelGetBufferHostSize_8u_-C3RSfs, [2216](#)
 - nppiFullNormLevelGetBufferHostSize_8u_-C4RSfs, [2216](#)
- CrossCorrSame_Norm, [2172](#)
- CrossCorrSame_NormLevel, [2217](#)
- crosscorrsmenorm
 - nppiCrossCorrSame_Norm_16u32f_AC4R, [2174](#)
 - nppiCrossCorrSame_Norm_16u32f_C1R, [2174](#)
 - nppiCrossCorrSame_Norm_16u32f_C3R, [2174](#)
 - nppiCrossCorrSame_Norm_16u32f_C4R, [2175](#)
 - nppiCrossCorrSame_Norm_32f_AC4R, [2175](#)
 - nppiCrossCorrSame_Norm_32f_C1R, [2176](#)
 - nppiCrossCorrSame_Norm_32f_C3R, [2176](#)
 - nppiCrossCorrSame_Norm_32f_C4R, [2177](#)
 - nppiCrossCorrSame_Norm_8s32f_AC4R, [2177](#)
 - nppiCrossCorrSame_Norm_8s32f_C1R, [2177](#)
 - nppiCrossCorrSame_Norm_8s32f_C3R, [2178](#)
 - nppiCrossCorrSame_Norm_8s32f_C4R, [2178](#)

- nppiCrossCorrSame_Norm_8u32f_AC4R, [2179](#)
- nppiCrossCorrSame_Norm_8u32f_C1R, [2179](#)
- nppiCrossCorrSame_Norm_8u32f_C3R, [2180](#)
- nppiCrossCorrSame_Norm_8u32f_C4R, [2180](#)
- nppiCrossCorrSame_Norm_8u_AC4RSfs, [2180](#)
- nppiCrossCorrSame_Norm_8u_C1RSfs, [2181](#)
- nppiCrossCorrSame_Norm_8u_C3RSfs, [2181](#)
- nppiCrossCorrSame_Norm_8u_C4RSfs, [2182](#)
- crosscorrnamenormlevel
 - nppiCrossCorrSame_NormLevel_16u32f_-AC4R, [2221](#)
 - nppiCrossCorrSame_NormLevel_16u32f_-C1R, [2221](#)
 - nppiCrossCorrSame_NormLevel_16u32f_-C3R, [2221](#)
 - nppiCrossCorrSame_NormLevel_16u32f_-C4R, [2222](#)
 - nppiCrossCorrSame_NormLevel_32f_AC4R, [2222](#)
 - nppiCrossCorrSame_NormLevel_32f_C1R, [2223](#)
 - nppiCrossCorrSame_NormLevel_32f_C3R, [2223](#)
 - nppiCrossCorrSame_NormLevel_32f_C4R, [2224](#)
 - nppiCrossCorrSame_NormLevel_8s32f_-AC4R, [2224](#)
 - nppiCrossCorrSame_NormLevel_8s32f_C1R, [2225](#)
 - nppiCrossCorrSame_NormLevel_8s32f_C3R, [2225](#)
 - nppiCrossCorrSame_NormLevel_8s32f_C4R, [2226](#)
 - nppiCrossCorrSame_NormLevel_8u32f_-AC4R, [2226](#)
 - nppiCrossCorrSame_NormLevel_8u32f_C1R, [2227](#)
 - nppiCrossCorrSame_NormLevel_8u32f_C3R, [2227](#)
 - nppiCrossCorrSame_NormLevel_8u32f_C4R, [2228](#)
 - nppiCrossCorrSame_NormLevel_8u_-AC4RSfs, [2228](#)
 - nppiCrossCorrSame_NormLevel_8u_C1RSfs, [2229](#)
 - nppiCrossCorrSame_NormLevel_8u_C3RSfs, [2229](#)
 - nppiCrossCorrSame_NormLevel_8u_C4RSfs, [2230](#)
 - nppiSameNormLevelGetBufferHostSize_-16u32f_AC4R, [2230](#)
 - nppiSameNormLevelGetBufferHostSize_-16u32f_C1R, [2231](#)
 - nppiSameNormLevelGetBufferHostSize_-16u32f_C3R, [2231](#)
 - nppiSameNormLevelGetBufferHostSize_-16u32f_C4R, [2231](#)
 - nppiSameNormLevelGetBufferHostSize_-32f_AC4R, [2232](#)
 - nppiSameNormLevelGetBufferHostSize_-32f_C1R, [2232](#)
 - nppiSameNormLevelGetBufferHostSize_-32f_C3R, [2232](#)
 - nppiSameNormLevelGetBufferHostSize_-32f_C4R, [2232](#)
 - nppiSameNormLevelGetBufferHostSize_-8s32f_AC4R, [2233](#)
 - nppiSameNormLevelGetBufferHostSize_-8s32f_C1R, [2233](#)
 - nppiSameNormLevelGetBufferHostSize_-8s32f_C3R, [2233](#)
 - nppiSameNormLevelGetBufferHostSize_-8s32f_C4R, [2234](#)
 - nppiSameNormLevelGetBufferHostSize_-8u32f_AC4R, [2234](#)
 - nppiSameNormLevelGetBufferHostSize_-8u32f_C1R, [2234](#)
 - nppiSameNormLevelGetBufferHostSize_-8u32f_C3R, [2234](#)
 - nppiSameNormLevelGetBufferHostSize_-8u32f_C4R, [2235](#)
 - nppiSameNormLevelGetBufferHostSize_8u_-AC4RSfs, [2235](#)
 - nppiSameNormLevelGetBufferHostSize_8u_-C1RSfs, [2235](#)
 - nppiSameNormLevelGetBufferHostSize_8u_-C3RSfs, [2236](#)
 - nppiSameNormLevelGetBufferHostSize_8u_-C4RSfs, [2236](#)
- CrossCorrValid, [2194](#)
- crosscorrvalid
 - nppiCrossCorrValid_16u32f_C1R, [2194](#)
 - nppiCrossCorrValid_32f_C1R, [2195](#)
 - nppiCrossCorrValid_8s32f_C1R, [2195](#)
 - nppiCrossCorrValid_8u32f_C1R, [2195](#)
- CrossCorrValid_Norm, [2183](#)
- CrossCorrValid_NormLevel, [2237](#)
- crosscorrvalidnorm
 - nppiCrossCorrValid_Norm_16u32f_AC4R, [2185](#)
 - nppiCrossCorrValid_Norm_16u32f_C1R, [2185](#)
 - nppiCrossCorrValid_Norm_16u32f_C3R, [2185](#)

- nppiCrossCorrValid_Norm_16u32f_C4R, [2186](#)
- nppiCrossCorrValid_Norm_32f_AC4R, [2186](#)
- nppiCrossCorrValid_Norm_32f_C1R, [2187](#)
- nppiCrossCorrValid_Norm_32f_C3R, [2187](#)
- nppiCrossCorrValid_Norm_32f_C4R, [2188](#)
- nppiCrossCorrValid_Norm_8s32f_AC4R, [2188](#)
- nppiCrossCorrValid_Norm_8s32f_C1R, [2188](#)
- nppiCrossCorrValid_Norm_8s32f_C3R, [2189](#)
- nppiCrossCorrValid_Norm_8s32f_C4R, [2189](#)
- nppiCrossCorrValid_Norm_8u32f_AC4R, [2190](#)
- nppiCrossCorrValid_Norm_8u32f_C1R, [2190](#)
- nppiCrossCorrValid_Norm_8u32f_C3R, [2191](#)
- nppiCrossCorrValid_Norm_8u32f_C4R, [2191](#)
- nppiCrossCorrValid_Norm_8u_AC4RSfs, [2191](#)
- nppiCrossCorrValid_Norm_8u_C1RSfs, [2192](#)
- nppiCrossCorrValid_Norm_8u_C3RSfs, [2192](#)
- nppiCrossCorrValid_Norm_8u_C4RSfs, [2193](#)
- crosscorrvalidnormlevel
 - nppiCrossCorrValid_NormLevel_16u32f_AC4R, [2241](#)
 - nppiCrossCorrValid_NormLevel_16u32f_C1R, [2241](#)
 - nppiCrossCorrValid_NormLevel_16u32f_C3R, [2241](#)
 - nppiCrossCorrValid_NormLevel_16u32f_C4R, [2242](#)
 - nppiCrossCorrValid_NormLevel_32f_AC4R, [2242](#)
 - nppiCrossCorrValid_NormLevel_32f_C1R, [2243](#)
 - nppiCrossCorrValid_NormLevel_32f_C3R, [2243](#)
 - nppiCrossCorrValid_NormLevel_32f_C4R, [2244](#)
 - nppiCrossCorrValid_NormLevel_8s32f_AC4R, [2244](#)
 - nppiCrossCorrValid_NormLevel_8s32f_C1R, [2245](#)
 - nppiCrossCorrValid_NormLevel_8s32f_C3R, [2245](#)
 - nppiCrossCorrValid_NormLevel_8s32f_C4R, [2246](#)
 - nppiCrossCorrValid_NormLevel_8u32f_AC4R, [2246](#)
 - nppiCrossCorrValid_NormLevel_8u32f_C1R, [2247](#)
 - nppiCrossCorrValid_NormLevel_8u32f_C3R, [2247](#)
 - nppiCrossCorrValid_NormLevel_8u32f_C4R, [2248](#)
 - nppiCrossCorrValid_NormLevel_8u_AC4RSfs, [2248](#)
 - nppiCrossCorrValid_NormLevel_8u_C1RSfs, [2249](#)
 - nppiCrossCorrValid_NormLevel_8u_C3RSfs, [2249](#)
 - nppiCrossCorrValid_NormLevel_8u_C4RSfs, [2250](#)
 - nppiValidNormLevelGetBufferHostSize_16u32f_AC4R, [2250](#)
 - nppiValidNormLevelGetBufferHostSize_16u32f_C1R, [2251](#)
 - nppiValidNormLevelGetBufferHostSize_16u32f_C3R, [2251](#)
 - nppiValidNormLevelGetBufferHostSize_16u32f_C4R, [2251](#)
 - nppiValidNormLevelGetBufferHostSize_32f_AC4R, [2252](#)
 - nppiValidNormLevelGetBufferHostSize_32f_C1R, [2252](#)
 - nppiValidNormLevelGetBufferHostSize_32f_C3R, [2252](#)
 - nppiValidNormLevelGetBufferHostSize_32f_C4R, [2252](#)
 - nppiValidNormLevelGetBufferHostSize_8s32f_AC4R, [2253](#)
 - nppiValidNormLevelGetBufferHostSize_8s32f_C1R, [2253](#)
 - nppiValidNormLevelGetBufferHostSize_8s32f_C3R, [2253](#)
 - nppiValidNormLevelGetBufferHostSize_8s32f_C4R, [2254](#)
 - nppiValidNormLevelGetBufferHostSize_8u32f_AC4R, [2254](#)
 - nppiValidNormLevelGetBufferHostSize_8u32f_C1R, [2254](#)
 - nppiValidNormLevelGetBufferHostSize_8u32f_C3R, [2254](#)
 - nppiValidNormLevelGetBufferHostSize_8u32f_C4R, [2255](#)
 - nppiValidNormLevelGetBufferHostSize_8u_AC4RSfs, [2255](#)
 - nppiValidNormLevelGetBufferHostSize_8u_C1RSfs, [2255](#)
 - nppiValidNormLevelGetBufferHostSize_8u_C3RSfs, [2256](#)
 - nppiValidNormLevelGetBufferHostSize_8u_C4RSfs, [2256](#)
- Cubrt, [2603](#)
- Data Exchange and Initialization, [738](#)
- Dilate3x3, [1594](#)
- Dilate3x3Border, [1600](#)
- Dilation, [1579](#)

Dilation with border control, [1586](#)

Div, [277](#), [2575](#)

Div_Round, [306](#), [2583](#)

DivC, [141](#), [2527](#)

DivCRev, [2534](#)

Dot Product, [2790](#)

DotProd, [2043](#)

Duplicate Channel, [928](#)

Erode, [1607](#)

Erode3x3, [1622](#)

Erode3x3Border, [1628](#)

Erosion with border control, [1614](#)

Exp, [364](#), [2604](#)

Filtering Functions, [960](#), [2683](#)

Fixed Filters, [1348](#)

fixed_filters

 nppiFilterPrewittHoriz_16s_AC4R, [1358](#)

 nppiFilterPrewittHoriz_16s_C1R, [1358](#)

 nppiFilterPrewittHoriz_16s_C3R, [1359](#)

 nppiFilterPrewittHoriz_16s_C4R, [1359](#)

 nppiFilterPrewittHoriz_32f_AC4R, [1359](#)

 nppiFilterPrewittHoriz_32f_C1R, [1360](#)

 nppiFilterPrewittHoriz_32f_C3R, [1360](#)

 nppiFilterPrewittHoriz_32f_C4R, [1360](#)

 nppiFilterPrewittHoriz_8u_AC4R, [1361](#)

 nppiFilterPrewittHoriz_8u_C1R, [1361](#)

 nppiFilterPrewittHoriz_8u_C3R, [1361](#)

 nppiFilterPrewittHoriz_8u_C4R, [1362](#)

 nppiFilterPrewittHorizBorder_16s_AC4R,
 [1362](#)

 nppiFilterPrewittHorizBorder_16s_C1R, [1363](#)

 nppiFilterPrewittHorizBorder_16s_C3R, [1363](#)

 nppiFilterPrewittHorizBorder_16s_C4R, [1363](#)

 nppiFilterPrewittHorizBorder_32f_AC4R,
 [1364](#)

 nppiFilterPrewittHorizBorder_32f_C1R, [1364](#)

 nppiFilterPrewittHorizBorder_32f_C3R, [1365](#)

 nppiFilterPrewittHorizBorder_32f_C4R, [1365](#)

 nppiFilterPrewittHorizBorder_8u_AC4R,
 [1366](#)

 nppiFilterPrewittHorizBorder_8u_C1R, [1366](#)

 nppiFilterPrewittHorizBorder_8u_C3R, [1366](#)

 nppiFilterPrewittHorizBorder_8u_C4R, [1367](#)

 nppiFilterPrewittVert_16s_AC4R, [1367](#)

 nppiFilterPrewittVert_16s_C1R, [1368](#)

 nppiFilterPrewittVert_16s_C3R, [1368](#)

 nppiFilterPrewittVert_16s_C4R, [1368](#)

 nppiFilterPrewittVert_32f_AC4R, [1369](#)

 nppiFilterPrewittVert_32f_C1R, [1369](#)

 nppiFilterPrewittVert_32f_C3R, [1369](#)

 nppiFilterPrewittVert_32f_C4R, [1370](#)

 nppiFilterPrewittVert_8u_AC4R, [1370](#)

 nppiFilterPrewittVert_8u_C1R, [1370](#)

 nppiFilterPrewittVert_8u_C3R, [1371](#)

 nppiFilterPrewittVert_8u_C4R, [1371](#)

 nppiFilterPrewittVertBorder_16s_AC4R, [1371](#)

 nppiFilterPrewittVertBorder_16s_C1R, [1372](#)

 nppiFilterPrewittVertBorder_16s_C3R, [1372](#)

 nppiFilterPrewittVertBorder_16s_C4R, [1373](#)

 nppiFilterPrewittVertBorder_32f_AC4R, [1373](#)

 nppiFilterPrewittVertBorder_32f_C1R, [1374](#)

 nppiFilterPrewittVertBorder_32f_C3R, [1374](#)

 nppiFilterPrewittVertBorder_32f_C4R, [1374](#)

 nppiFilterPrewittVertBorder_8u_AC4R, [1375](#)

 nppiFilterPrewittVertBorder_8u_C1R, [1375](#)

 nppiFilterPrewittVertBorder_8u_C3R, [1376](#)

 nppiFilterPrewittVertBorder_8u_C4R, [1376](#)

 nppiFilterScharrHoriz_32f_C1R, [1377](#)

 nppiFilterScharrHoriz_8s16s_C1R, [1377](#)

 nppiFilterScharrHoriz_8u16s_C1R, [1377](#)

 nppiFilterScharrHorizBorder_32f_C1R, [1378](#)

 nppiFilterScharrHorizBorder_8s16s_C1R,
 [1378](#)

 nppiFilterScharrHorizBorder_8u16s_C1R,
 [1379](#)

 nppiFilterScharrVert_32f_C1R, [1379](#)

 nppiFilterScharrVert_8s16s_C1R, [1379](#)

 nppiFilterScharrVert_8u16s_C1R, [1380](#)

 nppiFilterScharrVertBorder_32f_C1R, [1380](#)

 nppiFilterScharrVertBorder_8s16s_C1R, [1381](#)

 nppiFilterScharrVertBorder_8u16s_C1R, [1381](#)

 nppiFilterSobelHoriz_16s_AC4R, [1381](#)

 nppiFilterSobelHoriz_16s_C1R, [1382](#)

 nppiFilterSobelHoriz_16s_C3R, [1382](#)

 nppiFilterSobelHoriz_16s_C4R, [1382](#)

 nppiFilterSobelHoriz_32f_AC4R, [1383](#)

 nppiFilterSobelHoriz_32f_C1R, [1383](#)

 nppiFilterSobelHoriz_32f_C3R, [1383](#)

 nppiFilterSobelHoriz_32f_C4R, [1384](#)

 nppiFilterSobelHoriz_8s16s_C1R, [1384](#)

 nppiFilterSobelHoriz_8u16s_C1R, [1384](#)

 nppiFilterSobelHoriz_8u_AC4R, [1385](#)

 nppiFilterSobelHoriz_8u_C1R, [1385](#)

 nppiFilterSobelHoriz_8u_C3R, [1385](#)

 nppiFilterSobelHoriz_8u_C4R, [1386](#)

 nppiFilterSobelHorizMask_32f_C1R, [1386](#)

 nppiFilterSobelHorizSecond_32f_C1R, [1386](#)

 nppiFilterSobelHorizSecond_8s16s_C1R,
 [1387](#)

 nppiFilterSobelHorizSecond_8u16s_C1R,
 [1387](#)

 nppiFilterSobelVert_16s_AC4R, [1388](#)

 nppiFilterSobelVert_16s_C1R, [1388](#)

 nppiFilterSobelVert_16s_C3R, [1388](#)

 nppiFilterSobelVert_16s_C4R, [1389](#)

 nppiFilterSobelVert_32f_AC4R, [1389](#)

- npfiFilterSobelVert_32f_C1R, 1389
 - npfiFilterSobelVert_32f_C3R, 1390
 - npfiFilterSobelVert_32f_C4R, 1390
 - npfiFilterSobelVert_8s16s_C1R, 1390
 - npfiFilterSobelVert_8u16s_C1R, 1391
 - npfiFilterSobelVert_8u_AC4R, 1391
 - npfiFilterSobelVert_8u_C1R, 1391
 - npfiFilterSobelVert_8u_C3R, 1392
 - npfiFilterSobelVert_8u_C4R, 1392
 - npfiFilterSobelVertMask_32f_C1R, 1392
- Fourier Transforms, 1576
- Free, 2865
- Geometry Transforms, 1394
- GraphCut, 731
- haarBuffer
 - NppiHaarBuffer, 2871
- haarBufferSize
 - NppiHaarBuffer, 2871
- height
 - NppiRect, 2874
 - NppiSize, 2875
- HistogramEven, 2096
- HistogramRange, 2109
- im
 - NPP_ALIGN_16, 2867
 - NPP_ALIGN_8, 2869
- Image Norms, 1839
- Image Proximity, 2125
- Image Quality Index, 2257
- image_1D_linear_filter
 - npfiFilterColumn32f_16s_AC4R, 1120
 - npfiFilterColumn32f_16s_C1R, 1120
 - npfiFilterColumn32f_16s_C3R, 1121
 - npfiFilterColumn32f_16s_C4R, 1121
 - npfiFilterColumn32f_16u_AC4R, 1122
 - npfiFilterColumn32f_16u_C1R, 1122
 - npfiFilterColumn32f_16u_C3R, 1123
 - npfiFilterColumn32f_16u_C4R, 1123
 - npfiFilterColumn32f_8u_AC4R, 1124
 - npfiFilterColumn32f_8u_C1R, 1124
 - npfiFilterColumn32f_8u_C3R, 1125
 - npfiFilterColumn32f_8u_C4R, 1125
 - npfiFilterColumn_16s_AC4R, 1126
 - npfiFilterColumn_16s_C1R, 1126
 - npfiFilterColumn_16s_C3R, 1127
 - npfiFilterColumn_16s_C4R, 1127
 - npfiFilterColumn_16u_AC4R, 1128
 - npfiFilterColumn_16u_C1R, 1128
 - npfiFilterColumn_16u_C3R, 1129
 - npfiFilterColumn_16u_C4R, 1129
 - npfiFilterColumn_32f_AC4R, 1130
 - npfiFilterColumn_32f_C1R, 1130
 - npfiFilterColumn_32f_C3R, 1131
 - npfiFilterColumn_32f_C4R, 1131
 - npfiFilterColumn_64f_C1R, 1132
 - npfiFilterColumn_8u_AC4R, 1132
 - npfiFilterColumn_8u_C1R, 1133
 - npfiFilterColumn_8u_C3R, 1133
 - npfiFilterColumn_8u_C4R, 1134
 - npfiFilterColumnBorder32f_16s_AC4R, 1134
 - npfiFilterColumnBorder32f_16s_C1R, 1135
 - npfiFilterColumnBorder32f_16s_C3R, 1135
 - npfiFilterColumnBorder32f_16s_C4R, 1136
 - npfiFilterColumnBorder32f_16u_AC4R, 1136
 - npfiFilterColumnBorder32f_16u_C1R, 1137
 - npfiFilterColumnBorder32f_16u_C3R, 1137
 - npfiFilterColumnBorder32f_16u_C4R, 1138
 - npfiFilterColumnBorder32f_8u_AC4R, 1138
 - npfiFilterColumnBorder32f_8u_C1R, 1139
 - npfiFilterColumnBorder32f_8u_C3R, 1139
 - npfiFilterColumnBorder32f_8u_C4R, 1140
 - npfiFilterColumnBorder_16s_AC4R, 1140
 - npfiFilterColumnBorder_16s_C1R, 1141
 - npfiFilterColumnBorder_16s_C3R, 1142
 - npfiFilterColumnBorder_16s_C4R, 1142
 - npfiFilterColumnBorder_16u_AC4R, 1143
 - npfiFilterColumnBorder_16u_C1R, 1143
 - npfiFilterColumnBorder_16u_C3R, 1144
 - npfiFilterColumnBorder_16u_C4R, 1145
 - npfiFilterColumnBorder_32f_AC4R, 1145
 - npfiFilterColumnBorder_32f_C1R, 1146
 - npfiFilterColumnBorder_32f_C3R, 1146
 - npfiFilterColumnBorder_32f_C4R, 1147
 - npfiFilterColumnBorder_8u_AC4R, 1147
 - npfiFilterColumnBorder_8u_C1R, 1148
 - npfiFilterColumnBorder_8u_C3R, 1149
 - npfiFilterColumnBorder_8u_C4R, 1149
 - npfiFilterRow32f_16s_AC4R, 1150
 - npfiFilterRow32f_16s_C1R, 1150
 - npfiFilterRow32f_16s_C3R, 1151
 - npfiFilterRow32f_16s_C4R, 1151
 - npfiFilterRow32f_16u_AC4R, 1152
 - npfiFilterRow32f_16u_C1R, 1152
 - npfiFilterRow32f_16u_C3R, 1153
 - npfiFilterRow32f_16u_C4R, 1153
 - npfiFilterRow32f_8u_AC4R, 1153
 - npfiFilterRow32f_8u_C1R, 1154
 - npfiFilterRow32f_8u_C3R, 1154
 - npfiFilterRow32f_8u_C4R, 1155
 - npfiFilterRow_16s_AC4R, 1155
 - npfiFilterRow_16s_C1R, 1156
 - npfiFilterRow_16s_C3R, 1156
 - npfiFilterRow_16s_C4R, 1157
 - npfiFilterRow_16u_AC4R, 1157

- [nppiFilterRow_16u_C1R](#), [1158](#)
- [nppiFilterRow_16u_C3R](#), [1158](#)
- [nppiFilterRow_16u_C4R](#), [1159](#)
- [nppiFilterRow_32f_AC4R](#), [1159](#)
- [nppiFilterRow_32f_C1R](#), [1160](#)
- [nppiFilterRow_32f_C3R](#), [1160](#)
- [nppiFilterRow_32f_C4R](#), [1161](#)
- [nppiFilterRow_64f_C1R](#), [1161](#)
- [nppiFilterRow_8u_AC4R](#), [1162](#)
- [nppiFilterRow_8u_C1R](#), [1162](#)
- [nppiFilterRow_8u_C3R](#), [1163](#)
- [nppiFilterRow_8u_C4R](#), [1163](#)
- [nppiFilterRowBorder32f_16s_AC4R](#), [1164](#)
- [nppiFilterRowBorder32f_16s_C1R](#), [1164](#)
- [nppiFilterRowBorder32f_16s_C3R](#), [1165](#)
- [nppiFilterRowBorder32f_16s_C4R](#), [1165](#)
- [nppiFilterRowBorder32f_16u_AC4R](#), [1166](#)
- [nppiFilterRowBorder32f_16u_C1R](#), [1166](#)
- [nppiFilterRowBorder32f_16u_C3R](#), [1167](#)
- [nppiFilterRowBorder32f_16u_C4R](#), [1167](#)
- [nppiFilterRowBorder32f_8u_AC4R](#), [1168](#)
- [nppiFilterRowBorder32f_8u_C1R](#), [1168](#)
- [nppiFilterRowBorder32f_8u_C3R](#), [1169](#)
- [nppiFilterRowBorder32f_8u_C4R](#), [1169](#)
- [nppiFilterRowBorder_16s_AC4R](#), [1170](#)
- [nppiFilterRowBorder_16s_C1R](#), [1171](#)
- [nppiFilterRowBorder_16s_C3R](#), [1171](#)
- [nppiFilterRowBorder_16s_C4R](#), [1172](#)
- [nppiFilterRowBorder_16u_AC4R](#), [1172](#)
- [nppiFilterRowBorder_16u_C1R](#), [1173](#)
- [nppiFilterRowBorder_16u_C3R](#), [1174](#)
- [nppiFilterRowBorder_16u_C4R](#), [1174](#)
- [nppiFilterRowBorder_32f_AC4R](#), [1175](#)
- [nppiFilterRowBorder_32f_C1R](#), [1175](#)
- [nppiFilterRowBorder_32f_C3R](#), [1176](#)
- [nppiFilterRowBorder_32f_C4R](#), [1176](#)
- [nppiFilterRowBorder_8u_AC4R](#), [1177](#)
- [nppiFilterRowBorder_8u_C1R](#), [1178](#)
- [nppiFilterRowBorder_8u_C3R](#), [1178](#)
- [nppiFilterRowBorder_8u_C4R](#), [1179](#)
- [nppiFilterSobelCross_32f_C1R](#), [1179](#)
- [nppiFilterSobelCross_8s16s_C1R](#), [1180](#)
- [nppiFilterSobelCross_8u16s_C1R](#), [1180](#)
- [nppiFilterSobelHorizBorder_16s_AC4R](#), [1180](#)
- [nppiFilterSobelHorizBorder_16s_C1R](#), [1181](#)
- [nppiFilterSobelHorizBorder_16s_C3R](#), [1181](#)
- [nppiFilterSobelHorizBorder_16s_C4R](#), [1182](#)
- [nppiFilterSobelHorizBorder_32f_AC4R](#), [1182](#)
- [nppiFilterSobelHorizBorder_32f_C1R](#), [1183](#)
- [nppiFilterSobelHorizBorder_32f_C3R](#), [1183](#)
- [nppiFilterSobelHorizBorder_32f_C4R](#), [1183](#)
- [nppiFilterSobelHorizBorder_8s16s_C1R](#), [1184](#)
- [nppiFilterSobelHorizBorder_8u16s_C1R](#), [1184](#)
- [nppiFilterSobelHorizBorder_8u_AC4R](#), [1185](#)
- [nppiFilterSobelHorizBorder_8u_C1R](#), [1185](#)
- [nppiFilterSobelHorizBorder_8u_C3R](#), [1186](#)
- [nppiFilterSobelHorizBorder_8u_C4R](#), [1186](#)
- [nppiFilterSobelHorizMaskBorder_32f_C1R](#), [1187](#)
- [nppiFilterSobelHorizSecondBorder_32f_C1R](#), [1187](#)
- [nppiFilterSobelHorizSecondBorder_8s16s_C1R](#), [1188](#)
- [nppiFilterSobelHorizSecondBorder_8u16s_C1R](#), [1188](#)
- [nppiFilterSobelVertBorder_16s_AC4R](#), [1189](#)
- [nppiFilterSobelVertBorder_16s_C1R](#), [1189](#)
- [nppiFilterSobelVertBorder_16s_C3R](#), [1189](#)
- [nppiFilterSobelVertBorder_16s_C4R](#), [1190](#)
- [nppiFilterSobelVertBorder_32f_AC4R](#), [1190](#)
- [nppiFilterSobelVertBorder_32f_C1R](#), [1191](#)
- [nppiFilterSobelVertBorder_32f_C3R](#), [1191](#)
- [nppiFilterSobelVertBorder_32f_C4R](#), [1192](#)
- [nppiFilterSobelVertBorder_8s16s_C1R](#), [1192](#)
- [nppiFilterSobelVertBorder_8u16s_C1R](#), [1192](#)
- [nppiFilterSobelVertBorder_8u_AC4R](#), [1193](#)
- [nppiFilterSobelVertBorder_8u_C1R](#), [1193](#)
- [nppiFilterSobelVertBorder_8u_C3R](#), [1194](#)
- [nppiFilterSobelVertBorder_8u_C4R](#), [1194](#)
- [nppiFilterSobelVertMaskBorder_32f_C1R](#), [1195](#)
- [nppiFilterSobelVertSecond_32f_C1R](#), [1195](#)
- [nppiFilterSobelVertSecond_8s16s_C1R](#), [1196](#)
- [nppiFilterSobelVertSecond_8u16s_C1R](#), [1196](#)
- [image_1D_window_sum](#)
 - [nppiSumWindowColumn_16s32f_C1R](#), [1198](#)
 - [nppiSumWindowColumn_16s32f_C3R](#), [1199](#)
 - [nppiSumWindowColumn_16s32f_C4R](#), [1199](#)
 - [nppiSumWindowColumn_16u32f_C1R](#), [1200](#)
 - [nppiSumWindowColumn_16u32f_C3R](#), [1200](#)
 - [nppiSumWindowColumn_16u32f_C4R](#), [1201](#)
 - [nppiSumWindowColumn_8u32f_C1R](#), [1201](#)
 - [nppiSumWindowColumn_8u32f_C3R](#), [1201](#)
 - [nppiSumWindowColumn_8u32f_C4R](#), [1202](#)
 - [nppiSumWindowRow_16s32f_C1R](#), [1202](#)
 - [nppiSumWindowRow_16s32f_C3R](#), [1203](#)
 - [nppiSumWindowRow_16s32f_C4R](#), [1203](#)
 - [nppiSumWindowRow_16u32f_C1R](#), [1204](#)
 - [nppiSumWindowRow_16u32f_C3R](#), [1204](#)
 - [nppiSumWindowRow_16u32f_C4R](#), [1205](#)
 - [nppiSumWindowRow_8u32f_C1R](#), [1205](#)
 - [nppiSumWindowRow_8u32f_C3R](#), [1206](#)
 - [nppiSumWindowRow_8u32f_C4R](#), [1206](#)
- [image_1D_window_sum_border](#)

- nppiSumWindowColumnBorder_16s32f_C1R, [1210](#)
- nppiSumWindowColumnBorder_16s32f_C3R, [1210](#)
- nppiSumWindowColumnBorder_16s32f_C4R, [1211](#)
- nppiSumWindowColumnBorder_16u32f_C1R, [1211](#)
- nppiSumWindowColumnBorder_16u32f_C3R, [1212](#)
- nppiSumWindowColumnBorder_16u32f_C4R, [1212](#)
- nppiSumWindowColumnBorder_8u32f_C1R, [1213](#)
- nppiSumWindowColumnBorder_8u32f_C3R, [1214](#)
- nppiSumWindowColumnBorder_8u32f_C4R, [1214](#)
- nppiSumWindowRowBorder_16s32f_C1R, [1215](#)
- nppiSumWindowRowBorder_16s32f_C3R, [1215](#)
- nppiSumWindowRowBorder_16s32f_C4R, [1216](#)
- nppiSumWindowRowBorder_16u32f_C1R, [1216](#)
- nppiSumWindowRowBorder_16u32f_C3R, [1217](#)
- nppiSumWindowRowBorder_16u32f_C4R, [1218](#)
- nppiSumWindowRowBorder_8u32f_C1R, [1218](#)
- nppiSumWindowRowBorder_8u32f_C3R, [1219](#)
- nppiSumWindowRowBorder_8u32f_C4R, [1219](#)
- image_2D_fixed_linear_filters
 - nppiFilterBox_16s_AC4R, [1283](#)
 - nppiFilterBox_16s_C1R, [1283](#)
 - nppiFilterBox_16s_C3R, [1283](#)
 - nppiFilterBox_16s_C4R, [1284](#)
 - nppiFilterBox_16u_AC4R, [1284](#)
 - nppiFilterBox_16u_C1R, [1285](#)
 - nppiFilterBox_16u_C3R, [1285](#)
 - nppiFilterBox_16u_C4R, [1285](#)
 - nppiFilterBox_32f_AC4R, [1286](#)
 - nppiFilterBox_32f_C1R, [1286](#)
 - nppiFilterBox_32f_C3R, [1287](#)
 - nppiFilterBox_32f_C4R, [1287](#)
 - nppiFilterBox_64f_C1R, [1287](#)
 - nppiFilterBox_8u_AC4R, [1288](#)
 - nppiFilterBox_8u_C1R, [1288](#)
 - nppiFilterBox_8u_C3R, [1289](#)
 - nppiFilterBox_8u_C4R, [1289](#)
- nppiFilterBoxBorder_16s_AC4R, [1289](#)
- nppiFilterBoxBorder_16s_C1R, [1290](#)
- nppiFilterBoxBorder_16s_C3R, [1290](#)
- nppiFilterBoxBorder_16s_C4R, [1291](#)
- nppiFilterBoxBorder_16u_AC4R, [1291](#)
- nppiFilterBoxBorder_16u_C1R, [1292](#)
- nppiFilterBoxBorder_16u_C3R, [1292](#)
- nppiFilterBoxBorder_16u_C4R, [1293](#)
- nppiFilterBoxBorder_32f_AC4R, [1293](#)
- nppiFilterBoxBorder_32f_C1R, [1294](#)
- nppiFilterBoxBorder_32f_C3R, [1294](#)
- nppiFilterBoxBorder_32f_C4R, [1295](#)
- nppiFilterBoxBorder_8u_AC4R, [1295](#)
- nppiFilterBoxBorder_8u_C1R, [1296](#)
- nppiFilterBoxBorder_8u_C3R, [1296](#)
- nppiFilterBoxBorder_8u_C4R, [1297](#)
- image_abs
 - nppiAbs_16s_AC4IR, [322](#)
 - nppiAbs_16s_AC4R, [322](#)
 - nppiAbs_16s_C1IR, [322](#)
 - nppiAbs_16s_C1R, [323](#)
 - nppiAbs_16s_C3IR, [323](#)
 - nppiAbs_16s_C3R, [323](#)
 - nppiAbs_16s_C4IR, [324](#)
 - nppiAbs_16s_C4R, [324](#)
 - nppiAbs_32f_AC4IR, [324](#)
 - nppiAbs_32f_AC4R, [325](#)
 - nppiAbs_32f_C1IR, [325](#)
 - nppiAbs_32f_C1R, [325](#)
 - nppiAbs_32f_C3IR, [326](#)
 - nppiAbs_32f_C3R, [326](#)
 - nppiAbs_32f_C4IR, [326](#)
 - nppiAbs_32f_C4R, [327](#)
- image_absdiff
 - nppiAbsDiff_16u_C1R, [328](#)
 - nppiAbsDiff_32f_C1R, [329](#)
 - nppiAbsDiff_8u_C1R, [329](#)
 - nppiAbsDiff_8u_C3R, [329](#)
 - nppiAbsDiff_8u_C4R, [330](#)
- image_absdiffc
 - nppiAbsDiffC_16u_C1R, [167](#)
 - nppiAbsDiffC_32f_C1R, [167](#)
 - nppiAbsDiffC_8u_C1R, [168](#)
- image_add
 - nppiAdd_16s_AC4RSfs, [174](#)
 - nppiAdd_16s_AC4RSfs, [174](#)
 - nppiAdd_16s_C1RSfs, [175](#)
 - nppiAdd_16s_C1RSfs, [175](#)
 - nppiAdd_16s_C3RSfs, [176](#)
 - nppiAdd_16s_C3RSfs, [176](#)
 - nppiAdd_16s_C4RSfs, [176](#)
 - nppiAdd_16s_C4RSfs, [177](#)
 - nppiAdd_16sc_AC4RSfs, [177](#)
 - nppiAdd_16sc_AC4RSfs, [178](#)

- nppiAdd_16sc_C1IRSfs, 178
- nppiAdd_16sc_C1RSfs, 178
- nppiAdd_16sc_C3IRSfs, 179
- nppiAdd_16sc_C3RSfs, 179
- nppiAdd_16u_AC4IRSfs, 180
- nppiAdd_16u_AC4RSfs, 180
- nppiAdd_16u_C1IRSfs, 181
- nppiAdd_16u_C1RSfs, 181
- nppiAdd_16u_C3IRSfs, 181
- nppiAdd_16u_C3RSfs, 182
- nppiAdd_16u_C4IRSfs, 182
- nppiAdd_16u_C4RSfs, 183
- nppiAdd_32f_AC4IR, 183
- nppiAdd_32f_AC4R, 183
- nppiAdd_32f_C1IR, 184
- nppiAdd_32f_C1R, 184
- nppiAdd_32f_C3IR, 185
- nppiAdd_32f_C3R, 185
- nppiAdd_32f_C4IR, 185
- nppiAdd_32f_C4R, 186
- nppiAdd_32fc_AC4IR, 186
- nppiAdd_32fc_AC4R, 186
- nppiAdd_32fc_C1IR, 187
- nppiAdd_32fc_C1R, 187
- nppiAdd_32fc_C3IR, 188
- nppiAdd_32fc_C3R, 188
- nppiAdd_32fc_C4IR, 188
- nppiAdd_32fc_C4R, 189
- nppiAdd_32s_C1IRSfs, 189
- nppiAdd_32s_C1R, 190
- nppiAdd_32s_C1RSfs, 190
- nppiAdd_32s_C3IRSfs, 190
- nppiAdd_32s_C3RSfs, 191
- nppiAdd_32sc_AC4IRSfs, 191
- nppiAdd_32sc_AC4RSfs, 192
- nppiAdd_32sc_C1IRSfs, 192
- nppiAdd_32sc_C1RSfs, 192
- nppiAdd_32sc_C3IRSfs, 193
- nppiAdd_32sc_C3RSfs, 193
- nppiAdd_8u_AC4IRSfs, 194
- nppiAdd_8u_AC4RSfs, 194
- nppiAdd_8u_C1IRSfs, 195
- nppiAdd_8u_C1RSfs, 195
- nppiAdd_8u_C3IRSfs, 195
- nppiAdd_8u_C3RSfs, 196
- nppiAdd_8u_C4IRSfs, 196
- nppiAdd_8u_C4RSfs, 197
- image_addc
 - nppiAddC_16s_AC4IRSfs, 61
 - nppiAddC_16s_AC4RSfs, 61
 - nppiAddC_16s_C1IRSfs, 61
 - nppiAddC_16s_C1RSfs, 62
 - nppiAddC_16s_C3IRSfs, 62
 - nppiAddC_16s_C3RSfs, 62
- nppiAddC_16s_C4IRSfs, 63
- nppiAddC_16s_C4RSfs, 63
- nppiAddC_16sc_AC4IRSfs, 64
- nppiAddC_16sc_AC4RSfs, 64
- nppiAddC_16sc_C1IRSfs, 64
- nppiAddC_16sc_C1RSfs, 65
- nppiAddC_16sc_C3IRSfs, 65
- nppiAddC_16sc_C3RSfs, 66
- nppiAddC_16u_AC4IRSfs, 66
- nppiAddC_16u_AC4RSfs, 66
- nppiAddC_16u_C1IRSfs, 67
- nppiAddC_16u_C1RSfs, 67
- nppiAddC_16u_C3IRSfs, 68
- nppiAddC_16u_C3RSfs, 68
- nppiAddC_16u_C4IRSfs, 68
- nppiAddC_16u_C4RSfs, 69
- nppiAddC_32f_AC4IR, 69
- nppiAddC_32f_AC4R, 69
- nppiAddC_32f_C1IR, 70
- nppiAddC_32f_C1R, 70
- nppiAddC_32f_C3IR, 70
- nppiAddC_32f_C3R, 71
- nppiAddC_32f_C4IR, 71
- nppiAddC_32f_C4R, 71
- nppiAddC_32fc_AC4IR, 72
- nppiAddC_32fc_AC4R, 72
- nppiAddC_32fc_C1IR, 72
- nppiAddC_32fc_C1R, 73
- nppiAddC_32fc_C3IR, 73
- nppiAddC_32fc_C3R, 73
- nppiAddC_32fc_C4IR, 74
- nppiAddC_32fc_C4R, 74
- nppiAddC_32s_C1IRSfs, 75
- nppiAddC_32s_C1RSfs, 75
- nppiAddC_32s_C3IRSfs, 75
- nppiAddC_32s_C3RSfs, 76
- nppiAddC_32sc_AC4IRSfs, 76
- nppiAddC_32sc_AC4RSfs, 76
- nppiAddC_32sc_C1IRSfs, 77
- nppiAddC_32sc_C1RSfs, 77
- nppiAddC_32sc_C3IRSfs, 78
- nppiAddC_32sc_C3RSfs, 78
- nppiAddC_8u_AC4IRSfs, 78
- nppiAddC_8u_AC4RSfs, 79
- nppiAddC_8u_C1IRSfs, 79
- nppiAddC_8u_C1RSfs, 80
- nppiAddC_8u_C3IRSfs, 80
- nppiAddC_8u_C3RSfs, 80
- nppiAddC_8u_C4IRSfs, 81
- nppiAddC_8u_C4RSfs, 81
- image_addproduct
 - nppiAddProduct_16u32f_C1IMR, 201
 - nppiAddProduct_16u32f_C1IR, 202
 - nppiAddProduct_32f_C1IMR, 202

- nppiAddProduct_32f_C1IR, 203
- nppiAddProduct_8u32f_C1IMR, 203
- nppiAddProduct_8u32f_C1IR, 203
- image_addsquare
 - nppiAddSquare_16u32f_C1IMR, 198
 - nppiAddSquare_16u32f_C1IR, 199
 - nppiAddSquare_32f_C1IMR, 199
 - nppiAddSquare_32f_C1IR, 199
 - nppiAddSquare_8u32f_C1IMR, 200
 - nppiAddSquare_8u32f_C1IR, 200
- image_addweighted
 - nppiAddWeighted_16u32f_C1IMR, 205
 - nppiAddWeighted_16u32f_C1IR, 206
 - nppiAddWeighted_32f_C1IMR, 206
 - nppiAddWeighted_32f_C1IR, 207
 - nppiAddWeighted_8u32f_C1IMR, 207
 - nppiAddWeighted_8u32f_C1IR, 207
- image_affine_transform
 - nppiGetAffineBound, 1488
 - nppiGetAffineQuad, 1488
 - nppiGetAffineTransform, 1489
 - nppiWarpAffine_16u_AC4R, 1489
 - nppiWarpAffine_16u_C1R, 1490
 - nppiWarpAffine_16u_C3R, 1490
 - nppiWarpAffine_16u_C4R, 1491
 - nppiWarpAffine_16u_P3R, 1491
 - nppiWarpAffine_16u_P4R, 1492
 - nppiWarpAffine_32f_AC4R, 1492
 - nppiWarpAffine_32f_C1R, 1493
 - nppiWarpAffine_32f_C3R, 1493
 - nppiWarpAffine_32f_C4R, 1494
 - nppiWarpAffine_32f_P3R, 1494
 - nppiWarpAffine_32f_P4R, 1495
 - nppiWarpAffine_32s_AC4R, 1495
 - nppiWarpAffine_32s_C1R, 1496
 - nppiWarpAffine_32s_C3R, 1496
 - nppiWarpAffine_32s_C4R, 1497
 - nppiWarpAffine_32s_P3R, 1497
 - nppiWarpAffine_32s_P4R, 1498
 - nppiWarpAffine_64f_AC4R, 1498
 - nppiWarpAffine_64f_C1R, 1499
 - nppiWarpAffine_64f_C3R, 1499
 - nppiWarpAffine_64f_C4R, 1500
 - nppiWarpAffine_64f_P3R, 1500
 - nppiWarpAffine_64f_P4R, 1501
 - nppiWarpAffine_8u_AC4R, 1501
 - nppiWarpAffine_8u_C1R, 1502
 - nppiWarpAffine_8u_C3R, 1502
 - nppiWarpAffine_8u_C4R, 1503
 - nppiWarpAffine_8u_P3R, 1503
 - nppiWarpAffine_8u_P4R, 1504
 - nppiWarpAffineBack_16u_AC4R, 1504
 - nppiWarpAffineBack_16u_C1R, 1505
 - nppiWarpAffineBack_16u_C3R, 1505
 - nppiWarpAffineBack_16u_C4R, 1506
 - nppiWarpAffineBack_16u_P3R, 1506
 - nppiWarpAffineBack_16u_P4R, 1507
 - nppiWarpAffineBack_32f_AC4R, 1507
 - nppiWarpAffineBack_32f_C1R, 1508
 - nppiWarpAffineBack_32f_C3R, 1508
 - nppiWarpAffineBack_32f_C4R, 1509
 - nppiWarpAffineBack_32f_P3R, 1509
 - nppiWarpAffineBack_32f_P4R, 1510
 - nppiWarpAffineBack_32s_AC4R, 1510
 - nppiWarpAffineBack_32s_C1R, 1511
 - nppiWarpAffineBack_32s_C3R, 1511
 - nppiWarpAffineBack_32s_C4R, 1512
 - nppiWarpAffineBack_32s_P3R, 1512
 - nppiWarpAffineBack_32s_P4R, 1513
 - nppiWarpAffineBack_8u_AC4R, 1513
 - nppiWarpAffineBack_8u_C1R, 1514
 - nppiWarpAffineBack_8u_C3R, 1514
 - nppiWarpAffineBack_8u_C4R, 1515
 - nppiWarpAffineBack_8u_P3R, 1515
 - nppiWarpAffineBack_8u_P4R, 1516
 - nppiWarpAffineQuad_16u_AC4R, 1516
 - nppiWarpAffineQuad_16u_C1R, 1517
 - nppiWarpAffineQuad_16u_C3R, 1517
 - nppiWarpAffineQuad_16u_C4R, 1518
 - nppiWarpAffineQuad_16u_P3R, 1518
 - nppiWarpAffineQuad_16u_P4R, 1519
 - nppiWarpAffineQuad_32f_AC4R, 1519
 - nppiWarpAffineQuad_32f_C1R, 1520
 - nppiWarpAffineQuad_32f_C3R, 1520
 - nppiWarpAffineQuad_32f_C4R, 1521
 - nppiWarpAffineQuad_32f_P3R, 1521
 - nppiWarpAffineQuad_32f_P4R, 1522
 - nppiWarpAffineQuad_32s_AC4R, 1522
 - nppiWarpAffineQuad_32s_C1R, 1523
 - nppiWarpAffineQuad_32s_C3R, 1523
 - nppiWarpAffineQuad_32s_C4R, 1524
 - nppiWarpAffineQuad_32s_P3R, 1524
 - nppiWarpAffineQuad_32s_P4R, 1525
 - nppiWarpAffineQuad_8u_AC4R, 1525
 - nppiWarpAffineQuad_8u_C1R, 1526
 - nppiWarpAffineQuad_8u_C3R, 1526
 - nppiWarpAffineQuad_8u_C4R, 1527
 - nppiWarpAffineQuad_8u_P3R, 1527
 - nppiWarpAffineQuad_8u_P4R, 1528
- image_alphacomp
 - nppiAlphaComp_16s_AC1R, 490
 - nppiAlphaComp_16u_AC1R, 490
 - nppiAlphaComp_16u_AC4R, 491
 - nppiAlphaComp_32f_AC1R, 491
 - nppiAlphaComp_32f_AC4R, 492
 - nppiAlphaComp_32s_AC1R, 492
 - nppiAlphaComp_32s_AC4R, 492
 - nppiAlphaComp_32u_AC1R, 493

- [nppiAlphaComp_32u_AC4R](#), [493](#)
 - [nppiAlphaComp_8s_AC1R](#), [494](#)
 - [nppiAlphaComp_8u_AC1R](#), [494](#)
 - [nppiAlphaComp_8u_AC4R](#), [495](#)
- [image_alphacompc](#)
 - [nppiAlphaCompC_16s_C1R](#), [475](#)
 - [nppiAlphaCompC_16u_AC4R](#), [475](#)
 - [nppiAlphaCompC_16u_C1R](#), [476](#)
 - [nppiAlphaCompC_16u_C3R](#), [476](#)
 - [nppiAlphaCompC_16u_C4R](#), [477](#)
 - [nppiAlphaCompC_32f_C1R](#), [477](#)
 - [nppiAlphaCompC_32s_C1R](#), [478](#)
 - [nppiAlphaCompC_32u_C1R](#), [478](#)
 - [nppiAlphaCompC_8s_C1R](#), [479](#)
 - [nppiAlphaCompC_8u_AC4R](#), [479](#)
 - [nppiAlphaCompC_8u_C1R](#), [480](#)
 - [nppiAlphaCompC_8u_C3R](#), [480](#)
 - [nppiAlphaCompC_8u_C4R](#), [481](#)
- [image_alphapremul](#)
 - [nppiAlphaPremul_16u_AC4IR](#), [496](#)
 - [nppiAlphaPremul_16u_AC4R](#), [496](#)
 - [nppiAlphaPremul_8u_AC4IR](#), [497](#)
 - [nppiAlphaPremul_8u_AC4R](#), [497](#)
- [image_alphapremulc](#)
 - [nppiAlphaPremulC_16u_AC4IR](#), [483](#)
 - [nppiAlphaPremulC_16u_AC4R](#), [483](#)
 - [nppiAlphaPremulC_16u_C1IR](#), [484](#)
 - [nppiAlphaPremulC_16u_C1R](#), [484](#)
 - [nppiAlphaPremulC_16u_C3IR](#), [484](#)
 - [nppiAlphaPremulC_16u_C3R](#), [485](#)
 - [nppiAlphaPremulC_16u_C4IR](#), [485](#)
 - [nppiAlphaPremulC_16u_C4R](#), [485](#)
 - [nppiAlphaPremulC_8u_AC4IR](#), [486](#)
 - [nppiAlphaPremulC_8u_AC4R](#), [486](#)
 - [nppiAlphaPremulC_8u_C1IR](#), [486](#)
 - [nppiAlphaPremulC_8u_C1R](#), [487](#)
 - [nppiAlphaPremulC_8u_C3IR](#), [487](#)
 - [nppiAlphaPremulC_8u_C3R](#), [487](#)
 - [nppiAlphaPremulC_8u_C4IR](#), [488](#)
 - [nppiAlphaPremulC_8u_C4R](#), [488](#)
- [image_and](#)
 - [nppiAnd_16u_AC4IR](#), [435](#)
 - [nppiAnd_16u_AC4R](#), [435](#)
 - [nppiAnd_16u_C1IR](#), [435](#)
 - [nppiAnd_16u_C1R](#), [436](#)
 - [nppiAnd_16u_C3IR](#), [436](#)
 - [nppiAnd_16u_C3R](#), [436](#)
 - [nppiAnd_16u_C4IR](#), [437](#)
 - [nppiAnd_16u_C4R](#), [437](#)
 - [nppiAnd_32s_AC4IR](#), [438](#)
 - [nppiAnd_32s_AC4R](#), [438](#)
 - [nppiAnd_32s_C1IR](#), [438](#)
 - [nppiAnd_32s_C1R](#), [439](#)
 - [nppiAnd_32s_C3IR](#), [439](#)
 - [nppiAnd_32s_C3R](#), [439](#)
 - [nppiAnd_32s_C4IR](#), [440](#)
 - [nppiAnd_32s_C4R](#), [440](#)
 - [nppiAnd_8u_AC4IR](#), [441](#)
 - [nppiAnd_8u_AC4R](#), [441](#)
 - [nppiAnd_8u_C1IR](#), [441](#)
 - [nppiAnd_8u_C1R](#), [442](#)
 - [nppiAnd_8u_C3IR](#), [442](#)
 - [nppiAnd_8u_C3R](#), [442](#)
 - [nppiAnd_8u_C4IR](#), [443](#)
 - [nppiAnd_8u_C4R](#), [443](#)
- [image_andc](#)
 - [nppiAndC_16u_AC4IR](#), [374](#)
 - [nppiAndC_16u_AC4R](#), [374](#)
 - [nppiAndC_16u_C1IR](#), [374](#)
 - [nppiAndC_16u_C1R](#), [375](#)
 - [nppiAndC_16u_C3IR](#), [375](#)
 - [nppiAndC_16u_C3R](#), [375](#)
 - [nppiAndC_16u_C4IR](#), [376](#)
 - [nppiAndC_16u_C4R](#), [376](#)
 - [nppiAndC_32s_AC4IR](#), [376](#)
 - [nppiAndC_32s_AC4R](#), [377](#)
 - [nppiAndC_32s_C1IR](#), [377](#)
 - [nppiAndC_32s_C1R](#), [377](#)
 - [nppiAndC_32s_C3IR](#), [378](#)
 - [nppiAndC_32s_C3R](#), [378](#)
 - [nppiAndC_32s_C4IR](#), [378](#)
 - [nppiAndC_32s_C4R](#), [379](#)
 - [nppiAndC_8u_AC4IR](#), [379](#)
 - [nppiAndC_8u_AC4R](#), [379](#)
 - [nppiAndC_8u_C1IR](#), [380](#)
 - [nppiAndC_8u_C1R](#), [380](#)
 - [nppiAndC_8u_C3IR](#), [380](#)
 - [nppiAndC_8u_C3R](#), [381](#)
 - [nppiAndC_8u_C4IR](#), [381](#)
 - [nppiAndC_8u_C4R](#), [381](#)
- [image_average_error](#)
 - [nppiAverageError_16s_C1R](#), [2292](#)
 - [nppiAverageError_16s_C2R](#), [2293](#)
 - [nppiAverageError_16s_C3R](#), [2293](#)
 - [nppiAverageError_16s_C4R](#), [2294](#)
 - [nppiAverageError_16sc_C1R](#), [2294](#)
 - [nppiAverageError_16sc_C2R](#), [2294](#)
 - [nppiAverageError_16sc_C3R](#), [2295](#)
 - [nppiAverageError_16sc_C4R](#), [2295](#)
 - [nppiAverageError_16u_C1R](#), [2296](#)
 - [nppiAverageError_16u_C2R](#), [2296](#)
 - [nppiAverageError_16u_C3R](#), [2297](#)
 - [nppiAverageError_16u_C4R](#), [2297](#)
 - [nppiAverageError_32f_C1R](#), [2297](#)
 - [nppiAverageError_32f_C2R](#), [2298](#)
 - [nppiAverageError_32f_C3R](#), [2298](#)
 - [nppiAverageError_32f_C4R](#), [2299](#)
 - [nppiAverageError_32fc_C1R](#), [2299](#)

- nppiAverageError_32fc_C2R, [2300](#)
- nppiAverageError_32fc_C3R, [2300](#)
- nppiAverageError_32fc_C4R, [2301](#)
- nppiAverageError_32s_C1R, [2301](#)
- nppiAverageError_32s_C2R, [2301](#)
- nppiAverageError_32s_C3R, [2302](#)
- nppiAverageError_32s_C4R, [2302](#)
- nppiAverageError_32sc_C1R, [2303](#)
- nppiAverageError_32sc_C2R, [2303](#)
- nppiAverageError_32sc_C3R, [2304](#)
- nppiAverageError_32sc_C4R, [2304](#)
- nppiAverageError_32u_C1R, [2304](#)
- nppiAverageError_32u_C2R, [2305](#)
- nppiAverageError_32u_C3R, [2305](#)
- nppiAverageError_32u_C4R, [2306](#)
- nppiAverageError_64f_C1R, [2306](#)
- nppiAverageError_64f_C2R, [2307](#)
- nppiAverageError_64f_C3R, [2307](#)
- nppiAverageError_64f_C4R, [2308](#)
- nppiAverageError_8s_C1R, [2308](#)
- nppiAverageError_8s_C2R, [2308](#)
- nppiAverageError_8s_C3R, [2309](#)
- nppiAverageError_8s_C4R, [2309](#)
- nppiAverageError_8u_C1R, [2310](#)
- nppiAverageError_8u_C2R, [2310](#)
- nppiAverageError_8u_C3R, [2311](#)
- nppiAverageError_8u_C4R, [2311](#)
- image_average_relative_error
 - nppiAverageRelativeError_16s_C1R, [2339](#)
 - nppiAverageRelativeError_16s_C2R, [2340](#)
 - nppiAverageRelativeError_16s_C3R, [2340](#)
 - nppiAverageRelativeError_16s_C4R, [2341](#)
 - nppiAverageRelativeError_16sc_C1R, [2341](#)
 - nppiAverageRelativeError_16sc_C2R, [2342](#)
 - nppiAverageRelativeError_16sc_C3R, [2342](#)
 - nppiAverageRelativeError_16sc_C4R, [2342](#)
 - nppiAverageRelativeError_16u_C1R, [2343](#)
 - nppiAverageRelativeError_16u_C2R, [2343](#)
 - nppiAverageRelativeError_16u_C3R, [2344](#)
 - nppiAverageRelativeError_16u_C4R, [2344](#)
 - nppiAverageRelativeError_32f_C1R, [2345](#)
 - nppiAverageRelativeError_32f_C2R, [2345](#)
 - nppiAverageRelativeError_32f_C3R, [2346](#)
 - nppiAverageRelativeError_32f_C4R, [2346](#)
 - nppiAverageRelativeError_32fc_C1R, [2347](#)
 - nppiAverageRelativeError_32fc_C2R, [2347](#)
 - nppiAverageRelativeError_32fc_C3R, [2347](#)
 - nppiAverageRelativeError_32fc_C4R, [2348](#)
 - nppiAverageRelativeError_32s_C1R, [2348](#)
 - nppiAverageRelativeError_32s_C2R, [2349](#)
 - nppiAverageRelativeError_32s_C3R, [2349](#)
 - nppiAverageRelativeError_32s_C4R, [2350](#)
 - nppiAverageRelativeError_32sc_C1R, [2350](#)
 - nppiAverageRelativeError_32sc_C2R, [2351](#)
- nppiAverageRelativeError_32sc_C3R, [2351](#)
- nppiAverageRelativeError_32sc_C4R, [2352](#)
- nppiAverageRelativeError_32u_C1R, [2352](#)
- nppiAverageRelativeError_32u_C2R, [2352](#)
- nppiAverageRelativeError_32u_C3R, [2353](#)
- nppiAverageRelativeError_32u_C4R, [2353](#)
- nppiAverageRelativeError_64f_C1R, [2354](#)
- nppiAverageRelativeError_64f_C2R, [2354](#)
- nppiAverageRelativeError_64f_C3R, [2355](#)
- nppiAverageRelativeError_64f_C4R, [2355](#)
- nppiAverageRelativeError_8s_C1R, [2356](#)
- nppiAverageRelativeError_8s_C2R, [2356](#)
- nppiAverageRelativeError_8s_C3R, [2357](#)
- nppiAverageRelativeError_8s_C4R, [2357](#)
- nppiAverageRelativeError_8u_C1R, [2357](#)
- nppiAverageRelativeError_8u_C2R, [2358](#)
- nppiAverageRelativeError_8u_C3R, [2358](#)
- nppiAverageRelativeError_8u_C4R, [2359](#)
- image_color_gamma_correction
 - nppiGammaFwd_8u_AC4IR, [615](#)
 - nppiGammaFwd_8u_AC4R, [615](#)
 - nppiGammaFwd_8u_C3IR, [615](#)
 - nppiGammaFwd_8u_C3R, [616](#)
 - nppiGammaFwd_8u_IP3R, [616](#)
 - nppiGammaFwd_8u_P3R, [616](#)
 - nppiGammaInv_8u_AC4IR, [617](#)
 - nppiGammaInv_8u_AC4R, [617](#)
 - nppiGammaInv_8u_C3IR, [617](#)
 - nppiGammaInv_8u_C3R, [618](#)
 - nppiGammaInv_8u_IP3R, [618](#)
 - nppiGammaInv_8u_P3R, [618](#)
- image_color_model_conversion
 - nppiBGRTToCbYCr422_709HDTV_8u_-AC4C2R, [527](#)
 - nppiBGRTToCbYCr422_709HDTV_8u_-C3C2R, [528](#)
 - nppiBGRTToCbYCr422_8u_AC4C2R, [528](#)
 - nppiBGRTToHLS_8u_AC4P4R, [528](#)
 - nppiBGRTToHLS_8u_AC4R, [529](#)
 - nppiBGRTToHLS_8u_AP4C4R, [529](#)
 - nppiBGRTToHLS_8u_AP4R, [529](#)
 - nppiBGRTToHLS_8u_C3P3R, [530](#)
 - nppiBGRTToHLS_8u_P3C3R, [530](#)
 - nppiBGRTToHLS_8u_P3R, [530](#)
 - nppiBGRTToLab_8u_C3R, [531](#)
 - nppiBGRTToYCbCr411_8u_AC4P3R, [531](#)
 - nppiBGRTToYCbCr411_8u_C3P3R, [531](#)
 - nppiBGRTToYCbCr420_709CSC_8u_-AC4P3R, [532](#)
 - nppiBGRTToYCbCr420_709CSC_8u_C3P3R, [532](#)
 - nppiBGRTToYCbCr420_709HDTV_8u_-AC4P3R, [533](#)
 - nppiBGRTToYCbCr420_8u_AC4P3R, [533](#)

- nppiBGRToYCbCr420_8u_C3P3R, [533](#)
nppiBGRToYCbCr422_8u_AC4C2R, [534](#)
nppiBGRToYCbCr422_8u_AC4P3R, [534](#)
nppiBGRToYCbCr422_8u_C3C2R, [535](#)
nppiBGRToYCbCr422_8u_C3P3R, [535](#)
nppiBGRToYCbCr_8u_AC4P3R, [535](#)
nppiBGRToYCbCr_8u_AC4P4R, [536](#)
nppiBGRToYCbCr_8u_C3P3R, [536](#)
nppiBGRToYCrCb420_709CSC_8u_-
AC4P3R, [537](#)
nppiBGRToYCrCb420_709CSC_8u_C3P3R,
[537](#)
nppiBGRToYCrCb420_8u_AC4P3R, [537](#)
nppiBGRToYCrCb420_8u_C3P3R, [538](#)
nppiBGRToYUV420_8u_AC4P3R, [538](#)
nppiBGRToYUV_8u_AC4P4R, [539](#)
nppiBGRToYUV_8u_AC4R, [539](#)
nppiBGRToYUV_8u_C3P3R, [539](#)
nppiBGRToYUV_8u_C3R, [540](#)
nppiBGRToYUV_8u_P3R, [540](#)
nppiCbYCr422ToBGR_709HDTV_8u_-
C2C3R, [540](#)
nppiCbYCr422ToBGR_709HDTV_8u_-
C2C4R, [541](#)
nppiCbYCr422ToBGR_8u_C2C4R, [541](#)
nppiCbYCr422ToRGB_8u_C2C3R, [542](#)
nppiCFAToRGB_16u_C1C3R, [542](#)
nppiCFAToRGB_8u_C1C3R, [542](#)
nppiCFAToRGBA_16u_C1AC4R, [543](#)
nppiCFAToRGBA_8u_C1AC4R, [543](#)
nppiColorToGray_16s_AC4C1R, [544](#)
nppiColorToGray_16s_C3C1R, [544](#)
nppiColorToGray_16s_C4C1R, [545](#)
nppiColorToGray_16u_AC4C1R, [545](#)
nppiColorToGray_16u_C3C1R, [545](#)
nppiColorToGray_16u_C4C1R, [546](#)
nppiColorToGray_32f_AC4C1R, [546](#)
nppiColorToGray_32f_C3C1R, [547](#)
nppiColorToGray_32f_C4C1R, [547](#)
nppiColorToGray_8u_AC4C1R, [547](#)
nppiColorToGray_8u_C3C1R, [548](#)
nppiColorToGray_8u_C4C1R, [548](#)
nppiHLSToBGR_8u_AC4P4R, [548](#)
nppiHLSToBGR_8u_AC4R, [549](#)
nppiHLSToBGR_8u_AP4C4R, [549](#)
nppiHLSToBGR_8u_AP4R, [549](#)
nppiHLSToBGR_8u_C3P3R, [550](#)
nppiHLSToBGR_8u_P3C3R, [550](#)
nppiHLSToBGR_8u_P3R, [550](#)
nppiHLSToRGB_8u_AC4R, [551](#)
nppiHLSToRGB_8u_C3R, [551](#)
nppiHSVToRGB_8u_AC4R, [551](#)
nppiHSVToRGB_8u_C3R, [552](#)
nppiLabToBGR_8u_C3R, [552](#)
nppiLUVToRGB_8u_AC4R, [552](#)
nppiLUVToRGB_8u_C3R, [553](#)
nppiNV21ToBGR_8u_P2C4R, [553](#)
nppiNV21ToRGB_8u_P2C4R, [553](#)
nppiRGBToCbYCr422_8u_C3C2R, [554](#)
nppiRGBToCbYCr422Gamma_8u_C3C2R,
[554](#)
nppiRGBToGray_16s_AC4C1R, [555](#)
nppiRGBToGray_16s_C3C1R, [555](#)
nppiRGBToGray_16u_AC4C1R, [555](#)
nppiRGBToGray_16u_C3C1R, [556](#)
nppiRGBToGray_32f_AC4C1R, [556](#)
nppiRGBToGray_32f_C3C1R, [556](#)
nppiRGBToGray_8u_AC4C1R, [557](#)
nppiRGBToGray_8u_C3C1R, [557](#)
nppiRGBToHLS_8u_AC4R, [557](#)
nppiRGBToHLS_8u_C3R, [558](#)
nppiRGBToHSV_8u_AC4R, [558](#)
nppiRGBToHSV_8u_C3R, [558](#)
nppiRGBToLUV_8u_AC4R, [559](#)
nppiRGBToLUV_8u_C3R, [559](#)
nppiRGBToXYZ_8u_AC4R, [559](#)
nppiRGBToXYZ_8u_C3R, [560](#)
nppiRGBToYCbCr420_8u_C3P3R, [560](#)
nppiRGBToYCbCr422_8u_C3C2R, [560](#)
nppiRGBToYCbCr422_8u_C3P3R, [561](#)
nppiRGBToYCbCr422_8u_P3C2R, [561](#)
nppiRGBToYCbCr_8u_AC4P3R, [562](#)
nppiRGBToYCbCr_8u_AC4R, [562](#)
nppiRGBToYCbCr_8u_C3P3R, [562](#)
nppiRGBToYCbCr_8u_C3R, [563](#)
nppiRGBToYCbCr_8u_P3R, [563](#)
nppiRGBToYCC_8u_AC4R, [563](#)
nppiRGBToYCC_8u_C3R, [564](#)
nppiRGBToYCrCb420_8u_AC4P3R, [564](#)
nppiRGBToYCrCb422_8u_C3C2R, [564](#)
nppiRGBToYCrCb422_8u_P3C2R, [565](#)
nppiRGBToYUV420_8u_C3P3R, [565](#)
nppiRGBToYUV420_8u_P3R, [565](#)
nppiRGBToYUV422_8u_C3C2R, [566](#)
nppiRGBToYUV422_8u_C3P3R, [566](#)
nppiRGBToYUV422_8u_P3R, [566](#)
nppiRGBToYUV_8u_AC4P4R, [567](#)
nppiRGBToYUV_8u_AC4R, [567](#)
nppiRGBToYUV_8u_C3P3R, [568](#)
nppiRGBToYUV_8u_C3R, [568](#)
nppiRGBToYUV_8u_P3R, [568](#)
nppiXYZToRGB_8u_AC4R, [569](#)
nppiXYZToRGB_8u_C3R, [569](#)
nppiYCbCr411ToBGR_8u_P3C3R, [569](#)
nppiYCbCr411ToBGR_8u_P3C4R, [570](#)
nppiYCbCr420ToBGR_709CSC_8u_P3C3R,
[570](#)

- nppiYCbCr420ToBGR_709HDTV_8u_-P3C4R, [570](#)
- nppiYCbCr420ToBGR_8u_P3C3R, [571](#)
- nppiYCbCr420ToBGR_8u_P3C4R, [571](#)
- nppiYCbCr420ToRGB_8u_P3C3R, [572](#)
- nppiYCbCr422ToBGR_8u_C2C3R, [572](#)
- nppiYCbCr422ToBGR_8u_C2C4R, [572](#)
- nppiYCbCr422ToBGR_8u_P3C3R, [573](#)
- nppiYCbCr422ToRGB_8u_C2C3R, [573](#)
- nppiYCbCr422ToRGB_8u_C2P3R, [573](#)
- nppiYCbCr422ToRGB_8u_P3C3R, [574](#)
- nppiYCbCrToBGR_709CSC_8u_P3C3R, [574](#)
- nppiYCbCrToBGR_709CSC_8u_P3C4R, [574](#)
- nppiYCbCrToBGR_8u_P3C3R, [575](#)
- nppiYCbCrToBGR_8u_P3C4R, [575](#)
- nppiYCbCrToRGB_8u_AC4R, [576](#)
- nppiYCbCrToRGB_8u_C3R, [576](#)
- nppiYCbCrToRGB_8u_P3C3R, [576](#)
- nppiYCbCrToRGB_8u_P3C4R, [577](#)
- nppiYCbCrToRGB_8u_P3R, [577](#)
- nppiYCCToRGB_8u_AC4R, [577](#)
- nppiYCCToRGB_8u_C3R, [578](#)
- nppiYCrCb420ToRGB_8u_P3C4R, [578](#)
- nppiYCrCb422ToRGB_8u_C2C3R, [578](#)
- nppiYCrCb422ToRGB_8u_C2P3R, [579](#)
- nppiYUV420ToBGR_8u_P3C3R, [579](#)
- nppiYUV420ToBGR_8u_P3C4R, [579](#)
- nppiYUV420ToRGB_8u_P3AC4R, [580](#)
- nppiYUV420ToRGB_8u_P3C3R, [580](#)
- nppiYUV420ToRGB_8u_P3C4R, [580](#)
- nppiYUV420ToRGB_8u_P3R, [581](#)
- nppiYUV422ToRGB_8u_C2C3R, [581](#)
- nppiYUV422ToRGB_8u_P3AC4R, [581](#)
- nppiYUV422ToRGB_8u_P3C3R, [582](#)
- nppiYUV422ToRGB_8u_P3R, [582](#)
- nppiYUVToBGR_8u_AC4R, [582](#)
- nppiYUVToBGR_8u_C3R, [583](#)
- nppiYUVToBGR_8u_P3C3R, [583](#)
- nppiYUVToBGR_8u_P3R, [583](#)
- nppiYUVToRGB_8u_AC4R, [584](#)
- nppiYUVToRGB_8u_C3R, [584](#)
- nppiYUVToRGB_8u_P3C3R, [584](#)
- nppiYUVToRGB_8u_P3R, [585](#)
- image_color_processing
 - nppiColorTwist32f_16s_AC4IR, [637](#)
 - nppiColorTwist32f_16s_AC4R, [638](#)
 - nppiColorTwist32f_16s_C1IR, [638](#)
 - nppiColorTwist32f_16s_C1R, [638](#)
 - nppiColorTwist32f_16s_C2IR, [639](#)
 - nppiColorTwist32f_16s_C2R, [639](#)
 - nppiColorTwist32f_16s_C3IR, [640](#)
 - nppiColorTwist32f_16s_C3R, [640](#)
 - nppiColorTwist32f_16s_IP3R, [640](#)
 - nppiColorTwist32f_16s_P3R, [641](#)
 - nppiColorTwist32f_16u_AC4IR, [641](#)
 - nppiColorTwist32f_16u_AC4R, [642](#)
 - nppiColorTwist32f_16u_C1IR, [642](#)
 - nppiColorTwist32f_16u_C1R, [642](#)
 - nppiColorTwist32f_16u_C2IR, [643](#)
 - nppiColorTwist32f_16u_C2R, [643](#)
 - nppiColorTwist32f_16u_C3IR, [643](#)
 - nppiColorTwist32f_16u_C3R, [644](#)
 - nppiColorTwist32f_16u_IP3R, [644](#)
 - nppiColorTwist32f_16u_P3R, [644](#)
 - nppiColorTwist32f_8s_AC4IR, [645](#)
 - nppiColorTwist32f_8s_AC4R, [645](#)
 - nppiColorTwist32f_8s_C1IR, [646](#)
 - nppiColorTwist32f_8s_C1R, [646](#)
 - nppiColorTwist32f_8s_C2IR, [646](#)
 - nppiColorTwist32f_8s_C2R, [647](#)
 - nppiColorTwist32f_8s_C3IR, [647](#)
 - nppiColorTwist32f_8s_C3R, [647](#)
 - nppiColorTwist32f_8s_C4IR, [648](#)
 - nppiColorTwist32f_8s_C4R, [648](#)
 - nppiColorTwist32f_8s_IP3R, [649](#)
 - nppiColorTwist32f_8s_P3R, [649](#)
 - nppiColorTwist32f_8u_AC4IR, [649](#)
 - nppiColorTwist32f_8u_AC4R, [650](#)
 - nppiColorTwist32f_8u_C1IR, [650](#)
 - nppiColorTwist32f_8u_C1R, [651](#)
 - nppiColorTwist32f_8u_C2IR, [651](#)
 - nppiColorTwist32f_8u_C2R, [651](#)
 - nppiColorTwist32f_8u_C3IR, [652](#)
 - nppiColorTwist32f_8u_C3R, [652](#)
 - nppiColorTwist32f_8u_C4IR, [653](#)
 - nppiColorTwist32f_8u_C4R, [653](#)
 - nppiColorTwist32f_8u_IP3R, [653](#)
 - nppiColorTwist32f_8u_P3R, [654](#)
 - nppiColorTwist32fC_8u_C4IR, [654](#)
 - nppiColorTwist32fC_8u_C4R, [655](#)
 - nppiColorTwist_32f_AC4IR, [655](#)
 - nppiColorTwist_32f_AC4R, [656](#)
 - nppiColorTwist_32f_C1IR, [656](#)
 - nppiColorTwist_32f_C1R, [656](#)
 - nppiColorTwist_32f_C2IR, [657](#)
 - nppiColorTwist_32f_C2R, [657](#)
 - nppiColorTwist_32f_C3IR, [658](#)
 - nppiColorTwist_32f_C3R, [658](#)
 - nppiColorTwist_32f_C4IR, [658](#)
 - nppiColorTwist_32f_C4R, [659](#)
 - nppiColorTwist_32f_IP3R, [659](#)
 - nppiColorTwist_32f_P3R, [660](#)
 - nppiColorTwist_32fC_C4IR, [660](#)
 - nppiColorTwist_32fC_C4R, [660](#)
 - nppiLUT_16s_AC4IR, [661](#)
 - nppiLUT_16s_AC4R, [661](#)
 - nppiLUT_16s_C1IR, [662](#)
 - nppiLUT_16s_C1R, [662](#)

nppiLUT_16s_C3IR, 663
nppiLUT_16s_C3R, 663
nppiLUT_16s_C4IR, 664
nppiLUT_16s_C4R, 664
nppiLUT_16u_AC4IR, 665
nppiLUT_16u_AC4R, 665
nppiLUT_16u_C1IR, 666
nppiLUT_16u_C1R, 666
nppiLUT_16u_C3IR, 667
nppiLUT_16u_C3R, 667
nppiLUT_16u_C4IR, 668
nppiLUT_16u_C4R, 668
nppiLUT_32f_AC4IR, 669
nppiLUT_32f_AC4R, 669
nppiLUT_32f_C1IR, 670
nppiLUT_32f_C1R, 670
nppiLUT_32f_C3IR, 671
nppiLUT_32f_C3R, 671
nppiLUT_32f_C4IR, 672
nppiLUT_32f_C4R, 672
nppiLUT_8u_AC4IR, 673
nppiLUT_8u_AC4R, 673
nppiLUT_8u_C1IR, 674
nppiLUT_8u_C1R, 674
nppiLUT_8u_C3IR, 675
nppiLUT_8u_C3R, 675
nppiLUT_8u_C4IR, 676
nppiLUT_8u_C4R, 676
nppiLUT_Cubic_16s_AC4IR, 677
nppiLUT_Cubic_16s_AC4R, 677
nppiLUT_Cubic_16s_C1IR, 678
nppiLUT_Cubic_16s_C1R, 678
nppiLUT_Cubic_16s_C3IR, 679
nppiLUT_Cubic_16s_C3R, 679
nppiLUT_Cubic_16s_C4IR, 680
nppiLUT_Cubic_16s_C4R, 680
nppiLUT_Cubic_16u_AC4IR, 681
nppiLUT_Cubic_16u_AC4R, 681
nppiLUT_Cubic_16u_C1IR, 682
nppiLUT_Cubic_16u_C1R, 682
nppiLUT_Cubic_16u_C3IR, 683
nppiLUT_Cubic_16u_C3R, 683
nppiLUT_Cubic_16u_C4IR, 684
nppiLUT_Cubic_16u_C4R, 684
nppiLUT_Cubic_32f_AC4IR, 685
nppiLUT_Cubic_32f_AC4R, 685
nppiLUT_Cubic_32f_C1IR, 686
nppiLUT_Cubic_32f_C1R, 686
nppiLUT_Cubic_32f_C3IR, 687
nppiLUT_Cubic_32f_C3R, 687
nppiLUT_Cubic_32f_C4IR, 688
nppiLUT_Cubic_32f_C4R, 688
nppiLUT_Cubic_8u_AC4IR, 689
nppiLUT_Cubic_8u_AC4R, 689
nppiLUT_Cubic_8u_C1IR, 690
nppiLUT_Cubic_8u_C1R, 690
nppiLUT_Cubic_8u_C3IR, 691
nppiLUT_Cubic_8u_C3R, 691
nppiLUT_Cubic_8u_C4IR, 692
nppiLUT_Cubic_8u_C4R, 692
nppiLUT_Linear_16s_AC4IR, 693
nppiLUT_Linear_16s_AC4R, 693
nppiLUT_Linear_16s_C1IR, 694
nppiLUT_Linear_16s_C1R, 694
nppiLUT_Linear_16s_C3IR, 695
nppiLUT_Linear_16s_C3R, 695
nppiLUT_Linear_16s_C4IR, 696
nppiLUT_Linear_16s_C4R, 696
nppiLUT_Linear_16u_AC4IR, 697
nppiLUT_Linear_16u_AC4R, 697
nppiLUT_Linear_16u_C1IR, 698
nppiLUT_Linear_16u_C1R, 698
nppiLUT_Linear_16u_C3IR, 699
nppiLUT_Linear_16u_C3R, 699
nppiLUT_Linear_16u_C4IR, 700
nppiLUT_Linear_16u_C4R, 700
nppiLUT_Linear_32f_AC4IR, 701
nppiLUT_Linear_32f_AC4R, 701
nppiLUT_Linear_32f_C1IR, 702
nppiLUT_Linear_32f_C1R, 702
nppiLUT_Linear_32f_C3IR, 703
nppiLUT_Linear_32f_C3R, 703
nppiLUT_Linear_32f_C4IR, 704
nppiLUT_Linear_32f_C4R, 704
nppiLUT_Linear_8u_AC4IR, 705
nppiLUT_Linear_8u_AC4R, 705
nppiLUT_Linear_8u_C1IR, 706
nppiLUT_Linear_8u_C1R, 707
nppiLUT_Linear_8u_C3IR, 707
nppiLUT_Linear_8u_C3R, 708
nppiLUT_Linear_8u_C4IR, 708
nppiLUT_Linear_8u_C4R, 709
nppiLUT_Trilinear_8u_AC4IR, 709
nppiLUT_Trilinear_8u_AC4R, 710
nppiLUT_Trilinear_8u_C4R, 711
nppiLUTPalette_16u24u_C1R, 711
nppiLUTPalette_16u32u_C1R, 712
nppiLUTPalette_16u8u_C1R, 712
nppiLUTPalette_16u_AC4R, 713
nppiLUTPalette_16u_C1R, 713
nppiLUTPalette_16u_C3R, 714
nppiLUTPalette_16u_C4R, 714
nppiLUTPalette_8u24u_C1R, 715
nppiLUTPalette_8u32u_C1R, 715
nppiLUTPalette_8u_AC4R, 716
nppiLUTPalette_8u_C1R, 716
nppiLUTPalette_8u_C3R, 717
nppiLUTPalette_8u_C4R, 717

- nppiLUTPaletteSwap_16u_C3A0C4R, [718](#)
 - nppiLUTPaletteSwap_8u_C3A0C4R, [718](#)
- image_color_sampling_format_conversion
 - nppiCbYCr422ToYCbCr411_8u_C2P3R, [593](#)
 - nppiCbYCr422ToYCbCr420_8u_C2P2R, [594](#)
 - nppiCbYCr422ToYCbCr420_8u_C2P3R, [594](#)
 - nppiCbYCr422ToYCbCr422_8u_C2P3R, [594](#)
 - nppiCbYCr422ToYCbCr422_8u_C2R, [595](#)
 - nppiCbYCr422ToYCrCb420_8u_C2P3R, [595](#)
 - nppiYCbCr411_8u_P2P3R, [596](#)
 - nppiYCbCr411_8u_P3P2R, [596](#)
 - nppiYCbCr411ToYCbCr420_8u_P2P3R, [596](#)
 - nppiYCbCr411ToYCbCr420_8u_P3P2R, [597](#)
 - nppiYCbCr411ToYCbCr420_8u_P3R, [597](#)
 - nppiYCbCr411ToYCbCr422_8u_P2C2R, [598](#)
 - nppiYCbCr411ToYCbCr422_8u_P2P3R, [598](#)
 - nppiYCbCr411ToYCbCr422_8u_P3C2R, [598](#)
 - nppiYCbCr411ToYCbCr422_8u_P3R, [599](#)
 - nppiYCbCr411ToYCrCb420_8u_P2P3R, [599](#)
 - nppiYCbCr411ToYCrCb422_8u_P3C2R, [600](#)
 - nppiYCbCr411ToYCrCb422_8u_P3R, [600](#)
 - nppiYCbCr420_8u_P2P3R, [600](#)
 - nppiYCbCr420_8u_P3P2R, [601](#)
 - nppiYCbCr420ToCbYCr422_8u_P2C2R, [601](#)
 - nppiYCbCr420ToYCbCr411_8u_P2P3R, [602](#)
 - nppiYCbCr420ToYCbCr411_8u_P3P2R, [602](#)
 - nppiYCbCr420ToYCbCr422_8u_P2C2R, [603](#)
 - nppiYCbCr420ToYCbCr422_8u_P2P3R, [603](#)
 - nppiYCbCr420ToYCbCr422_8u_P3R, [603](#)
 - nppiYCbCr420ToYCrCb420_8u_P2P3R, [604](#)
 - nppiYCbCr422_8u_C2P3R, [604](#)
 - nppiYCbCr422_8u_P3C2R, [605](#)
 - nppiYCbCr422ToCbYCr422_8u_C2R, [605](#)
 - nppiYCbCr422ToYCbCr411_8u_C2P2R, [605](#)
 - nppiYCbCr422ToYCbCr411_8u_C2P3R, [606](#)
 - nppiYCbCr422ToYCbCr411_8u_P3P2R, [606](#)
 - nppiYCbCr422ToYCbCr411_8u_P3R, [607](#)
 - nppiYCbCr422ToYCbCr420_8u_C2P2R, [607](#)
 - nppiYCbCr422ToYCbCr420_8u_C2P3R, [608](#)
 - nppiYCbCr422ToYCbCr420_8u_P3P2R, [608](#)
 - nppiYCbCr422ToYCbCr420_8u_P3R, [608](#)
 - nppiYCbCr422ToYCrCb420_8u_C2P3R, [609](#)
 - nppiYCbCr422ToYCrCb422_8u_C2R, [609](#)
 - nppiYCbCr422ToYCrCb422_8u_P3C2R, [610](#)
 - nppiYCrCb420ToCbYCr422_8u_P3C2R, [610](#)
 - nppiYCrCb420ToYCbCr411_8u_P3P2R, [610](#)
 - nppiYCrCb420ToYCbCr420_8u_P3P2R, [611](#)
 - nppiYCrCb420ToYCbCr422_8u_P3C2R, [611](#)
 - nppiYCrCb420ToYCbCr422_8u_P3R, [612](#)
 - nppiYCrCb422ToYCbCr411_8u_C2P3R, [612](#)
 - nppiYCrCb422ToYCbCr420_8u_C2P3R, [613](#)
 - nppiYCrCb422ToYCbCr422_8u_C2P3R, [613](#)
- image_compare_operations
 - nppiCompare_16s_AC4R, [2465](#)
 - nppiCompare_16s_C1R, [2466](#)
 - nppiCompare_16s_C3R, [2466](#)
 - nppiCompare_16s_C4R, [2466](#)
 - nppiCompare_16u_AC4R, [2467](#)
 - nppiCompare_16u_C1R, [2467](#)
 - nppiCompare_16u_C3R, [2468](#)
 - nppiCompare_16u_C4R, [2468](#)
 - nppiCompare_32f_AC4R, [2469](#)
 - nppiCompare_32f_C1R, [2469](#)
 - nppiCompare_32f_C3R, [2470](#)
 - nppiCompare_32f_C4R, [2470](#)
 - nppiCompare_8u_AC4R, [2471](#)
 - nppiCompare_8u_C1R, [2471](#)
 - nppiCompare_8u_C3R, [2472](#)
 - nppiCompare_8u_C4R, [2472](#)
 - nppiCompareC_16s_AC4R, [2473](#)
 - nppiCompareC_16s_C1R, [2473](#)
 - nppiCompareC_16s_C3R, [2474](#)
 - nppiCompareC_16s_C4R, [2474](#)
 - nppiCompareC_16u_AC4R, [2475](#)
 - nppiCompareC_16u_C1R, [2475](#)
 - nppiCompareC_16u_C3R, [2475](#)
 - nppiCompareC_16u_C4R, [2476](#)
 - nppiCompareC_32f_AC4R, [2476](#)
 - nppiCompareC_32f_C1R, [2477](#)
 - nppiCompareC_32f_C3R, [2477](#)
 - nppiCompareC_32f_C4R, [2478](#)
 - nppiCompareC_8u_AC4R, [2478](#)
 - nppiCompareC_8u_C1R, [2478](#)
 - nppiCompareC_8u_C3R, [2479](#)
 - nppiCompareC_8u_C4R, [2479](#)
 - nppiCompareEqualEps_32f_AC4R, [2480](#)
 - nppiCompareEqualEps_32f_C1R, [2480](#)
 - nppiCompareEqualEps_32f_C3R, [2481](#)
 - nppiCompareEqualEps_32f_C4R, [2481](#)
 - nppiCompareEqualEpsC_32f_AC4R, [2482](#)
 - nppiCompareEqualEpsC_32f_C1R, [2482](#)
 - nppiCompareEqualEpsC_32f_C3R, [2483](#)
 - nppiCompareEqualEpsC_32f_C4R, [2483](#)
- image_complement_color_key
 - nppiAlphaCompColorKey_8u_AC4R, [620](#)
 - nppiCompColorKey_8u_C1R, [621](#)
 - nppiCompColorKey_8u_C3R, [621](#)
 - nppiCompColorKey_8u_C4R, [622](#)
- image_compression
 - nppiDecodeHuffmanScanHost_JPEG_-8u16s_P1R, [721](#)
 - nppiDecodeHuffmanScanHost_JPEG_-8u16s_P3R, [721](#)
 - NppiDecodeHuffmanSpec, [721](#)
 - nppiDecodeHuffmanSpecFreeHost_JPEG, [722](#)
 - nppiDecodeHuffmanSpecGetBufSize_JPEG, [722](#)

image_convolution
 nppiFilter32f_16s_AC4R, [1230](#)
 nppiFilter32f_16s_C1R, [1231](#)
 nppiFilter32f_16s_C3R, [1231](#)
 nppiFilter32f_16s_C4R, [1231](#)

- [nppiFilter32f_16u_AC4R](#), [1232](#)
- [nppiFilter32f_16u_C1R](#), [1232](#)
- [nppiFilter32f_16u_C3R](#), [1233](#)
- [nppiFilter32f_16u_C4R](#), [1233](#)
- [nppiFilter32f_32s_AC4R](#), [1234](#)
- [nppiFilter32f_32s_C1R](#), [1234](#)
- [nppiFilter32f_32s_C3R](#), [1235](#)
- [nppiFilter32f_32s_C4R](#), [1235](#)
- [nppiFilter32f_8s16s_AC4R](#), [1236](#)
- [nppiFilter32f_8s16s_C1R](#), [1236](#)
- [nppiFilter32f_8s16s_C3R](#), [1237](#)
- [nppiFilter32f_8s16s_C4R](#), [1237](#)
- [nppiFilter32f_8s_AC4R](#), [1238](#)
- [nppiFilter32f_8s_C1R](#), [1238](#)
- [nppiFilter32f_8s_C2R](#), [1239](#)
- [nppiFilter32f_8s_C3R](#), [1239](#)
- [nppiFilter32f_8s_C4R](#), [1240](#)
- [nppiFilter32f_8u16s_AC4R](#), [1240](#)
- [nppiFilter32f_8u16s_C1R](#), [1241](#)
- [nppiFilter32f_8u16s_C3R](#), [1241](#)
- [nppiFilter32f_8u16s_C4R](#), [1242](#)
- [nppiFilter32f_8u_AC4R](#), [1242](#)
- [nppiFilter32f_8u_C1R](#), [1243](#)
- [nppiFilter32f_8u_C2R](#), [1243](#)
- [nppiFilter32f_8u_C3R](#), [1244](#)
- [nppiFilter32f_8u_C4R](#), [1244](#)
- [nppiFilter_16s_AC4R](#), [1245](#)
- [nppiFilter_16s_C1R](#), [1245](#)
- [nppiFilter_16s_C3R](#), [1246](#)
- [nppiFilter_16s_C4R](#), [1246](#)
- [nppiFilter_16u_AC4R](#), [1247](#)
- [nppiFilter_16u_C1R](#), [1247](#)
- [nppiFilter_16u_C3R](#), [1248](#)
- [nppiFilter_16u_C4R](#), [1248](#)
- [nppiFilter_32f_AC4R](#), [1249](#)
- [nppiFilter_32f_C1R](#), [1249](#)
- [nppiFilter_32f_C2R](#), [1250](#)
- [nppiFilter_32f_C3R](#), [1250](#)
- [nppiFilter_32f_C4R](#), [1251](#)
- [nppiFilter_64f_C1R](#), [1251](#)
- [nppiFilter_8u_AC4R](#), [1252](#)
- [nppiFilter_8u_C1R](#), [1252](#)
- [nppiFilter_8u_C3R](#), [1253](#)
- [nppiFilter_8u_C4R](#), [1253](#)
- [nppiFilterBorder32f_16s_AC4R](#), [1254](#)
- [nppiFilterBorder32f_16s_C1R](#), [1254](#)
- [nppiFilterBorder32f_16s_C3R](#), [1255](#)
- [nppiFilterBorder32f_16s_C4R](#), [1255](#)
- [nppiFilterBorder32f_16u_AC4R](#), [1256](#)
- [nppiFilterBorder32f_16u_C1R](#), [1256](#)
- [nppiFilterBorder32f_16u_C3R](#), [1257](#)
- [nppiFilterBorder32f_16u_C4R](#), [1257](#)
- [nppiFilterBorder32f_32s_AC4R](#), [1258](#)
- [nppiFilterBorder32f_32s_C1R](#), [1258](#)
- [nppiFilterBorder32f_32s_C3R](#), [1259](#)
- [nppiFilterBorder32f_32s_C4R](#), [1259](#)
- [nppiFilterBorder32f_8s16s_AC4R](#), [1260](#)
- [nppiFilterBorder32f_8s16s_C1R](#), [1260](#)
- [nppiFilterBorder32f_8s16s_C3R](#), [1261](#)
- [nppiFilterBorder32f_8s16s_C4R](#), [1261](#)
- [nppiFilterBorder32f_8s_AC4R](#), [1262](#)
- [nppiFilterBorder32f_8s_C1R](#), [1262](#)
- [nppiFilterBorder32f_8s_C2R](#), [1263](#)
- [nppiFilterBorder32f_8s_C3R](#), [1263](#)
- [nppiFilterBorder32f_8s_C4R](#), [1264](#)
- [nppiFilterBorder32f_8u16s_AC4R](#), [1264](#)
- [nppiFilterBorder32f_8u16s_C1R](#), [1265](#)
- [nppiFilterBorder32f_8u16s_C3R](#), [1265](#)
- [nppiFilterBorder32f_8u16s_C4R](#), [1266](#)
- [nppiFilterBorder32f_8u_AC4R](#), [1266](#)
- [nppiFilterBorder32f_8u_C1R](#), [1267](#)
- [nppiFilterBorder32f_8u_C2R](#), [1267](#)
- [nppiFilterBorder32f_8u_C3R](#), [1268](#)
- [nppiFilterBorder32f_8u_C4R](#), [1268](#)
- [nppiFilterBorder_16s_AC4R](#), [1269](#)
- [nppiFilterBorder_16s_C1R](#), [1270](#)
- [nppiFilterBorder_16s_C3R](#), [1270](#)
- [nppiFilterBorder_16s_C4R](#), [1271](#)
- [nppiFilterBorder_16u_AC4R](#), [1271](#)
- [nppiFilterBorder_16u_C1R](#), [1272](#)
- [nppiFilterBorder_16u_C3R](#), [1273](#)
- [nppiFilterBorder_16u_C4R](#), [1273](#)
- [nppiFilterBorder_32f_AC4R](#), [1274](#)
- [nppiFilterBorder_32f_C1R](#), [1274](#)
- [nppiFilterBorder_32f_C2R](#), [1275](#)
- [nppiFilterBorder_32f_C3R](#), [1275](#)
- [nppiFilterBorder_32f_C4R](#), [1276](#)
- [nppiFilterBorder_8u_AC4R](#), [1276](#)
- [nppiFilterBorder_8u_C1R](#), [1277](#)
- [nppiFilterBorder_8u_C3R](#), [1278](#)
- [nppiFilterBorder_8u_C4R](#), [1278](#)
- image_copy
 - [nppiCopy_16s_AC4MR](#), [782](#)
 - [nppiCopy_16s_AC4R](#), [783](#)
 - [nppiCopy_16s_C1C3R](#), [783](#)
 - [nppiCopy_16s_C1C4R](#), [783](#)
 - [nppiCopy_16s_C1MR](#), [784](#)
 - [nppiCopy_16s_C1R](#), [784](#)
 - [nppiCopy_16s_C3C1R](#), [784](#)
 - [nppiCopy_16s_C3CR](#), [785](#)
 - [nppiCopy_16s_C3MR](#), [785](#)
 - [nppiCopy_16s_C3P3R](#), [785](#)
 - [nppiCopy_16s_C3R](#), [786](#)
 - [nppiCopy_16s_C4C1R](#), [786](#)
 - [nppiCopy_16s_C4CR](#), [786](#)
 - [nppiCopy_16s_C4MR](#), [787](#)
 - [nppiCopy_16s_C4P4R](#), [787](#)
 - [nppiCopy_16s_C4R](#), [787](#)

- [nppiCopy_16s_P3C3R, 788](#)
- [nppiCopy_16s_P4C4R, 788](#)
- [nppiCopy_16sc_AC4R, 788](#)
- [nppiCopy_16sc_C1R, 789](#)
- [nppiCopy_16sc_C2R, 789](#)
- [nppiCopy_16sc_C3R, 789](#)
- [nppiCopy_16sc_C4R, 790](#)
- [nppiCopy_16u_AC4MR, 790](#)
- [nppiCopy_16u_AC4R, 790](#)
- [nppiCopy_16u_C1C3R, 791](#)
- [nppiCopy_16u_C1C4R, 791](#)
- [nppiCopy_16u_C1MR, 791](#)
- [nppiCopy_16u_C1R, 792](#)
- [nppiCopy_16u_C3C1R, 792](#)
- [nppiCopy_16u_C3CR, 792](#)
- [nppiCopy_16u_C3MR, 793](#)
- [nppiCopy_16u_C3P3R, 793](#)
- [nppiCopy_16u_C3R, 793](#)
- [nppiCopy_16u_C4C1R, 794](#)
- [nppiCopy_16u_C4CR, 794](#)
- [nppiCopy_16u_C4MR, 794](#)
- [nppiCopy_16u_C4P4R, 795](#)
- [nppiCopy_16u_C4R, 795](#)
- [nppiCopy_16u_P3C3R, 795](#)
- [nppiCopy_16u_P4C4R, 796](#)
- [nppiCopy_32f_AC4MR, 796](#)
- [nppiCopy_32f_AC4R, 796](#)
- [nppiCopy_32f_C1C3R, 797](#)
- [nppiCopy_32f_C1C4R, 797](#)
- [nppiCopy_32f_C1MR, 797](#)
- [nppiCopy_32f_C1R, 798](#)
- [nppiCopy_32f_C3C1R, 798](#)
- [nppiCopy_32f_C3CR, 798](#)
- [nppiCopy_32f_C3MR, 799](#)
- [nppiCopy_32f_C3P3R, 799](#)
- [nppiCopy_32f_C3R, 799](#)
- [nppiCopy_32f_C4C1R, 800](#)
- [nppiCopy_32f_C4CR, 800](#)
- [nppiCopy_32f_C4MR, 800](#)
- [nppiCopy_32f_C4P4R, 801](#)
- [nppiCopy_32f_C4R, 801](#)
- [nppiCopy_32f_P3C3R, 801](#)
- [nppiCopy_32f_P4C4R, 802](#)
- [nppiCopy_32fc_AC4R, 802](#)
- [nppiCopy_32fc_C1R, 802](#)
- [nppiCopy_32fc_C2R, 803](#)
- [nppiCopy_32fc_C3R, 803](#)
- [nppiCopy_32fc_C4R, 803](#)
- [nppiCopy_32s_AC4MR, 804](#)
- [nppiCopy_32s_AC4R, 804](#)
- [nppiCopy_32s_C1C3R, 804](#)
- [nppiCopy_32s_C1C4R, 805](#)
- [nppiCopy_32s_C1MR, 805](#)
- [nppiCopy_32s_C1R, 805](#)
- [nppiCopy_32s_C3C1R, 806](#)
- [nppiCopy_32s_C3CR, 806](#)
- [nppiCopy_32s_C3MR, 806](#)
- [nppiCopy_32s_C3P3R, 807](#)
- [nppiCopy_32s_C3R, 807](#)
- [nppiCopy_32s_C4C1R, 807](#)
- [nppiCopy_32s_C4CR, 808](#)
- [nppiCopy_32s_C4MR, 808](#)
- [nppiCopy_32s_C4P4R, 808](#)
- [nppiCopy_32s_C4R, 809](#)
- [nppiCopy_32s_P3C3R, 809](#)
- [nppiCopy_32s_P4C4R, 809](#)
- [nppiCopy_32sc_AC4R, 810](#)
- [nppiCopy_32sc_C1R, 810](#)
- [nppiCopy_32sc_C2R, 810](#)
- [nppiCopy_32sc_C3R, 811](#)
- [nppiCopy_32sc_C4R, 811](#)
- [nppiCopy_8s_AC4R, 811](#)
- [nppiCopy_8s_C1R, 812](#)
- [nppiCopy_8s_C2R, 812](#)
- [nppiCopy_8s_C3R, 812](#)
- [nppiCopy_8s_C4R, 813](#)
- [nppiCopy_8u_AC4MR, 813](#)
- [nppiCopy_8u_AC4R, 813](#)
- [nppiCopy_8u_C1C3R, 814](#)
- [nppiCopy_8u_C1C4R, 814](#)
- [nppiCopy_8u_C1MR, 814](#)
- [nppiCopy_8u_C1R, 815](#)
- [nppiCopy_8u_C3C1R, 815](#)
- [nppiCopy_8u_C3CR, 815](#)
- [nppiCopy_8u_C3MR, 816](#)
- [nppiCopy_8u_C3P3R, 816](#)
- [nppiCopy_8u_C3R, 816](#)
- [nppiCopy_8u_C4C1R, 817](#)
- [nppiCopy_8u_C4CR, 817](#)
- [nppiCopy_8u_C4MR, 817](#)
- [nppiCopy_8u_C4P4R, 818](#)
- [nppiCopy_8u_C4R, 818](#)
- [nppiCopy_8u_P3C3R, 818](#)
- [nppiCopy_8u_P4C4R, 819](#)
- [image_copy_constant_border](#)
 - [nppiCopyConstBorder_16s_AC4R, 881](#)
 - [nppiCopyConstBorder_16s_C1R, 881](#)
 - [nppiCopyConstBorder_16s_C3R, 882](#)
 - [nppiCopyConstBorder_16s_C4R, 882](#)
 - [nppiCopyConstBorder_16u_AC4R, 883](#)
 - [nppiCopyConstBorder_16u_C1R, 883](#)
 - [nppiCopyConstBorder_16u_C3R, 884](#)
 - [nppiCopyConstBorder_16u_C4R, 884](#)
 - [nppiCopyConstBorder_32f_AC4R, 885](#)
 - [nppiCopyConstBorder_32f_C1R, 885](#)
 - [nppiCopyConstBorder_32f_C3R, 886](#)
 - [nppiCopyConstBorder_32f_C4R, 886](#)
 - [nppiCopyConstBorder_32s_AC4R, 887](#)

- nppiCopyConstBorder_32s_C1R, 887
- nppiCopyConstBorder_32s_C3R, 888
- nppiCopyConstBorder_32s_C4R, 888
- nppiCopyConstBorder_8u_AC4R, 889
- nppiCopyConstBorder_8u_C1R, 889
- nppiCopyConstBorder_8u_C3R, 890
- nppiCopyConstBorder_8u_C4R, 890
- image_copy_replicate_border
 - nppiCopyReplicateBorder_16s_AC4R, 894
 - nppiCopyReplicateBorder_16s_C1R, 894
 - nppiCopyReplicateBorder_16s_C3R, 895
 - nppiCopyReplicateBorder_16s_C4R, 895
 - nppiCopyReplicateBorder_16u_AC4R, 896
 - nppiCopyReplicateBorder_16u_C1R, 896
 - nppiCopyReplicateBorder_16u_C3R, 897
 - nppiCopyReplicateBorder_16u_C4R, 897
 - nppiCopyReplicateBorder_32f_AC4R, 897
 - nppiCopyReplicateBorder_32f_C1R, 898
 - nppiCopyReplicateBorder_32f_C3R, 898
 - nppiCopyReplicateBorder_32f_C4R, 899
 - nppiCopyReplicateBorder_32s_AC4R, 899
 - nppiCopyReplicateBorder_32s_C1R, 900
 - nppiCopyReplicateBorder_32s_C3R, 900
 - nppiCopyReplicateBorder_32s_C4R, 901
 - nppiCopyReplicateBorder_8u_AC4R, 901
 - nppiCopyReplicateBorder_8u_C1R, 902
 - nppiCopyReplicateBorder_8u_C3R, 902
 - nppiCopyReplicateBorder_8u_C4R, 903
- image_copy_sub_pixel
 - nppiCopySubpix_16s_AC4R, 918
 - nppiCopySubpix_16s_C1R, 919
 - nppiCopySubpix_16s_C3R, 919
 - nppiCopySubpix_16s_C4R, 920
 - nppiCopySubpix_16u_AC4R, 920
 - nppiCopySubpix_16u_C1R, 920
 - nppiCopySubpix_16u_C3R, 921
 - nppiCopySubpix_16u_C4R, 921
 - nppiCopySubpix_32f_AC4R, 922
 - nppiCopySubpix_32f_C1R, 922
 - nppiCopySubpix_32f_C3R, 922
 - nppiCopySubpix_32f_C4R, 923
 - nppiCopySubpix_32s_AC4R, 923
 - nppiCopySubpix_32s_C1R, 924
 - nppiCopySubpix_32s_C3R, 924
 - nppiCopySubpix_32s_C4R, 925
 - nppiCopySubpix_8u_AC4R, 925
 - nppiCopySubpix_8u_C1R, 925
 - nppiCopySubpix_8u_C3R, 926
 - nppiCopySubpix_8u_C4R, 926
- image_copy_wrap_border
 - nppiCopyWrapBorder_16s_AC4R, 906
 - nppiCopyWrapBorder_16s_C1R, 906
 - nppiCopyWrapBorder_16s_C3R, 907
 - nppiCopyWrapBorder_16s_C4R, 907
 - nppiCopyWrapBorder_16u_AC4R, 908
 - nppiCopyWrapBorder_16u_C1R, 908
 - nppiCopyWrapBorder_16u_C3R, 909
 - nppiCopyWrapBorder_16u_C4R, 909
 - nppiCopyWrapBorder_32f_AC4R, 910
 - nppiCopyWrapBorder_32f_C1R, 910
 - nppiCopyWrapBorder_32f_C3R, 911
 - nppiCopyWrapBorder_32f_C4R, 911
 - nppiCopyWrapBorder_32s_AC4R, 912
 - nppiCopyWrapBorder_32s_C1R, 912
 - nppiCopyWrapBorder_32s_C3R, 913
 - nppiCopyWrapBorder_32s_C4R, 913
 - nppiCopyWrapBorder_8u_AC4R, 914
 - nppiCopyWrapBorder_8u_C1R, 914
 - nppiCopyWrapBorder_8u_C3R, 915
 - nppiCopyWrapBorder_8u_C4R, 915
- image_count_in_range
 - nppiCountInRange_32f_AC4R, 2069
 - nppiCountInRange_32f_C1R, 2069
 - nppiCountInRange_32f_C3R, 2070
 - nppiCountInRange_8u_AC4R, 2070
 - nppiCountInRange_8u_C1R, 2071
 - nppiCountInRange_8u_C3R, 2071
 - nppiCountInRangeGetBufferHostSize_32f_-AC4R, 2072
 - nppiCountInRangeGetBufferHostSize_32f_-C1R, 2072
 - nppiCountInRangeGetBufferHostSize_32f_-C3R, 2072
 - nppiCountInRangeGetBufferHostSize_8u_-AC4R, 2072
 - nppiCountInRangeGetBufferHostSize_8u_-C1R, 2073
 - nppiCountInRangeGetBufferHostSize_8u_-C3R, 2073
- image_dilate
 - nppiDilate_16u_AC4R, 1580
 - nppiDilate_16u_C1R, 1580
 - nppiDilate_16u_C3R, 1581
 - nppiDilate_16u_C4R, 1581
 - nppiDilate_32f_AC4R, 1581
 - nppiDilate_32f_C1R, 1582
 - nppiDilate_32f_C3R, 1582
 - nppiDilate_32f_C4R, 1583
 - nppiDilate_8u_AC4R, 1583
 - nppiDilate_8u_C1R, 1584
 - nppiDilate_8u_C3R, 1584
 - nppiDilate_8u_C4R, 1584
- image_dilate_3x3
 - nppiDilate3x3_16u_AC4R, 1595
 - nppiDilate3x3_16u_C1R, 1595
 - nppiDilate3x3_16u_C3R, 1595
 - nppiDilate3x3_16u_C4R, 1596
 - nppiDilate3x3_32f_AC4R, 1596

- nppiDilate3x3_32f_C1R, 1596
- nppiDilate3x3_32f_C3R, 1597
- nppiDilate3x3_32f_C4R, 1597
- nppiDilate3x3_64f_C1R, 1597
- nppiDilate3x3_8u_AC4R, 1598
- nppiDilate3x3_8u_C1R, 1598
- nppiDilate3x3_8u_C3R, 1598
- nppiDilate3x3_8u_C4R, 1599
- image_dilate_3x3_border
 - nppiDilate3x3Border_16u_AC4R, 1601
 - nppiDilate3x3Border_16u_C1R, 1601
 - nppiDilate3x3Border_16u_C3R, 1602
 - nppiDilate3x3Border_16u_C4R, 1602
 - nppiDilate3x3Border_32f_AC4R, 1603
 - nppiDilate3x3Border_32f_C1R, 1603
 - nppiDilate3x3Border_32f_C3R, 1604
 - nppiDilate3x3Border_32f_C4R, 1604
 - nppiDilate3x3Border_8u_AC4R, 1604
 - nppiDilate3x3Border_8u_C1R, 1605
 - nppiDilate3x3Border_8u_C3R, 1605
 - nppiDilate3x3Border_8u_C4R, 1606
- image_dilate_border
 - nppiDilateBorder_16u_AC4R, 1587
 - nppiDilateBorder_16u_C1R, 1588
 - nppiDilateBorder_16u_C3R, 1588
 - nppiDilateBorder_16u_C4R, 1589
 - nppiDilateBorder_32f_AC4R, 1589
 - nppiDilateBorder_32f_C1R, 1590
 - nppiDilateBorder_32f_C3R, 1590
 - nppiDilateBorder_32f_C4R, 1591
 - nppiDilateBorder_8u_AC4R, 1591
 - nppiDilateBorder_8u_C1R, 1592
 - nppiDilateBorder_8u_C3R, 1592
 - nppiDilateBorder_8u_C4R, 1593
- image_div
 - nppiDiv_16s_AC4IRSfs, 282
 - nppiDiv_16s_AC4RSfs, 282
 - nppiDiv_16s_C1IRSfs, 283
 - nppiDiv_16s_C1RSfs, 283
 - nppiDiv_16s_C3IRSfs, 283
 - nppiDiv_16s_C3RSfs, 284
 - nppiDiv_16s_C4IRSfs, 284
 - nppiDiv_16s_C4RSfs, 285
 - nppiDiv_16sc_AC4IRSfs, 285
 - nppiDiv_16sc_AC4RSfs, 285
 - nppiDiv_16sc_C1IRSfs, 286
 - nppiDiv_16sc_C1RSfs, 286
 - nppiDiv_16sc_C3IRSfs, 287
 - nppiDiv_16sc_C3RSfs, 287
 - nppiDiv_16u_AC4IRSfs, 288
 - nppiDiv_16u_AC4RSfs, 288
 - nppiDiv_16u_C1IRSfs, 288
 - nppiDiv_16u_C1RSfs, 289
 - nppiDiv_16u_C3IRSfs, 289
- nppiDiv_16u_C3RSfs, 290
- nppiDiv_16u_C4IRSfs, 290
- nppiDiv_16u_C4RSfs, 290
- nppiDiv_32f_AC4IR, 291
- nppiDiv_32f_AC4R, 291
- nppiDiv_32f_C1IR, 292
- nppiDiv_32f_C1R, 292
- nppiDiv_32f_C3IR, 292
- nppiDiv_32f_C3R, 293
- nppiDiv_32f_C4IR, 293
- nppiDiv_32f_C4R, 293
- nppiDiv_32fc_AC4IR, 294
- nppiDiv_32fc_AC4R, 294
- nppiDiv_32fc_C1IR, 295
- nppiDiv_32fc_C1R, 295
- nppiDiv_32fc_C3IR, 295
- nppiDiv_32fc_C3R, 296
- nppiDiv_32fc_C4IR, 296
- nppiDiv_32fc_C4R, 296
- nppiDiv_32s_C1IRSfs, 297
- nppiDiv_32s_C1R, 297
- nppiDiv_32s_C1RSfs, 298
- nppiDiv_32s_C3IRSfs, 298
- nppiDiv_32s_C3RSfs, 298
- nppiDiv_32sc_AC4IRSfs, 299
- nppiDiv_32sc_AC4RSfs, 299
- nppiDiv_32sc_C1IRSfs, 300
- nppiDiv_32sc_C1RSfs, 300
- nppiDiv_32sc_C3IRSfs, 301
- nppiDiv_32sc_C3RSfs, 301
- nppiDiv_8u_AC4IRSfs, 301
- nppiDiv_8u_AC4RSfs, 302
- nppiDiv_8u_C1IRSfs, 302
- nppiDiv_8u_C1RSfs, 303
- nppiDiv_8u_C3IRSfs, 303
- nppiDiv_8u_C3RSfs, 303
- nppiDiv_8u_C4IRSfs, 304
- nppiDiv_8u_C4RSfs, 304
- image_divc
 - nppiDivC_16s_AC4IRSfs, 146
 - nppiDivC_16s_AC4RSfs, 146
 - nppiDivC_16s_C1IRSfs, 146
 - nppiDivC_16s_C1RSfs, 147
 - nppiDivC_16s_C3IRSfs, 147
 - nppiDivC_16s_C3RSfs, 147
 - nppiDivC_16s_C4IRSfs, 148
 - nppiDivC_16s_C4RSfs, 148
 - nppiDivC_16sc_AC4IRSfs, 149
 - nppiDivC_16sc_AC4RSfs, 149
 - nppiDivC_16sc_C1IRSfs, 149
 - nppiDivC_16sc_C1RSfs, 150
 - nppiDivC_16sc_C3IRSfs, 150
 - nppiDivC_16sc_C3RSfs, 151
 - nppiDivC_16u_AC4IRSfs, 151

- nppiDivC_16u_AC4RSfs, [151](#)
- nppiDivC_16u_C1IRSfs, [152](#)
- nppiDivC_16u_C1RSfs, [152](#)
- nppiDivC_16u_C3IRSfs, [153](#)
- nppiDivC_16u_C3RSfs, [153](#)
- nppiDivC_16u_C4IRSfs, [153](#)
- nppiDivC_16u_C4RSfs, [154](#)
- nppiDivC_32f_AC4IR, [154](#)
- nppiDivC_32f_AC4R, [154](#)
- nppiDivC_32f_C1IR, [155](#)
- nppiDivC_32f_C1R, [155](#)
- nppiDivC_32f_C3IR, [155](#)
- nppiDivC_32f_C3R, [156](#)
- nppiDivC_32f_C4IR, [156](#)
- nppiDivC_32f_C4R, [156](#)
- nppiDivC_32fc_AC4IR, [157](#)
- nppiDivC_32fc_AC4R, [157](#)
- nppiDivC_32fc_C1IR, [157](#)
- nppiDivC_32fc_C1R, [158](#)
- nppiDivC_32fc_C3IR, [158](#)
- nppiDivC_32fc_C3R, [158](#)
- nppiDivC_32fc_C4IR, [159](#)
- nppiDivC_32fc_C4R, [159](#)
- nppiDivC_32s_C1IRSfs, [160](#)
- nppiDivC_32s_C1RSfs, [160](#)
- nppiDivC_32s_C3IRSfs, [160](#)
- nppiDivC_32s_C3RSfs, [161](#)
- nppiDivC_32sc_AC4IRSfs, [161](#)
- nppiDivC_32sc_AC4RSfs, [161](#)
- nppiDivC_32sc_C1IRSfs, [162](#)
- nppiDivC_32sc_C1RSfs, [162](#)
- nppiDivC_32sc_C3IRSfs, [163](#)
- nppiDivC_32sc_C3RSfs, [163](#)
- nppiDivC_8u_AC4IRSfs, [163](#)
- nppiDivC_8u_AC4RSfs, [164](#)
- nppiDivC_8u_C1IRSfs, [164](#)
- nppiDivC_8u_C1RSfs, [165](#)
- nppiDivC_8u_C3IRSfs, [165](#)
- nppiDivC_8u_C3RSfs, [165](#)
- nppiDivC_8u_C4IRSfs, [166](#)
- nppiDivC_8u_C4RSfs, [166](#)
- image_divround
 - nppiDiv_Round_16s_AC4IRSfs, [308](#)
 - nppiDiv_Round_16s_AC4RSfs, [309](#)
 - nppiDiv_Round_16s_C1IRSfs, [309](#)
 - nppiDiv_Round_16s_C1RSfs, [310](#)
 - nppiDiv_Round_16s_C3IRSfs, [310](#)
 - nppiDiv_Round_16s_C3RSfs, [310](#)
 - nppiDiv_Round_16s_C4IRSfs, [311](#)
 - nppiDiv_Round_16s_C4RSfs, [311](#)
 - nppiDiv_Round_16u_AC4IRSfs, [312](#)
 - nppiDiv_Round_16u_AC4RSfs, [312](#)
 - nppiDiv_Round_16u_C1IRSfs, [313](#)
 - nppiDiv_Round_16u_C1RSfs, [313](#)
 - nppiDiv_Round_16u_C3IRSfs, [314](#)
 - nppiDiv_Round_16u_C3RSfs, [314](#)
 - nppiDiv_Round_16u_C4IRSfs, [315](#)
 - nppiDiv_Round_16u_C4RSfs, [315](#)
 - nppiDiv_Round_8u_AC4IRSfs, [316](#)
 - nppiDiv_Round_8u_AC4RSfs, [316](#)
 - nppiDiv_Round_8u_C1IRSfs, [317](#)
 - nppiDiv_Round_8u_C1RSfs, [317](#)
 - nppiDiv_Round_8u_C3IRSfs, [318](#)
 - nppiDiv_Round_8u_C3RSfs, [318](#)
 - nppiDiv_Round_8u_C4IRSfs, [319](#)
 - nppiDiv_Round_8u_C4RSfs, [319](#)
- image_dot_prod
 - nppiDotProd_16s64f_AC4R, [2047](#)
 - nppiDotProd_16s64f_C1R, [2047](#)
 - nppiDotProd_16s64f_C3R, [2048](#)
 - nppiDotProd_16s64f_C4R, [2048](#)
 - nppiDotProd_16u64f_AC4R, [2049](#)
 - nppiDotProd_16u64f_C1R, [2049](#)
 - nppiDotProd_16u64f_C3R, [2050](#)
 - nppiDotProd_16u64f_C4R, [2050](#)
 - nppiDotProd_32f64f_AC4R, [2050](#)
 - nppiDotProd_32f64f_C1R, [2051](#)
 - nppiDotProd_32f64f_C3R, [2051](#)
 - nppiDotProd_32f64f_C4R, [2052](#)
 - nppiDotProd_32s64f_AC4R, [2052](#)
 - nppiDotProd_32s64f_C1R, [2053](#)
 - nppiDotProd_32s64f_C3R, [2053](#)
 - nppiDotProd_32s64f_C4R, [2053](#)
 - nppiDotProd_32u64f_AC4R, [2054](#)
 - nppiDotProd_32u64f_C1R, [2054](#)
 - nppiDotProd_32u64f_C3R, [2055](#)
 - nppiDotProd_32u64f_C4R, [2055](#)
 - nppiDotProd_8s64f_AC4R, [2056](#)
 - nppiDotProd_8s64f_C1R, [2056](#)
 - nppiDotProd_8s64f_C3R, [2056](#)
 - nppiDotProd_8s64f_C4R, [2057](#)
 - nppiDotProd_8u64f_AC4R, [2057](#)
 - nppiDotProd_8u64f_C1R, [2058](#)
 - nppiDotProd_8u64f_C3R, [2058](#)
 - nppiDotProd_8u64f_C4R, [2059](#)
 - nppiDotProdGetBufferSize_16s64f_-
AC4R, [2059](#)
 - nppiDotProdGetBufferSize_16s64f_C1R,
[2059](#)
 - nppiDotProdGetBufferSize_16s64f_C3R,
[2060](#)
 - nppiDotProdGetBufferSize_16s64f_C4R,
[2060](#)
 - nppiDotProdGetBufferSize_16u64f_-
AC4R, [2060](#)
 - nppiDotProdGetBufferSize_16u64f_C1R,
[2060](#)

- [nppiDotProdGetBufferHostSize_16u64f_C3R, 2061](#)
- [nppiDotProdGetBufferHostSize_16u64f_C4R, 2061](#)
- [nppiDotProdGetBufferHostSize_32f64f_-AC4R, 2061](#)
- [nppiDotProdGetBufferHostSize_32f64f_C1R, 2062](#)
- [nppiDotProdGetBufferHostSize_32f64f_C3R, 2062](#)
- [nppiDotProdGetBufferHostSize_32f64f_C4R, 2062](#)
- [nppiDotProdGetBufferHostSize_32s64f_-AC4R, 2062](#)
- [nppiDotProdGetBufferHostSize_32s64f_C1R, 2063](#)
- [nppiDotProdGetBufferHostSize_32s64f_C3R, 2063](#)
- [nppiDotProdGetBufferHostSize_32s64f_C4R, 2063](#)
- [nppiDotProdGetBufferHostSize_32u64f_-AC4R, 2064](#)
- [nppiDotProdGetBufferHostSize_32u64f_C1R, 2064](#)
- [nppiDotProdGetBufferHostSize_32u64f_C3R, 2064](#)
- [nppiDotProdGetBufferHostSize_32u64f_C4R, 2064](#)
- [nppiDotProdGetBufferHostSize_8s64f_-AC4R, 2065](#)
- [nppiDotProdGetBufferHostSize_8s64f_C1R, 2065](#)
- [nppiDotProdGetBufferHostSize_8s64f_C3R, 2065](#)
- [nppiDotProdGetBufferHostSize_8s64f_C4R, 2066](#)
- [nppiDotProdGetBufferHostSize_8u64f_-AC4R, 2066](#)
- [nppiDotProdGetBufferHostSize_8u64f_C1R, 2066](#)
- [nppiDotProdGetBufferHostSize_8u64f_C3R, 2066](#)
- [nppiDotProdGetBufferHostSize_8u64f_C4R, 2067](#)
- [image_duplicate_channel](#)
 - [nppiDup_16s_C1AC4R, 929](#)
 - [nppiDup_16s_C1C3R, 929](#)
 - [nppiDup_16s_C1C4R, 930](#)
 - [nppiDup_16u_C1AC4R, 930](#)
 - [nppiDup_16u_C1C3R, 930](#)
 - [nppiDup_16u_C1C4R, 931](#)
 - [nppiDup_32f_C1AC4R, 931](#)
 - [nppiDup_32f_C1C3R, 931](#)
 - [nppiDup_32f_C1C4R, 932](#)
 - [nppiDup_32s_C1AC4R, 932](#)
 - [nppiDup_32s_C1C3R, 932](#)
 - [nppiDup_32s_C1C4R, 933](#)
 - [nppiDup_8u_C1AC4R, 933](#)
 - [nppiDup_8u_C1C3R, 933](#)
 - [nppiDup_8u_C1C4R, 934](#)
- [image_erode](#)
 - [nppiErode_16u_AC4R, 1608](#)
 - [nppiErode_16u_C1R, 1608](#)
 - [nppiErode_16u_C3R, 1609](#)
 - [nppiErode_16u_C4R, 1609](#)
 - [nppiErode_32f_AC4R, 1609](#)
 - [nppiErode_32f_C1R, 1610](#)
 - [nppiErode_32f_C3R, 1610](#)
 - [nppiErode_32f_C4R, 1611](#)
 - [nppiErode_8u_AC4R, 1611](#)
 - [nppiErode_8u_C1R, 1612](#)
 - [nppiErode_8u_C3R, 1612](#)
 - [nppiErode_8u_C4R, 1612](#)
- [image_erode_3x3](#)
 - [nppiErode3x3_16u_AC4R, 1623](#)
 - [nppiErode3x3_16u_C1R, 1623](#)
 - [nppiErode3x3_16u_C3R, 1623](#)
 - [nppiErode3x3_16u_C4R, 1624](#)
 - [nppiErode3x3_32f_AC4R, 1624](#)
 - [nppiErode3x3_32f_C1R, 1624](#)
 - [nppiErode3x3_32f_C3R, 1625](#)
 - [nppiErode3x3_32f_C4R, 1625](#)
 - [nppiErode3x3_64f_C1R, 1625](#)
 - [nppiErode3x3_8u_AC4R, 1626](#)
 - [nppiErode3x3_8u_C1R, 1626](#)
 - [nppiErode3x3_8u_C3R, 1626](#)
 - [nppiErode3x3_8u_C4R, 1627](#)
- [image_erode_3x3_border](#)
 - [nppiErode3x3Border_16u_AC4R, 1629](#)
 - [nppiErode3x3Border_16u_C1R, 1629](#)
 - [nppiErode3x3Border_16u_C3R, 1630](#)
 - [nppiErode3x3Border_16u_C4R, 1630](#)
 - [nppiErode3x3Border_32f_AC4R, 1631](#)
 - [nppiErode3x3Border_32f_C1R, 1631](#)
 - [nppiErode3x3Border_32f_C3R, 1632](#)
 - [nppiErode3x3Border_32f_C4R, 1632](#)
 - [nppiErode3x3Border_8u_AC4R, 1632](#)
 - [nppiErode3x3Border_8u_C1R, 1633](#)
 - [nppiErode3x3Border_8u_C3R, 1633](#)
 - [nppiErode3x3Border_8u_C4R, 1634](#)
- [image_erode_border](#)
 - [nppiErodeBorder_16u_AC4R, 1615](#)
 - [nppiErodeBorder_16u_C1R, 1616](#)
 - [nppiErodeBorder_16u_C3R, 1616](#)
 - [nppiErodeBorder_16u_C4R, 1617](#)
 - [nppiErodeBorder_32f_AC4R, 1617](#)
 - [nppiErodeBorder_32f_C1R, 1618](#)
 - [nppiErodeBorder_32f_C3R, 1618](#)

- nppiErodeBorder_32f_C4R, [1619](#)
- nppiErodeBorder_8u_AC4R, [1619](#)
- nppiErodeBorder_8u_C1R, [1620](#)
- nppiErodeBorder_8u_C3R, [1620](#)
- nppiErodeBorder_8u_C4R, [1621](#)
- image_exp
 - nppiExp_16s_C1RSfs, [365](#)
 - nppiExp_16s_C1RSfs, [365](#)
 - nppiExp_16s_C3RSfs, [366](#)
 - nppiExp_16s_C3RSfs, [366](#)
 - nppiExp_16u_C1RSfs, [366](#)
 - nppiExp_16u_C1RSfs, [367](#)
 - nppiExp_16u_C3RSfs, [367](#)
 - nppiExp_16u_C3RSfs, [367](#)
 - nppiExp_32f_C1R, [368](#)
 - nppiExp_32f_C1R, [368](#)
 - nppiExp_32f_C3R, [369](#)
 - nppiExp_8u_C1RSfs, [369](#)
 - nppiExp_8u_C1RSfs, [369](#)
 - nppiExp_8u_C3RSfs, [370](#)
 - nppiExp_8u_C3RSfs, [370](#)
- image_filtering_functions
 - nppiFilterGauss_16s_AC4R, [989](#)
 - nppiFilterGauss_16s_C1R, [989](#)
 - nppiFilterGauss_16s_C3R, [989](#)
 - nppiFilterGauss_16s_C4R, [990](#)
 - nppiFilterGauss_16u_AC4R, [990](#)
 - nppiFilterGauss_16u_C1R, [990](#)
 - nppiFilterGauss_16u_C3R, [991](#)
 - nppiFilterGauss_16u_C4R, [991](#)
 - nppiFilterGauss_32f_AC4R, [991](#)
 - nppiFilterGauss_32f_C1R, [992](#)
 - nppiFilterGauss_32f_C3R, [992](#)
 - nppiFilterGauss_32f_C4R, [992](#)
 - nppiFilterGauss_8u_AC4R, [993](#)
 - nppiFilterGauss_8u_C1R, [993](#)
 - nppiFilterGauss_8u_C3R, [993](#)
 - nppiFilterGauss_8u_C4R, [994](#)
 - nppiFilterGaussAdvanced_16s_AC4R, [994](#)
 - nppiFilterGaussAdvanced_16s_C1R, [995](#)
 - nppiFilterGaussAdvanced_16s_C3R, [995](#)
 - nppiFilterGaussAdvanced_16s_C4R, [995](#)
 - nppiFilterGaussAdvanced_16u_AC4R, [996](#)
 - nppiFilterGaussAdvanced_16u_C1R, [996](#)
 - nppiFilterGaussAdvanced_16u_C3R, [997](#)
 - nppiFilterGaussAdvanced_16u_C4R, [997](#)
 - nppiFilterGaussAdvanced_32f_AC4R, [997](#)
 - nppiFilterGaussAdvanced_32f_C1R, [998](#)
 - nppiFilterGaussAdvanced_32f_C3R, [998](#)
 - nppiFilterGaussAdvanced_32f_C4R, [999](#)
 - nppiFilterGaussAdvanced_8u_AC4R, [999](#)
 - nppiFilterGaussAdvanced_8u_C1R, [999](#)
 - nppiFilterGaussAdvanced_8u_C3R, [1000](#)
 - nppiFilterGaussAdvanced_8u_C4R, [1000](#)
 - nppiFilterGaussAdvancedBorder_16s_AC4R, [1001](#)
 - nppiFilterGaussAdvancedBorder_16s_C1R, [1001](#)
 - nppiFilterGaussAdvancedBorder_16s_C3R, [1002](#)
 - nppiFilterGaussAdvancedBorder_16s_C4R, [1002](#)
 - nppiFilterGaussAdvancedBorder_16u_AC4R, [1003](#)
 - nppiFilterGaussAdvancedBorder_16u_C1R, [1003](#)
 - nppiFilterGaussAdvancedBorder_16u_C3R, [1004](#)
 - nppiFilterGaussAdvancedBorder_16u_C4R, [1004](#)
 - nppiFilterGaussAdvancedBorder_32f_AC4R, [1005](#)
 - nppiFilterGaussAdvancedBorder_32f_C1R, [1005](#)
 - nppiFilterGaussAdvancedBorder_32f_C3R, [1006](#)
 - nppiFilterGaussAdvancedBorder_32f_C4R, [1006](#)
 - nppiFilterGaussAdvancedBorder_8u_AC4R, [1007](#)
 - nppiFilterGaussAdvancedBorder_8u_C1R, [1007](#)
 - nppiFilterGaussAdvancedBorder_8u_C3R, [1008](#)
 - nppiFilterGaussAdvancedBorder_8u_C4R, [1008](#)
 - nppiFilterGaussBorder_16s_AC4R, [1009](#)
 - nppiFilterGaussBorder_16s_C1R, [1009](#)
 - nppiFilterGaussBorder_16s_C3R, [1010](#)
 - nppiFilterGaussBorder_16s_C4R, [1010](#)
 - nppiFilterGaussBorder_16u_AC4R, [1011](#)
 - nppiFilterGaussBorder_16u_C1R, [1011](#)
 - nppiFilterGaussBorder_16u_C3R, [1011](#)
 - nppiFilterGaussBorder_16u_C4R, [1012](#)
 - nppiFilterGaussBorder_32f_AC4R, [1012](#)
 - nppiFilterGaussBorder_32f_C1R, [1013](#)
 - nppiFilterGaussBorder_32f_C3R, [1013](#)
 - nppiFilterGaussBorder_32f_C4R, [1014](#)
 - nppiFilterGaussBorder_8u_AC4R, [1014](#)
 - nppiFilterGaussBorder_8u_C1R, [1015](#)
 - nppiFilterGaussBorder_8u_C3R, [1015](#)
 - nppiFilterGaussBorder_8u_C4R, [1016](#)
 - nppiFilterHighPass_16s_AC4R, [1016](#)
 - nppiFilterHighPass_16s_C1R, [1017](#)
 - nppiFilterHighPass_16s_C3R, [1017](#)
 - nppiFilterHighPass_16s_C4R, [1017](#)
 - nppiFilterHighPass_16u_AC4R, [1018](#)

- [nppiFilterHighPass_16u_C1R](#), [1018](#)
- [nppiFilterHighPass_16u_C3R](#), [1018](#)
- [nppiFilterHighPass_16u_C4R](#), [1019](#)
- [nppiFilterHighPass_32f_AC4R](#), [1019](#)
- [nppiFilterHighPass_32f_C1R](#), [1019](#)
- [nppiFilterHighPass_32f_C3R](#), [1020](#)
- [nppiFilterHighPass_32f_C4R](#), [1020](#)
- [nppiFilterHighPass_8u_AC4R](#), [1020](#)
- [nppiFilterHighPass_8u_C1R](#), [1021](#)
- [nppiFilterHighPass_8u_C3R](#), [1021](#)
- [nppiFilterHighPass_8u_C4R](#), [1021](#)
- [nppiFilterHighPassBorder_16s_AC4R](#), [1022](#)
- [nppiFilterHighPassBorder_16s_C1R](#), [1022](#)
- [nppiFilterHighPassBorder_16s_C3R](#), [1023](#)
- [nppiFilterHighPassBorder_16s_C4R](#), [1023](#)
- [nppiFilterHighPassBorder_16u_AC4R](#), [1024](#)
- [nppiFilterHighPassBorder_16u_C1R](#), [1024](#)
- [nppiFilterHighPassBorder_16u_C3R](#), [1025](#)
- [nppiFilterHighPassBorder_16u_C4R](#), [1025](#)
- [nppiFilterHighPassBorder_32f_AC4R](#), [1025](#)
- [nppiFilterHighPassBorder_32f_C1R](#), [1026](#)
- [nppiFilterHighPassBorder_32f_C3R](#), [1026](#)
- [nppiFilterHighPassBorder_32f_C4R](#), [1027](#)
- [nppiFilterHighPassBorder_8u_AC4R](#), [1027](#)
- [nppiFilterHighPassBorder_8u_C1R](#), [1028](#)
- [nppiFilterHighPassBorder_8u_C3R](#), [1028](#)
- [nppiFilterHighPassBorder_8u_C4R](#), [1029](#)
- [nppiFilterLaplace_16s_AC4R](#), [1029](#)
- [nppiFilterLaplace_16s_C1R](#), [1030](#)
- [nppiFilterLaplace_16s_C3R](#), [1030](#)
- [nppiFilterLaplace_16s_C4R](#), [1030](#)
- [nppiFilterLaplace_32f_AC4R](#), [1031](#)
- [nppiFilterLaplace_32f_C1R](#), [1031](#)
- [nppiFilterLaplace_32f_C3R](#), [1031](#)
- [nppiFilterLaplace_32f_C4R](#), [1032](#)
- [nppiFilterLaplace_8s16s_C1R](#), [1032](#)
- [nppiFilterLaplace_8u16s_C1R](#), [1032](#)
- [nppiFilterLaplace_8u_AC4R](#), [1033](#)
- [nppiFilterLaplace_8u_C1R](#), [1033](#)
- [nppiFilterLaplace_8u_C3R](#), [1033](#)
- [nppiFilterLaplace_8u_C4R](#), [1034](#)
- [nppiFilterLaplaceBorder_16s_AC4R](#), [1034](#)
- [nppiFilterLaplaceBorder_16s_C1R](#), [1035](#)
- [nppiFilterLaplaceBorder_16s_C3R](#), [1035](#)
- [nppiFilterLaplaceBorder_16s_C4R](#), [1036](#)
- [nppiFilterLaplaceBorder_32f_AC4R](#), [1036](#)
- [nppiFilterLaplaceBorder_32f_C1R](#), [1036](#)
- [nppiFilterLaplaceBorder_32f_C3R](#), [1037](#)
- [nppiFilterLaplaceBorder_32f_C4R](#), [1037](#)
- [nppiFilterLaplaceBorder_8s16s_C1R](#), [1038](#)
- [nppiFilterLaplaceBorder_8u16s_C1R](#), [1038](#)
- [nppiFilterLaplaceBorder_8u_AC4R](#), [1039](#)
- [nppiFilterLaplaceBorder_8u_C1R](#), [1039](#)
- [nppiFilterLaplaceBorder_8u_C3R](#), [1040](#)
- [nppiFilterLaplaceBorder_8u_C4R](#), [1040](#)
- [nppiFilterLowPass_16s_AC4R](#), [1041](#)
- [nppiFilterLowPass_16s_C1R](#), [1041](#)
- [nppiFilterLowPass_16s_C3R](#), [1042](#)
- [nppiFilterLowPass_16s_C4R](#), [1042](#)
- [nppiFilterLowPass_16u_AC4R](#), [1042](#)
- [nppiFilterLowPass_16u_C1R](#), [1043](#)
- [nppiFilterLowPass_16u_C3R](#), [1043](#)
- [nppiFilterLowPass_16u_C4R](#), [1043](#)
- [nppiFilterLowPass_32f_AC4R](#), [1044](#)
- [nppiFilterLowPass_32f_C1R](#), [1044](#)
- [nppiFilterLowPass_32f_C3R](#), [1044](#)
- [nppiFilterLowPass_32f_C4R](#), [1045](#)
- [nppiFilterLowPass_8u_AC4R](#), [1045](#)
- [nppiFilterLowPass_8u_C1R](#), [1045](#)
- [nppiFilterLowPass_8u_C3R](#), [1046](#)
- [nppiFilterLowPass_8u_C4R](#), [1046](#)
- [nppiFilterLowPassBorder_16s_AC4R](#), [1046](#)
- [nppiFilterLowPassBorder_16s_C1R](#), [1047](#)
- [nppiFilterLowPassBorder_16s_C3R](#), [1047](#)
- [nppiFilterLowPassBorder_16s_C4R](#), [1048](#)
- [nppiFilterLowPassBorder_16u_AC4R](#), [1048](#)
- [nppiFilterLowPassBorder_16u_C1R](#), [1049](#)
- [nppiFilterLowPassBorder_16u_C3R](#), [1049](#)
- [nppiFilterLowPassBorder_16u_C4R](#), [1050](#)
- [nppiFilterLowPassBorder_32f_AC4R](#), [1050](#)
- [nppiFilterLowPassBorder_32f_C1R](#), [1051](#)
- [nppiFilterLowPassBorder_32f_C3R](#), [1051](#)
- [nppiFilterLowPassBorder_32f_C4R](#), [1052](#)
- [nppiFilterLowPassBorder_8u_AC4R](#), [1052](#)
- [nppiFilterLowPassBorder_8u_C1R](#), [1053](#)
- [nppiFilterLowPassBorder_8u_C3R](#), [1053](#)
- [nppiFilterLowPassBorder_8u_C4R](#), [1054](#)
- [nppiFilterRobertsDown_16s_AC4R](#), [1054](#)
- [nppiFilterRobertsDown_16s_C1R](#), [1055](#)
- [nppiFilterRobertsDown_16s_C3R](#), [1055](#)
- [nppiFilterRobertsDown_16s_C4R](#), [1055](#)
- [nppiFilterRobertsDown_32f_AC4R](#), [1056](#)
- [nppiFilterRobertsDown_32f_C1R](#), [1056](#)
- [nppiFilterRobertsDown_32f_C3R](#), [1056](#)
- [nppiFilterRobertsDown_32f_C4R](#), [1057](#)
- [nppiFilterRobertsDown_8u_AC4R](#), [1057](#)
- [nppiFilterRobertsDown_8u_C1R](#), [1057](#)
- [nppiFilterRobertsDown_8u_C3R](#), [1058](#)
- [nppiFilterRobertsDown_8u_C4R](#), [1058](#)
- [nppiFilterRobertsDownBorder_16s_AC4R](#), [1058](#)
- [nppiFilterRobertsDownBorder_16s_C1R](#), [1059](#)
- [nppiFilterRobertsDownBorder_16s_C3R](#), [1059](#)
- [nppiFilterRobertsDownBorder_16s_C4R](#), [1060](#)

- [nppiFilterRobertsDownBorder_32f_AC4R, 1060](#)
- [nppiFilterRobertsDownBorder_32f_C1R, 1061](#)
- [nppiFilterRobertsDownBorder_32f_C3R, 1061](#)
- [nppiFilterRobertsDownBorder_32f_C4R, 1061](#)
- [nppiFilterRobertsDownBorder_8u_AC4R, 1062](#)
- [nppiFilterRobertsDownBorder_8u_C1R, 1062](#)
- [nppiFilterRobertsDownBorder_8u_C3R, 1063](#)
- [nppiFilterRobertsDownBorder_8u_C4R, 1063](#)
- [nppiFilterRobertsUp_16s_AC4R, 1064](#)
- [nppiFilterRobertsUp_16s_C1R, 1064](#)
- [nppiFilterRobertsUp_16s_C3R, 1064](#)
- [nppiFilterRobertsUp_16s_C4R, 1065](#)
- [nppiFilterRobertsUp_32f_AC4R, 1065](#)
- [nppiFilterRobertsUp_32f_C1R, 1065](#)
- [nppiFilterRobertsUp_32f_C3R, 1066](#)
- [nppiFilterRobertsUp_32f_C4R, 1066](#)
- [nppiFilterRobertsUp_8u_AC4R, 1066](#)
- [nppiFilterRobertsUp_8u_C1R, 1067](#)
- [nppiFilterRobertsUp_8u_C3R, 1067](#)
- [nppiFilterRobertsUp_8u_C4R, 1067](#)
- [nppiFilterRobertsUpBorder_16s_AC4R, 1068](#)
- [nppiFilterRobertsUpBorder_16s_C1R, 1068](#)
- [nppiFilterRobertsUpBorder_16s_C3R, 1069](#)
- [nppiFilterRobertsUpBorder_16s_C4R, 1069](#)
- [nppiFilterRobertsUpBorder_32f_AC4R, 1069](#)
- [nppiFilterRobertsUpBorder_32f_C1R, 1070](#)
- [nppiFilterRobertsUpBorder_32f_C3R, 1070](#)
- [nppiFilterRobertsUpBorder_32f_C4R, 1071](#)
- [nppiFilterRobertsUpBorder_8u_AC4R, 1071](#)
- [nppiFilterRobertsUpBorder_8u_C1R, 1072](#)
- [nppiFilterRobertsUpBorder_8u_C3R, 1072](#)
- [nppiFilterRobertsUpBorder_8u_C4R, 1072](#)
- [nppiFilterSharpen_16s_AC4R, 1073](#)
- [nppiFilterSharpen_16s_C1R, 1073](#)
- [nppiFilterSharpen_16s_C3R, 1074](#)
- [nppiFilterSharpen_16s_C4R, 1074](#)
- [nppiFilterSharpen_16u_AC4R, 1074](#)
- [nppiFilterSharpen_16u_C1R, 1075](#)
- [nppiFilterSharpen_16u_C3R, 1075](#)
- [nppiFilterSharpen_16u_C4R, 1075](#)
- [nppiFilterSharpen_32f_AC4R, 1076](#)
- [nppiFilterSharpen_32f_C1R, 1076](#)
- [nppiFilterSharpen_32f_C3R, 1076](#)
- [nppiFilterSharpen_32f_C4R, 1077](#)
- [nppiFilterSharpen_8u_AC4R, 1077](#)
- [nppiFilterSharpen_8u_C1R, 1077](#)
- [nppiFilterSharpen_8u_C3R, 1078](#)
- [nppiFilterSharpen_8u_C4R, 1078](#)
- [nppiFilterSharpenBorder_16s_AC4R, 1078](#)
- [nppiFilterSharpenBorder_16s_C1R, 1079](#)
- [nppiFilterSharpenBorder_16s_C3R, 1079](#)
- [nppiFilterSharpenBorder_16s_C4R, 1080](#)
- [nppiFilterSharpenBorder_16u_AC4R, 1080](#)
- [nppiFilterSharpenBorder_16u_C1R, 1081](#)
- [nppiFilterSharpenBorder_16u_C3R, 1081](#)
- [nppiFilterSharpenBorder_16u_C4R, 1081](#)
- [nppiFilterSharpenBorder_32f_AC4R, 1082](#)
- [nppiFilterSharpenBorder_32f_C1R, 1082](#)
- [nppiFilterSharpenBorder_32f_C3R, 1083](#)
- [nppiFilterSharpenBorder_32f_C4R, 1083](#)
- [nppiFilterSharpenBorder_8u_AC4R, 1084](#)
- [nppiFilterSharpenBorder_8u_C1R, 1084](#)
- [nppiFilterSharpenBorder_8u_C3R, 1084](#)
- [nppiFilterSharpenBorder_8u_C4R, 1085](#)
- [nppiFilterSobelCrossBorder_32f_C1R, 1085](#)
- [nppiFilterSobelCrossBorder_8s16s_C1R, 1086](#)
- [nppiFilterSobelCrossBorder_8u16s_C1R, 1086](#)
- [nppiFilterSobelVertSecondBorder_32f_C1R, 1087](#)
- [nppiFilterSobelVertSecondBorder_8s16s_C1R, 1087](#)
- [nppiFilterSobelVertSecondBorder_8u16s_C1R, 1088](#)
- [nppiFilterUnsharpBorder_16s_AC4R, 1088](#)
- [nppiFilterUnsharpBorder_16s_C1R, 1089](#)
- [nppiFilterUnsharpBorder_16s_C3R, 1089](#)
- [nppiFilterUnsharpBorder_16s_C4R, 1090](#)
- [nppiFilterUnsharpBorder_16u_AC4R, 1090](#)
- [nppiFilterUnsharpBorder_16u_C1R, 1091](#)
- [nppiFilterUnsharpBorder_16u_C3R, 1092](#)
- [nppiFilterUnsharpBorder_16u_C4R, 1092](#)
- [nppiFilterUnsharpBorder_32f_AC4R, 1093](#)
- [nppiFilterUnsharpBorder_32f_C1R, 1093](#)
- [nppiFilterUnsharpBorder_32f_C3R, 1094](#)
- [nppiFilterUnsharpBorder_32f_C4R, 1094](#)
- [nppiFilterUnsharpBorder_8u_AC4R, 1095](#)
- [nppiFilterUnsharpBorder_8u_C1R, 1096](#)
- [nppiFilterUnsharpBorder_8u_C3R, 1096](#)
- [nppiFilterUnsharpBorder_8u_C4R, 1097](#)
- [nppiFilterUnsharpGetBufferSize_16s_AC4R, 1097](#)
- [nppiFilterUnsharpGetBufferSize_16s_C1R, 1098](#)
- [nppiFilterUnsharpGetBufferSize_16s_C3R, 1098](#)
- [nppiFilterUnsharpGetBufferSize_16s_C4R, 1098](#)
- [nppiFilterUnsharpGetBufferSize_16u_AC4R, 1098](#)
- [nppiFilterUnsharpGetBufferSize_16u_C1R, 1099](#)

- Copyright ©2009–2014 NVIDIA Corporation

- nppiHistogramRangeGetBufferSize_32f_C1R, 2122
- nppiHistogramRangeGetBufferSize_32f_C3R, 2122
- nppiHistogramRangeGetBufferSize_32f_C4R, 2122
- nppiHistogramRangeGetBufferSize_8u_-AC4R, 2123
- nppiHistogramRangeGetBufferSize_8u_C1R, 2123
- nppiHistogramRangeGetBufferSize_8u_C3R, 2123
- nppiHistogramRangeGetBufferSize_8u_C4R, 2124
- image_inf_norm
 - nppiNorm_Inf_16s_AC4R, 1845
 - nppiNorm_Inf_16s_C1R, 1845
 - nppiNorm_Inf_16s_C3R, 1845
 - nppiNorm_Inf_16s_C4R, 1846
 - nppiNorm_Inf_16u_AC4R, 1846
 - nppiNorm_Inf_16u_C1MR, 1846
 - nppiNorm_Inf_16u_C1R, 1847
 - nppiNorm_Inf_16u_C3CMR, 1847
 - nppiNorm_Inf_16u_C3R, 1848
 - nppiNorm_Inf_16u_C4R, 1848
 - nppiNorm_Inf_32f_AC4R, 1848
 - nppiNorm_Inf_32f_C1MR, 1849
 - nppiNorm_Inf_32f_C1R, 1849
 - nppiNorm_Inf_32f_C3CMR, 1850
 - nppiNorm_Inf_32f_C3R, 1850
 - nppiNorm_Inf_32f_C4R, 1850
 - nppiNorm_Inf_32s_C1R, 1851
 - nppiNorm_Inf_8s_C1MR, 1851
 - nppiNorm_Inf_8s_C3CMR, 1852
 - nppiNorm_Inf_8u_AC4R, 1852
 - nppiNorm_Inf_8u_C1MR, 1852
 - nppiNorm_Inf_8u_C1R, 1853
 - nppiNorm_Inf_8u_C3CMR, 1853
 - nppiNorm_Inf_8u_C3R, 1854
 - nppiNorm_Inf_8u_C4R, 1854
 - nppiNormInfGetBufferHostSize_16s_AC4R, 1854
 - nppiNormInfGetBufferHostSize_16s_C1R, 1855
 - nppiNormInfGetBufferHostSize_16s_C3R, 1855
 - nppiNormInfGetBufferHostSize_16s_C4R, 1855
 - nppiNormInfGetBufferHostSize_16u_AC4R, 1856
 - nppiNormInfGetBufferHostSize_16u_C1MR, 1856
 - nppiNormInfGetBufferHostSize_16u_C1R, 1856
 - nppiNormInfGetBufferHostSize_16u_-C3CMR, 1856
 - nppiNormInfGetBufferHostSize_16u_C3R, 1857
 - nppiNormInfGetBufferHostSize_16u_C4R, 1857
 - nppiNormInfGetBufferHostSize_32f_AC4R, 1857
 - nppiNormInfGetBufferHostSize_32f_C1MR, 1858
 - nppiNormInfGetBufferHostSize_32f_C1R, 1858
 - nppiNormInfGetBufferHostSize_32f_-C3CMR, 1858
 - nppiNormInfGetBufferHostSize_32f_C3R, 1858
 - nppiNormInfGetBufferHostSize_32f_C4R, 1859
 - nppiNormInfGetBufferHostSize_32s_C1R, 1859
 - nppiNormInfGetBufferHostSize_8s_C1MR, 1859
 - nppiNormInfGetBufferHostSize_8s_C3CMR, 1860
 - nppiNormInfGetBufferHostSize_8u_AC4R, 1860
 - nppiNormInfGetBufferHostSize_8u_C1MR, 1860
 - nppiNormInfGetBufferHostSize_8u_C1R, 1860
 - nppiNormInfGetBufferHostSize_8u_C3CMR, 1861
 - nppiNormInfGetBufferHostSize_8u_C3R, 1861
 - nppiNormInfGetBufferHostSize_8u_C4R, 1861
- image_inf_normdiff
 - nppiNormDiff_Inf_16s_AC4R, 1909
 - nppiNormDiff_Inf_16s_C1R, 1909
 - nppiNormDiff_Inf_16s_C3R, 1910
 - nppiNormDiff_Inf_16s_C4R, 1910
 - nppiNormDiff_Inf_16u_AC4R, 1911
 - nppiNormDiff_Inf_16u_C1MR, 1911
 - nppiNormDiff_Inf_16u_C1R, 1912
 - nppiNormDiff_Inf_16u_C3CMR, 1912
 - nppiNormDiff_Inf_16u_C3R, 1913
 - nppiNormDiff_Inf_16u_C4R, 1913
 - nppiNormDiff_Inf_32f_AC4R, 1913
 - nppiNormDiff_Inf_32f_C1MR, 1914
 - nppiNormDiff_Inf_32f_C1R, 1914
 - nppiNormDiff_Inf_32f_C3CMR, 1915
 - nppiNormDiff_Inf_32f_C3R, 1915
 - nppiNormDiff_Inf_32f_C4R, 1916
 - nppiNormDiff_Inf_8s_C1MR, 1916

- nppiNormDiff_Inf_8s_C3CMR, [1917](#)
- nppiNormDiff_Inf_8u_AC4R, [1917](#)
- nppiNormDiff_Inf_8u_C1MR, [1918](#)
- nppiNormDiff_Inf_8u_C1R, [1918](#)
- nppiNormDiff_Inf_8u_C3CMR, [1919](#)
- nppiNormDiff_Inf_8u_C3R, [1919](#)
- nppiNormDiff_Inf_8u_C4R, [1920](#)
- nppiNormDiffInfGetBufferHostSize_16s_-AC4R, [1920](#)
- nppiNormDiffInfGetBufferHostSize_16s_-C1R, [1920](#)
- nppiNormDiffInfGetBufferHostSize_16s_-C3R, [1921](#)
- nppiNormDiffInfGetBufferHostSize_16s_-C4R, [1921](#)
- nppiNormDiffInfGetBufferHostSize_16u_-AC4R, [1921](#)
- nppiNormDiffInfGetBufferHostSize_16u_-C1MR, [1922](#)
- nppiNormDiffInfGetBufferHostSize_16u_-C1R, [1922](#)
- nppiNormDiffInfGetBufferHostSize_16u_-C3CMR, [1922](#)
- nppiNormDiffInfGetBufferHostSize_16u_-C3R, [1922](#)
- nppiNormDiffInfGetBufferHostSize_16u_-C4R, [1923](#)
- nppiNormDiffInfGetBufferHostSize_32f_-AC4R, [1923](#)
- nppiNormDiffInfGetBufferHostSize_32f_-C1MR, [1923](#)
- nppiNormDiffInfGetBufferHostSize_32f_-C1R, [1924](#)
- nppiNormDiffInfGetBufferHostSize_32f_-C3CMR, [1924](#)
- nppiNormDiffInfGetBufferHostSize_32f_-C3R, [1924](#)
- nppiNormDiffInfGetBufferHostSize_32f_-C4R, [1924](#)
- nppiNormDiffInfGetBufferHostSize_8s_-C1MR, [1925](#)
- nppiNormDiffInfGetBufferHostSize_8s_-C3CMR, [1925](#)
- nppiNormDiffInfGetBufferHostSize_8u_-AC4R, [1925](#)
- nppiNormDiffInfGetBufferHostSize_8u_-C1MR, [1926](#)
- nppiNormDiffInfGetBufferHostSize_8u_C1R, [1926](#)
- nppiNormDiffInfGetBufferHostSize_8u_-C3CMR, [1926](#)
- nppiNormDiffInfGetBufferHostSize_8u_C3R, [1926](#)
- nppiNormDiffInfGetBufferHostSize_8u_C4R, [1927](#)
- image_inf_normrel
 - nppiNormRel_Inf_16s_AC4R, [1978](#)
 - nppiNormRel_Inf_16s_C1R, [1978](#)
 - nppiNormRel_Inf_16s_C3R, [1979](#)
 - nppiNormRel_Inf_16s_C4R, [1979](#)
 - nppiNormRel_Inf_16u_AC4R, [1980](#)
 - nppiNormRel_Inf_16u_C1MR, [1980](#)
 - nppiNormRel_Inf_16u_C1R, [1981](#)
 - nppiNormRel_Inf_16u_C3CMR, [1981](#)
 - nppiNormRel_Inf_16u_C3R, [1982](#)
 - nppiNormRel_Inf_16u_C4R, [1982](#)
 - nppiNormRel_Inf_32f_AC4R, [1982](#)
 - nppiNormRel_Inf_32f_C1MR, [1983](#)
 - nppiNormRel_Inf_32f_C1R, [1983](#)
 - nppiNormRel_Inf_32f_C3CMR, [1984](#)
 - nppiNormRel_Inf_32f_C3R, [1984](#)
 - nppiNormRel_Inf_32f_C4R, [1985](#)
 - nppiNormRel_Inf_8s_C1MR, [1985](#)
 - nppiNormRel_Inf_8s_C3CMR, [1986](#)
 - nppiNormRel_Inf_8u_AC4R, [1986](#)
 - nppiNormRel_Inf_8u_C1MR, [1987](#)
 - nppiNormRel_Inf_8u_C1R, [1987](#)
 - nppiNormRel_Inf_8u_C3CMR, [1988](#)
 - nppiNormRel_Inf_8u_C3R, [1988](#)
 - nppiNormRel_Inf_8u_C4R, [1989](#)
 - nppiNormRelInfGetBufferHostSize_16s_-AC4R, [1989](#)
 - nppiNormRelInfGetBufferHostSize_16s_-C1R, [1990](#)
 - nppiNormRelInfGetBufferHostSize_16s_-C3R, [1990](#)
 - nppiNormRelInfGetBufferHostSize_16s_-C4R, [1990](#)
 - nppiNormRelInfGetBufferHostSize_16u_-AC4R, [1990](#)
 - nppiNormRelInfGetBufferHostSize_16u_-C1MR, [1991](#)
 - nppiNormRelInfGetBufferHostSize_16u_-C1R, [1991](#)
 - nppiNormRelInfGetBufferHostSize_16u_-C3CMR, [1991](#)
 - nppiNormRelInfGetBufferHostSize_16u_-C3R, [1992](#)
 - nppiNormRelInfGetBufferHostSize_16u_-C4R, [1992](#)
 - nppiNormRelInfGetBufferHostSize_32f_-AC4R, [1992](#)
 - nppiNormRelInfGetBufferHostSize_32f_-C1MR, [1992](#)
 - nppiNormRelInfGetBufferHostSize_32f_C1R, [1993](#)

- nppiNormRelInfGetBufferHostSize_32f_-C3CMR, [1993](#)
- nppiNormRelInfGetBufferHostSize_32f_C3R, [1993](#)
- nppiNormRelInfGetBufferHostSize_32f_C4R, [1994](#)
- nppiNormRelInfGetBufferHostSize_32s_-C1R, [1994](#)
- nppiNormRelInfGetBufferHostSize_8s_-C1MR, [1994](#)
- nppiNormRelInfGetBufferHostSize_8s_-C3CMR, [1994](#)
- nppiNormRelInfGetBufferHostSize_8u_-AC4R, [1995](#)
- nppiNormRelInfGetBufferHostSize_8u_-C1MR, [1995](#)
- nppiNormRelInfGetBufferHostSize_8u_C1R, [1995](#)
- nppiNormRelInfGetBufferHostSize_8u_-C3CMR, [1996](#)
- nppiNormRelInfGetBufferHostSize_8u_C3R, [1996](#)
- nppiNormRelInfGetBufferHostSize_8u_C4R, [1996](#)
- image_integral
 - nppiIntegral_8u32f_C1R, [2088](#)
 - nppiIntegral_8u32s_C1R, [2088](#)
- image_L1_norm
 - nppiNorm_L1_16s_AC4R, [1867](#)
 - nppiNorm_L1_16s_C1R, [1867](#)
 - nppiNorm_L1_16s_C3R, [1867](#)
 - nppiNorm_L1_16s_C4R, [1868](#)
 - nppiNorm_L1_16u_AC4R, [1868](#)
 - nppiNorm_L1_16u_C1MR, [1868](#)
 - nppiNorm_L1_16u_C1R, [1869](#)
 - nppiNorm_L1_16u_C3CMR, [1869](#)
 - nppiNorm_L1_16u_C3R, [1870](#)
 - nppiNorm_L1_16u_C4R, [1870](#)
 - nppiNorm_L1_32f_AC4R, [1870](#)
 - nppiNorm_L1_32f_C1MR, [1871](#)
 - nppiNorm_L1_32f_C1R, [1871](#)
 - nppiNorm_L1_32f_C3CMR, [1871](#)
 - nppiNorm_L1_32f_C3R, [1872](#)
 - nppiNorm_L1_32f_C4R, [1872](#)
 - nppiNorm_L1_8s_C1MR, [1873](#)
 - nppiNorm_L1_8s_C3CMR, [1873](#)
 - nppiNorm_L1_8u_AC4R, [1873](#)
 - nppiNorm_L1_8u_C1MR, [1874](#)
 - nppiNorm_L1_8u_C1R, [1874](#)
 - nppiNorm_L1_8u_C3CMR, [1875](#)
 - nppiNorm_L1_8u_C3R, [1875](#)
 - nppiNorm_L1_8u_C4R, [1875](#)
 - nppiNormL1GetBufferHostSize_16s_AC4R, [1876](#)
 - nppiNormL1GetBufferHostSize_16s_C1R, [1876](#)
 - nppiNormL1GetBufferHostSize_16s_C3R, [1876](#)
 - nppiNormL1GetBufferHostSize_16s_C4R, [1877](#)
 - nppiNormL1GetBufferHostSize_16u_AC4R, [1877](#)
 - nppiNormL1GetBufferHostSize_16u_C1MR, [1877](#)
 - nppiNormL1GetBufferHostSize_16u_C1R, [1878](#)
 - nppiNormL1GetBufferHostSize_16u_-C3CMR, [1878](#)
 - nppiNormL1GetBufferHostSize_16u_C3R, [1878](#)
 - nppiNormL1GetBufferHostSize_16u_C4R, [1878](#)
 - nppiNormL1GetBufferHostSize_32f_AC4R, [1879](#)
 - nppiNormL1GetBufferHostSize_32f_C1MR, [1879](#)
 - nppiNormL1GetBufferHostSize_32f_C1R, [1879](#)
 - nppiNormL1GetBufferHostSize_32f_-C3CMR, [1880](#)
 - nppiNormL1GetBufferHostSize_32f_C3R, [1880](#)
 - nppiNormL1GetBufferHostSize_32f_C4R, [1880](#)
 - nppiNormL1GetBufferHostSize_8s_C1MR, [1880](#)
 - nppiNormL1GetBufferHostSize_8s_C3CMR, [1881](#)
 - nppiNormL1GetBufferHostSize_8u_AC4R, [1881](#)
 - nppiNormL1GetBufferHostSize_8u_C1MR, [1881](#)
 - nppiNormL1GetBufferHostSize_8u_C1R, [1882](#)
 - nppiNormL1GetBufferHostSize_8u_C3CMR, [1882](#)
 - nppiNormL1GetBufferHostSize_8u_C3R, [1882](#)
 - nppiNormL1GetBufferHostSize_8u_C4R, [1882](#)
 - image_L1_normdiff
 - nppiNormDiff_L1_16s_AC4R, [1932](#)
 - nppiNormDiff_L1_16s_C1R, [1932](#)
 - nppiNormDiff_L1_16s_C3R, [1933](#)
 - nppiNormDiff_L1_16s_C4R, [1933](#)
 - nppiNormDiff_L1_16u_AC4R, [1934](#)
 - nppiNormDiff_L1_16u_C1MR, [1934](#)
 - nppiNormDiff_L1_16u_C1R, [1934](#)

- [nppiNormDiff_L1_16u_C3CMR](#), 1935
- [nppiNormDiff_L1_16u_C3R](#), 1935
- [nppiNormDiff_L1_16u_C4R](#), 1936
- [nppiNormDiff_L1_32f_AC4R](#), 1936
- [nppiNormDiff_L1_32f_C1MR](#), 1937
- [nppiNormDiff_L1_32f_C1R](#), 1937
- [nppiNormDiff_L1_32f_C3CMR](#), 1938
- [nppiNormDiff_L1_32f_C3R](#), 1938
- [nppiNormDiff_L1_32f_C4R](#), 1939
- [nppiNormDiff_L1_8s_C1MR](#), 1939
- [nppiNormDiff_L1_8s_C3CMR](#), 1940
- [nppiNormDiff_L1_8u_AC4R](#), 1940
- [nppiNormDiff_L1_8u_C1MR](#), 1941
- [nppiNormDiff_L1_8u_C1R](#), 1941
- [nppiNormDiff_L1_8u_C3CMR](#), 1941
- [nppiNormDiff_L1_8u_C3R](#), 1942
- [nppiNormDiff_L1_8u_C4R](#), 1942
- [nppiNormDiffL1GetBufferHostSize_16s_-AC4R](#), 1943
- [nppiNormDiffL1GetBufferHostSize_16s_-C1R](#), 1943
- [nppiNormDiffL1GetBufferHostSize_16s_-C3R](#), 1943
- [nppiNormDiffL1GetBufferHostSize_16s_-C4R](#), 1944
- [nppiNormDiffL1GetBufferHostSize_16u_-AC4R](#), 1944
- [nppiNormDiffL1GetBufferHostSize_16u_-C1MR](#), 1944
- [nppiNormDiffL1GetBufferHostSize_16u_-C1R](#), 1945
- [nppiNormDiffL1GetBufferHostSize_16u_-C3CMR](#), 1945
- [nppiNormDiffL1GetBufferHostSize_16u_-C3R](#), 1945
- [nppiNormDiffL1GetBufferHostSize_16u_-C4R](#), 1945
- [nppiNormDiffL1GetBufferHostSize_32f_-AC4R](#), 1946
- [nppiNormDiffL1GetBufferHostSize_32f_-C1MR](#), 1946
- [nppiNormDiffL1GetBufferHostSize_32f_-C1R](#), 1946
- [nppiNormDiffL1GetBufferHostSize_32f_-C3CMR](#), 1947
- [nppiNormDiffL1GetBufferHostSize_32f_-C3R](#), 1947
- [nppiNormDiffL1GetBufferHostSize_32f_-C4R](#), 1947
- [nppiNormDiffL1GetBufferHostSize_8s_-C1MR](#), 1947
- [nppiNormDiffL1GetBufferHostSize_8s_-C3CMR](#), 1948
- [nppiNormDiffL1GetBufferHostSize_8u_-AC4R](#), 1948
- [nppiNormDiffL1GetBufferHostSize_8u_-C1MR](#), 1948
- [nppiNormDiffL1GetBufferHostSize_8u_C1R](#), 1949
- [nppiNormDiffL1GetBufferHostSize_8u_-C3CMR](#), 1949
- [nppiNormDiffL1GetBufferHostSize_8u_C3R](#), 1949
- [nppiNormDiffL1GetBufferHostSize_8u_C4R](#), 1949
- [image_L1_normrel](#)
 - [nppiNormRel_L1_16s_AC4R](#), 2001
 - [nppiNormRel_L1_16s_C1R](#), 2001
 - [nppiNormRel_L1_16s_C3R](#), 2002
 - [nppiNormRel_L1_16s_C4R](#), 2002
 - [nppiNormRel_L1_16u_AC4R](#), 2003
 - [nppiNormRel_L1_16u_C1MR](#), 2003
 - [nppiNormRel_L1_16u_C1R](#), 2004
 - [nppiNormRel_L1_16u_C3CMR](#), 2004
 - [nppiNormRel_L1_16u_C3R](#), 2004
 - [nppiNormRel_L1_16u_C4R](#), 2005
 - [nppiNormRel_L1_32f_AC4R](#), 2005
 - [nppiNormRel_L1_32f_C1MR](#), 2006
 - [nppiNormRel_L1_32f_C1R](#), 2006
 - [nppiNormRel_L1_32f_C3CMR](#), 2007
 - [nppiNormRel_L1_32f_C3R](#), 2007
 - [nppiNormRel_L1_32f_C4R](#), 2008
 - [nppiNormRel_L1_8s_C1MR](#), 2008
 - [nppiNormRel_L1_8s_C3CMR](#), 2009
 - [nppiNormRel_L1_8u_AC4R](#), 2009
 - [nppiNormRel_L1_8u_C1MR](#), 2010
 - [nppiNormRel_L1_8u_C1R](#), 2010
 - [nppiNormRel_L1_8u_C3CMR](#), 2011
 - [nppiNormRel_L1_8u_C3R](#), 2011
 - [nppiNormRel_L1_8u_C4R](#), 2012
 - [nppiNormRelL1GetBufferHostSize_16s_-AC4R](#), 2012
 - [nppiNormRelL1GetBufferHostSize_16s_C1R](#), 2012
 - [nppiNormRelL1GetBufferHostSize_16s_C3R](#), 2013
 - [nppiNormRelL1GetBufferHostSize_16s_C4R](#), 2013
 - [nppiNormRelL1GetBufferHostSize_16u_-AC4R](#), 2013
 - [nppiNormRelL1GetBufferHostSize_16u_-C1MR](#), 2014
 - [nppiNormRelL1GetBufferHostSize_16u_-C1R](#), 2014
 - [nppiNormRelL1GetBufferHostSize_16u_-C3CMR](#), 2014

- nppiNormRelL1GetBufferHostSize_16u_-C3R, [2014](#)
- nppiNormRelL1GetBufferHostSize_16u_-C4R, [2015](#)
- nppiNormRelL1GetBufferHostSize_32f_-AC4R, [2015](#)
- nppiNormRelL1GetBufferHostSize_32f_-C1MR, [2015](#)
- nppiNormRelL1GetBufferHostSize_32f_C1R, [2016](#)
- nppiNormRelL1GetBufferHostSize_32f_-C3CMR, [2016](#)
- nppiNormRelL1GetBufferHostSize_32f_C3R, [2016](#)
- nppiNormRelL1GetBufferHostSize_32f_C4R, [2016](#)
- nppiNormRelL1GetBufferHostSize_8s_-C1MR, [2017](#)
- nppiNormRelL1GetBufferHostSize_8s_-C3CMR, [2017](#)
- nppiNormRelL1GetBufferHostSize_8u_-AC4R, [2017](#)
- nppiNormRelL1GetBufferHostSize_8u_-C1MR, [2018](#)
- nppiNormRelL1GetBufferHostSize_8u_C1R, [2018](#)
- nppiNormRelL1GetBufferHostSize_8u_-C3CMR, [2018](#)
- nppiNormRelL1GetBufferHostSize_8u_C3R, [2018](#)
- nppiNormRelL1GetBufferHostSize_8u_C4R, [2019](#)
- image_L2_norm
 - nppiNorm_L2_16s_AC4R, [1888](#)
 - nppiNorm_L2_16s_C1R, [1888](#)
 - nppiNorm_L2_16s_C3R, [1888](#)
 - nppiNorm_L2_16s_C4R, [1889](#)
 - nppiNorm_L2_16u_AC4R, [1889](#)
 - nppiNorm_L2_16u_C1MR, [1889](#)
 - nppiNorm_L2_16u_C1R, [1890](#)
 - nppiNorm_L2_16u_C3CMR, [1890](#)
 - nppiNorm_L2_16u_C3R, [1891](#)
 - nppiNorm_L2_16u_C4R, [1891](#)
 - nppiNorm_L2_32f_AC4R, [1891](#)
 - nppiNorm_L2_32f_C1MR, [1892](#)
 - nppiNorm_L2_32f_C1R, [1892](#)
 - nppiNorm_L2_32f_C3CMR, [1892](#)
 - nppiNorm_L2_32f_C3R, [1893](#)
 - nppiNorm_L2_32f_C4R, [1893](#)
 - nppiNorm_L2_8s_C1MR, [1894](#)
 - nppiNorm_L2_8s_C3CMR, [1894](#)
 - nppiNorm_L2_8u_AC4R, [1894](#)
 - nppiNorm_L2_8u_C1MR, [1895](#)
 - nppiNorm_L2_8u_C1R, [1895](#)
 - nppiNorm_L2_8u_C3CMR, [1896](#)
 - nppiNorm_L2_8u_C3R, [1896](#)
 - nppiNorm_L2_8u_C4R, [1896](#)
 - nppiNormL2GetBufferHostSize_16s_AC4R, [1897](#)
 - nppiNormL2GetBufferHostSize_16s_C1R, [1897](#)
 - nppiNormL2GetBufferHostSize_16s_C3R, [1897](#)
 - nppiNormL2GetBufferHostSize_16s_C4R, [1898](#)
 - nppiNormL2GetBufferHostSize_16u_AC4R, [1898](#)
 - nppiNormL2GetBufferHostSize_16u_C1MR, [1898](#)
 - nppiNormL2GetBufferHostSize_16u_C1R, [1899](#)
 - nppiNormL2GetBufferHostSize_16u_-C3CMR, [1899](#)
 - nppiNormL2GetBufferHostSize_16u_C3R, [1899](#)
 - nppiNormL2GetBufferHostSize_16u_C4R, [1899](#)
 - nppiNormL2GetBufferHostSize_32f_AC4R, [1900](#)
 - nppiNormL2GetBufferHostSize_32f_C1MR, [1900](#)
 - nppiNormL2GetBufferHostSize_32f_C1R, [1900](#)
 - nppiNormL2GetBufferHostSize_32f_-C3CMR, [1901](#)
 - nppiNormL2GetBufferHostSize_32f_C3R, [1901](#)
 - nppiNormL2GetBufferHostSize_32f_C4R, [1901](#)
 - nppiNormL2GetBufferHostSize_8s_C1MR, [1901](#)
 - nppiNormL2GetBufferHostSize_8s_C3CMR, [1902](#)
 - nppiNormL2GetBufferHostSize_8u_AC4R, [1902](#)
 - nppiNormL2GetBufferHostSize_8u_C1MR, [1902](#)
 - nppiNormL2GetBufferHostSize_8u_C1R, [1903](#)
 - nppiNormL2GetBufferHostSize_8u_C3CMR, [1903](#)
 - nppiNormL2GetBufferHostSize_8u_C3R, [1903](#)
 - nppiNormL2GetBufferHostSize_8u_C4R, [1903](#)
- image_L2_normdiff
 - nppiNormDiff_L2_16s_AC4R, [1955](#)
 - nppiNormDiff_L2_16s_C1R, [1955](#)

- [nppiNormDiff_L2_16s_C3R](#), [1956](#)
- [nppiNormDiff_L2_16s_C4R](#), [1956](#)
- [nppiNormDiff_L2_16u_AC4R](#), [1957](#)
- [nppiNormDiff_L2_16u_C1MR](#), [1957](#)
- [nppiNormDiff_L2_16u_C1R](#), [1957](#)
- [nppiNormDiff_L2_16u_C3CMR](#), [1958](#)
- [nppiNormDiff_L2_16u_C3R](#), [1958](#)
- [nppiNormDiff_L2_16u_C4R](#), [1959](#)
- [nppiNormDiff_L2_32f_AC4R](#), [1959](#)
- [nppiNormDiff_L2_32f_C1MR](#), [1960](#)
- [nppiNormDiff_L2_32f_C1R](#), [1960](#)
- [nppiNormDiff_L2_32f_C3CMR](#), [1961](#)
- [nppiNormDiff_L2_32f_C3R](#), [1961](#)
- [nppiNormDiff_L2_32f_C4R](#), [1962](#)
- [nppiNormDiff_L2_8s_C1MR](#), [1962](#)
- [nppiNormDiff_L2_8s_C3CMR](#), [1963](#)
- [nppiNormDiff_L2_8u_AC4R](#), [1963](#)
- [nppiNormDiff_L2_8u_C1MR](#), [1964](#)
- [nppiNormDiff_L2_8u_C1R](#), [1964](#)
- [nppiNormDiff_L2_8u_C3CMR](#), [1964](#)
- [nppiNormDiff_L2_8u_C3R](#), [1965](#)
- [nppiNormDiff_L2_8u_C4R](#), [1965](#)
- [nppiNormDiffL2GetBufferHostSize_16s_-AC4R](#), [1966](#)
- [nppiNormDiffL2GetBufferHostSize_16s_-C1R](#), [1966](#)
- [nppiNormDiffL2GetBufferHostSize_16s_-C3R](#), [1966](#)
- [nppiNormDiffL2GetBufferHostSize_16s_-C4R](#), [1967](#)
- [nppiNormDiffL2GetBufferHostSize_16u_-AC4R](#), [1967](#)
- [nppiNormDiffL2GetBufferHostSize_16u_-C1MR](#), [1967](#)
- [nppiNormDiffL2GetBufferHostSize_16u_-C1R](#), [1968](#)
- [nppiNormDiffL2GetBufferHostSize_16u_-C3CMR](#), [1968](#)
- [nppiNormDiffL2GetBufferHostSize_16u_-C3R](#), [1968](#)
- [nppiNormDiffL2GetBufferHostSize_16u_-C4R](#), [1968](#)
- [nppiNormDiffL2GetBufferHostSize_32f_-AC4R](#), [1969](#)
- [nppiNormDiffL2GetBufferHostSize_32f_-C1MR](#), [1969](#)
- [nppiNormDiffL2GetBufferHostSize_32f_-C1R](#), [1969](#)
- [nppiNormDiffL2GetBufferHostSize_32f_-C3CMR](#), [1970](#)
- [nppiNormDiffL2GetBufferHostSize_32f_-C3R](#), [1970](#)
- [nppiNormDiffL2GetBufferHostSize_32f_-C4R](#), [1970](#)
- [nppiNormDiffL2GetBufferHostSize_8s_-C1MR](#), [1970](#)
- [nppiNormDiffL2GetBufferHostSize_8s_-C3CMR](#), [1971](#)
- [nppiNormDiffL2GetBufferHostSize_8u_-AC4R](#), [1971](#)
- [nppiNormDiffL2GetBufferHostSize_8u_-C1MR](#), [1971](#)
- [nppiNormDiffL2GetBufferHostSize_8u_C1R](#), [1972](#)
- [nppiNormDiffL2GetBufferHostSize_8u_-C3CMR](#), [1972](#)
- [nppiNormDiffL2GetBufferHostSize_8u_C3R](#), [1972](#)
- [nppiNormDiffL2GetBufferHostSize_8u_C4R](#), [1972](#)
- [image_L2_normrel](#)
 - [nppiNormRel_L2_16s_AC4R](#), [2024](#)
 - [nppiNormRel_L2_16s_C1R](#), [2024](#)
 - [nppiNormRel_L2_16s_C3R](#), [2025](#)
 - [nppiNormRel_L2_16s_C4R](#), [2025](#)
 - [nppiNormRel_L2_16u_AC4R](#), [2026](#)
 - [nppiNormRel_L2_16u_C1MR](#), [2026](#)
 - [nppiNormRel_L2_16u_C1R](#), [2027](#)
 - [nppiNormRel_L2_16u_C3CMR](#), [2027](#)
 - [nppiNormRel_L2_16u_C3R](#), [2027](#)
 - [nppiNormRel_L2_16u_C4R](#), [2028](#)
 - [nppiNormRel_L2_32f_AC4R](#), [2028](#)
 - [nppiNormRel_L2_32f_C1MR](#), [2029](#)
 - [nppiNormRel_L2_32f_C1R](#), [2029](#)
 - [nppiNormRel_L2_32f_C3CMR](#), [2030](#)
 - [nppiNormRel_L2_32f_C3R](#), [2030](#)
 - [nppiNormRel_L2_32f_C4R](#), [2031](#)
 - [nppiNormRel_L2_8s_C1MR](#), [2031](#)
 - [nppiNormRel_L2_8s_C3CMR](#), [2032](#)
 - [nppiNormRel_L2_8u_AC4R](#), [2032](#)
 - [nppiNormRel_L2_8u_C1MR](#), [2033](#)
 - [nppiNormRel_L2_8u_C1R](#), [2033](#)
 - [nppiNormRel_L2_8u_C3CMR](#), [2034](#)
 - [nppiNormRel_L2_8u_C3R](#), [2034](#)
 - [nppiNormRel_L2_8u_C4R](#), [2035](#)
 - [nppiNormRelL2GetBufferHostSize_16s_-AC4R](#), [2035](#)
 - [nppiNormRelL2GetBufferHostSize_16s_C1R](#), [2035](#)
 - [nppiNormRelL2GetBufferHostSize_16s_C3R](#), [2036](#)
 - [nppiNormRelL2GetBufferHostSize_16s_C4R](#), [2036](#)
 - [nppiNormRelL2GetBufferHostSize_16u_-AC4R](#), [2036](#)
 - [nppiNormRelL2GetBufferHostSize_16u_-C1MR](#), [2037](#)

- nppiNormRelL2GetBufferHostSize_16u_-C1R, [2037](#)
- nppiNormRelL2GetBufferHostSize_16u_-C3CMR, [2037](#)
- nppiNormRelL2GetBufferHostSize_16u_-C3R, [2037](#)
- nppiNormRelL2GetBufferHostSize_16u_-C4R, [2038](#)
- nppiNormRelL2GetBufferHostSize_32f_-AC4R, [2038](#)
- nppiNormRelL2GetBufferHostSize_32f_-C1MR, [2038](#)
- nppiNormRelL2GetBufferHostSize_32f_-C1R, [2039](#)
- nppiNormRelL2GetBufferHostSize_32f_-C3CMR, [2039](#)
- nppiNormRelL2GetBufferHostSize_32f_-C3R, [2039](#)
- nppiNormRelL2GetBufferHostSize_32f_-C4R, [2039](#)
- nppiNormRelL2GetBufferHostSize_8s_-C1MR, [2040](#)
- nppiNormRelL2GetBufferHostSize_8s_-C3CMR, [2040](#)
- nppiNormRelL2GetBufferHostSize_8u_-AC4R, [2040](#)
- nppiNormRelL2GetBufferHostSize_8u_-C1MR, [2041](#)
- nppiNormRelL2GetBufferHostSize_8u_-C1R, [2041](#)
- nppiNormRelL2GetBufferHostSize_8u_-C3CMR, [2041](#)
- nppiNormRelL2GetBufferHostSize_8u_-C3R, [2041](#)
- nppiNormRelL2GetBufferHostSize_8u_-C4R, [2042](#)
- image_labeling_and_segmentation
 - NppiGraphcutState, [730](#)
- image_ln
 - nppiLn_16s_C1IRSfs, [358](#)
 - nppiLn_16s_C1RSfs, [358](#)
 - nppiLn_16s_C3IRSfs, [359](#)
 - nppiLn_16s_C3RSfs, [359](#)
 - nppiLn_16u_C1IRSfs, [359](#)
 - nppiLn_16u_C1RSfs, [360](#)
 - nppiLn_16u_C3IRSfs, [360](#)
 - nppiLn_16u_C3RSfs, [360](#)
 - nppiLn_32f_C1IR, [361](#)
 - nppiLn_32f_C1R, [361](#)
 - nppiLn_32f_C3IR, [361](#)
 - nppiLn_32f_C3R, [362](#)
 - nppiLn_8u_C1IRSfs, [362](#)
 - nppiLn_8u_C1RSfs, [362](#)
 - nppiLn_8u_C3IRSfs, [363](#)
 - nppiLn_8u_C3RSfs, [363](#)
- image_lshiftc
 - nppiLShiftC_16u_AC4IR, [424](#)
 - nppiLShiftC_16u_AC4R, [424](#)
 - nppiLShiftC_16u_C1IR, [424](#)
 - nppiLShiftC_16u_C1R, [425](#)
 - nppiLShiftC_16u_C3IR, [425](#)
 - nppiLShiftC_16u_C3R, [425](#)
 - nppiLShiftC_16u_C4IR, [426](#)
 - nppiLShiftC_16u_C4R, [426](#)
 - nppiLShiftC_32s_AC4IR, [426](#)
 - nppiLShiftC_32s_AC4R, [427](#)
 - nppiLShiftC_32s_C1IR, [427](#)
 - nppiLShiftC_32s_C1R, [427](#)
 - nppiLShiftC_32s_C3IR, [428](#)
 - nppiLShiftC_32s_C3R, [428](#)
 - nppiLShiftC_32s_C4IR, [428](#)
 - nppiLShiftC_32s_C4R, [429](#)
 - nppiLShiftC_8u_AC4IR, [429](#)
 - nppiLShiftC_8u_AC4R, [429](#)
 - nppiLShiftC_8u_C1IR, [430](#)
 - nppiLShiftC_8u_C1R, [430](#)
 - nppiLShiftC_8u_C3IR, [430](#)
 - nppiLShiftC_8u_C3R, [431](#)
 - nppiLShiftC_8u_C4IR, [431](#)
 - nppiLShiftC_8u_C4R, [431](#)
- image_max
 - nppiMax_16s_AC4R, [1746](#)
 - nppiMax_16s_C1R, [1746](#)
 - nppiMax_16s_C3R, [1747](#)
 - nppiMax_16s_C4R, [1747](#)
 - nppiMax_16u_AC4R, [1747](#)
 - nppiMax_16u_C1R, [1748](#)
 - nppiMax_16u_C3R, [1748](#)
 - nppiMax_16u_C4R, [1749](#)
 - nppiMax_32f_AC4R, [1749](#)
 - nppiMax_32f_C1R, [1749](#)
 - nppiMax_32f_C3R, [1750](#)
 - nppiMax_32f_C4R, [1750](#)
 - nppiMax_8u_AC4R, [1750](#)
 - nppiMax_8u_C1R, [1751](#)
 - nppiMax_8u_C3R, [1751](#)
 - nppiMax_8u_C4R, [1752](#)
 - nppiMaxGetBufferHostSize_16s_AC4R, [1752](#)
 - nppiMaxGetBufferHostSize_16s_C1R, [1752](#)
 - nppiMaxGetBufferHostSize_16s_C3R, [1752](#)
 - nppiMaxGetBufferHostSize_16s_C4R, [1753](#)
 - nppiMaxGetBufferHostSize_16u_AC4R, [1753](#)
 - nppiMaxGetBufferHostSize_16u_C1R, [1753](#)
 - nppiMaxGetBufferHostSize_16u_C3R, [1754](#)
 - nppiMaxGetBufferHostSize_16u_C4R, [1754](#)
 - nppiMaxGetBufferHostSize_32f_AC4R, [1754](#)
 - nppiMaxGetBufferHostSize_32f_C1R, [1754](#)
 - nppiMaxGetBufferHostSize_32f_C3R, [1755](#)

- [nppiMaxGetBufferHostSize_32f_C4R, 1755](#)
- [nppiMaxGetBufferHostSize_8u_AC4R, 1755](#)
- [nppiMaxGetBufferHostSize_8u_C1R, 1756](#)
- [nppiMaxGetBufferHostSize_8u_C3R, 1756](#)
- [nppiMaxGetBufferHostSize_8u_C4R, 1756](#)
- image_max_index
 - [nppiMaxIndx_16s_AC4R, 1759](#)
 - [nppiMaxIndx_16s_C1R, 1760](#)
 - [nppiMaxIndx_16s_C3R, 1760](#)
 - [nppiMaxIndx_16s_C4R, 1760](#)
 - [nppiMaxIndx_16u_AC4R, 1761](#)
 - [nppiMaxIndx_16u_C1R, 1761](#)
 - [nppiMaxIndx_16u_C3R, 1762](#)
 - [nppiMaxIndx_16u_C4R, 1762](#)
 - [nppiMaxIndx_32f_AC4R, 1762](#)
 - [nppiMaxIndx_32f_C1R, 1763](#)
 - [nppiMaxIndx_32f_C3R, 1763](#)
 - [nppiMaxIndx_32f_C4R, 1764](#)
 - [nppiMaxIndx_8u_AC4R, 1764](#)
 - [nppiMaxIndx_8u_C1R, 1764](#)
 - [nppiMaxIndx_8u_C3R, 1765](#)
 - [nppiMaxIndx_8u_C4R, 1765](#)
 - [nppiMaxIndxGetBufferHostSize_16s_AC4R, 1766](#)
 - [nppiMaxIndxGetBufferHostSize_16s_C1R, 1766](#)
 - [nppiMaxIndxGetBufferHostSize_16s_C3R, 1766](#)
 - [nppiMaxIndxGetBufferHostSize_16s_C4R, 1767](#)
 - [nppiMaxIndxGetBufferHostSize_16u_AC4R, 1767](#)
 - [nppiMaxIndxGetBufferHostSize_16u_C1R, 1767](#)
 - [nppiMaxIndxGetBufferHostSize_16u_C3R, 1767](#)
 - [nppiMaxIndxGetBufferHostSize_16u_C4R, 1768](#)
 - [nppiMaxIndxGetBufferHostSize_32f_AC4R, 1768](#)
 - [nppiMaxIndxGetBufferHostSize_32f_C1R, 1768](#)
 - [nppiMaxIndxGetBufferHostSize_32f_C3R, 1769](#)
 - [nppiMaxIndxGetBufferHostSize_32f_C4R, 1769](#)
 - [nppiMaxIndxGetBufferHostSize_8u_AC4R, 1769](#)
 - [nppiMaxIndxGetBufferHostSize_8u_C1R, 1769](#)
 - [nppiMaxIndxGetBufferHostSize_8u_C3R, 1770](#)
 - [nppiMaxIndxGetBufferHostSize_8u_C4R, 1770](#)
- image_maxevery
 - [nppiMaxEvery_16s_AC4IR, 2075](#)
 - [nppiMaxEvery_16s_C1IR, 2075](#)
 - [nppiMaxEvery_16s_C3IR, 2076](#)
 - [nppiMaxEvery_16s_C4IR, 2076](#)
 - [nppiMaxEvery_16u_AC4IR, 2076](#)
 - [nppiMaxEvery_16u_C1IR, 2077](#)
 - [nppiMaxEvery_16u_C3IR, 2077](#)
 - [nppiMaxEvery_16u_C4IR, 2077](#)
 - [nppiMaxEvery_32f_AC4IR, 2078](#)
 - [nppiMaxEvery_32f_C1IR, 2078](#)
 - [nppiMaxEvery_32f_C3IR, 2078](#)
 - [nppiMaxEvery_32f_C4IR, 2079](#)
 - [nppiMaxEvery_8u_AC4IR, 2079](#)
 - [nppiMaxEvery_8u_C1IR, 2079](#)
 - [nppiMaxEvery_8u_C3IR, 2080](#)
 - [nppiMaxEvery_8u_C4IR, 2080](#)
- image_maximum_error
 - [nppiMaximumError_16s_C1R, 2269](#)
 - [nppiMaximumError_16s_C2R, 2270](#)
 - [nppiMaximumError_16s_C3R, 2270](#)
 - [nppiMaximumError_16s_C4R, 2270](#)
 - [nppiMaximumError_16sc_C1R, 2271](#)
 - [nppiMaximumError_16sc_C2R, 2271](#)
 - [nppiMaximumError_16sc_C3R, 2272](#)
 - [nppiMaximumError_16sc_C4R, 2272](#)
 - [nppiMaximumError_16u_C1R, 2273](#)
 - [nppiMaximumError_16u_C2R, 2273](#)
 - [nppiMaximumError_16u_C3R, 2273](#)
 - [nppiMaximumError_16u_C4R, 2274](#)
 - [nppiMaximumError_32f_C1R, 2274](#)
 - [nppiMaximumError_32f_C2R, 2275](#)
 - [nppiMaximumError_32f_C3R, 2275](#)
 - [nppiMaximumError_32f_C4R, 2276](#)
 - [nppiMaximumError_32fc_C1R, 2276](#)
 - [nppiMaximumError_32fc_C2R, 2277](#)
 - [nppiMaximumError_32fc_C3R, 2277](#)
 - [nppiMaximumError_32fc_C4R, 2277](#)
 - [nppiMaximumError_32s_C1R, 2278](#)
 - [nppiMaximumError_32s_C2R, 2278](#)
 - [nppiMaximumError_32s_C3R, 2279](#)
 - [nppiMaximumError_32s_C4R, 2279](#)
 - [nppiMaximumError_32sc_C1R, 2280](#)
 - [nppiMaximumError_32sc_C2R, 2280](#)
 - [nppiMaximumError_32sc_C3R, 2280](#)
 - [nppiMaximumError_32sc_C4R, 2281](#)
 - [nppiMaximumError_32u_C1R, 2281](#)
 - [nppiMaximumError_32u_C2R, 2282](#)
 - [nppiMaximumError_32u_C3R, 2282](#)
 - [nppiMaximumError_32u_C4R, 2283](#)
 - [nppiMaximumError_64f_C1R, 2283](#)
 - [nppiMaximumError_64f_C2R, 2283](#)
 - [nppiMaximumError_64f_C3R, 2284](#)
 - [nppiMaximumError_64f_C4R, 2284](#)

- nppiMaximumError_8s_C1R, 2285
- nppiMaximumError_8s_C2R, 2285
- nppiMaximumError_8s_C3R, 2286
- nppiMaximumError_8s_C4R, 2286
- nppiMaximumError_8u_C1R, 2286
- nppiMaximumError_8u_C2R, 2287
- nppiMaximumError_8u_C3R, 2287
- nppiMaximumError_8u_C4R, 2288
- image_maximum_relative_error
 - nppiMaximumRelativeError_16s_C1R, 2315
 - nppiMaximumRelativeError_16s_C2R, 2316
 - nppiMaximumRelativeError_16s_C3R, 2316
 - nppiMaximumRelativeError_16s_C4R, 2317
 - nppiMaximumRelativeError_16sc_C1R, 2317
 - nppiMaximumRelativeError_16sc_C2R, 2318
 - nppiMaximumRelativeError_16sc_C3R, 2318
 - nppiMaximumRelativeError_16sc_C4R, 2318
 - nppiMaximumRelativeError_16u_C1R, 2319
 - nppiMaximumRelativeError_16u_C2R, 2319
 - nppiMaximumRelativeError_16u_C3R, 2320
 - nppiMaximumRelativeError_16u_C4R, 2320
 - nppiMaximumRelativeError_32f_C1R, 2321
 - nppiMaximumRelativeError_32f_C2R, 2321
 - nppiMaximumRelativeError_32f_C3R, 2322
 - nppiMaximumRelativeError_32f_C4R, 2322
 - nppiMaximumRelativeError_32fc_C1R, 2323
 - nppiMaximumRelativeError_32fc_C2R, 2323
 - nppiMaximumRelativeError_32fc_C3R, 2323
 - nppiMaximumRelativeError_32fc_C4R, 2324
 - nppiMaximumRelativeError_32s_C1R, 2324
 - nppiMaximumRelativeError_32s_C2R, 2325
 - nppiMaximumRelativeError_32s_C3R, 2325
 - nppiMaximumRelativeError_32s_C4R, 2326
 - nppiMaximumRelativeError_32sc_C1R, 2326
 - nppiMaximumRelativeError_32sc_C2R, 2327
 - nppiMaximumRelativeError_32sc_C3R, 2327
 - nppiMaximumRelativeError_32sc_C4R, 2328
 - nppiMaximumRelativeError_32u_C1R, 2328
 - nppiMaximumRelativeError_32u_C2R, 2328
 - nppiMaximumRelativeError_32u_C3R, 2329
 - nppiMaximumRelativeError_32u_C4R, 2329
 - nppiMaximumRelativeError_64f_C1R, 2330
 - nppiMaximumRelativeError_64f_C2R, 2330
 - nppiMaximumRelativeError_64f_C3R, 2331
 - nppiMaximumRelativeError_64f_C4R, 2331
 - nppiMaximumRelativeError_8s_C1R, 2332
 - nppiMaximumRelativeError_8s_C2R, 2332
 - nppiMaximumRelativeError_8s_C3R, 2333
 - nppiMaximumRelativeError_8s_C4R, 2333
 - nppiMaximumRelativeError_8u_C1R, 2333
 - nppiMaximumRelativeError_8u_C2R, 2334
 - nppiMaximumRelativeError_8u_C3R, 2334
 - nppiMaximumRelativeError_8u_C4R, 2335
- image_mean
 - nppiMean_16s_AC4R, 1806
 - nppiMean_16s_C1R, 1806
 - nppiMean_16s_C3R, 1806
 - nppiMean_16s_C4R, 1807
 - nppiMean_16u_AC4R, 1807
 - nppiMean_16u_C1MR, 1807
 - nppiMean_16u_C1R, 1808
 - nppiMean_16u_C3CMR, 1808
 - nppiMean_16u_C3R, 1808
 - nppiMean_16u_C4R, 1809
 - nppiMean_32f_AC4R, 1809
 - nppiMean_32f_C1MR, 1810
 - nppiMean_32f_C1R, 1810
 - nppiMean_32f_C3CMR, 1810
 - nppiMean_32f_C3R, 1811
 - nppiMean_32f_C4R, 1811
 - nppiMean_8s_C1MR, 1812
 - nppiMean_8s_C3CMR, 1812
 - nppiMean_8u_AC4R, 1813
 - nppiMean_8u_C1MR, 1813
 - nppiMean_8u_C1R, 1813
 - nppiMean_8u_C3CMR, 1814
 - nppiMean_8u_C3R, 1814
 - nppiMean_8u_C4R, 1815
 - nppiMeanGetBufferHostSize_16s_AC4R, 1815
 - nppiMeanGetBufferHostSize_16s_C1R, 1815
 - nppiMeanGetBufferHostSize_16s_C3R, 1816
 - nppiMeanGetBufferHostSize_16s_C4R, 1816
 - nppiMeanGetBufferHostSize_16u_AC4R, 1816
 - nppiMeanGetBufferHostSize_16u_C1MR, 1816
 - nppiMeanGetBufferHostSize_16u_C1R, 1817
 - nppiMeanGetBufferHostSize_16u_C3CMR, 1817
 - nppiMeanGetBufferHostSize_16u_C3R, 1817
 - nppiMeanGetBufferHostSize_16u_C4R, 1818
 - nppiMeanGetBufferHostSize_32f_AC4R, 1818
 - nppiMeanGetBufferHostSize_32f_C1MR, 1818
 - nppiMeanGetBufferHostSize_32f_C1R, 1818
 - nppiMeanGetBufferHostSize_32f_C3CMR, 1819
 - nppiMeanGetBufferHostSize_32f_C3R, 1819
 - nppiMeanGetBufferHostSize_32f_C4R, 1819
 - nppiMeanGetBufferHostSize_8s_C1MR, 1820
 - nppiMeanGetBufferHostSize_8s_C3CMR, 1820
 - nppiMeanGetBufferHostSize_8u_AC4R, 1820
 - nppiMeanGetBufferHostSize_8u_C1MR, 1820

- [nppiMeanGetBufferHostSize_8u_C1R](#), [1821](#)
 - [nppiMeanGetBufferHostSize_8u_C3CMR](#), [1821](#)
 - [nppiMeanGetBufferHostSize_8u_C3R](#), [1821](#)
 - [nppiMeanGetBufferHostSize_8u_C4R](#), [1822](#)
- [image_mean_stddev](#)
 - [nppiMean_StdDev_16u_C1MR](#), [1826](#)
 - [nppiMean_StdDev_16u_C1R](#), [1826](#)
 - [nppiMean_StdDev_16u_C3CMR](#), [1827](#)
 - [nppiMean_StdDev_16u_C3CR](#), [1827](#)
 - [nppiMean_StdDev_32f_C1MR](#), [1828](#)
 - [nppiMean_StdDev_32f_C1R](#), [1828](#)
 - [nppiMean_StdDev_32f_C3CMR](#), [1829](#)
 - [nppiMean_StdDev_32f_C3CR](#), [1829](#)
 - [nppiMean_StdDev_8s_C1MR](#), [1830](#)
 - [nppiMean_StdDev_8s_C1R](#), [1830](#)
 - [nppiMean_StdDev_8s_C3CMR](#), [1831](#)
 - [nppiMean_StdDev_8s_C3CR](#), [1831](#)
 - [nppiMean_StdDev_8u_C1MR](#), [1832](#)
 - [nppiMean_StdDev_8u_C1R](#), [1832](#)
 - [nppiMean_StdDev_8u_C3CMR](#), [1833](#)
 - [nppiMean_StdDev_8u_C3CR](#), [1833](#)
 - [nppiMeanStdDevGetBufferHostSize_16u_C1MR](#), [1834](#)
 - [nppiMeanStdDevGetBufferHostSize_16u_C1R](#), [1834](#)
 - [nppiMeanStdDevGetBufferHostSize_16u_C3CMR](#), [1834](#)
 - [nppiMeanStdDevGetBufferHostSize_16u_C3CR](#), [1835](#)
 - [nppiMeanStdDevGetBufferHostSize_32f_C1MR](#), [1835](#)
 - [nppiMeanStdDevGetBufferHostSize_32f_C1R](#), [1835](#)
 - [nppiMeanStdDevGetBufferHostSize_32f_C3CMR](#), [1836](#)
 - [nppiMeanStdDevGetBufferHostSize_32f_C3CR](#), [1836](#)
 - [nppiMeanStdDevGetBufferHostSize_8s_C1MR](#), [1836](#)
 - [nppiMeanStdDevGetBufferHostSize_8s_C1R](#), [1836](#)
 - [nppiMeanStdDevGetBufferHostSize_8s_C3CMR](#), [1837](#)
 - [nppiMeanStdDevGetBufferHostSize_8s_C3CR](#), [1837](#)
 - [nppiMeanStdDevGetBufferHostSize_8u_C1MR](#), [1837](#)
 - [nppiMeanStdDevGetBufferHostSize_8u_C1R](#), [1838](#)
 - [nppiMeanStdDevGetBufferHostSize_8u_C3CMR](#), [1838](#)
 - [nppiMeanStdDevGetBufferHostSize_8u_C3CR](#), [1838](#)
- [image_memory_management](#)
 - [nppiFree](#), [2362](#)
 - [nppiMalloc_16s_C1](#), [2362](#)
 - [nppiMalloc_16s_C2](#), [2362](#)
 - [nppiMalloc_16s_C4](#), [2363](#)
 - [nppiMalloc_16sc_C1](#), [2363](#)
 - [nppiMalloc_16sc_C2](#), [2363](#)
 - [nppiMalloc_16sc_C3](#), [2364](#)
 - [nppiMalloc_16sc_C4](#), [2364](#)
 - [nppiMalloc_16u_C1](#), [2364](#)
 - [nppiMalloc_16u_C2](#), [2364](#)
 - [nppiMalloc_16u_C3](#), [2365](#)
 - [nppiMalloc_16u_C4](#), [2365](#)
 - [nppiMalloc_32f_C1](#), [2365](#)
 - [nppiMalloc_32f_C2](#), [2366](#)
 - [nppiMalloc_32f_C3](#), [2366](#)
 - [nppiMalloc_32f_C4](#), [2366](#)
 - [nppiMalloc_32fc_C1](#), [2366](#)
 - [nppiMalloc_32fc_C2](#), [2367](#)
 - [nppiMalloc_32fc_C3](#), [2367](#)
 - [nppiMalloc_32fc_C4](#), [2367](#)
 - [nppiMalloc_32s_C1](#), [2368](#)
 - [nppiMalloc_32s_C3](#), [2368](#)
 - [nppiMalloc_32s_C4](#), [2368](#)
 - [nppiMalloc_32sc_C1](#), [2368](#)
 - [nppiMalloc_32sc_C2](#), [2369](#)
 - [nppiMalloc_32sc_C3](#), [2369](#)
 - [nppiMalloc_32sc_C4](#), [2369](#)
 - [nppiMalloc_8u_C1](#), [2370](#)
 - [nppiMalloc_8u_C2](#), [2370](#)
 - [nppiMalloc_8u_C3](#), [2370](#)
 - [nppiMalloc_8u_C4](#), [2370](#)
- [image_min](#)
 - [nppiMin_16s_AC4R](#), [1719](#)
 - [nppiMin_16s_C1R](#), [1719](#)
 - [nppiMin_16s_C3R](#), [1720](#)
 - [nppiMin_16s_C4R](#), [1720](#)
 - [nppiMin_16u_AC4R](#), [1720](#)
 - [nppiMin_16u_C1R](#), [1721](#)
 - [nppiMin_16u_C3R](#), [1721](#)
 - [nppiMin_16u_C4R](#), [1722](#)
 - [nppiMin_32f_AC4R](#), [1722](#)
 - [nppiMin_32f_C1R](#), [1722](#)
 - [nppiMin_32f_C3R](#), [1723](#)
 - [nppiMin_32f_C4R](#), [1723](#)
 - [nppiMin_8u_AC4R](#), [1723](#)
 - [nppiMin_8u_C1R](#), [1724](#)
 - [nppiMin_8u_C3R](#), [1724](#)
 - [nppiMin_8u_C4R](#), [1725](#)
 - [nppiMinGetBufferHostSize_16s_AC4R](#), [1725](#)
 - [nppiMinGetBufferHostSize_16s_C1R](#), [1725](#)
 - [nppiMinGetBufferHostSize_16s_C3R](#), [1725](#)
 - [nppiMinGetBufferHostSize_16s_C4R](#), [1726](#)
 - [nppiMinGetBufferHostSize_16u_AC4R](#), [1726](#)

- nppiMinGetBufferHostSize_16u_C1R, [1726](#)
- nppiMinGetBufferHostSize_16u_C3R, [1727](#)
- nppiMinGetBufferHostSize_16u_C4R, [1727](#)
- nppiMinGetBufferHostSize_32f_AC4R, [1727](#)
- nppiMinGetBufferHostSize_32f_C1R, [1727](#)
- nppiMinGetBufferHostSize_32f_C3R, [1728](#)
- nppiMinGetBufferHostSize_32f_C4R, [1728](#)
- nppiMinGetBufferHostSize_8u_AC4R, [1728](#)
- nppiMinGetBufferHostSize_8u_C1R, [1729](#)
- nppiMinGetBufferHostSize_8u_C3R, [1729](#)
- nppiMinGetBufferHostSize_8u_C4R, [1729](#)
- image_min_index
 - nppiMinIndx_16s_AC4R, [1732](#)
 - nppiMinIndx_16s_C1R, [1733](#)
 - nppiMinIndx_16s_C3R, [1733](#)
 - nppiMinIndx_16s_C4R, [1733](#)
 - nppiMinIndx_16u_AC4R, [1734](#)
 - nppiMinIndx_16u_C1R, [1734](#)
 - nppiMinIndx_16u_C3R, [1735](#)
 - nppiMinIndx_16u_C4R, [1735](#)
 - nppiMinIndx_32f_AC4R, [1735](#)
 - nppiMinIndx_32f_C1R, [1736](#)
 - nppiMinIndx_32f_C3R, [1736](#)
 - nppiMinIndx_32f_C4R, [1737](#)
 - nppiMinIndx_8u_AC4R, [1737](#)
 - nppiMinIndx_8u_C1R, [1737](#)
 - nppiMinIndx_8u_C3R, [1738](#)
 - nppiMinIndx_8u_C4R, [1738](#)
 - nppiMinIndxGetBufferHostSize_16s_AC4R, [1739](#)
 - nppiMinIndxGetBufferHostSize_16s_C1R, [1739](#)
 - nppiMinIndxGetBufferHostSize_16s_C3R, [1739](#)
 - nppiMinIndxGetBufferHostSize_16s_C4R, [1740](#)
 - nppiMinIndxGetBufferHostSize_16u_AC4R, [1740](#)
 - nppiMinIndxGetBufferHostSize_16u_C1R, [1740](#)
 - nppiMinIndxGetBufferHostSize_16u_C3R, [1740](#)
 - nppiMinIndxGetBufferHostSize_16u_C4R, [1741](#)
 - nppiMinIndxGetBufferHostSize_32f_AC4R, [1741](#)
 - nppiMinIndxGetBufferHostSize_32f_C1R, [1741](#)
 - nppiMinIndxGetBufferHostSize_32f_C3R, [1742](#)
 - nppiMinIndxGetBufferHostSize_32f_C4R, [1742](#)
 - nppiMinIndxGetBufferHostSize_8u_AC4R, [1742](#)
- nppiMinIndxGetBufferHostSize_8u_C1R, [1742](#)
- nppiMinIndxGetBufferHostSize_8u_C3R, [1743](#)
- nppiMinIndxGetBufferHostSize_8u_C4R, [1743](#)
- image_min_max
 - nppiMinMax_16s_AC4R, [1773](#)
 - nppiMinMax_16s_C1R, [1773](#)
 - nppiMinMax_16s_C3R, [1774](#)
 - nppiMinMax_16s_C4R, [1774](#)
 - nppiMinMax_16u_AC4R, [1775](#)
 - nppiMinMax_16u_C1R, [1775](#)
 - nppiMinMax_16u_C3R, [1775](#)
 - nppiMinMax_16u_C4R, [1776](#)
 - nppiMinMax_32f_AC4R, [1776](#)
 - nppiMinMax_32f_C1R, [1777](#)
 - nppiMinMax_32f_C3R, [1777](#)
 - nppiMinMax_32f_C4R, [1777](#)
 - nppiMinMax_8u_AC4R, [1778](#)
 - nppiMinMax_8u_C1R, [1778](#)
 - nppiMinMax_8u_C3R, [1779](#)
 - nppiMinMax_8u_C4R, [1779](#)
 - nppiMinMaxGetBufferHostSize_16s_AC4R, [1779](#)
 - nppiMinMaxGetBufferHostSize_16s_C1R, [1780](#)
 - nppiMinMaxGetBufferHostSize_16s_C3R, [1780](#)
 - nppiMinMaxGetBufferHostSize_16s_C4R, [1780](#)
 - nppiMinMaxGetBufferHostSize_16u_AC4R, [1781](#)
 - nppiMinMaxGetBufferHostSize_16u_C1R, [1781](#)
 - nppiMinMaxGetBufferHostSize_16u_C3R, [1781](#)
 - nppiMinMaxGetBufferHostSize_16u_C4R, [1781](#)
 - nppiMinMaxGetBufferHostSize_32f_AC4R, [1782](#)
 - nppiMinMaxGetBufferHostSize_32f_C1R, [1782](#)
 - nppiMinMaxGetBufferHostSize_32f_C3R, [1782](#)
 - nppiMinMaxGetBufferHostSize_32f_C4R, [1783](#)
 - nppiMinMaxGetBufferHostSize_8u_AC4R, [1783](#)
 - nppiMinMaxGetBufferHostSize_8u_C1R, [1783](#)
 - nppiMinMaxGetBufferHostSize_8u_C3R, [1783](#)

- [nppiMinMaxGetBufferHostSize_8u_C4R, 1784](#)
- [image_min_max_index](#)
 - [nppiMinMaxIdx_16u_C1MR, 1788](#)
 - [nppiMinMaxIdx_16u_C1R, 1789](#)
 - [nppiMinMaxIdx_16u_C3CMR, 1789](#)
 - [nppiMinMaxIdx_16u_C3CR, 1790](#)
 - [nppiMinMaxIdx_32f_C1MR, 1790](#)
 - [nppiMinMaxIdx_32f_C1R, 1791](#)
 - [nppiMinMaxIdx_32f_C3CMR, 1791](#)
 - [nppiMinMaxIdx_32f_C3CR, 1792](#)
 - [nppiMinMaxIdx_8s_C1MR, 1793](#)
 - [nppiMinMaxIdx_8s_C1R, 1793](#)
 - [nppiMinMaxIdx_8s_C3CMR, 1794](#)
 - [nppiMinMaxIdx_8s_C3CR, 1794](#)
 - [nppiMinMaxIdx_8u_C1MR, 1795](#)
 - [nppiMinMaxIdx_8u_C1R, 1795](#)
 - [nppiMinMaxIdx_8u_C3CMR, 1796](#)
 - [nppiMinMaxIdx_8u_C3CR, 1796](#)
 - [nppiMinMaxIdxGetBufferHostSize_16u_C1MR, 1797](#)
 - [nppiMinMaxIdxGetBufferHostSize_16u_C1R, 1797](#)
 - [nppiMinMaxIdxGetBufferHostSize_16u_C3CMR, 1797](#)
 - [nppiMinMaxIdxGetBufferHostSize_16u_C3CR, 1798](#)
 - [nppiMinMaxIdxGetBufferHostSize_32f_C1MR, 1798](#)
 - [nppiMinMaxIdxGetBufferHostSize_32f_C1R, 1798](#)
 - [nppiMinMaxIdxGetBufferHostSize_32f_C3CMR, 1799](#)
 - [nppiMinMaxIdxGetBufferHostSize_32f_C3CR, 1799](#)
 - [nppiMinMaxIdxGetBufferHostSize_8s_C1MR, 1799](#)
 - [nppiMinMaxIdxGetBufferHostSize_8s_C1R, 1799](#)
 - [nppiMinMaxIdxGetBufferHostSize_8s_C3CMR, 1800](#)
 - [nppiMinMaxIdxGetBufferHostSize_8s_C3CR, 1800](#)
 - [nppiMinMaxIdxGetBufferHostSize_8u_C1MR, 1800](#)
 - [nppiMinMaxIdxGetBufferHostSize_8u_C1R, 1801](#)
 - [nppiMinMaxIdxGetBufferHostSize_8u_C3CMR, 1801](#)
 - [nppiMinMaxIdxGetBufferHostSize_8u_C3CR, 1801](#)
- [image_minevery](#)
 - [nppiMinEvery_16s_AC4IR, 2082](#)
 - [nppiMinEvery_16s_C1IR, 2082](#)
 - [nppiMinEvery_16s_C3IR, 2083](#)
 - [nppiMinEvery_16s_C4IR, 2083](#)
 - [nppiMinEvery_16u_AC4IR, 2083](#)
 - [nppiMinEvery_16u_C1IR, 2084](#)
 - [nppiMinEvery_16u_C3IR, 2084](#)
 - [nppiMinEvery_16u_C4IR, 2084](#)
 - [nppiMinEvery_32f_AC4IR, 2085](#)
 - [nppiMinEvery_32f_C1IR, 2085](#)
 - [nppiMinEvery_32f_C3IR, 2085](#)
 - [nppiMinEvery_32f_C4IR, 2086](#)
 - [nppiMinEvery_8u_AC4IR, 2086](#)
 - [nppiMinEvery_8u_C1IR, 2086](#)
 - [nppiMinEvery_8u_C3IR, 2087](#)
 - [nppiMinEvery_8u_C4IR, 2087](#)
- [image_mirror](#)
 - [nppiMirror_16s_AC4IR, 1465](#)
 - [nppiMirror_16s_AC4R, 1465](#)
 - [nppiMirror_16s_C1IR, 1466](#)
 - [nppiMirror_16s_C1R, 1466](#)
 - [nppiMirror_16s_C3IR, 1466](#)
 - [nppiMirror_16s_C3R, 1467](#)
 - [nppiMirror_16s_C4IR, 1467](#)
 - [nppiMirror_16s_C4R, 1467](#)
 - [nppiMirror_16u_AC4IR, 1468](#)
 - [nppiMirror_16u_AC4R, 1468](#)
 - [nppiMirror_16u_C1IR, 1468](#)
 - [nppiMirror_16u_C1R, 1469](#)
 - [nppiMirror_16u_C3IR, 1469](#)
 - [nppiMirror_16u_C3R, 1469](#)
 - [nppiMirror_16u_C4IR, 1470](#)
 - [nppiMirror_16u_C4R, 1470](#)
 - [nppiMirror_32f_AC4IR, 1470](#)
 - [nppiMirror_32f_AC4R, 1471](#)
 - [nppiMirror_32f_C1IR, 1471](#)
 - [nppiMirror_32f_C1R, 1471](#)
 - [nppiMirror_32f_C3IR, 1472](#)
 - [nppiMirror_32f_C3R, 1472](#)
 - [nppiMirror_32f_C4IR, 1472](#)
 - [nppiMirror_32f_C4R, 1473](#)
 - [nppiMirror_32s_AC4IR, 1473](#)
 - [nppiMirror_32s_AC4R, 1473](#)
 - [nppiMirror_32s_C1IR, 1474](#)
 - [nppiMirror_32s_C1R, 1474](#)
 - [nppiMirror_32s_C3IR, 1474](#)
 - [nppiMirror_32s_C3R, 1475](#)
 - [nppiMirror_32s_C4IR, 1475](#)
 - [nppiMirror_32s_C4R, 1475](#)
 - [nppiMirror_8u_AC4IR, 1476](#)
 - [nppiMirror_8u_AC4R, 1476](#)
 - [nppiMirror_8u_C1IR, 1476](#)
 - [nppiMirror_8u_C1R, 1477](#)
 - [nppiMirror_8u_C3IR, 1477](#)
 - [nppiMirror_8u_C3R, 1477](#)
 - [nppiMirror_8u_C4IR, 1478](#)

- nppiMirror_8u_C4R, 1478
- image_mul
 - nppiMul_16s_AC4IRSfs, 214
 - nppiMul_16s_AC4RSfs, 214
 - nppiMul_16s_C1IRSfs, 215
 - nppiMul_16s_C1RSfs, 215
 - nppiMul_16s_C3IRSfs, 216
 - nppiMul_16s_C3RSfs, 216
 - nppiMul_16s_C4IRSfs, 216
 - nppiMul_16s_C4RSfs, 217
 - nppiMul_16sc_AC4IRSfs, 217
 - nppiMul_16sc_AC4RSfs, 218
 - nppiMul_16sc_C1IRSfs, 218
 - nppiMul_16sc_C1RSfs, 218
 - nppiMul_16sc_C3IRSfs, 219
 - nppiMul_16sc_C3RSfs, 219
 - nppiMul_16u_AC4IRSfs, 220
 - nppiMul_16u_AC4RSfs, 220
 - nppiMul_16u_C1IRSfs, 221
 - nppiMul_16u_C1RSfs, 221
 - nppiMul_16u_C3IRSfs, 221
 - nppiMul_16u_C3RSfs, 222
 - nppiMul_16u_C4IRSfs, 222
 - nppiMul_16u_C4RSfs, 223
 - nppiMul_32f_AC4IR, 223
 - nppiMul_32f_AC4R, 223
 - nppiMul_32f_C1IR, 224
 - nppiMul_32f_C1R, 224
 - nppiMul_32f_C3IR, 225
 - nppiMul_32f_C3R, 225
 - nppiMul_32f_C4IR, 225
 - nppiMul_32f_C4R, 226
 - nppiMul_32fc_AC4IR, 226
 - nppiMul_32fc_AC4R, 226
 - nppiMul_32fc_C1IR, 227
 - nppiMul_32fc_C1R, 227
 - nppiMul_32fc_C3IR, 228
 - nppiMul_32fc_C3R, 228
 - nppiMul_32fc_C4IR, 228
 - nppiMul_32fc_C4R, 229
 - nppiMul_32s_C1IRSfs, 229
 - nppiMul_32s_C1R, 230
 - nppiMul_32s_C3IRSfs, 230
 - nppiMul_32s_C3RSfs, 231
 - nppiMul_32sc_AC4IRSfs, 231
 - nppiMul_32sc_AC4RSfs, 232
 - nppiMul_32sc_C1IRSfs, 232
 - nppiMul_32sc_C1RSfs, 232
 - nppiMul_32sc_C3IRSfs, 233
 - nppiMul_32sc_C3RSfs, 233
 - nppiMul_8u_AC4IRSfs, 234
 - nppiMul_8u_AC4RSfs, 234
 - nppiMul_8u_C1IRSfs, 235
- nppiMul_8u_C1RSfs, 235
- nppiMul_8u_C3IRSfs, 235
- nppiMul_8u_C3RSfs, 236
- nppiMul_8u_C4IRSfs, 236
- nppiMul_8u_C4RSfs, 237
- image_mulc
 - nppiMulC_16s_AC4IRSfs, 87
 - nppiMulC_16s_AC4RSfs, 87
 - nppiMulC_16s_C1IRSfs, 87
 - nppiMulC_16s_C1RSfs, 88
 - nppiMulC_16s_C3IRSfs, 88
 - nppiMulC_16s_C3RSfs, 88
 - nppiMulC_16s_C4IRSfs, 89
 - nppiMulC_16s_C4RSfs, 89
 - nppiMulC_16sc_AC4IRSfs, 90
 - nppiMulC_16sc_AC4RSfs, 90
 - nppiMulC_16sc_C1IRSfs, 90
 - nppiMulC_16sc_C1RSfs, 91
 - nppiMulC_16sc_C3IRSfs, 91
 - nppiMulC_16sc_C3RSfs, 92
 - nppiMulC_16u_AC4IRSfs, 92
 - nppiMulC_16u_AC4RSfs, 92
 - nppiMulC_16u_C1IRSfs, 93
 - nppiMulC_16u_C1RSfs, 93
 - nppiMulC_16u_C3IRSfs, 94
 - nppiMulC_16u_C3RSfs, 94
 - nppiMulC_16u_C4IRSfs, 94
 - nppiMulC_16u_C4RSfs, 95
 - nppiMulC_32f_AC4IR, 95
 - nppiMulC_32f_AC4R, 95
 - nppiMulC_32f_C1IR, 96
 - nppiMulC_32f_C1R, 96
 - nppiMulC_32f_C3IR, 96
 - nppiMulC_32f_C3R, 97
 - nppiMulC_32f_C4IR, 97
 - nppiMulC_32f_C4R, 97
 - nppiMulC_32fc_AC4IR, 98
 - nppiMulC_32fc_AC4R, 98
 - nppiMulC_32fc_C1IR, 98
 - nppiMulC_32fc_C1R, 99
 - nppiMulC_32fc_C3IR, 99
 - nppiMulC_32fc_C3R, 99
 - nppiMulC_32fc_C4IR, 100
 - nppiMulC_32fc_C4R, 100
 - nppiMulC_32s_C1IRSfs, 101
 - nppiMulC_32s_C1RSfs, 101
 - nppiMulC_32s_C3IRSfs, 101
 - nppiMulC_32s_C3RSfs, 102
 - nppiMulC_32sc_AC4IRSfs, 102
 - nppiMulC_32sc_AC4RSfs, 102
 - nppiMulC_32sc_C1IRSfs, 103
 - nppiMulC_32sc_C1RSfs, 103
 - nppiMulC_32sc_C3IRSfs, 104
 - nppiMulC_32sc_C3RSfs, 104

- [nppiMulC_8u_AC4IRSfs, 104](#)
 - [nppiMulC_8u_AC4RSfs, 105](#)
 - [nppiMulC_8u_C1IRSfs, 105](#)
 - [nppiMulC_8u_C1RSfs, 106](#)
 - [nppiMulC_8u_C3IRSfs, 106](#)
 - [nppiMulC_8u_C3RSfs, 106](#)
 - [nppiMulC_8u_C4IRSfs, 107](#)
 - [nppiMulC_8u_C4RSfs, 107](#)
- [image_mulcscale](#)
 - [nppiMulCScale_16u_AC4IR, 109](#)
 - [nppiMulCScale_16u_AC4R, 109](#)
 - [nppiMulCScale_16u_C1IR, 110](#)
 - [nppiMulCScale_16u_C1R, 110](#)
 - [nppiMulCScale_16u_C3IR, 110](#)
 - [nppiMulCScale_16u_C3R, 111](#)
 - [nppiMulCScale_16u_C4IR, 111](#)
 - [nppiMulCScale_16u_C4R, 111](#)
 - [nppiMulCScale_8u_AC4IR, 112](#)
 - [nppiMulCScale_8u_AC4R, 112](#)
 - [nppiMulCScale_8u_C1IR, 112](#)
 - [nppiMulCScale_8u_C1R, 113](#)
 - [nppiMulCScale_8u_C3IR, 113](#)
 - [nppiMulCScale_8u_C3R, 113](#)
 - [nppiMulCScale_8u_C4IR, 114](#)
 - [nppiMulCScale_8u_C4R, 114](#)
- [image_mulscale](#)
 - [nppiMulScale_16u_AC4IR, 239](#)
 - [nppiMulScale_16u_AC4R, 240](#)
 - [nppiMulScale_16u_C1IR, 240](#)
 - [nppiMulScale_16u_C1R, 240](#)
 - [nppiMulScale_16u_C3IR, 241](#)
 - [nppiMulScale_16u_C3R, 241](#)
 - [nppiMulScale_16u_C4IR, 242](#)
 - [nppiMulScale_16u_C4R, 242](#)
 - [nppiMulScale_8u_AC4IR, 242](#)
 - [nppiMulScale_8u_AC4R, 243](#)
 - [nppiMulScale_8u_C1IR, 243](#)
 - [nppiMulScale_8u_C1R, 244](#)
 - [nppiMulScale_8u_C3IR, 244](#)
 - [nppiMulScale_8u_C3R, 244](#)
 - [nppiMulScale_8u_C4IR, 245](#)
 - [nppiMulScale_8u_C4R, 245](#)
- [image_not](#)
 - [nppiNot_8u_AC4IR, 469](#)
 - [nppiNot_8u_AC4R, 470](#)
 - [nppiNot_8u_C1IR, 470](#)
 - [nppiNot_8u_C1R, 470](#)
 - [nppiNot_8u_C3IR, 470](#)
 - [nppiNot_8u_C3R, 471](#)
 - [nppiNot_8u_C4IR, 471](#)
 - [nppiNot_8u_C4R, 471](#)
- [image_or](#)
 - [nppiOr_16u_AC4IR, 447](#)
 - [nppiOr_16u_AC4R, 447](#)
 - [nppiOr_16u_C1IR, 447](#)
 - [nppiOr_16u_C1R, 448](#)
 - [nppiOr_16u_C3IR, 448](#)
 - [nppiOr_16u_C3R, 448](#)
 - [nppiOr_16u_C4IR, 449](#)
 - [nppiOr_16u_C4R, 449](#)
 - [nppiOr_32s_AC4IR, 450](#)
 - [nppiOr_32s_AC4R, 450](#)
 - [nppiOr_32s_C1IR, 450](#)
 - [nppiOr_32s_C1R, 451](#)
 - [nppiOr_32s_C3IR, 451](#)
 - [nppiOr_32s_C3R, 451](#)
 - [nppiOr_32s_C4IR, 452](#)
 - [nppiOr_32s_C4R, 452](#)
 - [nppiOr_8u_AC4IR, 453](#)
 - [nppiOr_8u_AC4R, 453](#)
 - [nppiOr_8u_C1IR, 453](#)
 - [nppiOr_8u_C1R, 454](#)
 - [nppiOr_8u_C3IR, 454](#)
 - [nppiOr_8u_C3R, 454](#)
 - [nppiOr_8u_C4IR, 455](#)
 - [nppiOr_8u_C4R, 455](#)
- [image_orc](#)
 - [nppiOrC_16u_AC4IR, 385](#)
 - [nppiOrC_16u_AC4R, 385](#)
 - [nppiOrC_16u_C1IR, 385](#)
 - [nppiOrC_16u_C1R, 386](#)
 - [nppiOrC_16u_C3IR, 386](#)
 - [nppiOrC_16u_C3R, 386](#)
 - [nppiOrC_16u_C4IR, 387](#)
 - [nppiOrC_16u_C4R, 387](#)
 - [nppiOrC_32s_AC4IR, 387](#)
 - [nppiOrC_32s_AC4R, 388](#)
 - [nppiOrC_32s_C1IR, 388](#)
 - [nppiOrC_32s_C1R, 388](#)
 - [nppiOrC_32s_C3IR, 389](#)
 - [nppiOrC_32s_C3R, 389](#)
 - [nppiOrC_32s_C4IR, 389](#)
 - [nppiOrC_32s_C4R, 390](#)
 - [nppiOrC_8u_AC4IR, 390](#)
 - [nppiOrC_8u_AC4R, 390](#)
 - [nppiOrC_8u_C1IR, 391](#)
 - [nppiOrC_8u_C1R, 391](#)
 - [nppiOrC_8u_C3IR, 391](#)
 - [nppiOrC_8u_C3R, 392](#)
 - [nppiOrC_8u_C4IR, 392](#)
 - [nppiOrC_8u_C4R, 392](#)
- [image_perspective_transforms](#)
 - [nppiGetPerspectiveBound, 1537](#)
 - [nppiGetPerspectiveQuad, 1537](#)
 - [nppiGetPerspectiveTransform, 1538](#)
 - [nppiWarpPerspective_16u_AC4R, 1538](#)
 - [nppiWarpPerspective_16u_C1R, 1539](#)
 - [nppiWarpPerspective_16u_C3R, 1539](#)

- [nppiWarpPerspective_16u_C4R](#), [1540](#)
- [nppiWarpPerspective_16u_P3R](#), [1540](#)
- [nppiWarpPerspective_16u_P4R](#), [1541](#)
- [nppiWarpPerspective_32f_AC4R](#), [1541](#)
- [nppiWarpPerspective_32f_C1R](#), [1542](#)
- [nppiWarpPerspective_32f_C3R](#), [1542](#)
- [nppiWarpPerspective_32f_C4R](#), [1543](#)
- [nppiWarpPerspective_32f_P3R](#), [1543](#)
- [nppiWarpPerspective_32f_P4R](#), [1544](#)
- [nppiWarpPerspective_32s_AC4R](#), [1544](#)
- [nppiWarpPerspective_32s_C1R](#), [1545](#)
- [nppiWarpPerspective_32s_C3R](#), [1545](#)
- [nppiWarpPerspective_32s_C4R](#), [1546](#)
- [nppiWarpPerspective_32s_P3R](#), [1546](#)
- [nppiWarpPerspective_32s_P4R](#), [1546](#)
- [nppiWarpPerspective_8u_AC4R](#), [1547](#)
- [nppiWarpPerspective_8u_C1R](#), [1547](#)
- [nppiWarpPerspective_8u_C3R](#), [1548](#)
- [nppiWarpPerspective_8u_C4R](#), [1548](#)
- [nppiWarpPerspective_8u_P3R](#), [1549](#)
- [nppiWarpPerspective_8u_P4R](#), [1549](#)
- [nppiWarpPerspectiveBack_16u_AC4R](#), [1550](#)
- [nppiWarpPerspectiveBack_16u_C1R](#), [1550](#)
- [nppiWarpPerspectiveBack_16u_C3R](#), [1551](#)
- [nppiWarpPerspectiveBack_16u_C4R](#), [1551](#)
- [nppiWarpPerspectiveBack_16u_P3R](#), [1552](#)
- [nppiWarpPerspectiveBack_16u_P4R](#), [1552](#)
- [nppiWarpPerspectiveBack_32f_AC4R](#), [1553](#)
- [nppiWarpPerspectiveBack_32f_C1R](#), [1553](#)
- [nppiWarpPerspectiveBack_32f_C3R](#), [1554](#)
- [nppiWarpPerspectiveBack_32f_C4R](#), [1554](#)
- [nppiWarpPerspectiveBack_32f_P3R](#), [1555](#)
- [nppiWarpPerspectiveBack_32f_P4R](#), [1555](#)
- [nppiWarpPerspectiveBack_32s_AC4R](#), [1556](#)
- [nppiWarpPerspectiveBack_32s_C1R](#), [1556](#)
- [nppiWarpPerspectiveBack_32s_C3R](#), [1557](#)
- [nppiWarpPerspectiveBack_32s_C4R](#), [1557](#)
- [nppiWarpPerspectiveBack_32s_P3R](#), [1558](#)
- [nppiWarpPerspectiveBack_32s_P4R](#), [1558](#)
- [nppiWarpPerspectiveBack_8u_AC4R](#), [1559](#)
- [nppiWarpPerspectiveBack_8u_C1R](#), [1559](#)
- [nppiWarpPerspectiveBack_8u_C3R](#), [1560](#)
- [nppiWarpPerspectiveBack_8u_C4R](#), [1560](#)
- [nppiWarpPerspectiveBack_8u_P3R](#), [1561](#)
- [nppiWarpPerspectiveBack_8u_P4R](#), [1561](#)
- [nppiWarpPerspectiveQuad_16u_AC4R](#), [1562](#)
- [nppiWarpPerspectiveQuad_16u_C1R](#), [1562](#)
- [nppiWarpPerspectiveQuad_16u_C3R](#), [1563](#)
- [nppiWarpPerspectiveQuad_16u_C4R](#), [1563](#)
- [nppiWarpPerspectiveQuad_16u_P3R](#), [1564](#)
- [nppiWarpPerspectiveQuad_16u_P4R](#), [1564](#)
- [nppiWarpPerspectiveQuad_32f_AC4R](#), [1565](#)
- [nppiWarpPerspectiveQuad_32f_C1R](#), [1565](#)
- [nppiWarpPerspectiveQuad_32f_C3R](#), [1566](#)
- [nppiWarpPerspectiveQuad_32f_C4R](#), [1566](#)
- [nppiWarpPerspectiveQuad_32f_P3R](#), [1567](#)
- [nppiWarpPerspectiveQuad_32f_P4R](#), [1567](#)
- [nppiWarpPerspectiveQuad_32s_AC4R](#), [1568](#)
- [nppiWarpPerspectiveQuad_32s_C1R](#), [1568](#)
- [nppiWarpPerspectiveQuad_32s_C3R](#), [1569](#)
- [nppiWarpPerspectiveQuad_32s_C4R](#), [1569](#)
- [nppiWarpPerspectiveQuad_32s_P3R](#), [1570](#)
- [nppiWarpPerspectiveQuad_32s_P4R](#), [1570](#)
- [nppiWarpPerspectiveQuad_8u_AC4R](#), [1571](#)
- [nppiWarpPerspectiveQuad_8u_C1R](#), [1571](#)
- [nppiWarpPerspectiveQuad_8u_C3R](#), [1572](#)
- [nppiWarpPerspectiveQuad_8u_C4R](#), [1572](#)
- [nppiWarpPerspectiveQuad_8u_P3R](#), [1573](#)
- [nppiWarpPerspectiveQuad_8u_P4R](#), [1573](#)
- [image_quality_index](#)
 - [nppiQualityIndex_16u32f_AC4R](#), [2259](#)
 - [nppiQualityIndex_16u32f_C1R](#), [2259](#)
 - [nppiQualityIndex_16u32f_C3R](#), [2260](#)
 - [nppiQualityIndex_32f_AC4R](#), [2260](#)
 - [nppiQualityIndex_32f_C1R](#), [2261](#)
 - [nppiQualityIndex_32f_C3R](#), [2261](#)
 - [nppiQualityIndex_8u32f_AC4R](#), [2261](#)
 - [nppiQualityIndex_8u32f_C1R](#), [2262](#)
 - [nppiQualityIndex_8u32f_C3R](#), [2262](#)
 - [nppiQualityIndexGetBufferHostSize_16u32f_AC4R](#), [2263](#)
 - [nppiQualityIndexGetBufferHostSize_16u32f_C1R](#), [2263](#)
 - [nppiQualityIndexGetBufferHostSize_16u32f_C3R](#), [2263](#)
 - [nppiQualityIndexGetBufferHostSize_32f_AC4R](#), [2264](#)
 - [nppiQualityIndexGetBufferHostSize_32f_C1R](#), [2264](#)
 - [nppiQualityIndexGetBufferHostSize_32f_C3R](#), [2264](#)
 - [nppiQualityIndexGetBufferHostSize_8u32f_AC4R](#), [2265](#)
 - [nppiQualityIndexGetBufferHostSize_8u32f_C1R](#), [2265](#)
 - [nppiQualityIndexGetBufferHostSize_8u32f_C3R](#), [2265](#)
- [image_quantization](#)
 - [nppiDCTFree](#), [725](#)
 - [nppiDCTInitAlloc](#), [725](#)
 - [nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R](#), [725](#)
 - [nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R_NEW](#), [726](#)
 - [nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R](#), [726](#)
 - [nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R_NEW](#), [727](#)

- NppiDCTState, [725](#)
- nppiQuantFwdRawTableInit_JPEG_8u, [727](#)
- nppiQuantFwdTableInit_JPEG_8u16u, [728](#)
- nppiQuantInvTableInit_JPEG_8u16u, [728](#)
- image_rank_filters
 - nppiFilterMax_16s_AC4R, [1307](#)
 - nppiFilterMax_16s_C1R, [1307](#)
 - nppiFilterMax_16s_C3R, [1307](#)
 - nppiFilterMax_16s_C4R, [1308](#)
 - nppiFilterMax_16u_AC4R, [1308](#)
 - nppiFilterMax_16u_C1R, [1309](#)
 - nppiFilterMax_16u_C3R, [1309](#)
 - nppiFilterMax_16u_C4R, [1309](#)
 - nppiFilterMax_32f_AC4R, [1310](#)
 - nppiFilterMax_32f_C1R, [1310](#)
 - nppiFilterMax_32f_C3R, [1311](#)
 - nppiFilterMax_32f_C4R, [1311](#)
 - nppiFilterMax_8u_AC4R, [1311](#)
 - nppiFilterMax_8u_C1R, [1312](#)
 - nppiFilterMax_8u_C3R, [1312](#)
 - nppiFilterMax_8u_C4R, [1313](#)
 - nppiFilterMaxBorder_16s_AC4R, [1313](#)
 - nppiFilterMaxBorder_16s_C1R, [1313](#)
 - nppiFilterMaxBorder_16s_C3R, [1314](#)
 - nppiFilterMaxBorder_16s_C4R, [1314](#)
 - nppiFilterMaxBorder_16u_AC4R, [1315](#)
 - nppiFilterMaxBorder_16u_C1R, [1315](#)
 - nppiFilterMaxBorder_16u_C3R, [1316](#)
 - nppiFilterMaxBorder_16u_C4R, [1316](#)
 - nppiFilterMaxBorder_32f_AC4R, [1317](#)
 - nppiFilterMaxBorder_32f_C1R, [1317](#)
 - nppiFilterMaxBorder_32f_C3R, [1318](#)
 - nppiFilterMaxBorder_32f_C4R, [1318](#)
 - nppiFilterMaxBorder_8u_AC4R, [1319](#)
 - nppiFilterMaxBorder_8u_C1R, [1319](#)
 - nppiFilterMaxBorder_8u_C3R, [1320](#)
 - nppiFilterMaxBorder_8u_C4R, [1320](#)
 - nppiFilterMedian_16s_AC4R, [1321](#)
 - nppiFilterMedian_16s_C1R, [1321](#)
 - nppiFilterMedian_16s_C3R, [1322](#)
 - nppiFilterMedian_16s_C4R, [1322](#)
 - nppiFilterMedian_16u_AC4R, [1323](#)
 - nppiFilterMedian_16u_C1R, [1323](#)
 - nppiFilterMedian_16u_C3R, [1324](#)
 - nppiFilterMedian_16u_C4R, [1324](#)
 - nppiFilterMedian_32f_AC4R, [1324](#)
 - nppiFilterMedian_32f_C1R, [1325](#)
 - nppiFilterMedian_32f_C3R, [1325](#)
 - nppiFilterMedian_32f_C4R, [1326](#)
 - nppiFilterMedian_8u_AC4R, [1326](#)
 - nppiFilterMedian_8u_C1R, [1327](#)
 - nppiFilterMedian_8u_C3R, [1327](#)
 - nppiFilterMedian_8u_C4R, [1327](#)
 - nppiFilterMedianGetBufferSize_16s_AC4R, [1328](#)
 - nppiFilterMedianGetBufferSize_16s_C1R, [1328](#)
 - nppiFilterMedianGetBufferSize_16s_C3R, [1328](#)
 - nppiFilterMedianGetBufferSize_16s_C4R, [1329](#)
 - nppiFilterMedianGetBufferSize_16u_AC4R, [1329](#)
 - nppiFilterMedianGetBufferSize_16u_C1R, [1329](#)
 - nppiFilterMedianGetBufferSize_16u_C3R, [1330](#)
 - nppiFilterMedianGetBufferSize_16u_C4R, [1330](#)
 - nppiFilterMedianGetBufferSize_32f_AC4R, [1330](#)
 - nppiFilterMedianGetBufferSize_32f_C1R, [1330](#)
 - nppiFilterMedianGetBufferSize_32f_C3R, [1331](#)
 - nppiFilterMedianGetBufferSize_32f_C4R, [1331](#)
 - nppiFilterMedianGetBufferSize_8u_AC4R, [1331](#)
 - nppiFilterMedianGetBufferSize_8u_C1R, [1332](#)
 - nppiFilterMedianGetBufferSize_8u_C3R, [1332](#)
 - nppiFilterMedianGetBufferSize_8u_C4R, [1332](#)
 - nppiFilterMin_16s_AC4R, [1332](#)
 - nppiFilterMin_16s_C1R, [1333](#)
 - nppiFilterMin_16s_C3R, [1333](#)
 - nppiFilterMin_16s_C4R, [1334](#)
 - nppiFilterMin_16u_AC4R, [1334](#)
 - nppiFilterMin_16u_C1R, [1334](#)
 - nppiFilterMin_16u_C3R, [1335](#)
 - nppiFilterMin_16u_C4R, [1335](#)
 - nppiFilterMin_32f_AC4R, [1336](#)
 - nppiFilterMin_32f_C1R, [1336](#)
 - nppiFilterMin_32f_C3R, [1336](#)
 - nppiFilterMin_32f_C4R, [1337](#)
 - nppiFilterMin_8u_AC4R, [1337](#)
 - nppiFilterMin_8u_C1R, [1338](#)
 - nppiFilterMin_8u_C3R, [1338](#)
 - nppiFilterMin_8u_C4R, [1338](#)
 - nppiFilterMinBorder_16s_AC4R, [1339](#)
 - nppiFilterMinBorder_16s_C1R, [1339](#)
 - nppiFilterMinBorder_16s_C3R, [1340](#)
 - nppiFilterMinBorder_16s_C4R, [1340](#)
 - nppiFilterMinBorder_16u_AC4R, [1341](#)
 - nppiFilterMinBorder_16u_C1R, [1341](#)

- [nppiFilterMinBorder_16u_C3R](#), [1342](#)
 - [nppiFilterMinBorder_16u_C4R](#), [1342](#)
 - [nppiFilterMinBorder_32f_AC4R](#), [1343](#)
 - [nppiFilterMinBorder_32f_C1R](#), [1343](#)
 - [nppiFilterMinBorder_32f_C3R](#), [1344](#)
 - [nppiFilterMinBorder_32f_C4R](#), [1344](#)
 - [nppiFilterMinBorder_8u_AC4R](#), [1345](#)
 - [nppiFilterMinBorder_8u_C1R](#), [1345](#)
 - [nppiFilterMinBorder_8u_C3R](#), [1346](#)
 - [nppiFilterMinBorder_8u_C4R](#), [1346](#)
- [image_rectstddev](#)
 - [nppiRectStdDev_32f_C1R](#), [2093](#)
 - [nppiRectStdDev_32s32f_C1R](#), [2094](#)
 - [nppiRectStdDev_32s_C1RSfs](#), [2094](#)
- [image_remap](#)
 - [nppiRemap_16s_AC4R](#), [1434](#)
 - [nppiRemap_16s_C1R](#), [1435](#)
 - [nppiRemap_16s_C3R](#), [1435](#)
 - [nppiRemap_16s_C4R](#), [1436](#)
 - [nppiRemap_16s_P3R](#), [1437](#)
 - [nppiRemap_16s_P4R](#), [1437](#)
 - [nppiRemap_16u_AC4R](#), [1438](#)
 - [nppiRemap_16u_C1R](#), [1438](#)
 - [nppiRemap_16u_C3R](#), [1439](#)
 - [nppiRemap_16u_C4R](#), [1440](#)
 - [nppiRemap_16u_P3R](#), [1440](#)
 - [nppiRemap_16u_P4R](#), [1441](#)
 - [nppiRemap_32f_AC4R](#), [1441](#)
 - [nppiRemap_32f_C1R](#), [1442](#)
 - [nppiRemap_32f_C3R](#), [1443](#)
 - [nppiRemap_32f_C4R](#), [1443](#)
 - [nppiRemap_32f_P3R](#), [1444](#)
 - [nppiRemap_32f_P4R](#), [1444](#)
 - [nppiRemap_64f_AC4R](#), [1445](#)
 - [nppiRemap_64f_C1R](#), [1446](#)
 - [nppiRemap_64f_C3R](#), [1446](#)
 - [nppiRemap_64f_C4R](#), [1447](#)
 - [nppiRemap_64f_P3R](#), [1447](#)
 - [nppiRemap_64f_P4R](#), [1448](#)
 - [nppiRemap_8u_AC4R](#), [1449](#)
 - [nppiRemap_8u_C1R](#), [1449](#)
 - [nppiRemap_8u_C3R](#), [1450](#)
 - [nppiRemap_8u_C4R](#), [1450](#)
 - [nppiRemap_8u_P3R](#), [1451](#)
 - [nppiRemap_8u_P4R](#), [1452](#)
- [image_resize](#)
 - [nppiResize_16u_AC4R](#), [1421](#)
 - [nppiResize_16u_C1R](#), [1422](#)
 - [nppiResize_16u_C3R](#), [1422](#)
 - [nppiResize_16u_C4R](#), [1423](#)
 - [nppiResize_16u_P3R](#), [1423](#)
 - [nppiResize_16u_P4R](#), [1424](#)
 - [nppiResize_32f_AC4R](#), [1424](#)
 - [nppiResize_32f_C1R](#), [1425](#)
 - [nppiResize_32f_C3R](#), [1425](#)
 - [nppiResize_32f_C4R](#), [1426](#)
 - [nppiResize_32f_P3R](#), [1426](#)
 - [nppiResize_32f_P4R](#), [1427](#)
 - [nppiResize_8u_AC4R](#), [1427](#)
 - [nppiResize_8u_C1R](#), [1428](#)
 - [nppiResize_8u_C3R](#), [1428](#)
 - [nppiResize_8u_C4R](#), [1429](#)
 - [nppiResize_8u_P3R](#), [1429](#)
 - [nppiResize_8u_P4R](#), [1430](#)
- [image_resize_square_pixel](#)
 - [nppiGetResizeRect](#), [1400](#)
 - [nppiResizeAdvancedGetBufferHostSize_8u_C1R](#), [1400](#)
 - [nppiResizeSqrPixel_16s_AC4R](#), [1400](#)
 - [nppiResizeSqrPixel_16s_C1R](#), [1401](#)
 - [nppiResizeSqrPixel_16s_C3R](#), [1401](#)
 - [nppiResizeSqrPixel_16s_C4R](#), [1402](#)
 - [nppiResizeSqrPixel_16s_P3R](#), [1402](#)
 - [nppiResizeSqrPixel_16s_P4R](#), [1403](#)
 - [nppiResizeSqrPixel_16u_AC4R](#), [1404](#)
 - [nppiResizeSqrPixel_16u_C1R](#), [1404](#)
 - [nppiResizeSqrPixel_16u_C3R](#), [1405](#)
 - [nppiResizeSqrPixel_16u_C4R](#), [1405](#)
 - [nppiResizeSqrPixel_16u_P3R](#), [1406](#)
 - [nppiResizeSqrPixel_16u_P4R](#), [1406](#)
 - [nppiResizeSqrPixel_32f_AC4R](#), [1407](#)
 - [nppiResizeSqrPixel_32f_C1R](#), [1408](#)
 - [nppiResizeSqrPixel_32f_C3R](#), [1408](#)
 - [nppiResizeSqrPixel_32f_C4R](#), [1409](#)
 - [nppiResizeSqrPixel_32f_P3R](#), [1409](#)
 - [nppiResizeSqrPixel_32f_P4R](#), [1410](#)
 - [nppiResizeSqrPixel_64f_AC4R](#), [1410](#)
 - [nppiResizeSqrPixel_64f_C1R](#), [1411](#)
 - [nppiResizeSqrPixel_64f_C3R](#), [1411](#)
 - [nppiResizeSqrPixel_64f_C4R](#), [1412](#)
 - [nppiResizeSqrPixel_64f_P3R](#), [1412](#)
 - [nppiResizeSqrPixel_64f_P4R](#), [1413](#)
 - [nppiResizeSqrPixel_8u_AC4R](#), [1414](#)
 - [nppiResizeSqrPixel_8u_C1R](#), [1414](#)
 - [nppiResizeSqrPixel_8u_C1R_Advanced](#), [1415](#)
 - [nppiResizeSqrPixel_8u_C3R](#), [1415](#)
 - [nppiResizeSqrPixel_8u_C4R](#), [1416](#)
 - [nppiResizeSqrPixel_8u_P3R](#), [1416](#)
 - [nppiResizeSqrPixel_8u_P4R](#), [1417](#)
- [image_rotate](#)
 - [nppiGetRotateBound](#), [1454](#)
 - [nppiGetRotateQuad](#), [1455](#)
 - [nppiRotate_16u_AC4R](#), [1455](#)
 - [nppiRotate_16u_C1R](#), [1456](#)
 - [nppiRotate_16u_C3R](#), [1456](#)
 - [nppiRotate_16u_C4R](#), [1457](#)
 - [nppiRotate_32f_AC4R](#), [1457](#)
 - [nppiRotate_32f_C1R](#), [1458](#)

- nppiRotate_32f_C3R, [1458](#)
- nppiRotate_32f_C4R, [1459](#)
- nppiRotate_8u_AC4R, [1459](#)
- nppiRotate_8u_C1R, [1460](#)
- nppiRotate_8u_C3R, [1460](#)
- nppiRotate_8u_C4R, [1461](#)
- image_rshifc
 - nppiRShiftC_16s_AC4IR, [408](#)
 - nppiRShiftC_16s_AC4R, [408](#)
 - nppiRShiftC_16s_C1IR, [409](#)
 - nppiRShiftC_16s_C1R, [409](#)
 - nppiRShiftC_16s_C3IR, [409](#)
 - nppiRShiftC_16s_C3R, [410](#)
 - nppiRShiftC_16s_C4IR, [410](#)
 - nppiRShiftC_16s_C4R, [410](#)
 - nppiRShiftC_16u_AC4IR, [411](#)
 - nppiRShiftC_16u_AC4R, [411](#)
 - nppiRShiftC_16u_C1IR, [411](#)
 - nppiRShiftC_16u_C1R, [412](#)
 - nppiRShiftC_16u_C3IR, [412](#)
 - nppiRShiftC_16u_C3R, [412](#)
 - nppiRShiftC_16u_C4IR, [413](#)
 - nppiRShiftC_16u_C4R, [413](#)
 - nppiRShiftC_32s_AC4IR, [413](#)
 - nppiRShiftC_32s_AC4R, [414](#)
 - nppiRShiftC_32s_C1IR, [414](#)
 - nppiRShiftC_32s_C1R, [414](#)
 - nppiRShiftC_32s_C3IR, [415](#)
 - nppiRShiftC_32s_C3R, [415](#)
 - nppiRShiftC_32s_C4IR, [415](#)
 - nppiRShiftC_32s_C4R, [416](#)
 - nppiRShiftC_8s_AC4IR, [416](#)
 - nppiRShiftC_8s_AC4R, [416](#)
 - nppiRShiftC_8s_C1IR, [417](#)
 - nppiRShiftC_8s_C1R, [417](#)
 - nppiRShiftC_8s_C3IR, [417](#)
 - nppiRShiftC_8s_C3R, [418](#)
 - nppiRShiftC_8s_C4IR, [418](#)
 - nppiRShiftC_8s_C4R, [418](#)
 - nppiRShiftC_8u_AC4IR, [419](#)
 - nppiRShiftC_8u_AC4R, [419](#)
 - nppiRShiftC_8u_C1IR, [419](#)
 - nppiRShiftC_8u_C1R, [420](#)
 - nppiRShiftC_8u_C3IR, [420](#)
 - nppiRShiftC_8u_C3R, [420](#)
 - nppiRShiftC_8u_C4IR, [421](#)
 - nppiRShiftC_8u_C4R, [421](#)
- image_scale
 - nppiScale_16s8u_AC4R, [867](#)
 - nppiScale_16s8u_C1R, [867](#)
 - nppiScale_16s8u_C3R, [867](#)
 - nppiScale_16s8u_C4R, [868](#)
 - nppiScale_16u8u_AC4R, [868](#)
 - nppiScale_16u8u_C1R, [868](#)
 - nppiScale_16u8u_C3R, [869](#)
 - nppiScale_16u8u_C4R, [869](#)
 - nppiScale_32f8u_AC4R, [869](#)
 - nppiScale_32f8u_C1R, [870](#)
 - nppiScale_32f8u_C3R, [870](#)
 - nppiScale_32f8u_C4R, [871](#)
 - nppiScale_32s8u_AC4R, [871](#)
 - nppiScale_32s8u_C1R, [871](#)
 - nppiScale_32s8u_C3R, [872](#)
 - nppiScale_32s8u_C4R, [872](#)
 - nppiScale_8u16s_AC4R, [872](#)
 - nppiScale_8u16s_C1R, [873](#)
 - nppiScale_8u16s_C3R, [873](#)
 - nppiScale_8u16s_C4R, [873](#)
 - nppiScale_8u16u_AC4R, [874](#)
 - nppiScale_8u16u_C1R, [874](#)
 - nppiScale_8u16u_C3R, [874](#)
 - nppiScale_8u16u_C4R, [875](#)
 - nppiScale_8u32f_AC4R, [875](#)
 - nppiScale_8u32f_C1R, [875](#)
 - nppiScale_8u32f_C3R, [876](#)
 - nppiScale_8u32f_C4R, [876](#)
 - nppiScale_8u32s_AC4R, [877](#)
 - nppiScale_8u32s_C1R, [877](#)
 - nppiScale_8u32s_C3R, [877](#)
 - nppiScale_8u32s_C4R, [878](#)
- image_set
 - nppiSet_16s_AC4MR, [745](#)
 - nppiSet_16s_AC4R, [746](#)
 - nppiSet_16s_C1MR, [746](#)
 - nppiSet_16s_C1R, [746](#)
 - nppiSet_16s_C2R, [747](#)
 - nppiSet_16s_C3CR, [747](#)
 - nppiSet_16s_C3MR, [747](#)
 - nppiSet_16s_C3R, [748](#)
 - nppiSet_16s_C4CR, [748](#)
 - nppiSet_16s_C4MR, [748](#)
 - nppiSet_16s_C4R, [749](#)
 - nppiSet_16sc_AC4R, [749](#)
 - nppiSet_16sc_C1R, [749](#)
 - nppiSet_16sc_C2R, [750](#)
 - nppiSet_16sc_C3R, [750](#)
 - nppiSet_16sc_C4R, [750](#)
 - nppiSet_16u_AC4MR, [751](#)
 - nppiSet_16u_AC4R, [751](#)
 - nppiSet_16u_C1MR, [751](#)
 - nppiSet_16u_C1R, [752](#)
 - nppiSet_16u_C2R, [752](#)
 - nppiSet_16u_C3CR, [752](#)
 - nppiSet_16u_C3MR, [753](#)
 - nppiSet_16u_C3R, [753](#)
 - nppiSet_16u_C4CR, [753](#)
 - nppiSet_16u_C4MR, [754](#)
 - nppiSet_16u_C4R, [754](#)

- nppiSet_32f_AC4MR, 754
- nppiSet_32f_AC4R, 755
- nppiSet_32f_C1MR, 755
- nppiSet_32f_C1R, 755
- nppiSet_32f_C2R, 756
- nppiSet_32f_C3CR, 756
- nppiSet_32f_C3MR, 756
- nppiSet_32f_C3R, 757
- nppiSet_32f_C4CR, 757
- nppiSet_32f_C4MR, 757
- nppiSet_32f_C4R, 758
- nppiSet_32fc_AC4R, 758
- nppiSet_32fc_C1R, 758
- nppiSet_32fc_C2R, 759
- nppiSet_32fc_C3R, 759
- nppiSet_32fc_C4R, 759
- nppiSet_32s_AC4MR, 760
- nppiSet_32s_AC4R, 760
- nppiSet_32s_C1MR, 760
- nppiSet_32s_C1R, 761
- nppiSet_32s_C2R, 761
- nppiSet_32s_C3CR, 761
- nppiSet_32s_C3MR, 762
- nppiSet_32s_C3R, 762
- nppiSet_32s_C4CR, 762
- nppiSet_32s_C4MR, 763
- nppiSet_32s_C4R, 763
- nppiSet_32sc_AC4R, 763
- nppiSet_32sc_C1R, 764
- nppiSet_32sc_C2R, 764
- nppiSet_32sc_C3R, 764
- nppiSet_32sc_C4R, 765
- nppiSet_32u_AC4R, 765
- nppiSet_32u_C1R, 765
- nppiSet_32u_C2R, 766
- nppiSet_32u_C3R, 766
- nppiSet_32u_C4R, 766
- nppiSet_8s_AC4R, 767
- nppiSet_8s_C1R, 767
- nppiSet_8s_C2R, 767
- nppiSet_8s_C3R, 768
- nppiSet_8s_C4R, 768
- nppiSet_8u_AC4MR, 768
- nppiSet_8u_AC4R, 769
- nppiSet_8u_C1MR, 769
- nppiSet_8u_C1R, 769
- nppiSet_8u_C2R, 770
- nppiSet_8u_C3CR, 770
- nppiSet_8u_C3MR, 770
- nppiSet_8u_C3R, 771
- nppiSet_8u_C4CR, 771
- nppiSet_8u_C4MR, 771
- nppiSet_8u_C4R, 772
- image_sqr
- nppiSqr_16s_AC4IRSfs, 334
- nppiSqr_16s_AC4RSfs, 334
- nppiSqr_16s_C1IRSfs, 334
- nppiSqr_16s_C1RSfs, 334
- nppiSqr_16s_C3IRSfs, 335
- nppiSqr_16s_C3RSfs, 335
- nppiSqr_16s_C4IRSfs, 335
- nppiSqr_16s_C4RSfs, 336
- nppiSqr_16u_AC4IRSfs, 336
- nppiSqr_16u_AC4RSfs, 336
- nppiSqr_16u_C1IRSfs, 337
- nppiSqr_16u_C1RSfs, 337
- nppiSqr_16u_C3IRSfs, 338
- nppiSqr_16u_C3RSfs, 338
- nppiSqr_16u_C4IRSfs, 338
- nppiSqr_16u_C4RSfs, 339
- nppiSqr_32f_AC4IR, 339
- nppiSqr_32f_AC4R, 339
- nppiSqr_32f_C1IR, 340
- nppiSqr_32f_C1R, 340
- nppiSqr_32f_C3IR, 340
- nppiSqr_32f_C3R, 340
- nppiSqr_32f_C4IR, 341
- nppiSqr_32f_C4R, 341
- nppiSqr_8u_AC4IRSfs, 341
- nppiSqr_8u_AC4RSfs, 342
- nppiSqr_8u_C1IRSfs, 342
- nppiSqr_8u_C1RSfs, 342
- nppiSqr_8u_C3IRSfs, 343
- nppiSqr_8u_C3RSfs, 343
- nppiSqr_8u_C4IRSfs, 343
- nppiSqr_8u_C4RSfs, 344
- image_sqrintegral
 - nppiSqrIntegral_8u32f64f_C1R, 2090
 - nppiSqrIntegral_8u32s64f_C1R, 2091
 - nppiSqrIntegral_8u32s_C1R, 2091
- image_sqrt
 - nppiSqrt_16s_AC4IRSfs, 347
 - nppiSqrt_16s_AC4RSfs, 347
 - nppiSqrt_16s_C1IRSfs, 348
 - nppiSqrt_16s_C1RSfs, 348
 - nppiSqrt_16s_C3IRSfs, 349
 - nppiSqrt_16s_C3RSfs, 349
 - nppiSqrt_16u_AC4IRSfs, 349
 - nppiSqrt_16u_AC4RSfs, 350
 - nppiSqrt_16u_C1IRSfs, 350
 - nppiSqrt_16u_C1RSfs, 350
 - nppiSqrt_16u_C3IRSfs, 351
 - nppiSqrt_16u_C3RSfs, 351
 - nppiSqrt_32f_AC4IR, 351
 - nppiSqrt_32f_AC4R, 352
 - nppiSqrt_32f_C1IR, 352
 - nppiSqrt_32f_C1R, 352
 - nppiSqrt_32f_C3IR, 353

- nppiSqrt_32f_C3R, [353](#)
- nppiSqrt_32f_C4IR, [353](#)
- nppiSqrt_32f_C4R, [354](#)
- nppiSqrt_8u_AC4IRSfs, [354](#)
- nppiSqrt_8u_AC4RSfs, [354](#)
- nppiSqrt_8u_C1IRSfs, [355](#)
- nppiSqrt_8u_C1RSfs, [355](#)
- nppiSqrt_8u_C3IRSfs, [356](#)
- nppiSqrt_8u_C3RSfs, [356](#)
- image_statistics_functions
 - nppiAverageErrorGetBufferHostSize_16s_-C1R, [1651](#)
 - nppiAverageErrorGetBufferHostSize_16s_-C2R, [1651](#)
 - nppiAverageErrorGetBufferHostSize_16s_-C3R, [1651](#)
 - nppiAverageErrorGetBufferHostSize_16s_-C4R, [1651](#)
 - nppiAverageErrorGetBufferHostSize_16sc_-C1R, [1652](#)
 - nppiAverageErrorGetBufferHostSize_16sc_-C2R, [1652](#)
 - nppiAverageErrorGetBufferHostSize_16sc_-C3R, [1652](#)
 - nppiAverageErrorGetBufferHostSize_16sc_-C4R, [1653](#)
 - nppiAverageErrorGetBufferHostSize_16u_-C1R, [1653](#)
 - nppiAverageErrorGetBufferHostSize_16u_-C2R, [1653](#)
 - nppiAverageErrorGetBufferHostSize_16u_-C3R, [1653](#)
 - nppiAverageErrorGetBufferHostSize_16u_-C4R, [1654](#)
 - nppiAverageErrorGetBufferHostSize_32f_-C1R, [1654](#)
 - nppiAverageErrorGetBufferHostSize_32f_-C2R, [1654](#)
 - nppiAverageErrorGetBufferHostSize_32f_-C3R, [1655](#)
 - nppiAverageErrorGetBufferHostSize_32f_-C4R, [1655](#)
 - nppiAverageErrorGetBufferHostSize_32fc_-C1R, [1655](#)
 - nppiAverageErrorGetBufferHostSize_32fc_-C2R, [1655](#)
 - nppiAverageErrorGetBufferHostSize_32fc_-C3R, [1656](#)
 - nppiAverageErrorGetBufferHostSize_32fc_-C4R, [1656](#)
 - nppiAverageErrorGetBufferHostSize_32s_-C1R, [1656](#)
 - nppiAverageErrorGetBufferHostSize_32s_-C2R, [1657](#)
 - nppiAverageErrorGetBufferHostSize_32s_-C3R, [1657](#)
 - nppiAverageErrorGetBufferHostSize_32s_-C4R, [1657](#)
 - nppiAverageErrorGetBufferHostSize_32sc_-C1R, [1657](#)
 - nppiAverageErrorGetBufferHostSize_32sc_-C2R, [1658](#)
 - nppiAverageErrorGetBufferHostSize_32sc_-C3R, [1658](#)
 - nppiAverageErrorGetBufferHostSize_32sc_-C4R, [1658](#)
 - nppiAverageErrorGetBufferHostSize_32u_-C1R, [1659](#)
 - nppiAverageErrorGetBufferHostSize_32u_-C2R, [1659](#)
 - nppiAverageErrorGetBufferHostSize_32u_-C3R, [1659](#)
 - nppiAverageErrorGetBufferHostSize_32u_-C4R, [1659](#)
 - nppiAverageErrorGetBufferHostSize_64f_-C1R, [1660](#)
 - nppiAverageErrorGetBufferHostSize_64f_-C2R, [1660](#)
 - nppiAverageErrorGetBufferHostSize_64f_-C3R, [1660](#)
 - nppiAverageErrorGetBufferHostSize_64f_-C4R, [1661](#)
 - nppiAverageErrorGetBufferHostSize_8s_-C1R, [1661](#)
 - nppiAverageErrorGetBufferHostSize_8s_-C2R, [1661](#)
 - nppiAverageErrorGetBufferHostSize_8s_-C3R, [1661](#)
 - nppiAverageErrorGetBufferHostSize_8s_-C4R, [1662](#)
 - nppiAverageErrorGetBufferHostSize_8u_-C1R, [1662](#)
 - nppiAverageErrorGetBufferHostSize_8u_-C2R, [1662](#)
 - nppiAverageErrorGetBufferHostSize_8u_-C3R, [1663](#)
 - nppiAverageErrorGetBufferHostSize_8u_-C4R, [1663](#)
 - nppiAverageRelativeErrorGetBufferHostSize_16s_C1R, [1663](#)
 - nppiAverageRelativeErrorGetBufferHostSize_16s_C2R, [1663](#)
 - nppiAverageRelativeErrorGetBufferHostSize_16s_C3R, [1664](#)
 - nppiAverageRelativeErrorGetBufferHostSize_16s_C4R, [1664](#)
 - nppiAverageRelativeErrorGetBufferHostSize_16sc_C1R, [1664](#)

- [nppiAverageRelativeErrorGetBufferHostSize_16sc_C2R](#), 1665
- [nppiAverageRelativeErrorGetBufferHostSize_16sc_C3R](#), 1665
- [nppiAverageRelativeErrorGetBufferHostSize_16sc_C4R](#), 1665
- [nppiAverageRelativeErrorGetBufferHostSize_16u_C1R](#), 1665
- [nppiAverageRelativeErrorGetBufferHostSize_16u_C2R](#), 1666
- [nppiAverageRelativeErrorGetBufferHostSize_16u_C3R](#), 1666
- [nppiAverageRelativeErrorGetBufferHostSize_16u_C4R](#), 1666
- [nppiAverageRelativeErrorGetBufferHostSize_32f_C1R](#), 1667
- [nppiAverageRelativeErrorGetBufferHostSize_32f_C2R](#), 1667
- [nppiAverageRelativeErrorGetBufferHostSize_32f_C3R](#), 1667
- [nppiAverageRelativeErrorGetBufferHostSize_32f_C4R](#), 1667
- [nppiAverageRelativeErrorGetBufferHostSize_32fc_C1R](#), 1668
- [nppiAverageRelativeErrorGetBufferHostSize_32fc_C2R](#), 1668
- [nppiAverageRelativeErrorGetBufferHostSize_32fc_C3R](#), 1668
- [nppiAverageRelativeErrorGetBufferHostSize_32fc_C4R](#), 1669
- [nppiAverageRelativeErrorGetBufferHostSize_32s_C1R](#), 1669
- [nppiAverageRelativeErrorGetBufferHostSize_32s_C2R](#), 1669
- [nppiAverageRelativeErrorGetBufferHostSize_32s_C3R](#), 1669
- [nppiAverageRelativeErrorGetBufferHostSize_32s_C4R](#), 1670
- [nppiAverageRelativeErrorGetBufferHostSize_32sc_C1R](#), 1670
- [nppiAverageRelativeErrorGetBufferHostSize_32sc_C2R](#), 1670
- [nppiAverageRelativeErrorGetBufferHostSize_32sc_C3R](#), 1671
- [nppiAverageRelativeErrorGetBufferHostSize_32sc_C4R](#), 1671
- [nppiAverageRelativeErrorGetBufferHostSize_32u_C1R](#), 1671
- [nppiAverageRelativeErrorGetBufferHostSize_32u_C2R](#), 1671
- [nppiAverageRelativeErrorGetBufferHostSize_32u_C3R](#), 1672
- [nppiAverageRelativeErrorGetBufferHostSize_32u_C4R](#), 1672
- [nppiAverageRelativeErrorGetBufferHostSize_64f_C1R](#), 1672
- [nppiAverageRelativeErrorGetBufferHostSize_64f_C2R](#), 1673
- [nppiAverageRelativeErrorGetBufferHostSize_64f_C3R](#), 1673
- [nppiAverageRelativeErrorGetBufferHostSize_64f_C4R](#), 1673
- [nppiAverageRelativeErrorGetBufferHostSize_8s_C1R](#), 1673
- [nppiAverageRelativeErrorGetBufferHostSize_8s_C2R](#), 1674
- [nppiAverageRelativeErrorGetBufferHostSize_8s_C3R](#), 1674
- [nppiAverageRelativeErrorGetBufferHostSize_8s_C4R](#), 1674
- [nppiAverageRelativeErrorGetBufferHostSize_8u_C1R](#), 1675
- [nppiAverageRelativeErrorGetBufferHostSize_8u_C2R](#), 1675
- [nppiAverageRelativeErrorGetBufferHostSize_8u_C3R](#), 1675
- [nppiAverageRelativeErrorGetBufferHostSize_8u_C4R](#), 1675
- [nppiMaximumErrorGetBufferHostSize_16s_C1R](#), 1676
- [nppiMaximumErrorGetBufferHostSize_16s_C2R](#), 1676
- [nppiMaximumErrorGetBufferHostSize_16s_C3R](#), 1676
- [nppiMaximumErrorGetBufferHostSize_16s_C4R](#), 1677
- [nppiMaximumErrorGetBufferHostSize_16sc_C1R](#), 1677
- [nppiMaximumErrorGetBufferHostSize_16sc_C2R](#), 1677
- [nppiMaximumErrorGetBufferHostSize_16sc_C3R](#), 1677
- [nppiMaximumErrorGetBufferHostSize_16sc_C4R](#), 1678
- [nppiMaximumErrorGetBufferHostSize_16u_C1R](#), 1678
- [nppiMaximumErrorGetBufferHostSize_16u_C2R](#), 1678
- [nppiMaximumErrorGetBufferHostSize_16u_C3R](#), 1679
- [nppiMaximumErrorGetBufferHostSize_16u_C4R](#), 1679
- [nppiMaximumErrorGetBufferHostSize_32f_C1R](#), 1679
- [nppiMaximumErrorGetBufferHostSize_32f_C2R](#), 1679
- [nppiMaximumErrorGetBufferHostSize_32f_C3R](#), 1680

- [nppiMaximumErrorGetBufferHostSize_32f_-C4R](#), 1680
- [nppiMaximumErrorGetBufferHostSize_-32fc_C1R](#), 1680
- [nppiMaximumErrorGetBufferHostSize_-32fc_C2R](#), 1681
- [nppiMaximumErrorGetBufferHostSize_-32fc_C3R](#), 1681
- [nppiMaximumErrorGetBufferHostSize_-32fc_C4R](#), 1681
- [nppiMaximumErrorGetBufferHostSize_32s_-C1R](#), 1681
- [nppiMaximumErrorGetBufferHostSize_32s_-C2R](#), 1682
- [nppiMaximumErrorGetBufferHostSize_32s_-C3R](#), 1682
- [nppiMaximumErrorGetBufferHostSize_32s_-C4R](#), 1682
- [nppiMaximumErrorGetBufferHostSize_-32sc_C1R](#), 1683
- [nppiMaximumErrorGetBufferHostSize_-32sc_C2R](#), 1683
- [nppiMaximumErrorGetBufferHostSize_-32sc_C3R](#), 1683
- [nppiMaximumErrorGetBufferHostSize_-32sc_C4R](#), 1683
- [nppiMaximumErrorGetBufferHostSize_32u_-C1R](#), 1684
- [nppiMaximumErrorGetBufferHostSize_32u_-C2R](#), 1684
- [nppiMaximumErrorGetBufferHostSize_32u_-C3R](#), 1684
- [nppiMaximumErrorGetBufferHostSize_32u_-C4R](#), 1685
- [nppiMaximumErrorGetBufferHostSize_64f_-C1R](#), 1685
- [nppiMaximumErrorGetBufferHostSize_64f_-C2R](#), 1685
- [nppiMaximumErrorGetBufferHostSize_64f_-C3R](#), 1685
- [nppiMaximumErrorGetBufferHostSize_64f_-C4R](#), 1686
- [nppiMaximumErrorGetBufferHostSize_8s_-C1R](#), 1686
- [nppiMaximumErrorGetBufferHostSize_8s_-C2R](#), 1686
- [nppiMaximumErrorGetBufferHostSize_8s_-C3R](#), 1687
- [nppiMaximumErrorGetBufferHostSize_8s_-C4R](#), 1687
- [nppiMaximumErrorGetBufferHostSize_8u_-C1R](#), 1687
- [nppiMaximumErrorGetBufferHostSize_8u_-C2R](#), 1687
- [nppiMaximumErrorGetBufferHostSize_8u_-C3R](#), 1688
- [nppiMaximumErrorGetBufferHostSize_8u_-C4R](#), 1688
- [nppiMaximumRelativeErrorGetBufferHostSize_-16s_C1R](#), 1688
- [nppiMaximumRelativeErrorGetBufferHostSize_-16s_C2R](#), 1689
- [nppiMaximumRelativeErrorGetBufferHostSize_-16s_C3R](#), 1689
- [nppiMaximumRelativeErrorGetBufferHostSize_-16s_C4R](#), 1689
- [nppiMaximumRelativeErrorGetBufferHostSize_-16sc_C1R](#), 1689
- [nppiMaximumRelativeErrorGetBufferHostSize_-16sc_C2R](#), 1690
- [nppiMaximumRelativeErrorGetBufferHostSize_-16sc_C3R](#), 1690
- [nppiMaximumRelativeErrorGetBufferHostSize_-16sc_C4R](#), 1690
- [nppiMaximumRelativeErrorGetBufferHostSize_-16u_C1R](#), 1691
- [nppiMaximumRelativeErrorGetBufferHostSize_-16u_C2R](#), 1691
- [nppiMaximumRelativeErrorGetBufferHostSize_-16u_C3R](#), 1691
- [nppiMaximumRelativeErrorGetBufferHostSize_-16u_C4R](#), 1691
- [nppiMaximumRelativeErrorGetBufferHostSize_-32f_C1R](#), 1692
- [nppiMaximumRelativeErrorGetBufferHostSize_-32f_C2R](#), 1692
- [nppiMaximumRelativeErrorGetBufferHostSize_-32f_C3R](#), 1692
- [nppiMaximumRelativeErrorGetBufferHostSize_-32f_C4R](#), 1693
- [nppiMaximumRelativeErrorGetBufferHostSize_-32fc_C1R](#), 1693
- [nppiMaximumRelativeErrorGetBufferHostSize_-32fc_C2R](#), 1693
- [nppiMaximumRelativeErrorGetBufferHostSize_-32fc_C3R](#), 1693
- [nppiMaximumRelativeErrorGetBufferHostSize_-32fc_C4R](#), 1694
- [nppiMaximumRelativeErrorGetBufferHostSize_-32s_C1R](#), 1694
- [nppiMaximumRelativeErrorGetBufferHostSize_-32s_C2R](#), 1694
- [nppiMaximumRelativeErrorGetBufferHostSize_-32s_C3R](#), 1695
- [nppiMaximumRelativeErrorGetBufferHostSize_-32s_C4R](#), 1695
- [nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C1R](#), 1695

- nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C2R, [1695](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C3R, [1696](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C4R, [1696](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C1R, [1696](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C2R, [1697](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C3R, [1697](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C4R, [1697](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C1R, [1697](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C2R, [1698](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C3R, [1698](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C4R, [1698](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C1R, [1699](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C2R, [1699](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C3R, [1699](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C4R, [1699](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C1R, [1700](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C2R, [1700](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C3R, [1700](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C4R, [1701](#)
- image_sub
 - nppiSub_16s_AC4IRSfs, [252](#)
 - nppiSub_16s_AC4RSfs, [253](#)
 - nppiSub_16s_C1IRSfs, [253](#)
 - nppiSub_16s_C3IRSfs, [254](#)
 - nppiSub_16s_C3RSfs, [254](#)
 - nppiSub_16s_C4IRSfs, [255](#)
 - nppiSub_16s_C4RSfs, [255](#)
 - nppiSub_16sc_AC4IRSfs, [255](#)
 - nppiSub_16sc_AC4RSfs, [256](#)
 - nppiSub_16sc_C1IRSfs, [256](#)
 - nppiSub_16sc_C3IRSfs, [257](#)
 - nppiSub_16sc_C3RSfs, [257](#)
 - nppiSub_16u_AC4IRSfs, [258](#)
 - nppiSub_16u_AC4RSfs, [258](#)
 - nppiSub_16u_C1IRSfs, [259](#)
 - nppiSub_16u_C1RSfs, [259](#)
 - nppiSub_16u_C3IRSfs, [260](#)
 - nppiSub_16u_C3RSfs, [260](#)
 - nppiSub_16u_C4IRSfs, [260](#)
 - nppiSub_16u_C4RSfs, [261](#)
 - nppiSub_32f_AC4IR, [261](#)
 - nppiSub_32f_AC4R, [262](#)
 - nppiSub_32f_C1IR, [262](#)
 - nppiSub_32f_C1R, [262](#)
 - nppiSub_32f_C3IR, [263](#)
 - nppiSub_32f_C3R, [263](#)
 - nppiSub_32f_C4IR, [264](#)
 - nppiSub_32f_C4R, [264](#)
 - nppiSub_32fc_AC4IR, [264](#)
 - nppiSub_32fc_AC4R, [265](#)
 - nppiSub_32fc_C1IR, [265](#)
 - nppiSub_32fc_C1R, [266](#)
 - nppiSub_32fc_C3IR, [266](#)
 - nppiSub_32fc_C3R, [266](#)
 - nppiSub_32fc_C4IR, [267](#)
 - nppiSub_32fc_C4R, [267](#)
 - nppiSub_32s_C1IRSfs, [268](#)
 - nppiSub_32s_C1R, [268](#)
 - nppiSub_32s_C1RSfs, [268](#)
 - nppiSub_32s_C3IRSfs, [269](#)
 - nppiSub_32s_C3RSfs, [269](#)
 - nppiSub_32s_C4IRSfs, [270](#)
 - nppiSub_32s_C4RSfs, [270](#)
 - nppiSub_32sc_AC4IRSfs, [271](#)
 - nppiSub_32sc_AC4RSfs, [271](#)
 - nppiSub_32sc_C1IRSfs, [271](#)
 - nppiSub_32sc_C1RSfs, [272](#)
 - nppiSub_32sc_C3IRSfs, [272](#)
 - nppiSub_32sc_C3RSfs, [273](#)
 - nppiSub_8u_AC4IRSfs, [273](#)
 - nppiSub_8u_AC4RSfs, [273](#)
 - nppiSub_8u_C1IRSfs, [274](#)
 - nppiSub_8u_C1RSfs, [274](#)
 - nppiSub_8u_C3IRSfs, [275](#)
 - nppiSub_8u_C3RSfs, [275](#)
 - nppiSub_8u_C4IRSfs, [275](#)
 - nppiSub_8u_C4RSfs, [276](#)
- image_subc
 - nppiSubC_16s_AC4IRSfs, [120](#)
 - nppiSubC_16s_AC4RSfs, [120](#)
 - nppiSubC_16s_C1IRSfs, [120](#)
 - nppiSubC_16s_C1RSfs, [121](#)
 - nppiSubC_16s_C3IRSfs, [121](#)
 - nppiSubC_16s_C3RSfs, [121](#)
 - nppiSubC_16s_C4IRSfs, [122](#)
 - nppiSubC_16s_C4RSfs, [122](#)
 - nppiSubC_16sc_AC4IRSfs, [123](#)

- nppiSubC_16sc_AC4RSfs, [123](#)
- nppiSubC_16sc_C1IRSfs, [123](#)
- nppiSubC_16sc_C1RSfs, [124](#)
- nppiSubC_16sc_C3IRSfs, [124](#)
- nppiSubC_16sc_C3RSfs, [125](#)
- nppiSubC_16u_AC4IRSfs, [125](#)
- nppiSubC_16u_AC4RSfs, [125](#)
- nppiSubC_16u_C1IRSfs, [126](#)
- nppiSubC_16u_C1RSfs, [126](#)
- nppiSubC_16u_C3IRSfs, [127](#)
- nppiSubC_16u_C3RSfs, [127](#)
- nppiSubC_16u_C4IRSfs, [127](#)
- nppiSubC_16u_C4RSfs, [128](#)
- nppiSubC_32f_AC4IR, [128](#)
- nppiSubC_32f_AC4R, [128](#)
- nppiSubC_32f_C1IR, [129](#)
- nppiSubC_32f_C1R, [129](#)
- nppiSubC_32f_C3IR, [129](#)
- nppiSubC_32f_C3R, [130](#)
- nppiSubC_32f_C4IR, [130](#)
- nppiSubC_32f_C4R, [130](#)
- nppiSubC_32fc_AC4IR, [131](#)
- nppiSubC_32fc_AC4R, [131](#)
- nppiSubC_32fc_C1IR, [131](#)
- nppiSubC_32fc_C1R, [132](#)
- nppiSubC_32fc_C3IR, [132](#)
- nppiSubC_32fc_C3R, [132](#)
- nppiSubC_32fc_C4IR, [133](#)
- nppiSubC_32fc_C4R, [133](#)
- nppiSubC_32s_C1IRSfs, [134](#)
- nppiSubC_32s_C1RSfs, [134](#)
- nppiSubC_32s_C3IRSfs, [134](#)
- nppiSubC_32s_C3RSfs, [135](#)
- nppiSubC_32sc_AC4IRSfs, [135](#)
- nppiSubC_32sc_AC4RSfs, [135](#)
- nppiSubC_32sc_C1IRSfs, [136](#)
- nppiSubC_32sc_C1RSfs, [137](#)
- nppiSubC_32sc_C3RSfs, [137](#)
- nppiSubC_8u_AC4IRSfs, [137](#)
- nppiSubC_8u_AC4RSfs, [138](#)
- nppiSubC_8u_C1IRSfs, [138](#)
- nppiSubC_8u_C1RSfs, [139](#)
- nppiSubC_8u_C3IRSfs, [139](#)
- nppiSubC_8u_C3RSfs, [139](#)
- nppiSubC_8u_C4IRSfs, [140](#)
- nppiSubC_8u_C4RSfs, [140](#)
- image_sum
 - nppiSum_16s_AC4R, [1705](#)
 - nppiSum_16s_C1R, [1705](#)
 - nppiSum_16s_C3R, [1705](#)
 - nppiSum_16s_C4R, [1706](#)
 - nppiSum_16u_AC4R, [1706](#)
 - nppiSum_16u_C1R, [1706](#)
 - nppiSum_16u_C3R, [1707](#)
 - nppiSum_16u_C4R, [1707](#)
 - nppiSum_32f_AC4R, [1707](#)
 - nppiSum_32f_C1R, [1708](#)
 - nppiSum_32f_C3R, [1708](#)
 - nppiSum_32f_C4R, [1708](#)
 - nppiSum_8u64s_C1R, [1709](#)
 - nppiSum_8u64s_C4R, [1709](#)
 - nppiSum_8u_AC4R, [1710](#)
 - nppiSum_8u_C1R, [1710](#)
 - nppiSum_8u_C3R, [1710](#)
 - nppiSum_8u_C4R, [1711](#)
 - nppiSumGetBufferHostSize_16s_AC4R, [1711](#)
 - nppiSumGetBufferHostSize_16s_C1R, [1711](#)
 - nppiSumGetBufferHostSize_16s_C3R, [1712](#)
 - nppiSumGetBufferHostSize_16s_C4R, [1712](#)
 - nppiSumGetBufferHostSize_16u_AC4R, [1712](#)
 - nppiSumGetBufferHostSize_16u_C1R, [1713](#)
 - nppiSumGetBufferHostSize_16u_C3R, [1713](#)
 - nppiSumGetBufferHostSize_16u_C4R, [1713](#)
 - nppiSumGetBufferHostSize_32f_AC4R, [1713](#)
 - nppiSumGetBufferHostSize_32f_C1R, [1714](#)
 - nppiSumGetBufferHostSize_32f_C3R, [1714](#)
 - nppiSumGetBufferHostSize_32f_C4R, [1714](#)
 - nppiSumGetBufferHostSize_8u64s_C1R, [1715](#)
 - nppiSumGetBufferHostSize_8u64s_C4R, [1715](#)
 - nppiSumGetBufferHostSize_8u_AC4R, [1715](#)
 - nppiSumGetBufferHostSize_8u_C1R, [1715](#)
 - nppiSumGetBufferHostSize_8u_C3R, [1716](#)
 - nppiSumGetBufferHostSize_8u_C4R, [1716](#)
- image_swap_channels
 - nppiSwapChannels_16s_AC4R, [945](#)
 - nppiSwapChannels_16s_C3C4R, [945](#)
 - nppiSwapChannels_16s_C3IR, [945](#)
 - nppiSwapChannels_16s_C3R, [946](#)
 - nppiSwapChannels_16s_C4C3R, [946](#)
 - nppiSwapChannels_16s_C4IR, [947](#)
 - nppiSwapChannels_16s_C4R, [947](#)
 - nppiSwapChannels_16u_AC4R, [947](#)
 - nppiSwapChannels_16u_C3C4R, [948](#)
 - nppiSwapChannels_16u_C3IR, [948](#)
 - nppiSwapChannels_16u_C3R, [949](#)
 - nppiSwapChannels_16u_C4C3R, [949](#)
 - nppiSwapChannels_16u_C4IR, [950](#)
 - nppiSwapChannels_16u_C4R, [950](#)
 - nppiSwapChannels_32f_AC4R, [950](#)
 - nppiSwapChannels_32f_C3C4R, [951](#)
 - nppiSwapChannels_32f_C3IR, [951](#)
 - nppiSwapChannels_32f_C3R, [952](#)
 - nppiSwapChannels_32f_C4C3R, [952](#)
 - nppiSwapChannels_32f_C4IR, [953](#)
 - nppiSwapChannels_32f_C4R, [953](#)

- nppiSwapChannels_32s_AC4R, [953](#)
- nppiSwapChannels_32s_C3C4R, [954](#)
- nppiSwapChannels_32s_C3IR, [954](#)
- nppiSwapChannels_32s_C3R, [955](#)
- nppiSwapChannels_32s_C4C3R, [955](#)
- nppiSwapChannels_32s_C4IR, [956](#)
- nppiSwapChannels_32s_C4R, [956](#)
- nppiSwapChannels_8u_AC4R, [956](#)
- nppiSwapChannels_8u_C3C4R, [957](#)
- nppiSwapChannels_8u_C3IR, [957](#)
- nppiSwapChannels_8u_C3R, [958](#)
- nppiSwapChannels_8u_C4C3R, [958](#)
- nppiSwapChannels_8u_C4IR, [959](#)
- nppiSwapChannels_8u_C4R, [959](#)
- image_threshold_operations
 - nppiThreshold_16s_AC4IR, [2387](#)
 - nppiThreshold_16s_AC4R, [2387](#)
 - nppiThreshold_16s_C1IR, [2388](#)
 - nppiThreshold_16s_C1R, [2388](#)
 - nppiThreshold_16s_C3IR, [2389](#)
 - nppiThreshold_16s_C3R, [2389](#)
 - nppiThreshold_16u_AC4IR, [2390](#)
 - nppiThreshold_16u_AC4R, [2390](#)
 - nppiThreshold_16u_C1IR, [2390](#)
 - nppiThreshold_16u_C1R, [2391](#)
 - nppiThreshold_16u_C3IR, [2391](#)
 - nppiThreshold_16u_C3R, [2392](#)
 - nppiThreshold_32f_AC4IR, [2392](#)
 - nppiThreshold_32f_AC4R, [2393](#)
 - nppiThreshold_32f_C1IR, [2393](#)
 - nppiThreshold_32f_C1R, [2394](#)
 - nppiThreshold_32f_C3IR, [2394](#)
 - nppiThreshold_32f_C3R, [2394](#)
 - nppiThreshold_8u_AC4IR, [2395](#)
 - nppiThreshold_8u_AC4R, [2395](#)
 - nppiThreshold_8u_C1IR, [2396](#)
 - nppiThreshold_8u_C1R, [2396](#)
 - nppiThreshold_8u_C3IR, [2397](#)
 - nppiThreshold_8u_C3R, [2397](#)
 - nppiThreshold_GT_16s_AC4IR, [2398](#)
 - nppiThreshold_GT_16s_AC4R, [2398](#)
 - nppiThreshold_GT_16s_C1IR, [2399](#)
 - nppiThreshold_GT_16s_C1R, [2399](#)
 - nppiThreshold_GT_16s_C3IR, [2399](#)
 - nppiThreshold_GT_16s_C3R, [2400](#)
 - nppiThreshold_GT_16u_AC4IR, [2400](#)
 - nppiThreshold_GT_16u_AC4R, [2401](#)
 - nppiThreshold_GT_16u_C1IR, [2401](#)
 - nppiThreshold_GT_16u_C1R, [2401](#)
 - nppiThreshold_GT_16u_C3IR, [2402](#)
 - nppiThreshold_GT_16u_C3R, [2402](#)
 - nppiThreshold_GT_32f_AC4IR, [2403](#)
 - nppiThreshold_GT_32f_AC4R, [2403](#)
 - nppiThreshold_GT_32f_C1IR, [2403](#)
 - nppiThreshold_GT_32f_C1R, [2404](#)
 - nppiThreshold_GT_32f_C3IR, [2404](#)
 - nppiThreshold_GT_32f_C3R, [2405](#)
 - nppiThreshold_GT_8u_AC4IR, [2405](#)
 - nppiThreshold_GT_8u_AC4R, [2405](#)
 - nppiThreshold_GT_8u_C1IR, [2406](#)
 - nppiThreshold_GT_8u_C1R, [2406](#)
 - nppiThreshold_GT_8u_C3IR, [2407](#)
 - nppiThreshold_GT_8u_C3R, [2407](#)
 - nppiThreshold_GTVVal_16s_AC4IR, [2407](#)
 - nppiThreshold_GTVVal_16s_AC4R, [2408](#)
 - nppiThreshold_GTVVal_16s_C1IR, [2408](#)
 - nppiThreshold_GTVVal_16s_C1R, [2409](#)
 - nppiThreshold_GTVVal_16s_C3IR, [2409](#)
 - nppiThreshold_GTVVal_16s_C3R, [2409](#)
 - nppiThreshold_GTVVal_16u_AC4IR, [2410](#)
 - nppiThreshold_GTVVal_16u_AC4R, [2410](#)
 - nppiThreshold_GTVVal_16u_C1IR, [2411](#)
 - nppiThreshold_GTVVal_16u_C1R, [2411](#)
 - nppiThreshold_GTVVal_16u_C3IR, [2412](#)
 - nppiThreshold_GTVVal_16u_C3R, [2412](#)
 - nppiThreshold_GTVVal_32f_AC4IR, [2412](#)
 - nppiThreshold_GTVVal_32f_AC4R, [2413](#)
 - nppiThreshold_GTVVal_32f_C1IR, [2413](#)
 - nppiThreshold_GTVVal_32f_C1R, [2414](#)
 - nppiThreshold_GTVVal_32f_C3IR, [2414](#)
 - nppiThreshold_GTVVal_32f_C3R, [2414](#)
 - nppiThreshold_GTVVal_8u_AC4IR, [2415](#)
 - nppiThreshold_GTVVal_8u_AC4R, [2415](#)
 - nppiThreshold_GTVVal_8u_C1IR, [2416](#)
 - nppiThreshold_GTVVal_8u_C1R, [2416](#)
 - nppiThreshold_GTVVal_8u_C3IR, [2417](#)
 - nppiThreshold_GTVVal_8u_C3R, [2417](#)
 - nppiThreshold_LT_16s_AC4IR, [2417](#)
 - nppiThreshold_LT_16s_AC4R, [2418](#)
 - nppiThreshold_LT_16s_C1IR, [2418](#)
 - nppiThreshold_LT_16s_C1R, [2419](#)
 - nppiThreshold_LT_16s_C3IR, [2419](#)
 - nppiThreshold_LT_16s_C3R, [2419](#)
 - nppiThreshold_LT_16u_AC4IR, [2420](#)
 - nppiThreshold_LT_16u_AC4R, [2420](#)
 - nppiThreshold_LT_16u_C1IR, [2421](#)
 - nppiThreshold_LT_16u_C1R, [2421](#)
 - nppiThreshold_LT_16u_C3IR, [2421](#)
 - nppiThreshold_LT_16u_C3R, [2422](#)
 - nppiThreshold_LT_32f_AC4IR, [2422](#)
 - nppiThreshold_LT_32f_AC4R, [2423](#)
 - nppiThreshold_LT_32f_C1IR, [2423](#)
 - nppiThreshold_LT_32f_C1R, [2423](#)
 - nppiThreshold_LT_32f_C3IR, [2424](#)
 - nppiThreshold_LT_32f_C3R, [2424](#)
 - nppiThreshold_LT_8u_AC4IR, [2425](#)
 - nppiThreshold_LT_8u_AC4R, [2425](#)
 - nppiThreshold_LT_8u_C1IR, [2425](#)

- nppiThreshold_LT_8u_C1R, [2426](#)
- nppiThreshold_LT_8u_C3IR, [2426](#)
- nppiThreshold_LT_8u_C3R, [2427](#)
- nppiThreshold_LTVa16s_AC4IR, [2427](#)
- nppiThreshold_LTVa16s_AC4R, [2427](#)
- nppiThreshold_LTVa16s_C1IR, [2428](#)
- nppiThreshold_LTVa16s_C1R, [2428](#)
- nppiThreshold_LTVa16s_C3IR, [2429](#)
- nppiThreshold_LTVa16s_C3R, [2429](#)
- nppiThreshold_LTVa16u_AC4IR, [2430](#)
- nppiThreshold_LTVa16u_AC4R, [2430](#)
- nppiThreshold_LTVa16u_C1IR, [2430](#)
- nppiThreshold_LTVa16u_C1R, [2431](#)
- nppiThreshold_LTVa16u_C3IR, [2431](#)
- nppiThreshold_LTVa16u_C3R, [2432](#)
- nppiThreshold_LTVa32f_AC4IR, [2432](#)
- nppiThreshold_LTVa32f_AC4R, [2432](#)
- nppiThreshold_LTVa32f_C1IR, [2433](#)
- nppiThreshold_LTVa32f_C1R, [2433](#)
- nppiThreshold_LTVa32f_C3IR, [2434](#)
- nppiThreshold_LTVa32f_C3R, [2434](#)
- nppiThreshold_LTVa8u_AC4IR, [2435](#)
- nppiThreshold_LTVa8u_AC4R, [2435](#)
- nppiThreshold_LTVa8u_C1IR, [2435](#)
- nppiThreshold_LTVa8u_C1R, [2436](#)
- nppiThreshold_LTVa8u_C3IR, [2436](#)
- nppiThreshold_LTVa8u_C3R, [2437](#)
- nppiThreshold_LTVaGTVal_16s_AC4IR, [2437](#)
- nppiThreshold_LTVaGTVal_16s_AC4R, [2438](#)
- nppiThreshold_LTVaGTVal_16s_C1IR, [2438](#)
- nppiThreshold_LTVaGTVal_16s_C1R, [2439](#)
- nppiThreshold_LTVaGTVal_16s_C3IR, [2439](#)
- nppiThreshold_LTVaGTVal_16s_C3R, [2440](#)
- nppiThreshold_LTVaGTVal_16u_AC4IR, [2440](#)
- nppiThreshold_LTVaGTVal_16u_AC4R, [2441](#)
- nppiThreshold_LTVaGTVal_16u_C1IR, [2441](#)
- nppiThreshold_LTVaGTVal_16u_C1R, [2442](#)
- nppiThreshold_LTVaGTVal_16u_C3IR, [2442](#)
- nppiThreshold_LTVaGTVal_16u_C3R, [2443](#)
- nppiThreshold_LTVaGTVal_32f_AC4IR, [2443](#)
- nppiThreshold_LTVaGTVal_32f_AC4R, [2444](#)
- nppiThreshold_LTVaGTVal_32f_C1IR, [2444](#)
- nppiThreshold_LTVaGTVal_32f_C1R, [2445](#)
- nppiThreshold_LTVaGTVal_32f_C3IR, [2445](#)
- nppiThreshold_LTVaGTVal_32f_C3R, [2446](#)
- nppiThreshold_LTVaGTVal_8u_AC4IR, [2446](#)
- nppiThreshold_LTVaGTVal_8u_AC4R, [2447](#)
- nppiThreshold_LTVaGTVal_8u_C1IR, [2447](#)
- nppiThreshold_LTVaGTVal_8u_C1R, [2448](#)
- nppiThreshold_LTVaGTVal_8u_C3IR, [2448](#)
- nppiThreshold_LTVaGTVal_8u_C3R, [2449](#)
- nppiThreshold_Val_16s_AC4IR, [2449](#)
- nppiThreshold_Val_16s_AC4R, [2450](#)
- nppiThreshold_Val_16s_C1IR, [2450](#)
- nppiThreshold_Val_16s_C1R, [2451](#)
- nppiThreshold_Val_16s_C3IR, [2451](#)
- nppiThreshold_Val_16s_C3R, [2452](#)
- nppiThreshold_Val_16u_AC4IR, [2452](#)
- nppiThreshold_Val_16u_AC4R, [2453](#)
- nppiThreshold_Val_16u_C1IR, [2453](#)
- nppiThreshold_Val_16u_C1R, [2454](#)
- nppiThreshold_Val_16u_C3IR, [2454](#)
- nppiThreshold_Val_16u_C3R, [2455](#)
- nppiThreshold_Val_32f_AC4IR, [2455](#)
- nppiThreshold_Val_32f_AC4R, [2456](#)
- nppiThreshold_Val_32f_C1IR, [2456](#)
- nppiThreshold_Val_32f_C1R, [2457](#)
- nppiThreshold_Val_32f_C3IR, [2457](#)
- nppiThreshold_Val_32f_C3R, [2458](#)
- nppiThreshold_Val_8u_AC4IR, [2458](#)
- nppiThreshold_Val_8u_AC4R, [2459](#)
- nppiThreshold_Val_8u_C1IR, [2459](#)
- nppiThreshold_Val_8u_C1R, [2460](#)
- nppiThreshold_Val_8u_C3IR, [2460](#)
- nppiThreshold_Val_8u_C3R, [2461](#)
- image_transpose
 - nppiTranspose_16s_C1R, [936](#)
 - nppiTranspose_16s_C3R, [936](#)
 - nppiTranspose_16s_C4R, [937](#)
 - nppiTranspose_16u_C1R, [937](#)
 - nppiTranspose_16u_C3R, [937](#)
 - nppiTranspose_16u_C4R, [938](#)
 - nppiTranspose_32f_C1R, [938](#)
 - nppiTranspose_32f_C3R, [938](#)
 - nppiTranspose_32f_C4R, [939](#)
 - nppiTranspose_32s_C1R, [939](#)
 - nppiTranspose_32s_C3R, [939](#)
 - nppiTranspose_32s_C4R, [940](#)
 - nppiTranspose_8u_C1R, [940](#)
 - nppiTranspose_8u_C3R, [940](#)
 - nppiTranspose_8u_C4R, [941](#)
- image_xor
 - nppiXor_16u_AC4IR, [459](#)
 - nppiXor_16u_AC4R, [459](#)
 - nppiXor_16u_C1IR, [459](#)
 - nppiXor_16u_C1R, [460](#)
 - nppiXor_16u_C3IR, [460](#)
 - nppiXor_16u_C3R, [460](#)
 - nppiXor_16u_C4IR, [461](#)
 - nppiXor_16u_C4R, [461](#)
 - nppiXor_32s_AC4IR, [462](#)

- nppiXor_32s_AC4R, [462](#)
 - nppiXor_32s_C1IR, [462](#)
 - nppiXor_32s_C1R, [463](#)
 - nppiXor_32s_C3IR, [463](#)
 - nppiXor_32s_C3R, [463](#)
 - nppiXor_32s_C4IR, [464](#)
 - nppiXor_32s_C4R, [464](#)
 - nppiXor_8u_AC4IR, [465](#)
 - nppiXor_8u_AC4R, [465](#)
 - nppiXor_8u_C1IR, [465](#)
 - nppiXor_8u_C1R, [466](#)
 - nppiXor_8u_C3IR, [466](#)
 - nppiXor_8u_C3R, [466](#)
 - nppiXor_8u_C4IR, [467](#)
 - nppiXor_8u_C4R, [467](#)
- image_xorc
 - nppiXorC_16u_AC4IR, [396](#)
 - nppiXorC_16u_AC4R, [396](#)
 - nppiXorC_16u_C1IR, [396](#)
 - nppiXorC_16u_C1R, [397](#)
 - nppiXorC_16u_C3IR, [397](#)
 - nppiXorC_16u_C3R, [397](#)
 - nppiXorC_16u_C4IR, [398](#)
 - nppiXorC_16u_C4R, [398](#)
 - nppiXorC_32s_AC4IR, [398](#)
 - nppiXorC_32s_AC4R, [399](#)
 - nppiXorC_32s_C1IR, [399](#)
 - nppiXorC_32s_C1R, [399](#)
 - nppiXorC_32s_C3IR, [400](#)
 - nppiXorC_32s_C3R, [400](#)
 - nppiXorC_32s_C4IR, [400](#)
 - nppiXorC_32s_C4R, [401](#)
 - nppiXorC_8u_AC4IR, [401](#)
 - nppiXorC_8u_AC4R, [401](#)
 - nppiXorC_8u_C1IR, [402](#)
 - nppiXorC_8u_C1R, [402](#)
 - nppiXorC_8u_C3IR, [402](#)
 - nppiXorC_8u_C3R, [403](#)
 - nppiXorC_8u_C4IR, [403](#)
 - nppiXorC_8u_C4R, [403](#)
- Infinity Norm, [2756](#)
- Infinity Norm Diff, [2773](#)
- Initialization, [2685](#)
- Integral, [2088](#), [2684](#)
- L1 Norm, [2761](#)
- L1 Norm Diff, [2778](#)
- L2 Norm, [2767](#)
- L2 Norm Diff, [2784](#)
- Labeling and Segmentation, [730](#)
- Linear Transforms, [1575](#)
- Ln, [357](#), [2608](#)
- Logical And Shift Operations, [2624](#)
- Logical Operations, [371](#)
- LShiftC, [422](#), [2646](#)
- major
 - NppLibraryVersion, [2876](#)
- Malloc, [2860](#)
- Max, [1744](#)
- MaxEvery, [2074](#)
- Maximum, [2711](#)
- MaximumError, [2266](#), [2813](#)
- MaximumRelativeError, [2312](#), [2835](#)
- MaxIndx, [1757](#)
- Mean, [1802](#), [2731](#)
- Mean And Standard Deviation, [2740](#)
- Mean_StdDev, [1823](#)
- Memory Management, [2360](#), [2859](#)
- Min, [1717](#)
- MinEvery, [2081](#)
- MinEvery And MaxEvery Functions, [2700](#)
- Minimum, [2721](#)
- Minimum_Maximum, [2744](#)
- MinIndx, [1730](#)
- MinMax, [1771](#)
- MinMaxIndx, [1785](#)
- minor
 - NppLibraryVersion, [2876](#)
- Mirror, [1462](#)
- Morphological Operations, [1578](#)
- Mul, [209](#), [2552](#)
- MulC, [82](#), [2499](#)
- MulCScale, [108](#)
- MulScale, [238](#)
- Norm_Inf, [1841](#)
- Norm_L1, [1863](#)
- Norm_L2, [1884](#)
- Normalize, [2619](#)
- NormDiff_Inf, [1905](#)
- NormDiff_L1, [1928](#)
- NormDiff_L2, [1951](#)
- NormRel_Inf, [1974](#)
- NormRel_L1, [1997](#)
- NormRel_L2, [2020](#)
- Not, [469](#), [2643](#)
- NPP Core, [31](#)
- NPP Image Processing, [52](#)
- NPP Signal Processing, [2485](#)
- NPP Type Definitions and Constants, [34](#)
- Npp16s
 - npp_basic_types, [49](#)
- Npp16sc
 - npp_basic_types, [51](#)
- Npp16u
 - npp_basic_types, [49](#)
- Npp16uc

- npp_basic_types, 51
- Npp32f
 - npp_basic_types, 49
- Npp32fc
 - npp_basic_types, 49
- Npp32s
 - npp_basic_types, 49
- Npp32sc
 - npp_basic_types, 49
- Npp32u
 - npp_basic_types, 50
- Npp32uc
 - npp_basic_types, 50
- Npp64f
 - npp_basic_types, 50
- Npp64fc
 - npp_basic_types, 50
- Npp64s
 - npp_basic_types, 50
- Npp64sc
 - npp_basic_types, 50
- Npp64u
 - npp_basic_types, 50
- Npp8s
 - npp_basic_types, 50
- Npp8u
 - npp_basic_types, 50
- Npp8uc
 - npp_basic_types, 51
- NPP_AFFINE_QUAD_INCORRECT_WARNING
 - typedefs_npp, 47
- NPP_ALG_HINT_ACCURATE
 - typedefs_npp, 42
- NPP_ALG_HINT_FAST
 - typedefs_npp, 42
- NPP_ALG_HINT_NONE
 - typedefs_npp, 42
- NPP_ALIGNMENT_ERROR
 - typedefs_npp, 45
- NPP_ANCHOR_ERROR
 - typedefs_npp, 46
- NPP_BAD_ARGUMENT_ERROR
 - typedefs_npp, 46
- NPP_BORDER_CONSTANT
 - typedefs_npp, 43
- NPP_BORDER_MIRROR
 - typedefs_npp, 43
- NPP_BORDER_NONE
 - typedefs_npp, 43
- NPP_BORDER_REPLICATE
 - typedefs_npp, 43
- NPP_BORDER_UNDEFINED
 - typedefs_npp, 43
- NPP_BORDER_WRAP
 - typedefs_npp, 43
- NPP_BOTH_AXIS
 - typedefs_npp, 43
- NPP_CHANNEL_ERROR
 - typedefs_npp, 46
- NPP_CHANNEL_ORDER_ERROR
 - typedefs_npp, 46
- NPP_CMP_EQ
 - typedefs_npp, 41
- NPP_CMP_GREATER
 - typedefs_npp, 41
- NPP_CMP_GREATER_EQ
 - typedefs_npp, 41
- NPP_CMP_LESS
 - typedefs_npp, 41
- NPP_CMP_LESS_EQ
 - typedefs_npp, 41
- NPP_COEFFICIENT_ERROR
 - typedefs_npp, 46
- NPP_COI_ERROR
 - typedefs_npp, 46
- NPP_CONTEXT_MATCH_ERROR
 - typedefs_npp, 46
- NPP_CORRUPTED_DATA_ERROR
 - typedefs_npp, 46
- NPP_CUDA_1_0
 - typedefs_npp, 41
- NPP_CUDA_1_1
 - typedefs_npp, 42
- NPP_CUDA_1_2
 - typedefs_npp, 42
- NPP_CUDA_1_3
 - typedefs_npp, 42
- NPP_CUDA_2_0
 - typedefs_npp, 42
- NPP_CUDA_2_1
 - typedefs_npp, 42
- NPP_CUDA_3_0
 - typedefs_npp, 42
- NPP_CUDA_3_2
 - typedefs_npp, 42
- NPP_CUDA_3_5
 - typedefs_npp, 42
- NPP_CUDA_3_7
 - typedefs_npp, 42
- NPP_CUDA_5_0
 - typedefs_npp, 42
- NPP_CUDA_5_2
 - typedefs_npp, 42
- NPP_CUDA_5_3
 - typedefs_npp, 42
- NPP_CUDA_6_0
 - typedefs_npp, 42
- NPP_CUDA_KERNEL_EXECUTION_ERROR

- typedefs_npp, [45](#)
- NPP_CUDA_NOT_CAPABLE
 - typedefs_npp, [41](#)
- NPP_CUDA_UNKNOWN_VERSION
 - typedefs_npp, [41](#)
- NPP_DATA_TYPE_ERROR
 - typedefs_npp, [46](#)
- NPP_DIVIDE_BY_ZERO_ERROR
 - typedefs_npp, [46](#)
- NPP_DIVIDE_BY_ZERO_WARNING
 - typedefs_npp, [47](#)
- NPP_DIVISOR_ERROR
 - typedefs_npp, [46](#)
- NPP_DOUBLE_SIZE_WARNING
 - typedefs_npp, [47](#)
- NPP_ERROR
 - typedefs_npp, [46](#)
- NPP_ERROR_RESERVED
 - typedefs_npp, [46](#)
- NPP_FFT_FLAG_ERROR
 - typedefs_npp, [46](#)
- NPP_FFT_ORDER_ERROR
 - typedefs_npp, [46](#)
- NPP_HAAR_CLASSIFIER_PIXEL_MATCH_-
ERROR
 - typedefs_npp, [45](#)
- NPP_HISTOGRAM_NUMBER_OF_LEVELS_-
ERROR
 - typedefs_npp, [46](#)
- NPP_HORIZONTAL_AXIS
 - typedefs_npp, [43](#)
- NPP_INTERPOLATION_ERROR
 - typedefs_npp, [46](#)
- NPP_INVALID_DEVICE_POINTER_ERROR
 - typedefs_npp, [45](#)
- NPP_INVALID_HOST_POINTER_ERROR
 - typedefs_npp, [45](#)
- NPP_LUT_NUMBER_OF_LEVELS_ERROR
 - typedefs_npp, [46](#)
- NPP_LUT_PALETTE_BITSIZE_ERROR
 - typedefs_npp, [45](#)
- NPP_MASK_SIZE_11_X_11
 - typedefs_npp, [44](#)
- NPP_MASK_SIZE_13_X_13
 - typedefs_npp, [44](#)
- NPP_MASK_SIZE_15_X_15
 - typedefs_npp, [44](#)
- NPP_MASK_SIZE_1_X_3
 - typedefs_npp, [44](#)
- NPP_MASK_SIZE_1_X_5
 - typedefs_npp, [44](#)
- NPP_MASK_SIZE_3_X_1
 - typedefs_npp, [44](#)
- NPP_MASK_SIZE_3_X_3
 - typedefs_npp, [44](#)
- NPP_MASK_SIZE_5_X_1
 - typedefs_npp, [44](#)
- NPP_MASK_SIZE_5_X_5
 - typedefs_npp, [44](#)
- NPP_MASK_SIZE_7_X_7
 - typedefs_npp, [44](#)
- NPP_MASK_SIZE_9_X_9
 - typedefs_npp, [44](#)
- NPP_MASK_SIZE_ERROR
 - typedefs_npp, [46](#)
- NPP_MEMCPY_ERROR
 - typedefs_npp, [45](#)
- NPP_MEMFREE_ERROR
 - typedefs_npp, [45](#)
- NPP_MEMORY_ALLOCATION_ERR
 - typedefs_npp, [46](#)
- NPP_MEMSET_ERROR
 - typedefs_npp, [45](#)
- NPP_MIRROR_FLIP_ERROR
 - typedefs_npp, [46](#)
- NPP_MISALIGNED_DST_ROI_WARNING
 - typedefs_npp, [47](#)
- NPP_MOMENT_00_ZERO_ERROR
 - typedefs_npp, [46](#)
- NPP_NO_ERROR
 - typedefs_npp, [46](#)
- NPP_NO_MEMORY_ERROR
 - typedefs_npp, [46](#)
- NPP_NO_OPERATION_WARNING
 - typedefs_npp, [47](#)
- NPP_NOT_EVEN_STEP_ERROR
 - typedefs_npp, [45](#)
- NPP_NOT_IMPLEMENTED_ERROR
 - typedefs_npp, [46](#)
- NPP_NOT_SUFFICIENT_COMPUTE_-
CAPABILITY
 - typedefs_npp, [45](#)
- NPP_NOT_SUPPORTED_MODE_ERROR
 - typedefs_npp, [45](#)
- NPP_NULL_POINTER_ERROR
 - typedefs_npp, [46](#)
- NPP_NUMBER_OF_CHANNELS_ERROR
 - typedefs_npp, [46](#)
- NPP_OUT_OFF_RANGE_ERROR
 - typedefs_npp, [46](#)
- NPP_OVERFLOW_ERROR
 - typedefs_npp, [45](#)
- NPP_QUADRANGLE_ERROR
 - typedefs_npp, [46](#)
- NPP_QUALITY_INDEX_ERROR
 - typedefs_npp, [45](#)
- NPP_RANGE_ERROR
 - typedefs_npp, [46](#)

- NPP_RECTANGLE_ERROR
 - typedefs_npp, [46](#)
- NPP_RESIZE_FACTOR_ERROR
 - typedefs_npp, [46](#)
- NPP_RESIZE_NO_OPERATION_ERROR
 - typedefs_npp, [45](#)
- NPP_RND_FINANCIAL
 - typedefs_npp, [45](#)
- NPP_RND_NEAR
 - typedefs_npp, [44](#)
- NPP_RND_ZERO
 - typedefs_npp, [45](#)
- NPP_ROUND_MODE_NOT_SUPPORTED_-
ERROR
 - typedefs_npp, [45](#)
- NPP_ROUND_NEAREST_TIES_AWAY_-
FROM_ZERO
 - typedefs_npp, [45](#)
- NPP_ROUND_NEAREST_TIES_TO_EVEN
 - typedefs_npp, [45](#)
- NPP_ROUND_TOWARD_ZERO
 - typedefs_npp, [45](#)
- NPP_SCALE_RANGE_ERROR
 - typedefs_npp, [46](#)
- NPP_SIZE_ERROR
 - typedefs_npp, [46](#)
- NPP_STEP_ERROR
 - typedefs_npp, [46](#)
- NPP_STRIDE_ERROR
 - typedefs_npp, [46](#)
- NPP_SUCCESS
 - typedefs_npp, [46](#)
- NPP_TEXTURE_BIND_ERROR
 - typedefs_npp, [45](#)
- NPP_THRESHOLD_ERROR
 - typedefs_npp, [46](#)
- NPP_THRESHOLD_NEGATIVE_LEVEL_-
ERROR
 - typedefs_npp, [46](#)
- NPP_VERTICAL_AXIS
 - typedefs_npp, [43](#)
- NPP_WRONG_INTERSECTION_QUAD_-
WARNING
 - typedefs_npp, [47](#)
- NPP_WRONG_INTERSECTION_ROI_ERROR
 - typedefs_npp, [45](#)
- NPP_WRONG_INTERSECTION_ROI_-
WARNING
 - typedefs_npp, [47](#)
- NPP_ZC_MODE_NOT_SUPPORTED_ERROR
 - typedefs_npp, [45](#)
- NPP_ZERO_MASK_VALUE_ERROR
 - typedefs_npp, [46](#)
- NPP_ALIGN_16, [2867](#)
 - im, [2867](#)
 - re, [2868](#)
- NPP_ALIGN_8, [2869](#)
 - im, [2869](#)
 - re, [2869](#), [2870](#)
- npp_basic_types
 - __align__, [50](#), [51](#)
 - Npp16s, [49](#)
 - Npp16sc, [51](#)
 - Npp16u, [49](#)
 - Npp16uc, [51](#)
 - Npp32f, [49](#)
 - Npp32fc, [49](#)
 - Npp32s, [49](#)
 - Npp32sc, [49](#)
 - Npp32u, [50](#)
 - Npp32uc, [50](#)
 - Npp64f, [50](#)
 - Npp64fc, [50](#)
 - Npp64s, [50](#)
 - Npp64sc, [50](#)
 - Npp64u, [50](#)
 - Npp8s, [50](#)
 - Npp8u, [50](#)
 - Npp8uc, [51](#)
- NPP_MAX_16S
 - typedefs_npp, [39](#)
- NPP_MAX_16U
 - typedefs_npp, [39](#)
- NPP_MAX_32S
 - typedefs_npp, [39](#)
- NPP_MAX_32U
 - typedefs_npp, [40](#)
- NPP_MAX_64S
 - typedefs_npp, [40](#)
- NPP_MAX_64U
 - typedefs_npp, [40](#)
- NPP_MAX_8S
 - typedefs_npp, [40](#)
- NPP_MAX_8U
 - typedefs_npp, [40](#)
- NPP_MAXABS_32F
 - typedefs_npp, [40](#)
- NPP_MAXABS_64F
 - typedefs_npp, [40](#)
- NPP_MIN_16S
 - typedefs_npp, [40](#)
- NPP_MIN_16U
 - typedefs_npp, [40](#)
- NPP_MIN_32S
 - typedefs_npp, [40](#)
- NPP_MIN_32U
 - typedefs_npp, [40](#)
- NPP_MIN_64S

- typedefs_npp, 41
- NPP_MIN_64U
 - typedefs_npp, 41
- NPP_MIN_8S
 - typedefs_npp, 41
- NPP_MIN_8U
 - typedefs_npp, 41
- NPP_MINABS_32F
 - typedefs_npp, 41
- NPP_MINABS_64F
 - typedefs_npp, 41
- NppCmpOp
 - typedefs_npp, 41
- nppGetGpuComputeCapability
 - core_npp, 32
- nppGetGpuName
 - core_npp, 32
- nppGetGpuNumSMs
 - core_npp, 32
- nppGetLibVersion
 - core_npp, 32
- nppGetMaxThreadsPerBlock
 - core_npp, 32
- nppGetMaxThreadsPerSM
 - core_npp, 32
- nppGetStream
 - core_npp, 33
- NppGpuComputeCapability
 - typedefs_npp, 41
- NppHintAlgorithm
 - typedefs_npp, 42
- NPPI_BAYER_BGGR
 - typedefs_npp, 43
- NPPI_BAYER_GBRG
 - typedefs_npp, 43
- NPPI_BAYER_GRBG
 - typedefs_npp, 43
- NPPI_BAYER_RGGB
 - typedefs_npp, 43
- NPPI_INTER_CUBIC
 - typedefs_npp, 43
- NPPI_INTER_CUBIC2P_B05C03
 - typedefs_npp, 44
- NPPI_INTER_CUBIC2P_BSPLINE
 - typedefs_npp, 44
- NPPI_INTER_CUBIC2P_CATMULLROM
 - typedefs_npp, 44
- NPPI_INTER_LANCZOS
 - typedefs_npp, 44
- NPPI_INTER_LANCZOS3_ADVANCED
 - typedefs_npp, 44
- NPPI_INTER_LINEAR
 - typedefs_npp, 43
- NPPI_INTER_NN
 - typedefs_npp, 43
- NPPI_INTER_SUPER
 - typedefs_npp, 44
- NPPI_INTER_UNDEFINED
 - typedefs_npp, 43
- NPPI_OP_ALPHA_ATOP
 - typedefs_npp, 42
- NPPI_OP_ALPHA_ATOP_PREMUL
 - typedefs_npp, 42
- NPPI_OP_ALPHA_IN
 - typedefs_npp, 42
- NPPI_OP_ALPHA_IN_PREMUL
 - typedefs_npp, 42
- NPPI_OP_ALPHA_OUT
 - typedefs_npp, 42
- NPPI_OP_ALPHA_OUT_PREMUL
 - typedefs_npp, 42
- NPPI_OP_ALPHA_OVER
 - typedefs_npp, 42
- NPPI_OP_ALPHA_OVER_PREMUL
 - typedefs_npp, 42
- NPPI_OP_ALPHA_PLUS
 - typedefs_npp, 42
- NPPI_OP_ALPHA_PLUS_PREMUL
 - typedefs_npp, 42
- NPPI_OP_ALPHA_PREMUL
 - typedefs_npp, 42
- NPPI_OP_ALPHA_XOR
 - typedefs_npp, 42
- NPPI_OP_ALPHA_XOR_PREMUL
 - typedefs_npp, 42
- NPPI_SMOOTH_EDGE
 - typedefs_npp, 44
- nppiAbs_16s_AC4IR
 - image_abs, 322
- nppiAbs_16s_AC4R
 - image_abs, 322
- nppiAbs_16s_C1IR
 - image_abs, 322
- nppiAbs_16s_C1R
 - image_abs, 323
- nppiAbs_16s_C3IR
 - image_abs, 323
- nppiAbs_16s_C3R
 - image_abs, 323
- nppiAbs_16s_C4IR
 - image_abs, 324
- nppiAbs_16s_C4R
 - image_abs, 324
- nppiAbs_32f_AC4IR
 - image_abs, 324
- nppiAbs_32f_AC4R
 - image_abs, 325
- nppiAbs_32f_C1IR

image_abs, [325](#)
nppiAbs_32f_C1R
image_abs, [325](#)
nppiAbs_32f_C3IR
image_abs, [326](#)
nppiAbs_32f_C3R
image_abs, [326](#)
nppiAbs_32f_C4IR
image_abs, [326](#)
nppiAbs_32f_C4R
image_abs, [327](#)
nppiAbsDiff_16u_C1R
image_absdiff, [328](#)
nppiAbsDiff_32f_C1R
image_absdiff, [329](#)
nppiAbsDiff_8u_C1R
image_absdiff, [329](#)
nppiAbsDiff_8u_C3R
image_absdiff, [329](#)
nppiAbsDiff_8u_C4R
image_absdiff, [330](#)
nppiAbsDiffC_16u_C1R
image_absdiffc, [167](#)
nppiAbsDiffC_32f_C1R
image_absdiffc, [167](#)
nppiAbsDiffC_8u_C1R
image_absdiffc, [168](#)
nppiACTable
typedefs_npp, [43](#)
nppiAdd_16s_AC4IRSfs
image_add, [174](#)
nppiAdd_16s_AC4RSfs
image_add, [174](#)
nppiAdd_16s_C1IRSfs
image_add, [175](#)
nppiAdd_16s_C1RSfs
image_add, [175](#)
nppiAdd_16s_C3IRSfs
image_add, [176](#)
nppiAdd_16s_C3RSfs
image_add, [176](#)
nppiAdd_16s_C4IRSfs
image_add, [176](#)
nppiAdd_16s_C4RSfs
image_add, [177](#)
nppiAdd_16sc_AC4IRSfs
image_add, [177](#)
nppiAdd_16sc_AC4RSfs
image_add, [178](#)
nppiAdd_16sc_C1IRSfs
image_add, [178](#)
nppiAdd_16sc_C1RSfs
image_add, [178](#)
nppiAdd_16sc_C3IRSfs
image_add, [179](#)
nppiAdd_16sc_C3RSfs
image_add, [179](#)
nppiAdd_16u_AC4IRSfs
image_add, [180](#)
nppiAdd_16u_AC4RSfs
image_add, [180](#)
nppiAdd_16u_C1IRSfs
image_add, [181](#)
nppiAdd_16u_C1RSfs
image_add, [181](#)
nppiAdd_16u_C3IRSfs
image_add, [181](#)
nppiAdd_16u_C3RSfs
image_add, [182](#)
nppiAdd_16u_C4IRSfs
image_add, [182](#)
nppiAdd_16u_C4RSfs
image_add, [183](#)
nppiAdd_32f_AC4IR
image_add, [183](#)
nppiAdd_32f_AC4R
image_add, [183](#)
nppiAdd_32f_C1IR
image_add, [184](#)
nppiAdd_32f_C1R
image_add, [184](#)
nppiAdd_32f_C3IR
image_add, [185](#)
nppiAdd_32f_C3R
image_add, [185](#)
nppiAdd_32f_C4IR
image_add, [185](#)
nppiAdd_32f_C4R
image_add, [186](#)
nppiAdd_32fc_AC4IR
image_add, [186](#)
nppiAdd_32fc_AC4R
image_add, [186](#)
nppiAdd_32fc_C1IR
image_add, [187](#)
nppiAdd_32fc_C1R
image_add, [187](#)
nppiAdd_32fc_C3IR
image_add, [188](#)
nppiAdd_32fc_C3R
image_add, [188](#)
nppiAdd_32fc_C4IR
image_add, [188](#)
nppiAdd_32fc_C4R
image_add, [189](#)
nppiAdd_32s_C1IRSfs
image_add, [189](#)
nppiAdd_32s_C1R

image_add, [190](#)
 nppiAdd_32s_C1RSfs
 image_add, [190](#)
 nppiAdd_32s_C3IRSfs
 image_add, [190](#)
 nppiAdd_32s_C3RSfs
 image_add, [191](#)
 nppiAdd_32sc_AC4IRSfs
 image_add, [191](#)
 nppiAdd_32sc_AC4RSfs
 image_add, [192](#)
 nppiAdd_32sc_C1IRSfs
 image_add, [192](#)
 nppiAdd_32sc_C1RSfs
 image_add, [192](#)
 nppiAdd_32sc_C3IRSfs
 image_add, [193](#)
 nppiAdd_32sc_C3RSfs
 image_add, [193](#)
 nppiAdd_8u_AC4IRSfs
 image_add, [194](#)
 nppiAdd_8u_AC4RSfs
 image_add, [194](#)
 nppiAdd_8u_C1IRSfs
 image_add, [195](#)
 nppiAdd_8u_C1RSfs
 image_add, [195](#)
 nppiAdd_8u_C3IRSfs
 image_add, [195](#)
 nppiAdd_8u_C3RSfs
 image_add, [196](#)
 nppiAdd_8u_C4IRSfs
 image_add, [196](#)
 nppiAdd_8u_C4RSfs
 image_add, [197](#)
 nppiAddC_16s_AC4IRSfs
 image_addc, [61](#)
 nppiAddC_16s_AC4RSfs
 image_addc, [61](#)
 nppiAddC_16s_C1IRSfs
 image_addc, [61](#)
 nppiAddC_16s_C1RSfs
 image_addc, [62](#)
 nppiAddC_16s_C3IRSfs
 image_addc, [62](#)
 nppiAddC_16s_C3RSfs
 image_addc, [62](#)
 nppiAddC_16s_C4IRSfs
 image_addc, [63](#)
 nppiAddC_16s_C4RSfs
 image_addc, [63](#)
 nppiAddC_16sc_AC4IRSfs
 image_addc, [64](#)
 nppiAddC_16sc_AC4RSfs

 image_addc, [64](#)
 nppiAddC_16sc_C1IRSfs
 image_addc, [64](#)
 nppiAddC_16sc_C1RSfs
 image_addc, [65](#)
 nppiAddC_16sc_C3IRSfs
 image_addc, [65](#)
 nppiAddC_16sc_C3RSfs
 image_addc, [66](#)
 nppiAddC_16u_AC4IRSfs
 image_addc, [66](#)
 nppiAddC_16u_AC4RSfs
 image_addc, [66](#)
 nppiAddC_16u_C1IRSfs
 image_addc, [67](#)
 nppiAddC_16u_C1RSfs
 image_addc, [67](#)
 nppiAddC_16u_C3IRSfs
 image_addc, [68](#)
 nppiAddC_16u_C3RSfs
 image_addc, [68](#)
 nppiAddC_16u_C4IRSfs
 image_addc, [68](#)
 nppiAddC_16u_C4RSfs
 image_addc, [69](#)
 nppiAddC_32f_AC4IR
 image_addc, [69](#)
 nppiAddC_32f_AC4R
 image_addc, [69](#)
 nppiAddC_32f_C1IR
 image_addc, [70](#)
 nppiAddC_32f_C1R
 image_addc, [70](#)
 nppiAddC_32f_C3IR
 image_addc, [70](#)
 nppiAddC_32f_C3R
 image_addc, [71](#)
 nppiAddC_32f_C4IR
 image_addc, [71](#)
 nppiAddC_32f_C4R
 image_addc, [71](#)
 nppiAddC_32fc_AC4IR
 image_addc, [72](#)
 nppiAddC_32fc_AC4R
 image_addc, [72](#)
 nppiAddC_32fc_C1IR
 image_addc, [72](#)
 nppiAddC_32fc_C1R
 image_addc, [73](#)
 nppiAddC_32fc_C3IR
 image_addc, [73](#)
 nppiAddC_32fc_C3R
 image_addc, [73](#)
 nppiAddC_32fc_C4IR

- image_addc, [74](#)
- nppiAddC_32fc_C4R
 - image_addc, [74](#)
- nppiAddC_32s_C1RSfs
 - image_addc, [75](#)
- nppiAddC_32s_C1RSfs
 - image_addc, [75](#)
- nppiAddC_32s_C3RSfs
 - image_addc, [75](#)
- nppiAddC_32s_C3RSfs
 - image_addc, [76](#)
- nppiAddC_32sc_AC4IRSfs
 - image_addc, [76](#)
- nppiAddC_32sc_AC4RSfs
 - image_addc, [76](#)
- nppiAddC_32sc_C1RSfs
 - image_addc, [77](#)
- nppiAddC_32sc_C1RSfs
 - image_addc, [77](#)
- nppiAddC_32sc_C3RSfs
 - image_addc, [78](#)
- nppiAddC_32sc_C3RSfs
 - image_addc, [78](#)
- nppiAddC_8u_AC4IRSfs
 - image_addc, [78](#)
- nppiAddC_8u_AC4RSfs
 - image_addc, [79](#)
- nppiAddC_8u_C1RSfs
 - image_addc, [79](#)
- nppiAddC_8u_C1RSfs
 - image_addc, [80](#)
- nppiAddC_8u_C3RSfs
 - image_addc, [80](#)
- nppiAddC_8u_C3RSfs
 - image_addc, [80](#)
- nppiAddC_8u_C4RSfs
 - image_addc, [81](#)
- nppiAddC_8u_C4RSfs
 - image_addc, [81](#)
- nppiAddProduct_16u32f_C1IMR
 - image_addproduct, [201](#)
- nppiAddProduct_16u32f_C1IR
 - image_addproduct, [202](#)
- nppiAddProduct_32f_C1IMR
 - image_addproduct, [202](#)
- nppiAddProduct_32f_C1IR
 - image_addproduct, [203](#)
- nppiAddProduct_8u32f_C1IMR
 - image_addproduct, [203](#)
- nppiAddProduct_8u32f_C1IR
 - image_addproduct, [203](#)
- nppiAddSquare_16u32f_C1IMR
 - image_addsquare, [198](#)
- nppiAddSquare_16u32f_C1IR
 - image_addsquare, [199](#)
- nppiAddSquare_32f_C1IMR
 - image_addsquare, [199](#)
- nppiAddSquare_32f_C1IR
 - image_addsquare, [199](#)
- nppiAddSquare_8u32f_C1IMR
 - image_addsquare, [200](#)
- nppiAddSquare_8u32f_C1IR
 - image_addsquare, [200](#)
- nppiAddWeighted_16u32f_C1IMR
 - image_addweighted, [205](#)
- nppiAddWeighted_16u32f_C1IR
 - image_addweighted, [206](#)
- nppiAddWeighted_32f_C1IMR
 - image_addweighted, [206](#)
- nppiAddWeighted_32f_C1IR
 - image_addweighted, [207](#)
- nppiAddWeighted_8u32f_C1IMR
 - image_addweighted, [207](#)
- nppiAddWeighted_8u32f_C1IR
 - image_addweighted, [207](#)
- nppiAlphaComp_16s_AC1R
 - image_alphacomp, [490](#)
- nppiAlphaComp_16u_AC1R
 - image_alphacomp, [490](#)
- nppiAlphaComp_16u_AC4R
 - image_alphacomp, [491](#)
- nppiAlphaComp_32f_AC1R
 - image_alphacomp, [491](#)
- nppiAlphaComp_32f_AC4R
 - image_alphacomp, [492](#)
- nppiAlphaComp_32s_AC1R
 - image_alphacomp, [492](#)
- nppiAlphaComp_32s_AC4R
 - image_alphacomp, [492](#)
- nppiAlphaComp_32u_AC1R
 - image_alphacomp, [493](#)
- nppiAlphaComp_32u_AC4R
 - image_alphacomp, [493](#)
- nppiAlphaComp_8s_AC1R
 - image_alphacomp, [494](#)
- nppiAlphaComp_8u_AC1R
 - image_alphacomp, [494](#)
- nppiAlphaComp_8u_AC4R
 - image_alphacomp, [495](#)
- nppiAlphaCompC_16s_C1R
 - image_alphacompc, [475](#)
- nppiAlphaCompC_16u_AC4R
 - image_alphacompc, [475](#)
- nppiAlphaCompC_16u_C1R
 - image_alphacompc, [476](#)
- nppiAlphaCompC_16u_C3R
 - image_alphacompc, [476](#)
- nppiAlphaCompC_16u_C4R

- image_alphacompc, [477](#)
- nppiAlphaCompC_32f_C1R
 - image_alphacompc, [477](#)
- nppiAlphaCompC_32s_C1R
 - image_alphacompc, [478](#)
- nppiAlphaCompC_32u_C1R
 - image_alphacompc, [478](#)
- nppiAlphaCompC_8s_C1R
 - image_alphacompc, [479](#)
- nppiAlphaCompC_8u_AC4R
 - image_alphacompc, [479](#)
- nppiAlphaCompC_8u_C1R
 - image_alphacompc, [480](#)
- nppiAlphaCompC_8u_C3R
 - image_alphacompc, [480](#)
- nppiAlphaCompC_8u_C4R
 - image_alphacompc, [481](#)
- nppiAlphaCompColorKey_8u_AC4R
 - image_complement_color_key, [620](#)
- NppiAlphaOp
 - typedefs_npp, [42](#)
- nppiAlphaPremul_16u_AC4IR
 - image_alphapremul, [496](#)
- nppiAlphaPremul_16u_AC4R
 - image_alphapremul, [496](#)
- nppiAlphaPremul_8u_AC4IR
 - image_alphapremul, [497](#)
- nppiAlphaPremul_8u_AC4R
 - image_alphapremul, [497](#)
- nppiAlphaPremulC_16u_AC4IR
 - image_alphapremulc, [483](#)
- nppiAlphaPremulC_16u_AC4R
 - image_alphapremulc, [483](#)
- nppiAlphaPremulC_16u_C1IR
 - image_alphapremulc, [484](#)
- nppiAlphaPremulC_16u_C1R
 - image_alphapremulc, [484](#)
- nppiAlphaPremulC_16u_C3IR
 - image_alphapremulc, [484](#)
- nppiAlphaPremulC_16u_C3R
 - image_alphapremulc, [485](#)
- nppiAlphaPremulC_16u_C4IR
 - image_alphapremulc, [485](#)
- nppiAlphaPremulC_16u_C4R
 - image_alphapremulc, [485](#)
- nppiAlphaPremulC_8u_AC4IR
 - image_alphapremulc, [486](#)
- nppiAlphaPremulC_8u_AC4R
 - image_alphapremulc, [486](#)
- nppiAlphaPremulC_8u_C1IR
 - image_alphapremulc, [486](#)
- nppiAlphaPremulC_8u_C1R
 - image_alphapremulc, [487](#)
- nppiAlphaPremulC_8u_C3IR
 - image_alphapremulc, [487](#)
- nppiAlphaPremulC_8u_C3R
 - image_alphapremulc, [487](#)
- nppiAlphaPremulC_8u_C4IR
 - image_alphapremulc, [488](#)
- nppiAlphaPremulC_8u_C4R
 - image_alphapremulc, [488](#)
- nppiAnd_16u_AC4IR
 - image_and, [435](#)
- nppiAnd_16u_AC4R
 - image_and, [435](#)
- nppiAnd_16u_C1IR
 - image_and, [435](#)
- nppiAnd_16u_C1R
 - image_and, [436](#)
- nppiAnd_16u_C3IR
 - image_and, [436](#)
- nppiAnd_16u_C3R
 - image_and, [436](#)
- nppiAnd_16u_C4IR
 - image_and, [437](#)
- nppiAnd_16u_C4R
 - image_and, [437](#)
- nppiAnd_32s_AC4IR
 - image_and, [438](#)
- nppiAnd_32s_AC4R
 - image_and, [438](#)
- nppiAnd_32s_C1IR
 - image_and, [438](#)
- nppiAnd_32s_C1R
 - image_and, [439](#)
- nppiAnd_32s_C3IR
 - image_and, [439](#)
- nppiAnd_32s_C3R
 - image_and, [439](#)
- nppiAnd_32s_C4IR
 - image_and, [440](#)
- nppiAnd_32s_C4R
 - image_and, [440](#)
- nppiAnd_8u_AC4IR
 - image_and, [441](#)
- nppiAnd_8u_AC4R
 - image_and, [441](#)
- nppiAnd_8u_C1IR
 - image_and, [441](#)
- nppiAnd_8u_C1R
 - image_and, [442](#)
- nppiAnd_8u_C3IR
 - image_and, [442](#)
- nppiAnd_8u_C3R
 - image_and, [442](#)
- nppiAnd_8u_C4IR
 - image_and, [443](#)
- nppiAnd_8u_C4R

- image_and, [443](#)
- npPiAndC_16u_AC4IR
 - image_andc, [374](#)
- npPiAndC_16u_AC4R
 - image_andc, [374](#)
- npPiAndC_16u_C1IR
 - image_andc, [374](#)
- npPiAndC_16u_C1R
 - image_andc, [375](#)
- npPiAndC_16u_C3IR
 - image_andc, [375](#)
- npPiAndC_16u_C3R
 - image_andc, [375](#)
- npPiAndC_16u_C4IR
 - image_andc, [376](#)
- npPiAndC_16u_C4R
 - image_andc, [376](#)
- npPiAndC_32s_AC4IR
 - image_andc, [376](#)
- npPiAndC_32s_AC4R
 - image_andc, [377](#)
- npPiAndC_32s_C1IR
 - image_andc, [377](#)
- npPiAndC_32s_C1R
 - image_andc, [377](#)
- npPiAndC_32s_C3IR
 - image_andc, [378](#)
- npPiAndC_32s_C3R
 - image_andc, [378](#)
- npPiAndC_32s_C4IR
 - image_andc, [378](#)
- npPiAndC_32s_C4R
 - image_andc, [379](#)
- npPiAndC_8u_AC4IR
 - image_andc, [379](#)
- npPiAndC_8u_AC4R
 - image_andc, [379](#)
- npPiAndC_8u_C1IR
 - image_andc, [380](#)
- npPiAndC_8u_C1R
 - image_andc, [380](#)
- npPiAndC_8u_C3IR
 - image_andc, [380](#)
- npPiAndC_8u_C3R
 - image_andc, [381](#)
- npPiAndC_8u_C4IR
 - image_andc, [381](#)
- npPiAndC_8u_C4R
 - image_andc, [381](#)
- npPiAverageError_16s_C1R
 - image_average_error, [2292](#)
- npPiAverageError_16s_C2R
 - image_average_error, [2293](#)
- npPiAverageError_16s_C3R
 - image_average_error, [2293](#)
- npPiAverageError_16s_C4R
 - image_average_error, [2294](#)
- npPiAverageError_16sc_C1R
 - image_average_error, [2294](#)
- npPiAverageError_16sc_C2R
 - image_average_error, [2294](#)
- npPiAverageError_16sc_C3R
 - image_average_error, [2295](#)
- npPiAverageError_16sc_C4R
 - image_average_error, [2295](#)
- npPiAverageError_16u_C1R
 - image_average_error, [2296](#)
- npPiAverageError_16u_C2R
 - image_average_error, [2296](#)
- npPiAverageError_16u_C3R
 - image_average_error, [2297](#)
- npPiAverageError_16u_C4R
 - image_average_error, [2297](#)
- npPiAverageError_32f_C1R
 - image_average_error, [2297](#)
- npPiAverageError_32f_C2R
 - image_average_error, [2298](#)
- npPiAverageError_32f_C3R
 - image_average_error, [2298](#)
- npPiAverageError_32f_C4R
 - image_average_error, [2299](#)
- npPiAverageError_32fc_C1R
 - image_average_error, [2299](#)
- npPiAverageError_32fc_C2R
 - image_average_error, [2300](#)
- npPiAverageError_32fc_C3R
 - image_average_error, [2300](#)
- npPiAverageError_32fc_C4R
 - image_average_error, [2301](#)
- npPiAverageError_32s_C1R
 - image_average_error, [2301](#)
- npPiAverageError_32s_C2R
 - image_average_error, [2301](#)
- npPiAverageError_32s_C3R
 - image_average_error, [2302](#)
- npPiAverageError_32s_C4R
 - image_average_error, [2302](#)
- npPiAverageError_32sc_C1R
 - image_average_error, [2303](#)
- npPiAverageError_32sc_C2R
 - image_average_error, [2303](#)
- npPiAverageError_32sc_C3R
 - image_average_error, [2304](#)
- npPiAverageError_32sc_C4R
 - image_average_error, [2304](#)
- npPiAverageError_32u_C1R
 - image_average_error, [2304](#)
- npPiAverageError_32u_C2R

- image_average_error, [2305](#)
- nppiAverageError_32u_C3R
 - image_average_error, [2305](#)
- nppiAverageError_32u_C4R
 - image_average_error, [2306](#)
- nppiAverageError_64f_C1R
 - image_average_error, [2306](#)
- nppiAverageError_64f_C2R
 - image_average_error, [2307](#)
- nppiAverageError_64f_C3R
 - image_average_error, [2307](#)
- nppiAverageError_64f_C4R
 - image_average_error, [2308](#)
- nppiAverageError_8s_C1R
 - image_average_error, [2308](#)
- nppiAverageError_8s_C2R
 - image_average_error, [2308](#)
- nppiAverageError_8s_C3R
 - image_average_error, [2309](#)
- nppiAverageError_8s_C4R
 - image_average_error, [2309](#)
- nppiAverageError_8u_C1R
 - image_average_error, [2310](#)
- nppiAverageError_8u_C2R
 - image_average_error, [2310](#)
- nppiAverageError_8u_C3R
 - image_average_error, [2311](#)
- nppiAverageError_8u_C4R
 - image_average_error, [2311](#)
- nppiAverageErrorGetBufferHostSize_16s_C1R
 - image_statistics_functions, [1651](#)
- nppiAverageErrorGetBufferHostSize_16s_C2R
 - image_statistics_functions, [1651](#)
- nppiAverageErrorGetBufferHostSize_16s_C3R
 - image_statistics_functions, [1651](#)
- nppiAverageErrorGetBufferHostSize_16s_C4R
 - image_statistics_functions, [1651](#)
- nppiAverageErrorGetBufferHostSize_16sc_C1R
 - image_statistics_functions, [1652](#)
- nppiAverageErrorGetBufferHostSize_16sc_C2R
 - image_statistics_functions, [1652](#)
- nppiAverageErrorGetBufferHostSize_16sc_C3R
 - image_statistics_functions, [1652](#)
- nppiAverageErrorGetBufferHostSize_16sc_C4R
 - image_statistics_functions, [1653](#)
- nppiAverageErrorGetBufferHostSize_16u_C1R
 - image_statistics_functions, [1653](#)
- nppiAverageErrorGetBufferHostSize_16u_C2R
 - image_statistics_functions, [1653](#)
- nppiAverageErrorGetBufferHostSize_16u_C3R
 - image_statistics_functions, [1653](#)
- nppiAverageErrorGetBufferHostSize_16u_C4R
 - image_statistics_functions, [1654](#)
- nppiAverageErrorGetBufferHostSize_32f_C1R
 - image_statistics_functions, [1654](#)
- nppiAverageErrorGetBufferHostSize_32f_C2R
 - image_statistics_functions, [1654](#)
- nppiAverageErrorGetBufferHostSize_32f_C3R
 - image_statistics_functions, [1655](#)
- nppiAverageErrorGetBufferHostSize_32f_C4R
 - image_statistics_functions, [1655](#)
- nppiAverageErrorGetBufferHostSize_32fc_C1R
 - image_statistics_functions, [1655](#)
- nppiAverageErrorGetBufferHostSize_32fc_C2R
 - image_statistics_functions, [1655](#)
- nppiAverageErrorGetBufferHostSize_32fc_C3R
 - image_statistics_functions, [1656](#)
- nppiAverageErrorGetBufferHostSize_32fc_C4R
 - image_statistics_functions, [1656](#)
- nppiAverageErrorGetBufferHostSize_32s_C1R
 - image_statistics_functions, [1656](#)
- nppiAverageErrorGetBufferHostSize_32s_C2R
 - image_statistics_functions, [1657](#)
- nppiAverageErrorGetBufferHostSize_32s_C3R
 - image_statistics_functions, [1657](#)
- nppiAverageErrorGetBufferHostSize_32s_C4R
 - image_statistics_functions, [1657](#)
- nppiAverageErrorGetBufferHostSize_32sc_C1R
 - image_statistics_functions, [1657](#)
- nppiAverageErrorGetBufferHostSize_32sc_C2R
 - image_statistics_functions, [1658](#)
- nppiAverageErrorGetBufferHostSize_32sc_C3R
 - image_statistics_functions, [1658](#)
- nppiAverageErrorGetBufferHostSize_32sc_C4R
 - image_statistics_functions, [1658](#)
- nppiAverageErrorGetBufferHostSize_32u_C1R
 - image_statistics_functions, [1659](#)
- nppiAverageErrorGetBufferHostSize_32u_C2R
 - image_statistics_functions, [1659](#)
- nppiAverageErrorGetBufferHostSize_32u_C3R
 - image_statistics_functions, [1659](#)
- nppiAverageErrorGetBufferHostSize_32u_C4R
 - image_statistics_functions, [1659](#)
- nppiAverageErrorGetBufferHostSize_64f_C1R
 - image_statistics_functions, [1660](#)
- nppiAverageErrorGetBufferHostSize_64f_C2R
 - image_statistics_functions, [1660](#)
- nppiAverageErrorGetBufferHostSize_64f_C3R
 - image_statistics_functions, [1660](#)
- nppiAverageErrorGetBufferHostSize_64f_C4R
 - image_statistics_functions, [1661](#)
- nppiAverageErrorGetBufferHostSize_8s_C1R
 - image_statistics_functions, [1661](#)
- nppiAverageErrorGetBufferHostSize_8s_C2R
 - image_statistics_functions, [1661](#)
- nppiAverageErrorGetBufferHostSize_8s_C3R
 - image_statistics_functions, [1661](#)
- nppiAverageErrorGetBufferHostSize_8s_C4R
 - image_statistics_functions, [1661](#)

- image_statistics_functions, [1662](#)
- nppiAverageErrorGetBufferHostSize_8u_C1R
 - image_statistics_functions, [1662](#)
- nppiAverageErrorGetBufferHostSize_8u_C2R
 - image_statistics_functions, [1662](#)
- nppiAverageErrorGetBufferHostSize_8u_C3R
 - image_statistics_functions, [1663](#)
- nppiAverageErrorGetBufferHostSize_8u_C4R
 - image_statistics_functions, [1663](#)
- nppiAverageRelativeError_16s_C1R
 - image_average_relative_error, [2339](#)
- nppiAverageRelativeError_16s_C2R
 - image_average_relative_error, [2340](#)
- nppiAverageRelativeError_16s_C3R
 - image_average_relative_error, [2340](#)
- nppiAverageRelativeError_16s_C4R
 - image_average_relative_error, [2341](#)
- nppiAverageRelativeError_16sc_C1R
 - image_average_relative_error, [2341](#)
- nppiAverageRelativeError_16sc_C2R
 - image_average_relative_error, [2342](#)
- nppiAverageRelativeError_16sc_C3R
 - image_average_relative_error, [2342](#)
- nppiAverageRelativeError_16sc_C4R
 - image_average_relative_error, [2342](#)
- nppiAverageRelativeError_16u_C1R
 - image_average_relative_error, [2343](#)
- nppiAverageRelativeError_16u_C2R
 - image_average_relative_error, [2343](#)
- nppiAverageRelativeError_16u_C3R
 - image_average_relative_error, [2344](#)
- nppiAverageRelativeError_16u_C4R
 - image_average_relative_error, [2344](#)
- nppiAverageRelativeError_32f_C1R
 - image_average_relative_error, [2345](#)
- nppiAverageRelativeError_32f_C2R
 - image_average_relative_error, [2345](#)
- nppiAverageRelativeError_32f_C3R
 - image_average_relative_error, [2346](#)
- nppiAverageRelativeError_32f_C4R
 - image_average_relative_error, [2346](#)
- nppiAverageRelativeError_32fc_C1R
 - image_average_relative_error, [2347](#)
- nppiAverageRelativeError_32fc_C2R
 - image_average_relative_error, [2347](#)
- nppiAverageRelativeError_32fc_C3R
 - image_average_relative_error, [2347](#)
- nppiAverageRelativeError_32fc_C4R
 - image_average_relative_error, [2348](#)
- nppiAverageRelativeError_32s_C1R
 - image_average_relative_error, [2348](#)
- nppiAverageRelativeError_32s_C2R
 - image_average_relative_error, [2349](#)
- nppiAverageRelativeError_32s_C3R
 - image_average_relative_error, [2349](#)
- nppiAverageRelativeError_32s_C4R
 - image_average_relative_error, [2349](#)
- nppiAverageRelativeError_32sc_C1R
 - image_average_relative_error, [2350](#)
- nppiAverageRelativeError_32sc_C2R
 - image_average_relative_error, [2350](#)
- nppiAverageRelativeError_32sc_C3R
 - image_average_relative_error, [2351](#)
- nppiAverageRelativeError_32sc_C4R
 - image_average_relative_error, [2351](#)
- nppiAverageRelativeError_32u_C1R
 - image_average_relative_error, [2352](#)
- nppiAverageRelativeError_32u_C2R
 - image_average_relative_error, [2352](#)
- nppiAverageRelativeError_32u_C3R
 - image_average_relative_error, [2352](#)
- nppiAverageRelativeError_32u_C4R
 - image_average_relative_error, [2353](#)
- nppiAverageRelativeError_64f_C1R
 - image_average_relative_error, [2354](#)
- nppiAverageRelativeError_64f_C2R
 - image_average_relative_error, [2354](#)
- nppiAverageRelativeError_64f_C3R
 - image_average_relative_error, [2355](#)
- nppiAverageRelativeError_64f_C4R
 - image_average_relative_error, [2355](#)
- nppiAverageRelativeError_8s_C1R
 - image_average_relative_error, [2356](#)
- nppiAverageRelativeError_8s_C2R
 - image_average_relative_error, [2356](#)
- nppiAverageRelativeError_8s_C3R
 - image_average_relative_error, [2357](#)
- nppiAverageRelativeError_8s_C4R
 - image_average_relative_error, [2357](#)
- nppiAverageRelativeError_8u_C1R
 - image_average_relative_error, [2357](#)
- nppiAverageRelativeError_8u_C2R
 - image_average_relative_error, [2358](#)
- nppiAverageRelativeError_8u_C3R
 - image_average_relative_error, [2358](#)
- nppiAverageRelativeError_8u_C4R
 - image_average_relative_error, [2359](#)
- nppiAverageRelativeErrorGetBufferHostSize_-16s_C1R
 - image_statistics_functions, [1663](#)
- nppiAverageRelativeErrorGetBufferHostSize_-16s_C2R
 - image_statistics_functions, [1663](#)
- nppiAverageRelativeErrorGetBufferHostSize_-16s_C3R
 - image_statistics_functions, [1664](#)
- nppiAverageRelativeErrorGetBufferHostSize_-16s_C4R

image_statistics_functions, [1664](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 16sc_C1R
 image_statistics_functions, [1664](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 16sc_C2R
 image_statistics_functions, [1665](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 16sc_C3R
 image_statistics_functions, [1665](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 16sc_C4R
 image_statistics_functions, [1665](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 16u_C1R
 image_statistics_functions, [1665](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 16u_C2R
 image_statistics_functions, [1666](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 16u_C3R
 image_statistics_functions, [1666](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 16u_C4R
 image_statistics_functions, [1666](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32f_C1R
 image_statistics_functions, [1667](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32f_C2R
 image_statistics_functions, [1667](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32f_C3R
 image_statistics_functions, [1667](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32f_C4R
 image_statistics_functions, [1667](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32fc_C1R
 image_statistics_functions, [1668](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32fc_C2R
 image_statistics_functions, [1668](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32fc_C3R
 image_statistics_functions, [1668](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32fc_C4R
 image_statistics_functions, [1669](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32s_C1R
 image_statistics_functions, [1669](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32s_C2R

image_statistics_functions, [1669](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32s_C3R
 image_statistics_functions, [1669](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32s_C4R
 image_statistics_functions, [1670](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32sc_C1R
 image_statistics_functions, [1670](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32sc_C2R
 image_statistics_functions, [1670](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32sc_C3R
 image_statistics_functions, [1671](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32sc_C4R
 image_statistics_functions, [1671](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32u_C1R
 image_statistics_functions, [1671](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32u_C2R
 image_statistics_functions, [1671](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32u_C3R
 image_statistics_functions, [1672](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 32u_C4R
 image_statistics_functions, [1672](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 64f_C1R
 image_statistics_functions, [1672](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 64f_C2R
 image_statistics_functions, [1673](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 64f_C3R
 image_statistics_functions, [1673](#)
 nppiAverageRelativeErrorGetBufferHostSize_-
 64f_C4R
 image_statistics_functions, [1673](#)
 nppiAverageRelativeErrorGetBufferHostSize_8s_-
 C1R
 image_statistics_functions, [1673](#)
 nppiAverageRelativeErrorGetBufferHostSize_8s_-
 C2R
 image_statistics_functions, [1674](#)
 nppiAverageRelativeErrorGetBufferHostSize_8s_-
 C3R
 image_statistics_functions, [1674](#)
 nppiAverageRelativeErrorGetBufferHostSize_8s_-
 C4R

- image_statistics_functions, [1674](#)
- nppiAverageRelativeErrorGetBufferHostSize_8u_C1R
 - image_statistics_functions, [1675](#)
- nppiAverageRelativeErrorGetBufferHostSize_8u_C2R
 - image_statistics_functions, [1675](#)
- nppiAverageRelativeErrorGetBufferHostSize_8u_C3R
 - image_statistics_functions, [1675](#)
- nppiAverageRelativeErrorGetBufferHostSize_8u_C4R
 - image_statistics_functions, [1675](#)
- NppiAxis
 - typedefs_npp, [42](#)
- NppiBayerGridPosition
 - typedefs_npp, [43](#)
- nppiBGRToCbYCr422_709HDTV_8u_AC4C2R
 - image_color_model_conversion, [527](#)
- nppiBGRToCbYCr422_709HDTV_8u_C3C2R
 - image_color_model_conversion, [528](#)
- nppiBGRToCbYCr422_8u_AC4C2R
 - image_color_model_conversion, [528](#)
- nppiBGRToHLS_8u_AC4P4R
 - image_color_model_conversion, [528](#)
- nppiBGRToHLS_8u_AC4R
 - image_color_model_conversion, [529](#)
- nppiBGRToHLS_8u_AP4C4R
 - image_color_model_conversion, [529](#)
- nppiBGRToHLS_8u_AP4R
 - image_color_model_conversion, [529](#)
- nppiBGRToHLS_8u_C3P3R
 - image_color_model_conversion, [530](#)
- nppiBGRToHLS_8u_P3C3R
 - image_color_model_conversion, [530](#)
- nppiBGRToHLS_8u_P3R
 - image_color_model_conversion, [530](#)
- nppiBGRToLab_8u_C3R
 - image_color_model_conversion, [531](#)
- nppiBGRToYCbCr411_8u_AC4P3R
 - image_color_model_conversion, [531](#)
- nppiBGRToYCbCr411_8u_C3P3R
 - image_color_model_conversion, [531](#)
- nppiBGRToYCbCr420_709CSC_8u_AC4P3R
 - image_color_model_conversion, [532](#)
- nppiBGRToYCbCr420_709CSC_8u_C3P3R
 - image_color_model_conversion, [532](#)
- nppiBGRToYCbCr420_709HDTV_8u_AC4P3R
 - image_color_model_conversion, [533](#)
- nppiBGRToYCbCr420_8u_AC4P3R
 - image_color_model_conversion, [533](#)
- nppiBGRToYCbCr420_8u_C3P3R
 - image_color_model_conversion, [533](#)
- nppiBGRToYCbCr422_8u_AC4C2R
 - image_color_model_conversion, [534](#)
- nppiBGRToYCbCr422_8u_AC4P3R
 - image_color_model_conversion, [534](#)
- nppiBGRToYCbCr422_8u_C3C2R
 - image_color_model_conversion, [535](#)
- nppiBGRToYCbCr422_8u_C3P3R
 - image_color_model_conversion, [535](#)
- nppiBGRToYCbCr_8u_AC4P3R
 - image_color_model_conversion, [535](#)
- nppiBGRToYCbCr_8u_AC4P4R
 - image_color_model_conversion, [536](#)
- nppiBGRToYCbCr_8u_C3P3R
 - image_color_model_conversion, [536](#)
- nppiBGRToYCrCb420_709CSC_8u_AC4P3R
 - image_color_model_conversion, [537](#)
- nppiBGRToYCrCb420_709CSC_8u_C3P3R
 - image_color_model_conversion, [537](#)
- nppiBGRToYCrCb420_8u_AC4P3R
 - image_color_model_conversion, [537](#)
- nppiBGRToYCrCb420_8u_C3P3R
 - image_color_model_conversion, [538](#)
- nppiBGRToYUV420_8u_AC4P3R
 - image_color_model_conversion, [538](#)
- nppiBGRToYUV_8u_AC4P4R
 - image_color_model_conversion, [539](#)
- nppiBGRToYUV_8u_AC4R
 - image_color_model_conversion, [539](#)
- nppiBGRToYUV_8u_C3P3R
 - image_color_model_conversion, [539](#)
- nppiBGRToYUV_8u_C3R
 - image_color_model_conversion, [540](#)
- nppiBGRToYUV_8u_P3R
 - image_color_model_conversion, [540](#)
- NppiBorderType
 - typedefs_npp, [43](#)
- nppiCbYCr422ToBGR_709HDTV_8u_C2C3R
 - image_color_model_conversion, [540](#)
- nppiCbYCr422ToBGR_709HDTV_8u_C2C4R
 - image_color_model_conversion, [541](#)
- nppiCbYCr422ToBGR_8u_C2C4R
 - image_color_model_conversion, [541](#)
- nppiCbYCr422ToRGB_8u_C2C3R
 - image_color_model_conversion, [542](#)
- nppiCbYCr422ToYCbCr411_8u_C2P3R
 - image_color_sampling_format_conversion, [593](#)
- nppiCbYCr422ToYCbCr420_8u_C2P2R
 - image_color_sampling_format_conversion, [594](#)
- nppiCbYCr422ToYCbCr420_8u_C2P3R
 - image_color_sampling_format_conversion, [594](#)
- nppiCbYCr422ToYCbCr422_8u_C2P3R

- image_color_sampling_format_conversion, 594
- npPiCbYCr422ToYCbCr422_8u_C2R
 - image_color_sampling_format_conversion, 595
- npPiCbYCr422ToYCrCb420_8u_C2P3R
 - image_color_sampling_format_conversion, 595
- npPiCFAToRGB_16u_C1C3R
 - image_color_model_conversion, 542
- npPiCFAToRGB_8u_C1C3R
 - image_color_model_conversion, 542
- npPiCFAToRGBA_16u_C1AC4R
 - image_color_model_conversion, 543
- npPiCFAToRGBA_8u_C1AC4R
 - image_color_model_conversion, 543
- npPiColorToGray_16s_AC4C1R
 - image_color_model_conversion, 544
- npPiColorToGray_16s_C3C1R
 - image_color_model_conversion, 544
- npPiColorToGray_16s_C4C1R
 - image_color_model_conversion, 545
- npPiColorToGray_16u_AC4C1R
 - image_color_model_conversion, 545
- npPiColorToGray_16u_C3C1R
 - image_color_model_conversion, 545
- npPiColorToGray_16u_C4C1R
 - image_color_model_conversion, 546
- npPiColorToGray_32f_AC4C1R
 - image_color_model_conversion, 546
- npPiColorToGray_32f_C3C1R
 - image_color_model_conversion, 547
- npPiColorToGray_32f_C4C1R
 - image_color_model_conversion, 547
- npPiColorToGray_8u_AC4C1R
 - image_color_model_conversion, 547
- npPiColorToGray_8u_C3C1R
 - image_color_model_conversion, 548
- npPiColorToGray_8u_C4C1R
 - image_model_conversion, 548
- npPiColorTwist32f_16s_AC4IR
 - image_color_processing, 637
- npPiColorTwist32f_16s_AC4R
 - image_color_processing, 638
- npPiColorTwist32f_16s_C1IR
 - image_color_processing, 638
- npPiColorTwist32f_16s_C1R
 - image_color_processing, 638
- npPiColorTwist32f_16s_C2IR
 - image_color_processing, 639
- npPiColorTwist32f_16s_C2R
 - image_color_processing, 639
- npPiColorTwist32f_16s_C3IR
 - image_color_processing, 640
- npPiColorTwist32f_16s_C3R
 - image_color_processing, 640
- npPiColorTwist32f_16s_IP3R
 - image_color_processing, 640
- npPiColorTwist32f_16s_P3R
 - image_color_processing, 641
- npPiColorTwist32f_16u_AC4IR
 - image_color_processing, 641
- npPiColorTwist32f_16u_AC4R
 - image_color_processing, 642
- npPiColorTwist32f_16u_C1IR
 - image_color_processing, 642
- npPiColorTwist32f_16u_C1R
 - image_color_processing, 642
- npPiColorTwist32f_16u_C2IR
 - image_color_processing, 643
- npPiColorTwist32f_16u_C2R
 - image_color_processing, 643
- npPiColorTwist32f_16u_C3IR
 - image_color_processing, 643
- npPiColorTwist32f_16u_C3R
 - image_color_processing, 644
- npPiColorTwist32f_16u_IP3R
 - image_color_processing, 644
- npPiColorTwist32f_16u_P3R
 - image_color_processing, 644
- npPiColorTwist32f_8s_AC4IR
 - image_color_processing, 645
- npPiColorTwist32f_8s_AC4R
 - image_color_processing, 645
- npPiColorTwist32f_8s_C1IR
 - image_color_processing, 646
- npPiColorTwist32f_8s_C1R
 - image_color_processing, 646
- npPiColorTwist32f_8s_C2IR
 - image_color_processing, 646
- npPiColorTwist32f_8s_C2R
 - image_color_processing, 647
- npPiColorTwist32f_8s_C3IR
 - image_color_processing, 647
- npPiColorTwist32f_8s_C3R
 - image_color_processing, 647
- npPiColorTwist32f_8s_C4IR
 - image_color_processing, 648
- npPiColorTwist32f_8s_C4R
 - image_color_processing, 648
- npPiColorTwist32f_8s_IP3R
 - image_color_processing, 649
- npPiColorTwist32f_8s_P3R
 - image_color_processing, 649
- npPiColorTwist32f_8u_AC4IR
 - image_color_processing, 649
- npPiColorTwist32f_8u_AC4R
 - image_color_processing, 650

- nppiColorTwist32f_8u_C1IR
 - image_color_processing, [650](#)
- nppiColorTwist32f_8u_C1R
 - image_color_processing, [651](#)
- nppiColorTwist32f_8u_C2IR
 - image_color_processing, [651](#)
- nppiColorTwist32f_8u_C2R
 - image_color_processing, [651](#)
- nppiColorTwist32f_8u_C3IR
 - image_color_processing, [652](#)
- nppiColorTwist32f_8u_C3R
 - image_color_processing, [652](#)
- nppiColorTwist32f_8u_C4IR
 - image_color_processing, [653](#)
- nppiColorTwist32f_8u_C4R
 - image_color_processing, [653](#)
- nppiColorTwist32f_8u_IP3R
 - image_color_processing, [653](#)
- nppiColorTwist32f_8u_P3R
 - image_color_processing, [654](#)
- nppiColorTwist32fC_8u_C4IR
 - image_color_processing, [654](#)
- nppiColorTwist32fC_8u_C4R
 - image_color_processing, [655](#)
- nppiColorTwist_32f_AC4IR
 - image_color_processing, [655](#)
- nppiColorTwist_32f_AC4R
 - image_color_processing, [656](#)
- nppiColorTwist_32f_C1IR
 - image_color_processing, [656](#)
- nppiColorTwist_32f_C1R
 - image_color_processing, [656](#)
- nppiColorTwist_32f_C2IR
 - image_color_processing, [657](#)
- nppiColorTwist_32f_C2R
 - image_color_processing, [657](#)
- nppiColorTwist_32f_C3IR
 - image_color_processing, [658](#)
- nppiColorTwist_32f_C3R
 - image_color_processing, [658](#)
- nppiColorTwist_32f_C4IR
 - image_color_processing, [658](#)
- nppiColorTwist_32f_C4R
 - image_color_processing, [659](#)
- nppiColorTwist_32f_IP3R
 - image_color_processing, [659](#)
- nppiColorTwist_32f_P3R
 - image_color_processing, [660](#)
- nppiColorTwist_32fC_C4IR
 - image_color_processing, [660](#)
- nppiColorTwist_32fC_C4R
 - image_color_processing, [660](#)
- nppiCompare_16s_AC4R
 - image_compare_operations, [2465](#)
- nppiCompare_16s_C1R
 - image_compare_operations, [2466](#)
- nppiCompare_16s_C3R
 - image_compare_operations, [2466](#)
- nppiCompare_16s_C4R
 - image_compare_operations, [2466](#)
- nppiCompare_16u_AC4R
 - image_compare_operations, [2467](#)
- nppiCompare_16u_C1R
 - image_compare_operations, [2467](#)
- nppiCompare_16u_C3R
 - image_compare_operations, [2468](#)
- nppiCompare_16u_C4R
 - image_compare_operations, [2468](#)
- nppiCompare_32f_AC4R
 - image_compare_operations, [2469](#)
- nppiCompare_32f_C1R
 - image_compare_operations, [2469](#)
- nppiCompare_32f_C3R
 - image_compare_operations, [2470](#)
- nppiCompare_32f_C4R
 - image_compare_operations, [2470](#)
- nppiCompare_8u_AC4R
 - image_compare_operations, [2471](#)
- nppiCompare_8u_C1R
 - image_compare_operations, [2471](#)
- nppiCompare_8u_C3R
 - image_compare_operations, [2472](#)
- nppiCompare_8u_C4R
 - image_compare_operations, [2472](#)
- nppiCompareC_16s_AC4R
 - image_compare_operations, [2473](#)
- nppiCompareC_16s_C1R
 - image_compare_operations, [2473](#)
- nppiCompareC_16s_C3R
 - image_compare_operations, [2474](#)
- nppiCompareC_16s_C4R
 - image_compare_operations, [2474](#)
- nppiCompareC_16u_AC4R
 - image_compare_operations, [2475](#)
- nppiCompareC_16u_C1R
 - image_compare_operations, [2475](#)
- nppiCompareC_16u_C3R
 - image_compare_operations, [2475](#)
- nppiCompareC_16u_C4R
 - image_compare_operations, [2476](#)
- nppiCompareC_32f_AC4R
 - image_compare_operations, [2476](#)
- nppiCompareC_32f_C1R
 - image_compare_operations, [2477](#)
- nppiCompareC_32f_C3R
 - image_compare_operations, [2477](#)
- nppiCompareC_32f_C4R
 - image_compare_operations, [2478](#)

- nppiCompareC_8u_AC4R
 - image_compare_operations, [2478](#)
- nppiCompareC_8u_C1R
 - image_compare_operations, [2478](#)
- nppiCompareC_8u_C3R
 - image_compare_operations, [2479](#)
- nppiCompareC_8u_C4R
 - image_compare_operations, [2479](#)
- nppiCompareEqualEps_32f_AC4R
 - image_compare_operations, [2480](#)
- nppiCompareEqualEps_32f_C1R
 - image_compare_operations, [2480](#)
- nppiCompareEqualEps_32f_C3R
 - image_compare_operations, [2481](#)
- nppiCompareEqualEps_32f_C4R
 - image_compare_operations, [2481](#)
- nppiCompareEqualEpsC_32f_AC4R
 - image_compare_operations, [2482](#)
- nppiCompareEqualEpsC_32f_C1R
 - image_compare_operations, [2482](#)
- nppiCompareEqualEpsC_32f_C3R
 - image_compare_operations, [2483](#)
- nppiCompareEqualEpsC_32f_C4R
 - image_compare_operations, [2483](#)
- nppiCompColorKey_8u_C1R
 - image_complement_color_key, [621](#)
- nppiCompColorKey_8u_C3R
 - image_complement_color_key, [621](#)
- nppiCompColorKey_8u_C4R
 - image_complement_color_key, [622](#)
- nppiConvert_16s16u_C1Rs
 - image_convert, [828](#)
- nppiConvert_16s32f_AC4R
 - image_convert, [828](#)
- nppiConvert_16s32f_C1R
 - image_convert, [829](#)
- nppiConvert_16s32f_C3R
 - image_convert, [829](#)
- nppiConvert_16s32f_C4R
 - image_convert, [829](#)
- nppiConvert_16s32s_AC4R
 - image_convert, [830](#)
- nppiConvert_16s32s_C1R
 - image_convert, [830](#)
- nppiConvert_16s32s_C3R
 - image_convert, [830](#)
- nppiConvert_16s32s_C4R
 - image_convert, [831](#)
- nppiConvert_16s32u_C1Rs
 - image_convert, [831](#)
- nppiConvert_16s8s_C1RSfs
 - image_convert, [831](#)
- nppiConvert_16s8u_AC4R
 - image_convert, [832](#)
- nppiConvert_16s8u_C1R
 - image_convert, [832](#)
- nppiConvert_16s8u_C3R
 - image_convert, [832](#)
- nppiConvert_16s8u_C4R
 - image_convert, [833](#)
- nppiConvert_16u16s_C1RSfs
 - image_convert, [833](#)
- nppiConvert_16u32f_AC4R
 - image_convert, [833](#)
- nppiConvert_16u32f_C1R
 - image_convert, [834](#)
- nppiConvert_16u32f_C3R
 - image_convert, [834](#)
- nppiConvert_16u32f_C4R
 - image_convert, [834](#)
- nppiConvert_16u32s_AC4R
 - image_convert, [835](#)
- nppiConvert_16u32s_C1R
 - image_convert, [835](#)
- nppiConvert_16u32s_C3R
 - image_convert, [835](#)
- nppiConvert_16u32s_C4R
 - image_convert, [836](#)
- nppiConvert_16u32u_C1R
 - image_convert, [836](#)
- nppiConvert_16u8s_C1RSfs
 - image_convert, [836](#)
- nppiConvert_16u8u_AC4R
 - image_convert, [837](#)
- nppiConvert_16u8u_C1R
 - image_convert, [837](#)
- nppiConvert_16u8u_C3R
 - image_convert, [837](#)
- nppiConvert_16u8u_C4R
 - image_convert, [838](#)
- nppiConvert_32f16s_AC4R
 - image_convert, [838](#)
- nppiConvert_32f16s_C1R
 - image_convert, [838](#)
- nppiConvert_32f16s_C1RSfs
 - image_convert, [839](#)
- nppiConvert_32f16s_C3R
 - image_convert, [839](#)
- nppiConvert_32f16s_C4R
 - image_convert, [840](#)
- nppiConvert_32f16u_AC4R
 - image_convert, [840](#)
- nppiConvert_32f16u_C1R
 - image_convert, [840](#)
- nppiConvert_32f16u_C1RSfs
 - image_convert, [841](#)
- nppiConvert_32f16u_C3R
 - image_convert, [841](#)

- npPiConvert_32f16u_C4R
 - image_convert, [842](#)
- npPiConvert_32f32s_C1RSfs
 - image_convert, [842](#)
- npPiConvert_32f32u_C1RSfs
 - image_convert, [842](#)
- npPiConvert_32f8s_AC4R
 - image_convert, [843](#)
- npPiConvert_32f8s_C1R
 - image_convert, [843](#)
- npPiConvert_32f8s_C1RSfs
 - image_convert, [844](#)
- npPiConvert_32f8s_C3R
 - image_convert, [844](#)
- npPiConvert_32f8s_C4R
 - image_convert, [844](#)
- npPiConvert_32f8u_AC4R
 - image_convert, [845](#)
- npPiConvert_32f8u_C1R
 - image_convert, [845](#)
- npPiConvert_32f8u_C1RSfs
 - image_convert, [845](#)
- npPiConvert_32f8u_C3R
 - image_convert, [846](#)
- npPiConvert_32f8u_C4R
 - image_convert, [846](#)
- npPiConvert_32s16s_C1RSfs
 - image_convert, [847](#)
- npPiConvert_32s16u_C1RSfs
 - image_convert, [847](#)
- npPiConvert_32s32f_C1R
 - image_convert, [847](#)
- npPiConvert_32s32u_C1Rs
 - image_convert, [848](#)
- npPiConvert_32s8s_AC4R
 - image_convert, [848](#)
- npPiConvert_32s8s_C1R
 - image_convert, [848](#)
- npPiConvert_32s8s_C3R
 - image_convert, [849](#)
- npPiConvert_32s8s_C4R
 - image_convert, [849](#)
- npPiConvert_32s8u_AC4R
 - image_convert, [849](#)
- npPiConvert_32s8u_C1R
 - image_convert, [850](#)
- npPiConvert_32s8u_C3R
 - image_convert, [850](#)
- npPiConvert_32s8u_C4R
 - image_convert, [850](#)
- npPiConvert_32u16s_C1RSfs
 - image_convert, [851](#)
- npPiConvert_32u16u_C1RSfs
 - image_convert, [851](#)
- npPiConvert_32u32f_C1R
 - image_convert, [852](#)
- npPiConvert_32u32s_C1RSfs
 - image_convert, [852](#)
- npPiConvert_32u8s_C1RSfs
 - image_convert, [852](#)
- npPiConvert_32u8u_C1RSfs
 - image_convert, [853](#)
- npPiConvert_8s16s_C1R
 - image_convert, [853](#)
- npPiConvert_8s16u_C1Rs
 - image_convert, [854](#)
- npPiConvert_8s32f_AC4R
 - image_convert, [854](#)
- npPiConvert_8s32f_C1R
 - image_convert, [854](#)
- npPiConvert_8s32f_C3R
 - image_convert, [855](#)
- npPiConvert_8s32f_C4R
 - image_convert, [855](#)
- npPiConvert_8s32s_AC4R
 - image_convert, [855](#)
- npPiConvert_8s32s_C1R
 - image_convert, [856](#)
- npPiConvert_8s32s_C3R
 - image_convert, [856](#)
- npPiConvert_8s32s_C4R
 - image_convert, [856](#)
- npPiConvert_8s32u_C1Rs
 - image_convert, [857](#)
- npPiConvert_8s8u_C1Rs
 - image_convert, [857](#)
- npPiConvert_8u16s_AC4R
 - image_convert, [857](#)
- npPiConvert_8u16s_C1R
 - image_convert, [858](#)
- npPiConvert_8u16s_C3R
 - image_convert, [858](#)
- npPiConvert_8u16s_C4R
 - image_convert, [858](#)
- npPiConvert_8u16u_AC4R
 - image_convert, [859](#)
- npPiConvert_8u16u_C1R
 - image_convert, [859](#)
- npPiConvert_8u16u_C3R
 - image_convert, [859](#)
- npPiConvert_8u16u_C4R
 - image_convert, [860](#)
- npPiConvert_8u32f_AC4R
 - image_convert, [860](#)
- npPiConvert_8u32f_C1R
 - image_convert, [860](#)
- npPiConvert_8u32f_C3R
 - image_convert, [861](#)

- npPiConvert_8u32f_C4R
image_convert, [861](#)
- npPiConvert_8u32s_AC4R
image_convert, [861](#)
- npPiConvert_8u32s_C1R
image_convert, [862](#)
- npPiConvert_8u32s_C3R
image_convert, [862](#)
- npPiConvert_8u32s_C4R
image_convert, [862](#)
- npPiConvert_8u8s_C1RSfs
image_convert, [863](#)
- npPiCopy_16s_AC4MR
image_copy, [782](#)
- npPiCopy_16s_AC4R
image_copy, [783](#)
- npPiCopy_16s_C1C3R
image_copy, [783](#)
- npPiCopy_16s_C1C4R
image_copy, [783](#)
- npPiCopy_16s_C1MR
image_copy, [784](#)
- npPiCopy_16s_C1R
image_copy, [784](#)
- npPiCopy_16s_C3C1R
image_copy, [784](#)
- npPiCopy_16s_C3CR
image_copy, [785](#)
- npPiCopy_16s_C3MR
image_copy, [785](#)
- npPiCopy_16s_C3P3R
image_copy, [785](#)
- npPiCopy_16s_C3R
image_copy, [786](#)
- npPiCopy_16s_C4C1R
image_copy, [786](#)
- npPiCopy_16s_C4CR
image_copy, [786](#)
- npPiCopy_16s_C4MR
image_copy, [787](#)
- npPiCopy_16s_C4P4R
image_copy, [787](#)
- npPiCopy_16s_C4R
image_copy, [787](#)
- npPiCopy_16s_P3C3R
image_copy, [788](#)
- npPiCopy_16s_P4C4R
image_copy, [788](#)
- npPiCopy_16sc_AC4R
image_copy, [788](#)
- npPiCopy_16sc_C1R
image_copy, [789](#)
- npPiCopy_16sc_C2R
image_copy, [789](#)
- npPiCopy_16sc_C3R
image_copy, [789](#)
- npPiCopy_16sc_C4R
image_copy, [790](#)
- npPiCopy_16u_AC4MR
image_copy, [790](#)
- npPiCopy_16u_AC4R
image_copy, [790](#)
- npPiCopy_16u_C1C3R
image_copy, [791](#)
- npPiCopy_16u_C1C4R
image_copy, [791](#)
- npPiCopy_16u_C1MR
image_copy, [791](#)
- npPiCopy_16u_C1R
image_copy, [792](#)
- npPiCopy_16u_C3C1R
image_copy, [792](#)
- npPiCopy_16u_C3CR
image_copy, [792](#)
- npPiCopy_16u_C3MR
image_copy, [793](#)
- npPiCopy_16u_C3P3R
image_copy, [793](#)
- npPiCopy_16u_C3R
image_copy, [793](#)
- npPiCopy_16u_C4C1R
image_copy, [794](#)
- npPiCopy_16u_C4CR
image_copy, [794](#)
- npPiCopy_16u_C4MR
image_copy, [794](#)
- npPiCopy_16u_C4P4R
image_copy, [795](#)
- npPiCopy_16u_C4R
image_copy, [795](#)
- npPiCopy_16u_P3C3R
image_copy, [795](#)
- npPiCopy_16u_P4C4R
image_copy, [796](#)
- npPiCopy_32f_AC4MR
image_copy, [796](#)
- npPiCopy_32f_AC4R
image_copy, [796](#)
- npPiCopy_32f_C1C3R
image_copy, [797](#)
- npPiCopy_32f_C1C4R
image_copy, [797](#)
- npPiCopy_32f_C1MR
image_copy, [797](#)
- npPiCopy_32f_C1R
image_copy, [798](#)
- npPiCopy_32f_C3C1R
image_copy, [798](#)

- npPiCopy_32f_C3CR
 - image_copy, [798](#)
- npPiCopy_32f_C3MR
 - image_copy, [799](#)
- npPiCopy_32f_C3P3R
 - image_copy, [799](#)
- npPiCopy_32f_C3R
 - image_copy, [799](#)
- npPiCopy_32f_C4C1R
 - image_copy, [800](#)
- npPiCopy_32f_C4CR
 - image_copy, [800](#)
- npPiCopy_32f_C4MR
 - image_copy, [800](#)
- npPiCopy_32f_C4P4R
 - image_copy, [801](#)
- npPiCopy_32f_C4R
 - image_copy, [801](#)
- npPiCopy_32f_P3C3R
 - image_copy, [801](#)
- npPiCopy_32f_P4C4R
 - image_copy, [802](#)
- npPiCopy_32fc_AC4R
 - image_copy, [802](#)
- npPiCopy_32fc_C1R
 - image_copy, [802](#)
- npPiCopy_32fc_C2R
 - image_copy, [803](#)
- npPiCopy_32fc_C3R
 - image_copy, [803](#)
- npPiCopy_32fc_C4R
 - image_copy, [803](#)
- npPiCopy_32s_AC4MR
 - image_copy, [804](#)
- npPiCopy_32s_AC4R
 - image_copy, [804](#)
- npPiCopy_32s_C1C3R
 - image_copy, [804](#)
- npPiCopy_32s_C1C4R
 - image_copy, [805](#)
- npPiCopy_32s_C1MR
 - image_copy, [805](#)
- npPiCopy_32s_C1R
 - image_copy, [805](#)
- npPiCopy_32s_C3C1R
 - image_copy, [806](#)
- npPiCopy_32s_C3CR
 - image_copy, [806](#)
- npPiCopy_32s_C3MR
 - image_copy, [806](#)
- npPiCopy_32s_C3P3R
 - image_copy, [807](#)
- npPiCopy_32s_C3R
 - image_copy, [807](#)
- npPiCopy_32s_C4C1R
 - image_copy, [807](#)
- npPiCopy_32s_C4CR
 - image_copy, [808](#)
- npPiCopy_32s_C4MR
 - image_copy, [808](#)
- npPiCopy_32s_C4P4R
 - image_copy, [808](#)
- npPiCopy_32s_C4R
 - image_copy, [809](#)
- npPiCopy_32s_P3C3R
 - image_copy, [809](#)
- npPiCopy_32s_P4C4R
 - image_copy, [809](#)
- npPiCopy_32sc_AC4R
 - image_copy, [810](#)
- npPiCopy_32sc_C1R
 - image_copy, [810](#)
- npPiCopy_32sc_C2R
 - image_copy, [810](#)
- npPiCopy_32sc_C3R
 - image_copy, [811](#)
- npPiCopy_32sc_C4R
 - image_copy, [811](#)
- npPiCopy_8s_AC4R
 - image_copy, [811](#)
- npPiCopy_8s_C1R
 - image_copy, [812](#)
- npPiCopy_8s_C2R
 - image_copy, [812](#)
- npPiCopy_8s_C3R
 - image_copy, [812](#)
- npPiCopy_8s_C4R
 - image_copy, [813](#)
- npPiCopy_8u_AC4MR
 - image_copy, [813](#)
- npPiCopy_8u_AC4R
 - image_copy, [813](#)
- npPiCopy_8u_C1C3R
 - image_copy, [814](#)
- npPiCopy_8u_C1C4R
 - image_copy, [814](#)
- npPiCopy_8u_C1MR
 - image_copy, [814](#)
- npPiCopy_8u_C1R
 - image_copy, [815](#)
- npPiCopy_8u_C3C1R
 - image_copy, [815](#)
- npPiCopy_8u_C3CR
 - image_copy, [815](#)
- npPiCopy_8u_C3MR
 - image_copy, [816](#)
- npPiCopy_8u_C3P3R
 - image_copy, [816](#)

- image_copy, [816](#)
- nppiCopy_8u_C4C1R
 - image_copy, [817](#)
- nppiCopy_8u_C4CR
 - image_copy, [817](#)
- nppiCopy_8u_C4MR
 - image_copy, [817](#)
- nppiCopy_8u_C4P4R
 - image_copy, [818](#)
- nppiCopy_8u_C4R
 - image_copy, [818](#)
- nppiCopy_8u_P3C3R
 - image_copy, [818](#)
- nppiCopy_8u_P4C4R
 - image_copy, [819](#)
- nppiCopyConstBorder_16s_AC4R
 - image_copy_constant_border, [881](#)
- nppiCopyConstBorder_16s_C1R
 - image_copy_constant_border, [881](#)
- nppiCopyConstBorder_16s_C3R
 - image_copy_constant_border, [882](#)
- nppiCopyConstBorder_16s_C4R
 - image_copy_constant_border, [882](#)
- nppiCopyConstBorder_16u_AC4R
 - image_copy_constant_border, [883](#)
- nppiCopyConstBorder_16u_C1R
 - image_copy_constant_border, [883](#)
- nppiCopyConstBorder_16u_C3R
 - image_copy_constant_border, [884](#)
- nppiCopyConstBorder_16u_C4R
 - image_copy_constant_border, [884](#)
- nppiCopyConstBorder_32f_AC4R
 - image_copy_constant_border, [885](#)
- nppiCopyConstBorder_32f_C1R
 - image_copy_constant_border, [885](#)
- nppiCopyConstBorder_32f_C3R
 - image_copy_constant_border, [886](#)
- nppiCopyConstBorder_32f_C4R
 - image_copy_constant_border, [886](#)
- nppiCopyConstBorder_32s_AC4R
 - image_copy_constant_border, [887](#)
- nppiCopyConstBorder_32s_C1R
 - image_copy_constant_border, [887](#)
- nppiCopyConstBorder_32s_C3R
 - image_copy_constant_border, [888](#)
- nppiCopyConstBorder_32s_C4R
 - image_copy_constant_border, [888](#)
- nppiCopyConstBorder_8u_AC4R
 - image_copy_constant_border, [889](#)
- nppiCopyConstBorder_8u_C1R
 - image_copy_constant_border, [889](#)
- nppiCopyConstBorder_8u_C3R
 - image_copy_constant_border, [890](#)
- nppiCopyConstBorder_8u_C4R
 - image_copy_constant_border, [890](#)
- nppiCopyReplicateBorder_16s_AC4R
 - image_copy_replicate_border, [894](#)
- nppiCopyReplicateBorder_16s_C1R
 - image_copy_replicate_border, [894](#)
- nppiCopyReplicateBorder_16s_C3R
 - image_copy_replicate_border, [895](#)
- nppiCopyReplicateBorder_16s_C4R
 - image_copy_replicate_border, [895](#)
- nppiCopyReplicateBorder_16u_AC4R
 - image_copy_replicate_border, [896](#)
- nppiCopyReplicateBorder_16u_C1R
 - image_copy_replicate_border, [896](#)
- nppiCopyReplicateBorder_16u_C3R
 - image_copy_replicate_border, [897](#)
- nppiCopyReplicateBorder_16u_C4R
 - image_copy_replicate_border, [897](#)
- nppiCopyReplicateBorder_32f_AC4R
 - image_copy_replicate_border, [897](#)
- nppiCopyReplicateBorder_32f_C1R
 - image_copy_replicate_border, [898](#)
- nppiCopyReplicateBorder_32f_C3R
 - image_copy_replicate_border, [898](#)
- nppiCopyReplicateBorder_32f_C4R
 - image_copy_replicate_border, [899](#)
- nppiCopyReplicateBorder_32s_AC4R
 - image_copy_replicate_border, [899](#)
- nppiCopyReplicateBorder_32s_C1R
 - image_copy_replicate_border, [900](#)
- nppiCopyReplicateBorder_32s_C3R
 - image_copy_replicate_border, [900](#)
- nppiCopyReplicateBorder_32s_C4R
 - image_copy_replicate_border, [901](#)
- nppiCopyReplicateBorder_8u_AC4R
 - image_copy_replicate_border, [901](#)
- nppiCopyReplicateBorder_8u_C1R
 - image_copy_replicate_border, [902](#)
- nppiCopyReplicateBorder_8u_C3R
 - image_copy_replicate_border, [902](#)
- nppiCopyReplicateBorder_8u_C4R
 - image_copy_replicate_border, [903](#)
- nppiCopySubpix_16s_AC4R
 - image_copy_sub_pixel, [918](#)
- nppiCopySubpix_16s_C1R
 - image_copy_sub_pixel, [919](#)
- nppiCopySubpix_16s_C3R
 - image_copy_sub_pixel, [919](#)
- nppiCopySubpix_16s_C4R
 - image_copy_sub_pixel, [920](#)
- nppiCopySubpix_16u_AC4R
 - image_copy_sub_pixel, [920](#)
- nppiCopySubpix_16u_C1R
 - image_copy_sub_pixel, [920](#)

- nppiCopySubpix_16u_C3R
 - image_copy_sub_pixel, [921](#)
- nppiCopySubpix_16u_C4R
 - image_copy_sub_pixel, [921](#)
- nppiCopySubpix_32f_AC4R
 - image_copy_sub_pixel, [922](#)
- nppiCopySubpix_32f_C1R
 - image_copy_sub_pixel, [922](#)
- nppiCopySubpix_32f_C3R
 - image_copy_sub_pixel, [922](#)
- nppiCopySubpix_32f_C4R
 - image_copy_sub_pixel, [923](#)
- nppiCopySubpix_32s_AC4R
 - image_copy_sub_pixel, [923](#)
- nppiCopySubpix_32s_C1R
 - image_copy_sub_pixel, [924](#)
- nppiCopySubpix_32s_C3R
 - image_copy_sub_pixel, [924](#)
- nppiCopySubpix_32s_C4R
 - image_copy_sub_pixel, [925](#)
- nppiCopySubpix_8u_AC4R
 - image_copy_sub_pixel, [925](#)
- nppiCopySubpix_8u_C1R
 - image_copy_sub_pixel, [925](#)
- nppiCopySubpix_8u_C3R
 - image_copy_sub_pixel, [926](#)
- nppiCopySubpix_8u_C4R
 - image_copy_sub_pixel, [926](#)
- nppiCopyWrapBorder_16s_AC4R
 - image_copy_wrap_border, [906](#)
- nppiCopyWrapBorder_16s_C1R
 - image_copy_wrap_border, [906](#)
- nppiCopyWrapBorder_16s_C3R
 - image_copy_wrap_border, [907](#)
- nppiCopyWrapBorder_16s_C4R
 - image_copy_wrap_border, [907](#)
- nppiCopyWrapBorder_16u_AC4R
 - image_copy_wrap_border, [908](#)
- nppiCopyWrapBorder_16u_C1R
 - image_copy_wrap_border, [908](#)
- nppiCopyWrapBorder_16u_C3R
 - image_copy_wrap_border, [909](#)
- nppiCopyWrapBorder_16u_C4R
 - image_copy_wrap_border, [909](#)
- nppiCopyWrapBorder_32f_AC4R
 - image_copy_wrap_border, [910](#)
- nppiCopyWrapBorder_32f_C1R
 - image_copy_wrap_border, [910](#)
- nppiCopyWrapBorder_32f_C3R
 - image_copy_wrap_border, [911](#)
- nppiCopyWrapBorder_32f_C4R
 - image_copy_wrap_border, [911](#)
- nppiCopyWrapBorder_32s_AC4R
 - image_copy_wrap_border, [912](#)
- nppiCopyWrapBorder_32s_C1R
 - image_copy_wrap_border, [912](#)
- nppiCopyWrapBorder_32s_C3R
 - image_copy_wrap_border, [913](#)
- nppiCopyWrapBorder_32s_C4R
 - image_copy_wrap_border, [913](#)
- nppiCopyWrapBorder_8u_AC4R
 - image_copy_wrap_border, [914](#)
- nppiCopyWrapBorder_8u_C1R
 - image_copy_wrap_border, [914](#)
- nppiCopyWrapBorder_8u_C3R
 - image_copy_wrap_border, [915](#)
- nppiCopyWrapBorder_8u_C4R
 - image_copy_wrap_border, [915](#)
- nppiCountInRange_32f_AC4R
 - image_count_in_range, [2069](#)
- nppiCountInRange_32f_C1R
 - image_count_in_range, [2069](#)
- nppiCountInRange_32f_C3R
 - image_count_in_range, [2070](#)
- nppiCountInRange_8u_AC4R
 - image_count_in_range, [2070](#)
- nppiCountInRange_8u_C1R
 - image_count_in_range, [2071](#)
- nppiCountInRange_8u_C3R
 - image_count_in_range, [2071](#)
- nppiCountInRangeGetBufferHostSize_32f_AC4R
 - image_count_in_range, [2072](#)
- nppiCountInRangeGetBufferHostSize_32f_C1R
 - image_count_in_range, [2072](#)
- nppiCountInRangeGetBufferHostSize_32f_C3R
 - image_count_in_range, [2072](#)
- nppiCountInRangeGetBufferHostSize_8u_AC4R
 - image_count_in_range, [2072](#)
- nppiCountInRangeGetBufferHostSize_8u_C1R
 - image_count_in_range, [2073](#)
- nppiCountInRangeGetBufferHostSize_8u_C3R
 - image_count_in_range, [2073](#)
- nppiCrossCorrFull_Norm_16u32f_AC4R
 - crosscorrfullnorm, [2163](#)
- nppiCrossCorrFull_Norm_16u32f_C1R
 - crosscorrfullnorm, [2163](#)
- nppiCrossCorrFull_Norm_16u32f_C3R
 - crosscorrfullnorm, [2163](#)
- nppiCrossCorrFull_Norm_16u32f_C4R
 - crosscorrfullnorm, [2164](#)
- nppiCrossCorrFull_Norm_32f_AC4R
 - crosscorrfullnorm, [2164](#)
- nppiCrossCorrFull_Norm_32f_C1R
 - crosscorrfullnorm, [2165](#)
- nppiCrossCorrFull_Norm_32f_C3R
 - crosscorrfullnorm, [2165](#)
- nppiCrossCorrFull_Norm_32f_C4R
 - crosscorrfullnorm, [2166](#)

- nppiCrossCorrFull_Norm_8s32f_AC4R
 crosscorrfullnorm, [2166](#)
 nppiCrossCorrFull_Norm_8s32f_C1R
 crosscorrfullnorm, [2166](#)
 nppiCrossCorrFull_Norm_8s32f_C3R
 crosscorrfullnorm, [2167](#)
 nppiCrossCorrFull_Norm_8s32f_C4R
 crosscorrfullnorm, [2167](#)
 nppiCrossCorrFull_Norm_8u32f_AC4R
 crosscorrfullnorm, [2168](#)
 nppiCrossCorrFull_Norm_8u32f_C1R
 crosscorrfullnorm, [2168](#)
 nppiCrossCorrFull_Norm_8u32f_C3R
 crosscorrfullnorm, [2169](#)
 nppiCrossCorrFull_Norm_8u32f_C4R
 crosscorrfullnorm, [2169](#)
 nppiCrossCorrFull_Norm_8u_AC4RSfs
 crosscorrfullnorm, [2169](#)
 nppiCrossCorrFull_Norm_8u_C1RSfs
 crosscorrfullnorm, [2170](#)
 nppiCrossCorrFull_Norm_8u_C3RSfs
 crosscorrfullnorm, [2170](#)
 nppiCrossCorrFull_Norm_8u_C4RSfs
 crosscorrfullnorm, [2171](#)
 nppiCrossCorrFull_NormLevel_16u32f_AC4R
 crosscorrfullnormlevel, [2201](#)
 nppiCrossCorrFull_NormLevel_16u32f_C1R
 crosscorrfullnormlevel, [2201](#)
 nppiCrossCorrFull_NormLevel_16u32f_C3R
 crosscorrfullnormlevel, [2201](#)
 nppiCrossCorrFull_NormLevel_16u32f_C4R
 crosscorrfullnormlevel, [2202](#)
 nppiCrossCorrFull_NormLevel_32f_AC4R
 crosscorrfullnormlevel, [2202](#)
 nppiCrossCorrFull_NormLevel_32f_C1R
 crosscorrfullnormlevel, [2203](#)
 nppiCrossCorrFull_NormLevel_32f_C3R
 crosscorrfullnormlevel, [2203](#)
 nppiCrossCorrFull_NormLevel_32f_C4R
 crosscorrfullnormlevel, [2204](#)
 nppiCrossCorrFull_NormLevel_8s32f_AC4R
 crosscorrfullnormlevel, [2204](#)
 nppiCrossCorrFull_NormLevel_8s32f_C1R
 crosscorrfullnormlevel, [2205](#)
 nppiCrossCorrFull_NormLevel_8s32f_C3R
 crosscorrfullnormlevel, [2205](#)
 nppiCrossCorrFull_NormLevel_8s32f_C4R
 crosscorrfullnormlevel, [2206](#)
 nppiCrossCorrFull_NormLevel_8u32f_AC4R
 crosscorrfullnormlevel, [2206](#)
 nppiCrossCorrFull_NormLevel_8u32f_C1R
 crosscorrfullnormlevel, [2207](#)
 nppiCrossCorrFull_NormLevel_8u32f_C3R
 crosscorrfullnormlevel, [2207](#)
 nppiCrossCorrFull_NormLevel_8u32f_C4R
 crosscorrfullnormlevel, [2208](#)
 nppiCrossCorrFull_NormLevel_8u_AC4RSfs
 crosscorrfullnormlevel, [2208](#)
 nppiCrossCorrFull_NormLevel_8u_C1RSfs
 crosscorrfullnormlevel, [2209](#)
 nppiCrossCorrFull_NormLevel_8u_C3RSfs
 crosscorrfullnormlevel, [2209](#)
 nppiCrossCorrFull_NormLevel_8u_C4RSfs
 crosscorrfullnormlevel, [2210](#)
 nppiCrossCorrSame_Norm_16u32f_AC4R
 crosscorrsmenorm, [2174](#)
 nppiCrossCorrSame_Norm_16u32f_C1R
 crosscorrsmenorm, [2174](#)
 nppiCrossCorrSame_Norm_16u32f_C3R
 crosscorrsmenorm, [2174](#)
 nppiCrossCorrSame_Norm_16u32f_C4R
 crosscorrsmenorm, [2175](#)
 nppiCrossCorrSame_Norm_32f_AC4R
 crosscorrsmenorm, [2175](#)
 nppiCrossCorrSame_Norm_32f_C1R
 crosscorrsmenorm, [2176](#)
 nppiCrossCorrSame_Norm_32f_C3R
 crosscorrsmenorm, [2176](#)
 nppiCrossCorrSame_Norm_32f_C4R
 crosscorrsmenorm, [2177](#)
 nppiCrossCorrSame_Norm_8s32f_AC4R
 crosscorrsmenorm, [2177](#)
 nppiCrossCorrSame_Norm_8s32f_C1R
 crosscorrsmenorm, [2177](#)
 nppiCrossCorrSame_Norm_8s32f_C3R
 crosscorrsmenorm, [2178](#)
 nppiCrossCorrSame_Norm_8s32f_C4R
 crosscorrsmenorm, [2178](#)
 nppiCrossCorrSame_Norm_8u32f_AC4R
 crosscorrsmenorm, [2179](#)
 nppiCrossCorrSame_Norm_8u32f_C1R
 crosscorrsmenorm, [2179](#)
 nppiCrossCorrSame_Norm_8u32f_C3R
 crosscorrsmenorm, [2180](#)
 nppiCrossCorrSame_Norm_8u32f_C4R
 crosscorrsmenorm, [2180](#)
 nppiCrossCorrSame_Norm_8u_AC4RSfs
 crosscorrsmenorm, [2180](#)
 nppiCrossCorrSame_Norm_8u_C1RSfs
 crosscorrsmenorm, [2181](#)
 nppiCrossCorrSame_Norm_8u_C3RSfs
 crosscorrsmenorm, [2181](#)
 nppiCrossCorrSame_Norm_8u_C4RSfs
 crosscorrsmenorm, [2182](#)
 nppiCrossCorrSame_NormLevel_16u32f_AC4R
 crosscorrsmenormlevel, [2221](#)
 nppiCrossCorrSame_NormLevel_16u32f_C1R
 crosscorrsmenormlevel, [2221](#)

- nppiCrossCorrSame_NormLevel_16u32f_C3R
crosscorrmenormlevel, [2221](#)
- nppiCrossCorrSame_NormLevel_16u32f_C4R
crosscorrmenormlevel, [2222](#)
- nppiCrossCorrSame_NormLevel_32f_AC4R
crosscorrmenormlevel, [2222](#)
- nppiCrossCorrSame_NormLevel_32f_C1R
crosscorrmenormlevel, [2223](#)
- nppiCrossCorrSame_NormLevel_32f_C3R
crosscorrmenormlevel, [2223](#)
- nppiCrossCorrSame_NormLevel_32f_C4R
crosscorrmenormlevel, [2224](#)
- nppiCrossCorrSame_NormLevel_8s32f_AC4R
crosscorrmenormlevel, [2224](#)
- nppiCrossCorrSame_NormLevel_8s32f_C1R
crosscorrmenormlevel, [2225](#)
- nppiCrossCorrSame_NormLevel_8s32f_C3R
crosscorrmenormlevel, [2225](#)
- nppiCrossCorrSame_NormLevel_8s32f_C4R
crosscorrmenormlevel, [2226](#)
- nppiCrossCorrSame_NormLevel_8u32f_AC4R
crosscorrmenormlevel, [2226](#)
- nppiCrossCorrSame_NormLevel_8u32f_C1R
crosscorrmenormlevel, [2227](#)
- nppiCrossCorrSame_NormLevel_8u32f_C3R
crosscorrmenormlevel, [2227](#)
- nppiCrossCorrSame_NormLevel_8u32f_C4R
crosscorrmenormlevel, [2228](#)
- nppiCrossCorrSame_NormLevel_8u_AC4RSfs
crosscorrmenormlevel, [2228](#)
- nppiCrossCorrSame_NormLevel_8u_C1RSfs
crosscorrmenormlevel, [2229](#)
- nppiCrossCorrSame_NormLevel_8u_C3RSfs
crosscorrmenormlevel, [2229](#)
- nppiCrossCorrSame_NormLevel_8u_C4RSfs
crosscorrmenormlevel, [2230](#)
- nppiCrossCorrValid_16u32f_C1R
crosscorrvalid, [2194](#)
- nppiCrossCorrValid_32f_C1R
crosscorrvalid, [2195](#)
- nppiCrossCorrValid_8s32f_C1R
crosscorrvalid, [2195](#)
- nppiCrossCorrValid_8u32f_C1R
crosscorrvalid, [2195](#)
- nppiCrossCorrValid_Norm_16u32f_AC4R
crosscorrvalidnorm, [2185](#)
- nppiCrossCorrValid_Norm_16u32f_C1R
crosscorrvalidnorm, [2185](#)
- nppiCrossCorrValid_Norm_16u32f_C3R
crosscorrvalidnorm, [2185](#)
- nppiCrossCorrValid_Norm_16u32f_C4R
crosscorrvalidnorm, [2186](#)
- nppiCrossCorrValid_Norm_32f_AC4R
crosscorrvalidnorm, [2186](#)
- nppiCrossCorrValid_Norm_32f_C1R
crosscorrvalidnorm, [2187](#)
- nppiCrossCorrValid_Norm_32f_C3R
crosscorrvalidnorm, [2187](#)
- nppiCrossCorrValid_Norm_32f_C4R
crosscorrvalidnorm, [2188](#)
- nppiCrossCorrValid_Norm_8s32f_AC4R
crosscorrvalidnorm, [2188](#)
- nppiCrossCorrValid_Norm_8s32f_C1R
crosscorrvalidnorm, [2188](#)
- nppiCrossCorrValid_Norm_8s32f_C3R
crosscorrvalidnorm, [2189](#)
- nppiCrossCorrValid_Norm_8s32f_C4R
crosscorrvalidnorm, [2189](#)
- nppiCrossCorrValid_Norm_8u32f_AC4R
crosscorrvalidnorm, [2190](#)
- nppiCrossCorrValid_Norm_8u32f_C1R
crosscorrvalidnorm, [2190](#)
- nppiCrossCorrValid_Norm_8u32f_C3R
crosscorrvalidnorm, [2191](#)
- nppiCrossCorrValid_Norm_8u32f_C4R
crosscorrvalidnorm, [2191](#)
- nppiCrossCorrValid_Norm_8u_AC4RSfs
crosscorrvalidnorm, [2191](#)
- nppiCrossCorrValid_Norm_8u_C1RSfs
crosscorrvalidnorm, [2192](#)
- nppiCrossCorrValid_Norm_8u_C3RSfs
crosscorrvalidnorm, [2192](#)
- nppiCrossCorrValid_Norm_8u_C4RSfs
crosscorrvalidnorm, [2193](#)
- nppiCrossCorrValid_NormLevel_16u32f_AC4R
crosscorrvalidnormlevel, [2241](#)
- nppiCrossCorrValid_NormLevel_16u32f_C1R
crosscorrvalidnormlevel, [2241](#)
- nppiCrossCorrValid_NormLevel_16u32f_C3R
crosscorrvalidnormlevel, [2241](#)
- nppiCrossCorrValid_NormLevel_16u32f_C4R
crosscorrvalidnormlevel, [2242](#)
- nppiCrossCorrValid_NormLevel_32f_AC4R
crosscorrvalidnormlevel, [2242](#)
- nppiCrossCorrValid_NormLevel_32f_C1R
crosscorrvalidnormlevel, [2243](#)
- nppiCrossCorrValid_NormLevel_32f_C3R
crosscorrvalidnormlevel, [2243](#)
- nppiCrossCorrValid_NormLevel_32f_C4R
crosscorrvalidnormlevel, [2244](#)
- nppiCrossCorrValid_NormLevel_8s32f_AC4R
crosscorrvalidnormlevel, [2244](#)
- nppiCrossCorrValid_NormLevel_8s32f_C1R
crosscorrvalidnormlevel, [2245](#)
- nppiCrossCorrValid_NormLevel_8s32f_C3R
crosscorrvalidnormlevel, [2245](#)
- nppiCrossCorrValid_NormLevel_8s32f_C4R
crosscorrvalidnormlevel, [2246](#)

- nppiCrossCorrValid_NormLevel_8u32f_AC4R
crosscorrvalidnormlevel, [2246](#)
- nppiCrossCorrValid_NormLevel_8u32f_C1R
crosscorrvalidnormlevel, [2247](#)
- nppiCrossCorrValid_NormLevel_8u32f_C3R
crosscorrvalidnormlevel, [2247](#)
- nppiCrossCorrValid_NormLevel_8u32f_C4R
crosscorrvalidnormlevel, [2248](#)
- nppiCrossCorrValid_NormLevel_8u_AC4RSfs
crosscorrvalidnormlevel, [2248](#)
- nppiCrossCorrValid_NormLevel_8u_C1RSfs
crosscorrvalidnormlevel, [2249](#)
- nppiCrossCorrValid_NormLevel_8u_C3RSfs
crosscorrvalidnormlevel, [2249](#)
- nppiCrossCorrValid_NormLevel_8u_C4RSfs
crosscorrvalidnormlevel, [2250](#)
- nppiDCTable
typedefs_npp, [43](#)
- nppiDCTFree
image_quantization, [725](#)
- nppiDCTInitAlloc
image_quantization, [725](#)
- nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R
image_quantization, [725](#)
- nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R_-
NEW
image_quantization, [726](#)
- nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R
image_quantization, [726](#)
- nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R_-
NEW
image_quantization, [727](#)
- NppiDCTState
image_quantization, [725](#)
- nppiDecodeHuffmanScanHost_JPEG_8u16s_P1R
image_compression, [721](#)
- nppiDecodeHuffmanScanHost_JPEG_8u16s_P3R
image_compression, [721](#)
- NppiDecodeHuffmanSpec
image_compression, [721](#)
- nppiDecodeHuffmanSpecFreeHost_JPEG
image_compression, [722](#)
- nppiDecodeHuffmanSpecGetBufSize_JPEG
image_compression, [722](#)
- nppiDecodeHuffmanSpecInitAllocHost_JPEG
image_compression, [722](#)
- nppiDecodeHuffmanSpecInitHost_JPEG
image_compression, [723](#)
- nppiDilate3x3_16u_AC4R
image_dilate_3x3, [1595](#)
- nppiDilate3x3_16u_C1R
image_dilate_3x3, [1595](#)
- nppiDilate3x3_16u_C3R
image_dilate_3x3, [1595](#)
- nppiDilate3x3_16u_C4R
image_dilate_3x3, [1596](#)
- nppiDilate3x3_32f_AC4R
image_dilate_3x3, [1596](#)
- nppiDilate3x3_32f_C1R
image_dilate_3x3, [1596](#)
- nppiDilate3x3_32f_C3R
image_dilate_3x3, [1597](#)
- nppiDilate3x3_32f_C4R
image_dilate_3x3, [1597](#)
- nppiDilate3x3_64f_C1R
image_dilate_3x3, [1597](#)
- nppiDilate3x3_8u_AC4R
image_dilate_3x3, [1598](#)
- nppiDilate3x3_8u_C1R
image_dilate_3x3, [1598](#)
- nppiDilate3x3_8u_C3R
image_dilate_3x3, [1598](#)
- nppiDilate3x3_8u_C4R
image_dilate_3x3, [1599](#)
- nppiDilate3x3Border_16u_AC4R
image_dilate_3x3_border, [1601](#)
- nppiDilate3x3Border_16u_C1R
image_dilate_3x3_border, [1601](#)
- nppiDilate3x3Border_16u_C3R
image_dilate_3x3_border, [1602](#)
- nppiDilate3x3Border_16u_C4R
image_dilate_3x3_border, [1602](#)
- nppiDilate3x3Border_32f_AC4R
image_dilate_3x3_border, [1603](#)
- nppiDilate3x3Border_32f_C1R
image_dilate_3x3_border, [1603](#)
- nppiDilate3x3Border_32f_C3R
image_dilate_3x3_border, [1604](#)
- nppiDilate3x3Border_32f_C4R
image_dilate_3x3_border, [1604](#)
- nppiDilate3x3Border_8u_AC4R
image_dilate_3x3_border, [1604](#)
- nppiDilate3x3Border_8u_C1R
image_dilate_3x3_border, [1605](#)
- nppiDilate3x3Border_8u_C3R
image_dilate_3x3_border, [1605](#)
- nppiDilate3x3Border_8u_C4R
image_dilate_3x3_border, [1606](#)
- nppiDilate_16u_AC4R
image_dilate, [1580](#)
- nppiDilate_16u_C1R
image_dilate, [1580](#)
- nppiDilate_16u_C3R
image_dilate, [1581](#)
- nppiDilate_16u_C4R
image_dilate, [1581](#)
- nppiDilate_32f_AC4R
image_dilate, [1581](#)

- nppiDilate_32f_C1R
 - image_dilate, [1582](#)
- nppiDilate_32f_C3R
 - image_dilate, [1582](#)
- nppiDilate_32f_C4R
 - image_dilate, [1583](#)
- nppiDilate_8u_AC4R
 - image_dilate, [1583](#)
- nppiDilate_8u_C1R
 - image_dilate, [1584](#)
- nppiDilate_8u_C3R
 - image_dilate, [1584](#)
- nppiDilate_8u_C4R
 - image_dilate, [1584](#)
- nppiDilateBorder_16u_AC4R
 - image_dilate_border, [1587](#)
- nppiDilateBorder_16u_C1R
 - image_dilate_border, [1588](#)
- nppiDilateBorder_16u_C3R
 - image_dilate_border, [1588](#)
- nppiDilateBorder_16u_C4R
 - image_dilate_border, [1589](#)
- nppiDilateBorder_32f_AC4R
 - image_dilate_border, [1589](#)
- nppiDilateBorder_32f_C1R
 - image_dilate_border, [1590](#)
- nppiDilateBorder_32f_C3R
 - image_dilate_border, [1590](#)
- nppiDilateBorder_32f_C4R
 - image_dilate_border, [1591](#)
- nppiDilateBorder_8u_AC4R
 - image_dilate_border, [1591](#)
- nppiDilateBorder_8u_C1R
 - image_dilate_border, [1592](#)
- nppiDilateBorder_8u_C3R
 - image_dilate_border, [1592](#)
- nppiDilateBorder_8u_C4R
 - image_dilate_border, [1593](#)
- nppiDiv_16s_AC4IRSfs
 - image_div, [282](#)
- nppiDiv_16s_AC4RSfs
 - image_div, [282](#)
- nppiDiv_16s_C1IRSfs
 - image_div, [283](#)
- nppiDiv_16s_C1RSfs
 - image_div, [283](#)
- nppiDiv_16s_C3IRSfs
 - image_div, [283](#)
- nppiDiv_16s_C3RSfs
 - image_div, [284](#)
- nppiDiv_16s_C4IRSfs
 - image_div, [284](#)
- nppiDiv_16s_C4RSfs
 - image_div, [285](#)
- nppiDiv_16sc_AC4IRSfs
 - image_div, [285](#)
- nppiDiv_16sc_AC4RSfs
 - image_div, [285](#)
- nppiDiv_16sc_C1IRSfs
 - image_div, [286](#)
- nppiDiv_16sc_C1RSfs
 - image_div, [286](#)
- nppiDiv_16sc_C3IRSfs
 - image_div, [287](#)
- nppiDiv_16sc_C3RSfs
 - image_div, [287](#)
- nppiDiv_16u_AC4IRSfs
 - image_div, [288](#)
- nppiDiv_16u_AC4RSfs
 - image_div, [288](#)
- nppiDiv_16u_C1IRSfs
 - image_div, [288](#)
- nppiDiv_16u_C1RSfs
 - image_div, [289](#)
- nppiDiv_16u_C3IRSfs
 - image_div, [289](#)
- nppiDiv_16u_C3RSfs
 - image_div, [290](#)
- nppiDiv_16u_C4IRSfs
 - image_div, [290](#)
- nppiDiv_16u_C4RSfs
 - image_div, [290](#)
- nppiDiv_32f_AC4IR
 - image_div, [291](#)
- nppiDiv_32f_AC4R
 - image_div, [291](#)
- nppiDiv_32f_C1IR
 - image_div, [292](#)
- nppiDiv_32f_C1R
 - image_div, [292](#)
- nppiDiv_32f_C3IR
 - image_div, [292](#)
- nppiDiv_32f_C3R
 - image_div, [293](#)
- nppiDiv_32f_C4IR
 - image_div, [293](#)
- nppiDiv_32f_C4R
 - image_div, [293](#)
- nppiDiv_32fc_AC4IR
 - image_div, [294](#)
- nppiDiv_32fc_AC4R
 - image_div, [294](#)
- nppiDiv_32fc_C1IR
 - image_div, [295](#)
- nppiDiv_32fc_C1R
 - image_div, [295](#)
- nppiDiv_32fc_C3IR
 - image_div, [295](#)

- nppiDiv_32fc_C3R
image_div, [296](#)
- nppiDiv_32fc_C4IR
image_div, [296](#)
- nppiDiv_32fc_C4R
image_div, [296](#)
- nppiDiv_32s_C1IRSfs
image_div, [297](#)
- nppiDiv_32s_C1R
image_div, [297](#)
- nppiDiv_32s_C1RSfs
image_div, [298](#)
- nppiDiv_32s_C3IRSfs
image_div, [298](#)
- nppiDiv_32s_C3RSfs
image_div, [298](#)
- nppiDiv_32sc_AC4IRSfs
image_div, [299](#)
- nppiDiv_32sc_AC4RSfs
image_div, [299](#)
- nppiDiv_32sc_C1IRSfs
image_div, [300](#)
- nppiDiv_32sc_C1RSfs
image_div, [300](#)
- nppiDiv_32sc_C3IRSfs
image_div, [301](#)
- nppiDiv_32sc_C3RSfs
image_div, [301](#)
- nppiDiv_8u_AC4IRSfs
image_div, [301](#)
- nppiDiv_8u_AC4RSfs
image_div, [302](#)
- nppiDiv_8u_C1IRSfs
image_div, [302](#)
- nppiDiv_8u_C1RSfs
image_div, [303](#)
- nppiDiv_8u_C3IRSfs
image_div, [303](#)
- nppiDiv_8u_C3RSfs
image_div, [303](#)
- nppiDiv_8u_C4IRSfs
image_div, [304](#)
- nppiDiv_8u_C4RSfs
image_div, [304](#)
- nppiDiv_Round_16s_AC4IRSfs
image_divround, [308](#)
- nppiDiv_Round_16s_AC4RSfs
image_divround, [309](#)
- nppiDiv_Round_16s_C1IRSfs
image_divround, [309](#)
- nppiDiv_Round_16s_C1RSfs
image_divround, [310](#)
- nppiDiv_Round_16s_C3IRSfs
image_divround, [310](#)
- nppiDiv_Round_16s_C3RSfs
image_divround, [310](#)
- nppiDiv_Round_16s_C4IRSfs
image_divround, [311](#)
- nppiDiv_Round_16s_C4RSfs
image_divround, [311](#)
- nppiDiv_Round_16u_AC4IRSfs
image_divround, [312](#)
- nppiDiv_Round_16u_AC4RSfs
image_divround, [312](#)
- nppiDiv_Round_16u_C1IRSfs
image_divround, [313](#)
- nppiDiv_Round_16u_C1RSfs
image_divround, [313](#)
- nppiDiv_Round_16u_C3IRSfs
image_divround, [314](#)
- nppiDiv_Round_16u_C3RSfs
image_divround, [314](#)
- nppiDiv_Round_16u_C4IRSfs
image_divround, [315](#)
- nppiDiv_Round_16u_C4RSfs
image_divround, [315](#)
- nppiDiv_Round_8u_AC4IRSfs
image_divround, [316](#)
- nppiDiv_Round_8u_AC4RSfs
image_divround, [316](#)
- nppiDiv_Round_8u_C1IRSfs
image_divround, [317](#)
- nppiDiv_Round_8u_C1RSfs
image_divround, [317](#)
- nppiDiv_Round_8u_C3IRSfs
image_divround, [318](#)
- nppiDiv_Round_8u_C3RSfs
image_divround, [318](#)
- nppiDiv_Round_8u_C4IRSfs
image_divround, [319](#)
- nppiDiv_Round_8u_C4RSfs
image_divround, [319](#)
- nppiDivC_16s_AC4IRSfs
image_divc, [146](#)
- nppiDivC_16s_AC4RSfs
image_divc, [146](#)
- nppiDivC_16s_C1IRSfs
image_divc, [146](#)
- nppiDivC_16s_C1RSfs
image_divc, [147](#)
- nppiDivC_16s_C3IRSfs
image_divc, [147](#)
- nppiDivC_16s_C3RSfs
image_divc, [147](#)
- nppiDivC_16s_C4IRSfs
image_divc, [148](#)
- nppiDivC_16s_C4RSfs
image_divc, [148](#)

- nppiDivC_16sc_AC4IRSfs
 - image_divc, [149](#)
- nppiDivC_16sc_AC4RSfs
 - image_divc, [149](#)
- nppiDivC_16sc_C1IRSfs
 - image_divc, [149](#)
- nppiDivC_16sc_C1RSfs
 - image_divc, [150](#)
- nppiDivC_16sc_C3IRSfs
 - image_divc, [150](#)
- nppiDivC_16sc_C3RSfs
 - image_divc, [151](#)
- nppiDivC_16u_AC4IRSfs
 - image_divc, [151](#)
- nppiDivC_16u_AC4RSfs
 - image_divc, [151](#)
- nppiDivC_16u_C1IRSfs
 - image_divc, [152](#)
- nppiDivC_16u_C1RSfs
 - image_divc, [152](#)
- nppiDivC_16u_C3IRSfs
 - image_divc, [153](#)
- nppiDivC_16u_C3RSfs
 - image_divc, [153](#)
- nppiDivC_16u_C4IRSfs
 - image_divc, [153](#)
- nppiDivC_16u_C4RSfs
 - image_divc, [154](#)
- nppiDivC_32f_AC4IR
 - image_divc, [154](#)
- nppiDivC_32f_AC4R
 - image_divc, [154](#)
- nppiDivC_32f_C1IR
 - image_divc, [155](#)
- nppiDivC_32f_C1R
 - image_divc, [155](#)
- nppiDivC_32f_C3IR
 - image_divc, [155](#)
- nppiDivC_32f_C3R
 - image_divc, [156](#)
- nppiDivC_32f_C4IR
 - image_divc, [156](#)
- nppiDivC_32f_C4R
 - image_divc, [156](#)
- nppiDivC_32fc_AC4IR
 - image_divc, [157](#)
- nppiDivC_32fc_AC4R
 - image_divc, [157](#)
- nppiDivC_32fc_C1IR
 - image_divc, [157](#)
- nppiDivC_32fc_C1R
 - image_divc, [158](#)
- nppiDivC_32fc_C3IR
 - image_divc, [158](#)
- nppiDivC_32fc_C3R
 - image_divc, [158](#)
- nppiDivC_32fc_C4IR
 - image_divc, [159](#)
- nppiDivC_32fc_C4R
 - image_divc, [159](#)
- nppiDivC_32s_C1IRSfs
 - image_divc, [160](#)
- nppiDivC_32s_C1RSfs
 - image_divc, [160](#)
- nppiDivC_32s_C3IRSfs
 - image_divc, [160](#)
- nppiDivC_32s_C3RSfs
 - image_divc, [161](#)
- nppiDivC_32sc_AC4IRSfs
 - image_divc, [161](#)
- nppiDivC_32sc_AC4RSfs
 - image_divc, [161](#)
- nppiDivC_32sc_C1IRSfs
 - image_divc, [162](#)
- nppiDivC_32sc_C1RSfs
 - image_divc, [162](#)
- nppiDivC_32sc_C3IRSfs
 - image_divc, [163](#)
- nppiDivC_32sc_C3RSfs
 - image_divc, [163](#)
- nppiDivC_8u_AC4IRSfs
 - image_divc, [163](#)
- nppiDivC_8u_AC4RSfs
 - image_divc, [164](#)
- nppiDivC_8u_C1IRSfs
 - image_divc, [164](#)
- nppiDivC_8u_C1RSfs
 - image_divc, [165](#)
- nppiDivC_8u_C3IRSfs
 - image_divc, [165](#)
- nppiDivC_8u_C3RSfs
 - image_divc, [165](#)
- nppiDivC_8u_C4IRSfs
 - image_divc, [166](#)
- nppiDivC_8u_C4RSfs
 - image_divc, [166](#)
- nppiDotProd_16s64f_AC4R
 - image_dot_prod, [2047](#)
- nppiDotProd_16s64f_C1R
 - image_dot_prod, [2047](#)
- nppiDotProd_16s64f_C3R
 - image_dot_prod, [2048](#)
- nppiDotProd_16s64f_C4R
 - image_dot_prod, [2048](#)
- nppiDotProd_16u64f_AC4R
 - image_dot_prod, [2049](#)
- nppiDotProd_16u64f_C1R
 - image_dot_prod, [2049](#)

- nppiDotProd_16u64f_C3R
image_dot_prod, [2050](#)
- nppiDotProd_16u64f_C4R
image_dot_prod, [2050](#)
- nppiDotProd_32f64f_AC4R
image_dot_prod, [2050](#)
- nppiDotProd_32f64f_C1R
image_dot_prod, [2051](#)
- nppiDotProd_32f64f_C3R
image_dot_prod, [2051](#)
- nppiDotProd_32f64f_C4R
image_dot_prod, [2052](#)
- nppiDotProd_32s64f_AC4R
image_dot_prod, [2052](#)
- nppiDotProd_32s64f_C1R
image_dot_prod, [2053](#)
- nppiDotProd_32s64f_C3R
image_dot_prod, [2053](#)
- nppiDotProd_32s64f_C4R
image_dot_prod, [2053](#)
- nppiDotProd_32u64f_AC4R
image_dot_prod, [2054](#)
- nppiDotProd_32u64f_C1R
image_dot_prod, [2054](#)
- nppiDotProd_32u64f_C3R
image_dot_prod, [2055](#)
- nppiDotProd_32u64f_C4R
image_dot_prod, [2055](#)
- nppiDotProd_8s64f_AC4R
image_dot_prod, [2056](#)
- nppiDotProd_8s64f_C1R
image_dot_prod, [2056](#)
- nppiDotProd_8s64f_C3R
image_dot_prod, [2056](#)
- nppiDotProd_8s64f_C4R
image_dot_prod, [2057](#)
- nppiDotProd_8u64f_AC4R
image_dot_prod, [2057](#)
- nppiDotProd_8u64f_C1R
image_dot_prod, [2058](#)
- nppiDotProd_8u64f_C3R
image_dot_prod, [2058](#)
- nppiDotProd_8u64f_C4R
image_dot_prod, [2059](#)
- nppiDotProdGetBufferHostSize_16s64f_AC4R
image_dot_prod, [2059](#)
- nppiDotProdGetBufferHostSize_16s64f_C1R
image_dot_prod, [2059](#)
- nppiDotProdGetBufferHostSize_16s64f_C3R
image_dot_prod, [2060](#)
- nppiDotProdGetBufferHostSize_16s64f_C4R
image_dot_prod, [2060](#)
- nppiDotProdGetBufferHostSize_16u64f_AC4R
image_dot_prod, [2060](#)
- nppiDotProdGetBufferHostSize_16u64f_C1R
image_dot_prod, [2060](#)
- nppiDotProdGetBufferHostSize_16u64f_C3R
image_dot_prod, [2061](#)
- nppiDotProdGetBufferHostSize_16u64f_C4R
image_dot_prod, [2061](#)
- nppiDotProdGetBufferHostSize_32f64f_AC4R
image_dot_prod, [2061](#)
- nppiDotProdGetBufferHostSize_32f64f_C1R
image_dot_prod, [2062](#)
- nppiDotProdGetBufferHostSize_32f64f_C3R
image_dot_prod, [2062](#)
- nppiDotProdGetBufferHostSize_32f64f_C4R
image_dot_prod, [2062](#)
- nppiDotProdGetBufferHostSize_32s64f_AC4R
image_dot_prod, [2062](#)
- nppiDotProdGetBufferHostSize_32s64f_C1R
image_dot_prod, [2063](#)
- nppiDotProdGetBufferHostSize_32s64f_C3R
image_dot_prod, [2063](#)
- nppiDotProdGetBufferHostSize_32s64f_C4R
image_dot_prod, [2063](#)
- nppiDotProdGetBufferHostSize_32u64f_AC4R
image_dot_prod, [2064](#)
- nppiDotProdGetBufferHostSize_32u64f_C1R
image_dot_prod, [2064](#)
- nppiDotProdGetBufferHostSize_32u64f_C3R
image_dot_prod, [2064](#)
- nppiDotProdGetBufferHostSize_32u64f_C4R
image_dot_prod, [2064](#)
- nppiDotProdGetBufferHostSize_8s64f_AC4R
image_dot_prod, [2065](#)
- nppiDotProdGetBufferHostSize_8s64f_C1R
image_dot_prod, [2065](#)
- nppiDotProdGetBufferHostSize_8s64f_C3R
image_dot_prod, [2065](#)
- nppiDotProdGetBufferHostSize_8s64f_C4R
image_dot_prod, [2066](#)
- nppiDotProdGetBufferHostSize_8u64f_AC4R
image_dot_prod, [2066](#)
- nppiDotProdGetBufferHostSize_8u64f_C1R
image_dot_prod, [2066](#)
- nppiDotProdGetBufferHostSize_8u64f_C3R
image_dot_prod, [2066](#)
- nppiDotProdGetBufferHostSize_8u64f_C4R
image_dot_prod, [2067](#)
- nppiDup_16s_C1AC4R
image_duplicate_channel, [929](#)
- nppiDup_16s_C1C3R
image_duplicate_channel, [929](#)
- nppiDup_16s_C1C4R
image_duplicate_channel, [930](#)
- nppiDup_16u_C1AC4R
image_duplicate_channel, [930](#)

- npDppDup_16u_C1C3R
 - image_duplicate_channel, 930
- npDppDup_16u_C1C4R
 - image_duplicate_channel, 931
- npDppDup_32f_C1AC4R
 - image_duplicate_channel, 931
- npDppDup_32f_C1C3R
 - image_duplicate_channel, 931
- npDppDup_32f_C1C4R
 - image_duplicate_channel, 932
- npDppDup_32s_C1AC4R
 - image_duplicate_channel, 932
- npDppDup_32s_C1C3R
 - image_duplicate_channel, 932
- npDppDup_32s_C1C4R
 - image_duplicate_channel, 933
- npDppDup_8u_C1AC4R
 - image_duplicate_channel, 933
- npDppDup_8u_C1C3R
 - image_duplicate_channel, 933
- npDppDup_8u_C1C4R
 - image_duplicate_channel, 934
- npDppErode3x3_16u_AC4R
 - image_erode_3x3, 1623
- npDppErode3x3_16u_C1R
 - image_erode_3x3, 1623
- npDppErode3x3_16u_C3R
 - image_erode_3x3, 1623
- npDppErode3x3_16u_C4R
 - image_erode_3x3, 1624
- npDppErode3x3_32f_AC4R
 - image_erode_3x3, 1624
- npDppErode3x3_32f_C1R
 - image_erode_3x3, 1624
- npDppErode3x3_32f_C3R
 - image_erode_3x3, 1625
- npDppErode3x3_32f_C4R
 - image_erode_3x3, 1625
- npDppErode3x3_64f_C1R
 - image_erode_3x3, 1625
- npDppErode3x3_8u_AC4R
 - image_erode_3x3, 1626
- npDppErode3x3_8u_C1R
 - image_erode_3x3, 1626
- npDppErode3x3_8u_C3R
 - image_erode_3x3, 1626
- npDppErode3x3_8u_C4R
 - image_erode_3x3, 1627
- npDppErode3x3Border_16u_AC4R
 - image_erode_3x3_border, 1629
- npDppErode3x3Border_16u_C1R
 - image_erode_3x3_border, 1629
- npDppErode3x3Border_16u_C3R
 - image_erode_3x3_border, 1630
- npDppErode3x3Border_16u_C4R
 - image_erode_3x3_border, 1630
- npDppErode3x3Border_32f_AC4R
 - image_erode_3x3_border, 1631
- npDppErode3x3Border_32f_C1R
 - image_erode_3x3_border, 1631
- npDppErode3x3Border_32f_C3R
 - image_erode_3x3_border, 1632
- npDppErode3x3Border_32f_C4R
 - image_erode_3x3_border, 1632
- npDppErode3x3Border_8u_AC4R
 - image_erode_3x3_border, 1632
- npDppErode3x3Border_8u_C1R
 - image_erode_3x3_border, 1633
- npDppErode3x3Border_8u_C3R
 - image_erode_3x3_border, 1633
- npDppErode3x3Border_8u_C4R
 - image_erode_3x3_border, 1634
- npDppErode_16u_AC4R
 - image_erode, 1608
- npDppErode_16u_C1R
 - image_erode, 1608
- npDppErode_16u_C3R
 - image_erode, 1609
- npDppErode_16u_C4R
 - image_erode, 1609
- npDppErode_32f_AC4R
 - image_erode, 1609
- npDppErode_32f_C1R
 - image_erode, 1610
- npDppErode_32f_C3R
 - image_erode, 1610
- npDppErode_32f_C4R
 - image_erode, 1611
- npDppErode_8u_AC4R
 - image_erode, 1611
- npDppErode_8u_C1R
 - image_erode, 1612
- npDppErode_8u_C3R
 - image_erode, 1612
- npDppErode_8u_C4R
 - image_erode, 1612
- npDppErodeBorder_16u_AC4R
 - image_erode_border, 1615
- npDppErodeBorder_16u_C1R
 - image_erode_border, 1616
- npDppErodeBorder_16u_C3R
 - image_erode_border, 1616
- npDppErodeBorder_16u_C4R
 - image_erode_border, 1617
- npDppErodeBorder_32f_AC4R
 - image_erode_border, 1617
- npDppErodeBorder_32f_C1R
 - image_erode_border, 1618

- npErodeBorder_32f_C3R
 - image_erode_border, 1618
- npErodeBorder_32f_C4R
 - image_erode_border, 1619
- npErodeBorder_8u_AC4R
 - image_erode_border, 1619
- npErodeBorder_8u_C1R
 - image_erode_border, 1620
- npErodeBorder_8u_C3R
 - image_erode_border, 1620
- npErodeBorder_8u_C4R
 - image_erode_border, 1621
- npEvenLevelsHost_32s
 - image_histogrameven, 2098
- npExp_16s_C1RSfs
 - image_exp, 365
- npExp_16s_C1RSfs
 - image_exp, 365
- npExp_16s_C3RSfs
 - image_exp, 366
- npExp_16s_C3RSfs
 - image_exp, 366
- npExp_16u_C1RSfs
 - image_exp, 366
- npExp_16u_C1RSfs
 - image_exp, 367
- npExp_16u_C3RSfs
 - image_exp, 367
- npExp_16u_C3RSfs
 - image_exp, 367
- npExp_32f_C1IR
 - image_exp, 368
- npExp_32f_C1R
 - image_exp, 368
- npExp_32f_C3IR
 - image_exp, 368
- npExp_32f_C3R
 - image_exp, 369
- npExp_8u_C1RSfs
 - image_exp, 369
- npExp_8u_C1RSfs
 - image_exp, 369
- npExp_8u_C3RSfs
 - image_exp, 370
- npExp_8u_C3RSfs
 - image_exp, 370
- npFilter32f_16s_AC4R
 - image_convolution, 1230
- npFilter32f_16s_C1R
 - image_convolution, 1231
- npFilter32f_16s_C3R
 - image_convolution, 1231
- npFilter32f_16s_C4R
 - image_convolution, 1231
- npFilter32f_16u_AC4R
 - image_convolution, 1232
- npFilter32f_16u_C1R
 - image_convolution, 1232
- npFilter32f_16u_C3R
 - image_convolution, 1233
- npFilter32f_16u_C4R
 - image_convolution, 1233
- npFilter32f_32s_AC4R
 - image_convolution, 1234
- npFilter32f_32s_C1R
 - image_convolution, 1234
- npFilter32f_32s_C3R
 - image_convolution, 1235
- npFilter32f_32s_C4R
 - image_convolution, 1235
- npFilter32f_8s16s_AC4R
 - image_convolution, 1236
- npFilter32f_8s16s_C1R
 - image_convolution, 1236
- npFilter32f_8s16s_C3R
 - image_convolution, 1237
- npFilter32f_8s16s_C4R
 - image_convolution, 1237
- npFilter32f_8s_AC4R
 - image_convolution, 1238
- npFilter32f_8s_C1R
 - image_convolution, 1238
- npFilter32f_8s_C2R
 - image_convolution, 1239
- npFilter32f_8s_C3R
 - image_convolution, 1239
- npFilter32f_8s_C4R
 - image_convolution, 1240
- npFilter32f_8u16s_AC4R
 - image_convolution, 1240
- npFilter32f_8u16s_C1R
 - image_convolution, 1241
- npFilter32f_8u16s_C3R
 - image_convolution, 1241
- npFilter32f_8u16s_C4R
 - image_convolution, 1242
- npFilter32f_8u_AC4R
 - image_convolution, 1242
- npFilter32f_8u_C1R
 - image_convolution, 1243
- npFilter32f_8u_C2R
 - image_convolution, 1243
- npFilter32f_8u_C3R
 - image_convolution, 1244
- npFilter32f_8u_C4R
 - image_convolution, 1244
- npFilter_16s_AC4R
 - image_convolution, 1245

- nppiFilter_16s_C1R
 - image_convolution, [1245](#)
- nppiFilter_16s_C3R
 - image_convolution, [1246](#)
- nppiFilter_16s_C4R
 - image_convolution, [1246](#)
- nppiFilter_16u_AC4R
 - image_convolution, [1247](#)
- nppiFilter_16u_C1R
 - image_convolution, [1247](#)
- nppiFilter_16u_C3R
 - image_convolution, [1248](#)
- nppiFilter_16u_C4R
 - image_convolution, [1248](#)
- nppiFilter_32f_AC4R
 - image_convolution, [1249](#)
- nppiFilter_32f_C1R
 - image_convolution, [1249](#)
- nppiFilter_32f_C2R
 - image_convolution, [1250](#)
- nppiFilter_32f_C3R
 - image_convolution, [1250](#)
- nppiFilter_32f_C4R
 - image_convolution, [1251](#)
- nppiFilter_64f_C1R
 - image_convolution, [1251](#)
- nppiFilter_8u_AC4R
 - image_convolution, [1252](#)
- nppiFilter_8u_C1R
 - image_convolution, [1252](#)
- nppiFilter_8u_C3R
 - image_convolution, [1253](#)
- nppiFilter_8u_C4R
 - image_convolution, [1253](#)
- nppiFilterBorder32f_16s_AC4R
 - image_convolution, [1254](#)
- nppiFilterBorder32f_16s_C1R
 - image_convolution, [1254](#)
- nppiFilterBorder32f_16s_C3R
 - image_convolution, [1255](#)
- nppiFilterBorder32f_16s_C4R
 - image_convolution, [1255](#)
- nppiFilterBorder32f_16u_AC4R
 - image_convolution, [1256](#)
- nppiFilterBorder32f_16u_C1R
 - image_convolution, [1256](#)
- nppiFilterBorder32f_16u_C3R
 - image_convolution, [1257](#)
- nppiFilterBorder32f_16u_C4R
 - image_convolution, [1257](#)
- nppiFilterBorder32f_32s_AC4R
 - image_convolution, [1258](#)
- nppiFilterBorder32f_32s_C1R
 - image_convolution, [1258](#)
- nppiFilterBorder32f_32s_C3R
 - image_convolution, [1259](#)
- nppiFilterBorder32f_32s_C4R
 - image_convolution, [1259](#)
- nppiFilterBorder32f_8s16s_AC4R
 - image_convolution, [1260](#)
- nppiFilterBorder32f_8s16s_C1R
 - image_convolution, [1260](#)
- nppiFilterBorder32f_8s16s_C3R
 - image_convolution, [1261](#)
- nppiFilterBorder32f_8s16s_C4R
 - image_convolution, [1261](#)
- nppiFilterBorder32f_8s_AC4R
 - image_convolution, [1262](#)
- nppiFilterBorder32f_8s_C1R
 - image_convolution, [1262](#)
- nppiFilterBorder32f_8s_C2R
 - image_convolution, [1263](#)
- nppiFilterBorder32f_8s_C3R
 - image_convolution, [1263](#)
- nppiFilterBorder32f_8s_C4R
 - image_convolution, [1264](#)
- nppiFilterBorder32f_8u16s_AC4R
 - image_convolution, [1264](#)
- nppiFilterBorder32f_8u16s_C1R
 - image_convolution, [1265](#)
- nppiFilterBorder32f_8u16s_C3R
 - image_convolution, [1265](#)
- nppiFilterBorder32f_8u16s_C4R
 - image_convolution, [1266](#)
- nppiFilterBorder32f_8u_AC4R
 - image_convolution, [1266](#)
- nppiFilterBorder32f_8u_C1R
 - image_convolution, [1267](#)
- nppiFilterBorder32f_8u_C2R
 - image_convolution, [1267](#)
- nppiFilterBorder32f_8u_C3R
 - image_convolution, [1268](#)
- nppiFilterBorder32f_8u_C4R
 - image_convolution, [1268](#)
- nppiFilterBorder_16s_AC4R
 - image_convolution, [1269](#)
- nppiFilterBorder_16s_C1R
 - image_convolution, [1270](#)
- nppiFilterBorder_16s_C3R
 - image_convolution, [1270](#)
- nppiFilterBorder_16s_C4R
 - image_convolution, [1271](#)
- nppiFilterBorder_16u_AC4R
 - image_convolution, [1271](#)
- nppiFilterBorder_16u_C1R
 - image_convolution, [1272](#)
- nppiFilterBorder_16u_C3R
 - image_convolution, [1273](#)

nppiFilterBorder_16u_C4R
 image_convolution, 1273
 nppiFilterBorder_32f_AC4R
 image_convolution, 1274
 nppiFilterBorder_32f_C1R
 image_convolution, 1274
 nppiFilterBorder_32f_C2R
 image_convolution, 1275
 nppiFilterBorder_32f_C3R
 image_convolution, 1275
 nppiFilterBorder_32f_C4R
 image_convolution, 1276
 nppiFilterBorder_8u_AC4R
 image_convolution, 1276
 nppiFilterBorder_8u_C1R
 image_convolution, 1277
 nppiFilterBorder_8u_C3R
 image_convolution, 1278
 nppiFilterBorder_8u_C4R
 image_convolution, 1278
 nppiFilterBox_16s_AC4R
 image_2D_fixed_linear_filters, 1283
 nppiFilterBox_16s_C1R
 image_2D_fixed_linear_filters, 1283
 nppiFilterBox_16s_C3R
 image_2D_fixed_linear_filters, 1283
 nppiFilterBox_16s_C4R
 image_2D_fixed_linear_filters, 1284
 nppiFilterBox_16u_AC4R
 image_2D_fixed_linear_filters, 1284
 nppiFilterBox_16u_C1R
 image_2D_fixed_linear_filters, 1285
 nppiFilterBox_16u_C3R
 image_2D_fixed_linear_filters, 1285
 nppiFilterBox_16u_C4R
 image_2D_fixed_linear_filters, 1285
 nppiFilterBox_32f_AC4R
 image_2D_fixed_linear_filters, 1286
 nppiFilterBox_32f_C1R
 image_2D_fixed_linear_filters, 1286
 nppiFilterBox_32f_C3R
 image_2D_fixed_linear_filters, 1287
 nppiFilterBox_32f_C4R
 image_2D_fixed_linear_filters, 1287
 nppiFilterBox_64f_C1R
 image_2D_fixed_linear_filters, 1287
 nppiFilterBox_8u_AC4R
 image_2D_fixed_linear_filters, 1288
 nppiFilterBox_8u_C1R
 image_2D_fixed_linear_filters, 1288
 nppiFilterBox_8u_C3R
 image_2D_fixed_linear_filters, 1289
 nppiFilterBox_8u_C4R
 image_2D_fixed_linear_filters, 1289

nppiFilterBoxBorder_16s_AC4R
 image_2D_fixed_linear_filters, 1289
 nppiFilterBoxBorder_16s_C1R
 image_2D_fixed_linear_filters, 1290
 nppiFilterBoxBorder_16s_C3R
 image_2D_fixed_linear_filters, 1290
 nppiFilterBoxBorder_16s_C4R
 image_2D_fixed_linear_filters, 1291
 nppiFilterBoxBorder_16u_AC4R
 image_2D_fixed_linear_filters, 1291
 nppiFilterBoxBorder_16u_C1R
 image_2D_fixed_linear_filters, 1292
 nppiFilterBoxBorder_16u_C3R
 image_2D_fixed_linear_filters, 1292
 nppiFilterBoxBorder_16u_C4R
 image_2D_fixed_linear_filters, 1293
 nppiFilterBoxBorder_32f_AC4R
 image_2D_fixed_linear_filters, 1293
 nppiFilterBoxBorder_32f_C1R
 image_2D_fixed_linear_filters, 1294
 nppiFilterBoxBorder_32f_C3R
 image_2D_fixed_linear_filters, 1294
 nppiFilterBoxBorder_32f_C4R
 image_2D_fixed_linear_filters, 1295
 nppiFilterBoxBorder_8u_AC4R
 image_2D_fixed_linear_filters, 1295
 nppiFilterBoxBorder_8u_C1R
 image_2D_fixed_linear_filters, 1296
 nppiFilterBoxBorder_8u_C3R
 image_2D_fixed_linear_filters, 1296
 nppiFilterBoxBorder_8u_C4R
 image_2D_fixed_linear_filters, 1297
 nppiFilterColumn32f_16s_AC4R
 image_1D_linear_filter, 1120
 nppiFilterColumn32f_16s_C1R
 image_1D_linear_filter, 1120
 nppiFilterColumn32f_16s_C3R
 image_1D_linear_filter, 1121
 nppiFilterColumn32f_16s_C4R
 image_1D_linear_filter, 1121
 nppiFilterColumn32f_16u_AC4R
 image_1D_linear_filter, 1122
 nppiFilterColumn32f_16u_C1R
 image_1D_linear_filter, 1122
 nppiFilterColumn32f_16u_C3R
 image_1D_linear_filter, 1123
 nppiFilterColumn32f_16u_C4R
 image_1D_linear_filter, 1123
 nppiFilterColumn32f_8u_AC4R
 image_1D_linear_filter, 1124
 nppiFilterColumn32f_8u_C1R
 image_1D_linear_filter, 1124
 nppiFilterColumn32f_8u_C3R
 image_1D_linear_filter, 1125

- nppiFilterColumn32f_8u_C4R
 - image_1D_linear_filter, [1125](#)
- nppiFilterColumn_16s_AC4R
 - image_1D_linear_filter, [1126](#)
- nppiFilterColumn_16s_C1R
 - image_1D_linear_filter, [1126](#)
- nppiFilterColumn_16s_C3R
 - image_1D_linear_filter, [1127](#)
- nppiFilterColumn_16s_C4R
 - image_1D_linear_filter, [1127](#)
- nppiFilterColumn_16u_AC4R
 - image_1D_linear_filter, [1128](#)
- nppiFilterColumn_16u_C1R
 - image_1D_linear_filter, [1128](#)
- nppiFilterColumn_16u_C3R
 - image_1D_linear_filter, [1129](#)
- nppiFilterColumn_16u_C4R
 - image_1D_linear_filter, [1129](#)
- nppiFilterColumn_32f_AC4R
 - image_1D_linear_filter, [1130](#)
- nppiFilterColumn_32f_C1R
 - image_1D_linear_filter, [1130](#)
- nppiFilterColumn_32f_C3R
 - image_1D_linear_filter, [1131](#)
- nppiFilterColumn_32f_C4R
 - image_1D_linear_filter, [1131](#)
- nppiFilterColumn_64f_C1R
 - image_1D_linear_filter, [1132](#)
- nppiFilterColumn_8u_AC4R
 - image_1D_linear_filter, [1132](#)
- nppiFilterColumn_8u_C1R
 - image_1D_linear_filter, [1133](#)
- nppiFilterColumn_8u_C3R
 - image_1D_linear_filter, [1133](#)
- nppiFilterColumn_8u_C4R
 - image_1D_linear_filter, [1134](#)
- nppiFilterColumnBorder32f_16s_AC4R
 - image_1D_linear_filter, [1134](#)
- nppiFilterColumnBorder32f_16s_C1R
 - image_1D_linear_filter, [1135](#)
- nppiFilterColumnBorder32f_16s_C3R
 - image_1D_linear_filter, [1135](#)
- nppiFilterColumnBorder32f_16s_C4R
 - image_1D_linear_filter, [1136](#)
- nppiFilterColumnBorder32f_16u_AC4R
 - image_1D_linear_filter, [1136](#)
- nppiFilterColumnBorder32f_16u_C1R
 - image_1D_linear_filter, [1137](#)
- nppiFilterColumnBorder32f_16u_C3R
 - image_1D_linear_filter, [1137](#)
- nppiFilterColumnBorder32f_16u_C4R
 - image_1D_linear_filter, [1138](#)
- nppiFilterColumnBorder32f_8u_AC4R
 - image_1D_linear_filter, [1138](#)
- nppiFilterColumnBorder32f_8u_C1R
 - image_1D_linear_filter, [1139](#)
- nppiFilterColumnBorder32f_8u_C3R
 - image_1D_linear_filter, [1139](#)
- nppiFilterColumnBorder32f_8u_C4R
 - image_1D_linear_filter, [1140](#)
- nppiFilterColumnBorder_16s_AC4R
 - image_1D_linear_filter, [1140](#)
- nppiFilterColumnBorder_16s_C1R
 - image_1D_linear_filter, [1141](#)
- nppiFilterColumnBorder_16s_C3R
 - image_1D_linear_filter, [1142](#)
- nppiFilterColumnBorder_16s_C4R
 - image_1D_linear_filter, [1142](#)
- nppiFilterColumnBorder_16u_AC4R
 - image_1D_linear_filter, [1143](#)
- nppiFilterColumnBorder_16u_C1R
 - image_1D_linear_filter, [1143](#)
- nppiFilterColumnBorder_16u_C3R
 - image_1D_linear_filter, [1144](#)
- nppiFilterColumnBorder_16u_C4R
 - image_1D_linear_filter, [1145](#)
- nppiFilterColumnBorder_32f_AC4R
 - image_1D_linear_filter, [1145](#)
- nppiFilterColumnBorder_32f_C1R
 - image_1D_linear_filter, [1146](#)
- nppiFilterColumnBorder_32f_C3R
 - image_1D_linear_filter, [1146](#)
- nppiFilterColumnBorder_32f_C4R
 - image_1D_linear_filter, [1147](#)
- nppiFilterColumnBorder_8u_AC4R
 - image_1D_linear_filter, [1147](#)
- nppiFilterColumnBorder_8u_C1R
 - image_1D_linear_filter, [1148](#)
- nppiFilterColumnBorder_8u_C3R
 - image_1D_linear_filter, [1149](#)
- nppiFilterColumnBorder_8u_C4R
 - image_1D_linear_filter, [1149](#)
- nppiFilterGauss_16s_AC4R
 - image_filtering_functions, [989](#)
- nppiFilterGauss_16s_C1R
 - image_filtering_functions, [989](#)
- nppiFilterGauss_16s_C3R
 - image_filtering_functions, [989](#)
- nppiFilterGauss_16s_C4R
 - image_filtering_functions, [990](#)
- nppiFilterGauss_16u_AC4R
 - image_filtering_functions, [990](#)
- nppiFilterGauss_16u_C1R
 - image_filtering_functions, [990](#)
- nppiFilterGauss_16u_C3R
 - image_filtering_functions, [991](#)
- nppiFilterGauss_16u_C4R
 - image_filtering_functions, [991](#)

- npippiFilterGauss_32f_AC4R
 - image_filtering_functions, [991](#)
- npippiFilterGauss_32f_C1R
 - image_filtering_functions, [992](#)
- npippiFilterGauss_32f_C3R
 - image_filtering_functions, [992](#)
- npippiFilterGauss_32f_C4R
 - image_filtering_functions, [992](#)
- npippiFilterGauss_8u_AC4R
 - image_filtering_functions, [993](#)
- npippiFilterGauss_8u_C1R
 - image_filtering_functions, [993](#)
- npippiFilterGauss_8u_C3R
 - image_filtering_functions, [993](#)
- npippiFilterGauss_8u_C4R
 - image_filtering_functions, [994](#)
- npippiFilterGaussAdvanced_16s_AC4R
 - image_filtering_functions, [994](#)
- npippiFilterGaussAdvanced_16s_C1R
 - image_filtering_functions, [995](#)
- npippiFilterGaussAdvanced_16s_C3R
 - image_filtering_functions, [995](#)
- npippiFilterGaussAdvanced_16s_C4R
 - image_filtering_functions, [995](#)
- npippiFilterGaussAdvanced_16u_AC4R
 - image_filtering_functions, [996](#)
- npippiFilterGaussAdvanced_16u_C1R
 - image_filtering_functions, [996](#)
- npippiFilterGaussAdvanced_16u_C3R
 - image_filtering_functions, [997](#)
- npippiFilterGaussAdvanced_16u_C4R
 - image_filtering_functions, [997](#)
- npippiFilterGaussAdvanced_32f_AC4R
 - image_filtering_functions, [997](#)
- npippiFilterGaussAdvanced_32f_C1R
 - image_filtering_functions, [998](#)
- npippiFilterGaussAdvanced_32f_C3R
 - image_filtering_functions, [998](#)
- npippiFilterGaussAdvanced_32f_C4R
 - image_filtering_functions, [999](#)
- npippiFilterGaussAdvanced_8u_AC4R
 - image_filtering_functions, [999](#)
- npippiFilterGaussAdvanced_8u_C1R
 - image_filtering_functions, [999](#)
- npippiFilterGaussAdvanced_8u_C3R
 - image_filtering_functions, [1000](#)
- npippiFilterGaussAdvanced_8u_C4R
 - image_filtering_functions, [1000](#)
- npippiFilterGaussAdvancedBorder_16s_AC4R
 - image_filtering_functions, [1001](#)
- npippiFilterGaussAdvancedBorder_16s_C1R
 - image_filtering_functions, [1001](#)
- npippiFilterGaussAdvancedBorder_16s_C3R
 - image_filtering_functions, [1002](#)
- npippiFilterGaussAdvancedBorder_16s_C4R
 - image_filtering_functions, [1002](#)
- npippiFilterGaussAdvancedBorder_16u_AC4R
 - image_filtering_functions, [1003](#)
- npippiFilterGaussAdvancedBorder_16u_C1R
 - image_filtering_functions, [1003](#)
- npippiFilterGaussAdvancedBorder_16u_C3R
 - image_filtering_functions, [1004](#)
- npippiFilterGaussAdvancedBorder_16u_C4R
 - image_filtering_functions, [1004](#)
- npippiFilterGaussAdvancedBorder_32f_AC4R
 - image_filtering_functions, [1005](#)
- npippiFilterGaussAdvancedBorder_32f_C1R
 - image_filtering_functions, [1005](#)
- npippiFilterGaussAdvancedBorder_32f_C3R
 - image_filtering_functions, [1006](#)
- npippiFilterGaussAdvancedBorder_32f_C4R
 - image_filtering_functions, [1006](#)
- npippiFilterGaussAdvancedBorder_8u_AC4R
 - image_filtering_functions, [1007](#)
- npippiFilterGaussAdvancedBorder_8u_C1R
 - image_filtering_functions, [1007](#)
- npippiFilterGaussAdvancedBorder_8u_C3R
 - image_filtering_functions, [1008](#)
- npippiFilterGaussAdvancedBorder_8u_C4R
 - image_filtering_functions, [1008](#)
- npippiFilterGaussBorder_16s_AC4R
 - image_filtering_functions, [1009](#)
- npippiFilterGaussBorder_16s_C1R
 - image_filtering_functions, [1009](#)
- npippiFilterGaussBorder_16s_C3R
 - image_filtering_functions, [1010](#)
- npippiFilterGaussBorder_16s_C4R
 - image_filtering_functions, [1010](#)
- npippiFilterGaussBorder_16u_AC4R
 - image_filtering_functions, [1011](#)
- npippiFilterGaussBorder_16u_C1R
 - image_filtering_functions, [1011](#)
- npippiFilterGaussBorder_16u_C3R
 - image_filtering_functions, [1011](#)
- npippiFilterGaussBorder_16u_C4R
 - image_filtering_functions, [1012](#)
- npippiFilterGaussBorder_32f_AC4R
 - image_filtering_functions, [1012](#)
- npippiFilterGaussBorder_32f_C1R
 - image_filtering_functions, [1013](#)
- npippiFilterGaussBorder_32f_C3R
 - image_filtering_functions, [1013](#)
- npippiFilterGaussBorder_32f_C4R
 - image_filtering_functions, [1014](#)
- npippiFilterGaussBorder_8u_AC4R
 - image_filtering_functions, [1014](#)
- npippiFilterGaussBorder_8u_C1R
 - image_filtering_functions, [1015](#)

- nppiFilterGaussBorder_8u_C3R
 - image_filtering_functions, [1015](#)
- nppiFilterGaussBorder_8u_C4R
 - image_filtering_functions, [1016](#)
- nppiFilterHighPass_16s_AC4R
 - image_filtering_functions, [1016](#)
- nppiFilterHighPass_16s_C1R
 - image_filtering_functions, [1017](#)
- nppiFilterHighPass_16s_C3R
 - image_filtering_functions, [1017](#)
- nppiFilterHighPass_16s_C4R
 - image_filtering_functions, [1017](#)
- nppiFilterHighPass_16u_AC4R
 - image_filtering_functions, [1018](#)
- nppiFilterHighPass_16u_C1R
 - image_filtering_functions, [1018](#)
- nppiFilterHighPass_16u_C3R
 - image_filtering_functions, [1018](#)
- nppiFilterHighPass_16u_C4R
 - image_filtering_functions, [1019](#)
- nppiFilterHighPass_32f_AC4R
 - image_filtering_functions, [1019](#)
- nppiFilterHighPass_32f_C1R
 - image_filtering_functions, [1019](#)
- nppiFilterHighPass_32f_C3R
 - image_filtering_functions, [1020](#)
- nppiFilterHighPass_32f_C4R
 - image_filtering_functions, [1020](#)
- nppiFilterHighPass_8u_AC4R
 - image_filtering_functions, [1020](#)
- nppiFilterHighPass_8u_C1R
 - image_filtering_functions, [1021](#)
- nppiFilterHighPass_8u_C3R
 - image_filtering_functions, [1021](#)
- nppiFilterHighPass_8u_C4R
 - image_filtering_functions, [1021](#)
- nppiFilterHighPassBorder_16s_AC4R
 - image_filtering_functions, [1022](#)
- nppiFilterHighPassBorder_16s_C1R
 - image_filtering_functions, [1022](#)
- nppiFilterHighPassBorder_16s_C3R
 - image_filtering_functions, [1023](#)
- nppiFilterHighPassBorder_16s_C4R
 - image_filtering_functions, [1023](#)
- nppiFilterHighPassBorder_16u_AC4R
 - image_filtering_functions, [1024](#)
- nppiFilterHighPassBorder_16u_C1R
 - image_filtering_functions, [1024](#)
- nppiFilterHighPassBorder_16u_C3R
 - image_filtering_functions, [1025](#)
- nppiFilterHighPassBorder_16u_C4R
 - image_filtering_functions, [1025](#)
- nppiFilterHighPassBorder_32f_AC4R
 - image_filtering_functions, [1025](#)
- nppiFilterHighPassBorder_32f_C1R
 - image_filtering_functions, [1026](#)
- nppiFilterHighPassBorder_32f_C3R
 - image_filtering_functions, [1026](#)
- nppiFilterHighPassBorder_32f_C4R
 - image_filtering_functions, [1027](#)
- nppiFilterHighPassBorder_8u_AC4R
 - image_filtering_functions, [1027](#)
- nppiFilterHighPassBorder_8u_C1R
 - image_filtering_functions, [1028](#)
- nppiFilterHighPassBorder_8u_C3R
 - image_filtering_functions, [1028](#)
- nppiFilterHighPassBorder_8u_C4R
 - image_filtering_functions, [1029](#)
- nppiFilterLaplace_16s_AC4R
 - image_filtering_functions, [1029](#)
- nppiFilterLaplace_16s_C1R
 - image_filtering_functions, [1030](#)
- nppiFilterLaplace_16s_C3R
 - image_filtering_functions, [1030](#)
- nppiFilterLaplace_16s_C4R
 - image_filtering_functions, [1030](#)
- nppiFilterLaplace_32f_AC4R
 - image_filtering_functions, [1031](#)
- nppiFilterLaplace_32f_C1R
 - image_filtering_functions, [1031](#)
- nppiFilterLaplace_32f_C3R
 - image_filtering_functions, [1031](#)
- nppiFilterLaplace_32f_C4R
 - image_filtering_functions, [1032](#)
- nppiFilterLaplace_8s16s_C1R
 - image_filtering_functions, [1032](#)
- nppiFilterLaplace_8u16s_C1R
 - image_filtering_functions, [1032](#)
- nppiFilterLaplace_8u_AC4R
 - image_filtering_functions, [1033](#)
- nppiFilterLaplace_8u_C1R
 - image_filtering_functions, [1033](#)
- nppiFilterLaplace_8u_C3R
 - image_filtering_functions, [1033](#)
- nppiFilterLaplace_8u_C4R
 - image_filtering_functions, [1034](#)
- nppiFilterLaplaceBorder_16s_AC4R
 - image_filtering_functions, [1034](#)
- nppiFilterLaplaceBorder_16s_C1R
 - image_filtering_functions, [1035](#)
- nppiFilterLaplaceBorder_16s_C3R
 - image_filtering_functions, [1035](#)
- nppiFilterLaplaceBorder_16s_C4R
 - image_filtering_functions, [1036](#)
- nppiFilterLaplaceBorder_32f_AC4R
 - image_filtering_functions, [1036](#)
- nppiFilterLaplaceBorder_32f_C1R
 - image_filtering_functions, [1036](#)

- [nppiFilterLaplaceBorder_32f_C3R](#)
 - [image_filtering_functions, 1037](#)
- [nppiFilterLaplaceBorder_32f_C4R](#)
 - [image_filtering_functions, 1037](#)
- [nppiFilterLaplaceBorder_8s16s_C1R](#)
 - [image_filtering_functions, 1038](#)
- [nppiFilterLaplaceBorder_8u16s_C1R](#)
 - [image_filtering_functions, 1038](#)
- [nppiFilterLaplaceBorder_8u_AC4R](#)
 - [image_filtering_functions, 1039](#)
- [nppiFilterLaplaceBorder_8u_C1R](#)
 - [image_filtering_functions, 1039](#)
- [nppiFilterLaplaceBorder_8u_C3R](#)
 - [image_filtering_functions, 1040](#)
- [nppiFilterLaplaceBorder_8u_C4R](#)
 - [image_filtering_functions, 1040](#)
- [nppiFilterLowPass_16s_AC4R](#)
 - [image_filtering_functions, 1041](#)
- [nppiFilterLowPass_16s_C1R](#)
 - [image_filtering_functions, 1041](#)
- [nppiFilterLowPass_16s_C3R](#)
 - [image_filtering_functions, 1042](#)
- [nppiFilterLowPass_16s_C4R](#)
 - [image_filtering_functions, 1042](#)
- [nppiFilterLowPass_16u_AC4R](#)
 - [image_filtering_functions, 1042](#)
- [nppiFilterLowPass_16u_C1R](#)
 - [image_filtering_functions, 1043](#)
- [nppiFilterLowPass_16u_C3R](#)
 - [image_filtering_functions, 1043](#)
- [nppiFilterLowPass_16u_C4R](#)
 - [image_filtering_functions, 1043](#)
- [nppiFilterLowPass_32f_AC4R](#)
 - [image_filtering_functions, 1044](#)
- [nppiFilterLowPass_32f_C1R](#)
 - [image_filtering_functions, 1044](#)
- [nppiFilterLowPass_32f_C3R](#)
 - [image_filtering_functions, 1044](#)
- [nppiFilterLowPass_32f_C4R](#)
 - [image_filtering_functions, 1045](#)
- [nppiFilterLowPass_8u_AC4R](#)
 - [image_filtering_functions, 1045](#)
- [nppiFilterLowPass_8u_C1R](#)
 - [image_filtering_functions, 1045](#)
- [nppiFilterLowPass_8u_C3R](#)
 - [image_filtering_functions, 1046](#)
- [nppiFilterLowPass_8u_C4R](#)
 - [image_filtering_functions, 1046](#)
- [nppiFilterLowPassBorder_16s_AC4R](#)
 - [image_filtering_functions, 1046](#)
- [nppiFilterLowPassBorder_16s_C1R](#)
 - [image_filtering_functions, 1047](#)
- [nppiFilterLowPassBorder_16s_C3R](#)
 - [image_filtering_functions, 1047](#)
- [nppiFilterLowPassBorder_16s_C4R](#)
 - [image_filtering_functions, 1048](#)
- [nppiFilterLowPassBorder_16u_AC4R](#)
 - [image_filtering_functions, 1048](#)
- [nppiFilterLowPassBorder_16u_C1R](#)
 - [image_filtering_functions, 1049](#)
- [nppiFilterLowPassBorder_16u_C3R](#)
 - [image_filtering_functions, 1049](#)
- [nppiFilterLowPassBorder_16u_C4R](#)
 - [image_filtering_functions, 1050](#)
- [nppiFilterLowPassBorder_32f_AC4R](#)
 - [image_filtering_functions, 1050](#)
- [nppiFilterLowPassBorder_32f_C1R](#)
 - [image_filtering_functions, 1051](#)
- [nppiFilterLowPassBorder_32f_C3R](#)
 - [image_filtering_functions, 1051](#)
- [nppiFilterLowPassBorder_32f_C4R](#)
 - [image_filtering_functions, 1052](#)
- [nppiFilterLowPassBorder_8u_AC4R](#)
 - [image_filtering_functions, 1052](#)
- [nppiFilterLowPassBorder_8u_C1R](#)
 - [image_filtering_functions, 1053](#)
- [nppiFilterLowPassBorder_8u_C3R](#)
 - [image_filtering_functions, 1053](#)
- [nppiFilterLowPassBorder_8u_C4R](#)
 - [image_filtering_functions, 1054](#)
- [nppiFilterMax_16s_AC4R](#)
 - [image_rank_filters, 1307](#)
- [nppiFilterMax_16s_C1R](#)
 - [image_rank_filters, 1307](#)
- [nppiFilterMax_16s_C3R](#)
 - [image_rank_filters, 1307](#)
- [nppiFilterMax_16s_C4R](#)
 - [image_rank_filters, 1308](#)
- [nppiFilterMax_16u_AC4R](#)
 - [image_rank_filters, 1308](#)
- [nppiFilterMax_16u_C1R](#)
 - [image_rank_filters, 1309](#)
- [nppiFilterMax_16u_C3R](#)
 - [image_rank_filters, 1309](#)
- [nppiFilterMax_16u_C4R](#)
 - [image_rank_filters, 1309](#)
- [nppiFilterMax_32f_AC4R](#)
 - [image_rank_filters, 1310](#)
- [nppiFilterMax_32f_C1R](#)
 - [image_rank_filters, 1310](#)
- [nppiFilterMax_32f_C3R](#)
 - [image_rank_filters, 1311](#)
- [nppiFilterMax_32f_C4R](#)
 - [image_rank_filters, 1311](#)
- [nppiFilterMax_8u_AC4R](#)
 - [image_rank_filters, 1311](#)
- [nppiFilterMax_8u_C1R](#)
 - [image_rank_filters, 1312](#)

- nppiFilterMax_8u_C3R
 - image_rank_filters, [1312](#)
- nppiFilterMax_8u_C4R
 - image_rank_filters, [1313](#)
- nppiFilterMaxBorder_16s_AC4R
 - image_rank_filters, [1313](#)
- nppiFilterMaxBorder_16s_C1R
 - image_rank_filters, [1313](#)
- nppiFilterMaxBorder_16s_C3R
 - image_rank_filters, [1314](#)
- nppiFilterMaxBorder_16s_C4R
 - image_rank_filters, [1314](#)
- nppiFilterMaxBorder_16u_AC4R
 - image_rank_filters, [1315](#)
- nppiFilterMaxBorder_16u_C1R
 - image_rank_filters, [1315](#)
- nppiFilterMaxBorder_16u_C3R
 - image_rank_filters, [1316](#)
- nppiFilterMaxBorder_16u_C4R
 - image_rank_filters, [1316](#)
- nppiFilterMaxBorder_32f_AC4R
 - image_rank_filters, [1317](#)
- nppiFilterMaxBorder_32f_C1R
 - image_rank_filters, [1317](#)
- nppiFilterMaxBorder_32f_C3R
 - image_rank_filters, [1318](#)
- nppiFilterMaxBorder_32f_C4R
 - image_rank_filters, [1318](#)
- nppiFilterMaxBorder_8u_AC4R
 - image_rank_filters, [1319](#)
- nppiFilterMaxBorder_8u_C1R
 - image_rank_filters, [1319](#)
- nppiFilterMaxBorder_8u_C3R
 - image_rank_filters, [1320](#)
- nppiFilterMaxBorder_8u_C4R
 - image_rank_filters, [1320](#)
- nppiFilterMedian_16s_AC4R
 - image_rank_filters, [1321](#)
- nppiFilterMedian_16s_C1R
 - image_rank_filters, [1321](#)
- nppiFilterMedian_16s_C3R
 - image_rank_filters, [1322](#)
- nppiFilterMedian_16s_C4R
 - image_rank_filters, [1322](#)
- nppiFilterMedian_16u_AC4R
 - image_rank_filters, [1323](#)
- nppiFilterMedian_16u_C1R
 - image_rank_filters, [1323](#)
- nppiFilterMedian_16u_C3R
 - image_rank_filters, [1324](#)
- nppiFilterMedian_16u_C4R
 - image_rank_filters, [1324](#)
- nppiFilterMedian_32f_AC4R
 - image_rank_filters, [1324](#)
- nppiFilterMedian_32f_C1R
 - image_rank_filters, [1325](#)
- nppiFilterMedian_32f_C3R
 - image_rank_filters, [1325](#)
- nppiFilterMedian_32f_C4R
 - image_rank_filters, [1326](#)
- nppiFilterMedian_8u_AC4R
 - image_rank_filters, [1326](#)
- nppiFilterMedian_8u_C1R
 - image_rank_filters, [1327](#)
- nppiFilterMedian_8u_C3R
 - image_rank_filters, [1327](#)
- nppiFilterMedian_8u_C4R
 - image_rank_filters, [1327](#)
- nppiFilterMedianGetBufferSize_16s_AC4R
 - image_rank_filters, [1328](#)
- nppiFilterMedianGetBufferSize_16s_C1R
 - image_rank_filters, [1328](#)
- nppiFilterMedianGetBufferSize_16s_C3R
 - image_rank_filters, [1328](#)
- nppiFilterMedianGetBufferSize_16s_C4R
 - image_rank_filters, [1329](#)
- nppiFilterMedianGetBufferSize_16u_AC4R
 - image_rank_filters, [1329](#)
- nppiFilterMedianGetBufferSize_16u_C1R
 - image_rank_filters, [1329](#)
- nppiFilterMedianGetBufferSize_16u_C3R
 - image_rank_filters, [1330](#)
- nppiFilterMedianGetBufferSize_16u_C4R
 - image_rank_filters, [1330](#)
- nppiFilterMedianGetBufferSize_32f_AC4R
 - image_rank_filters, [1330](#)
- nppiFilterMedianGetBufferSize_32f_C1R
 - image_rank_filters, [1330](#)
- nppiFilterMedianGetBufferSize_32f_C3R
 - image_rank_filters, [1331](#)
- nppiFilterMedianGetBufferSize_32f_C4R
 - image_rank_filters, [1331](#)
- nppiFilterMedianGetBufferSize_8u_AC4R
 - image_rank_filters, [1331](#)
- nppiFilterMedianGetBufferSize_8u_C1R
 - image_rank_filters, [1332](#)
- nppiFilterMedianGetBufferSize_8u_C3R
 - image_rank_filters, [1332](#)
- nppiFilterMedianGetBufferSize_8u_C4R
 - image_rank_filters, [1332](#)
- nppiFilterMin_16s_AC4R
 - image_rank_filters, [1332](#)
- nppiFilterMin_16s_C1R
 - image_rank_filters, [1333](#)
- nppiFilterMin_16s_C3R
 - image_rank_filters, [1333](#)
- nppiFilterMin_16s_C4R
 - image_rank_filters, [1334](#)

- nppiFilterMin_16u_AC4R
 - image_rank_filters, [1334](#)
- nppiFilterMin_16u_C1R
 - image_rank_filters, [1334](#)
- nppiFilterMin_16u_C3R
 - image_rank_filters, [1335](#)
- nppiFilterMin_16u_C4R
 - image_rank_filters, [1335](#)
- nppiFilterMin_32f_AC4R
 - image_rank_filters, [1336](#)
- nppiFilterMin_32f_C1R
 - image_rank_filters, [1336](#)
- nppiFilterMin_32f_C3R
 - image_rank_filters, [1336](#)
- nppiFilterMin_32f_C4R
 - image_rank_filters, [1337](#)
- nppiFilterMin_8u_AC4R
 - image_rank_filters, [1337](#)
- nppiFilterMin_8u_C1R
 - image_rank_filters, [1338](#)
- nppiFilterMin_8u_C3R
 - image_rank_filters, [1338](#)
- nppiFilterMin_8u_C4R
 - image_rank_filters, [1338](#)
- nppiFilterMinBorder_16s_AC4R
 - image_rank_filters, [1339](#)
- nppiFilterMinBorder_16s_C1R
 - image_rank_filters, [1339](#)
- nppiFilterMinBorder_16s_C3R
 - image_rank_filters, [1340](#)
- nppiFilterMinBorder_16s_C4R
 - image_rank_filters, [1340](#)
- nppiFilterMinBorder_16u_AC4R
 - image_rank_filters, [1341](#)
- nppiFilterMinBorder_16u_C1R
 - image_rank_filters, [1341](#)
- nppiFilterMinBorder_16u_C3R
 - image_rank_filters, [1342](#)
- nppiFilterMinBorder_16u_C4R
 - image_rank_filters, [1342](#)
- nppiFilterMinBorder_32f_AC4R
 - image_rank_filters, [1343](#)
- nppiFilterMinBorder_32f_C1R
 - image_rank_filters, [1343](#)
- nppiFilterMinBorder_32f_C3R
 - image_rank_filters, [1344](#)
- nppiFilterMinBorder_32f_C4R
 - image_rank_filters, [1344](#)
- nppiFilterMinBorder_8u_AC4R
 - image_rank_filters, [1345](#)
- nppiFilterMinBorder_8u_C1R
 - image_rank_filters, [1345](#)
- nppiFilterMinBorder_8u_C3R
 - image_rank_filters, [1346](#)
- nppiFilterMinBorder_8u_C4R
 - image_rank_filters, [1346](#)
- nppiFilterPrewittHoriz_16s_AC4R
 - fixed_filters, [1358](#)
- nppiFilterPrewittHoriz_16s_C1R
 - fixed_filters, [1358](#)
- nppiFilterPrewittHoriz_16s_C3R
 - fixed_filters, [1359](#)
- nppiFilterPrewittHoriz_16s_C4R
 - fixed_filters, [1359](#)
- nppiFilterPrewittHoriz_32f_AC4R
 - fixed_filters, [1359](#)
- nppiFilterPrewittHoriz_32f_C1R
 - fixed_filters, [1360](#)
- nppiFilterPrewittHoriz_32f_C3R
 - fixed_filters, [1360](#)
- nppiFilterPrewittHoriz_32f_C4R
 - fixed_filters, [1360](#)
- nppiFilterPrewittHoriz_8u_AC4R
 - fixed_filters, [1361](#)
- nppiFilterPrewittHoriz_8u_C1R
 - fixed_filters, [1361](#)
- nppiFilterPrewittHoriz_8u_C3R
 - fixed_filters, [1361](#)
- nppiFilterPrewittHoriz_8u_C4R
 - fixed_filters, [1362](#)
- nppiFilterPrewittHorizBorder_16s_AC4R
 - fixed_filters, [1362](#)
- nppiFilterPrewittHorizBorder_16s_C1R
 - fixed_filters, [1363](#)
- nppiFilterPrewittHorizBorder_16s_C3R
 - fixed_filters, [1363](#)
- nppiFilterPrewittHorizBorder_16s_C4R
 - fixed_filters, [1363](#)
- nppiFilterPrewittHorizBorder_32f_AC4R
 - fixed_filters, [1364](#)
- nppiFilterPrewittHorizBorder_32f_C1R
 - fixed_filters, [1364](#)
- nppiFilterPrewittHorizBorder_32f_C3R
 - fixed_filters, [1365](#)
- nppiFilterPrewittHorizBorder_32f_C4R
 - fixed_filters, [1365](#)
- nppiFilterPrewittHorizBorder_8u_AC4R
 - fixed_filters, [1366](#)
- nppiFilterPrewittHorizBorder_8u_C1R
 - fixed_filters, [1366](#)
- nppiFilterPrewittHorizBorder_8u_C3R
 - fixed_filters, [1366](#)
- nppiFilterPrewittHorizBorder_8u_C4R
 - fixed_filters, [1367](#)
- nppiFilterPrewittVert_16s_AC4R
 - fixed_filters, [1367](#)
- nppiFilterPrewittVert_16s_C1R
 - fixed_filters, [1368](#)

- nppiFilterPrewittVert_16s_C3R
 - fixed_filters, [1368](#)
- nppiFilterPrewittVert_16s_C4R
 - fixed_filters, [1368](#)
- nppiFilterPrewittVert_32f_AC4R
 - fixed_filters, [1369](#)
- nppiFilterPrewittVert_32f_C1R
 - fixed_filters, [1369](#)
- nppiFilterPrewittVert_32f_C3R
 - fixed_filters, [1369](#)
- nppiFilterPrewittVert_32f_C4R
 - fixed_filters, [1370](#)
- nppiFilterPrewittVert_8u_AC4R
 - fixed_filters, [1370](#)
- nppiFilterPrewittVert_8u_C1R
 - fixed_filters, [1370](#)
- nppiFilterPrewittVert_8u_C3R
 - fixed_filters, [1371](#)
- nppiFilterPrewittVert_8u_C4R
 - fixed_filters, [1371](#)
- nppiFilterPrewittVertBorder_16s_AC4R
 - fixed_filters, [1371](#)
- nppiFilterPrewittVertBorder_16s_C1R
 - fixed_filters, [1372](#)
- nppiFilterPrewittVertBorder_16s_C3R
 - fixed_filters, [1372](#)
- nppiFilterPrewittVertBorder_16s_C4R
 - fixed_filters, [1373](#)
- nppiFilterPrewittVertBorder_32f_AC4R
 - fixed_filters, [1373](#)
- nppiFilterPrewittVertBorder_32f_C1R
 - fixed_filters, [1374](#)
- nppiFilterPrewittVertBorder_32f_C3R
 - fixed_filters, [1374](#)
- nppiFilterPrewittVertBorder_32f_C4R
 - fixed_filters, [1374](#)
- nppiFilterPrewittVertBorder_8u_AC4R
 - fixed_filters, [1375](#)
- nppiFilterPrewittVertBorder_8u_C1R
 - fixed_filters, [1375](#)
- nppiFilterPrewittVertBorder_8u_C3R
 - fixed_filters, [1376](#)
- nppiFilterPrewittVertBorder_8u_C4R
 - fixed_filters, [1376](#)
- nppiFilterRobertsDown_16s_AC4R
 - image_filtering_functions, [1054](#)
- nppiFilterRobertsDown_16s_C1R
 - image_filtering_functions, [1055](#)
- nppiFilterRobertsDown_16s_C3R
 - image_filtering_functions, [1055](#)
- nppiFilterRobertsDown_16s_C4R
 - image_filtering_functions, [1055](#)
- nppiFilterRobertsDown_32f_AC4R
 - image_filtering_functions, [1056](#)
- nppiFilterRobertsDown_32f_C1R
 - image_filtering_functions, [1056](#)
- nppiFilterRobertsDown_32f_C3R
 - image_filtering_functions, [1056](#)
- nppiFilterRobertsDown_32f_C4R
 - image_filtering_functions, [1057](#)
- nppiFilterRobertsDown_8u_AC4R
 - image_filtering_functions, [1057](#)
- nppiFilterRobertsDown_8u_C1R
 - image_filtering_functions, [1057](#)
- nppiFilterRobertsDown_8u_C3R
 - image_filtering_functions, [1058](#)
- nppiFilterRobertsDown_8u_C4R
 - image_filtering_functions, [1058](#)
- nppiFilterRobertsDownBorder_16s_AC4R
 - image_filtering_functions, [1058](#)
- nppiFilterRobertsDownBorder_16s_C1R
 - image_filtering_functions, [1059](#)
- nppiFilterRobertsDownBorder_16s_C3R
 - image_filtering_functions, [1059](#)
- nppiFilterRobertsDownBorder_16s_C4R
 - image_filtering_functions, [1060](#)
- nppiFilterRobertsDownBorder_32f_AC4R
 - image_filtering_functions, [1060](#)
- nppiFilterRobertsDownBorder_32f_C1R
 - image_filtering_functions, [1061](#)
- nppiFilterRobertsDownBorder_32f_C3R
 - image_filtering_functions, [1061](#)
- nppiFilterRobertsDownBorder_32f_C4R
 - image_filtering_functions, [1061](#)
- nppiFilterRobertsDownBorder_8u_AC4R
 - image_filtering_functions, [1062](#)
- nppiFilterRobertsDownBorder_8u_C1R
 - image_filtering_functions, [1062](#)
- nppiFilterRobertsDownBorder_8u_C3R
 - image_filtering_functions, [1063](#)
- nppiFilterRobertsDownBorder_8u_C4R
 - image_filtering_functions, [1063](#)
- nppiFilterRobertsUp_16s_AC4R
 - image_filtering_functions, [1064](#)
- nppiFilterRobertsUp_16s_C1R
 - image_filtering_functions, [1064](#)
- nppiFilterRobertsUp_16s_C3R
 - image_filtering_functions, [1064](#)
- nppiFilterRobertsUp_16s_C4R
 - image_filtering_functions, [1065](#)
- nppiFilterRobertsUp_32f_AC4R
 - image_filtering_functions, [1065](#)
- nppiFilterRobertsUp_32f_C1R
 - image_filtering_functions, [1065](#)
- nppiFilterRobertsUp_32f_C3R
 - image_filtering_functions, [1066](#)
- nppiFilterRobertsUp_32f_C4R
 - image_filtering_functions, [1066](#)

- nppiFilterRobertsUp_8u_AC4R
 - image_filtering_functions, [1066](#)
- nppiFilterRobertsUp_8u_C1R
 - image_filtering_functions, [1067](#)
- nppiFilterRobertsUp_8u_C3R
 - image_filtering_functions, [1067](#)
- nppiFilterRobertsUp_8u_C4R
 - image_filtering_functions, [1067](#)
- nppiFilterRobertsUpBorder_16s_AC4R
 - image_filtering_functions, [1068](#)
- nppiFilterRobertsUpBorder_16s_C1R
 - image_filtering_functions, [1068](#)
- nppiFilterRobertsUpBorder_16s_C3R
 - image_filtering_functions, [1069](#)
- nppiFilterRobertsUpBorder_16s_C4R
 - image_filtering_functions, [1069](#)
- nppiFilterRobertsUpBorder_32f_AC4R
 - image_filtering_functions, [1069](#)
- nppiFilterRobertsUpBorder_32f_C1R
 - image_filtering_functions, [1070](#)
- nppiFilterRobertsUpBorder_32f_C3R
 - image_filtering_functions, [1070](#)
- nppiFilterRobertsUpBorder_32f_C4R
 - image_filtering_functions, [1071](#)
- nppiFilterRobertsUpBorder_8u_AC4R
 - image_filtering_functions, [1071](#)
- nppiFilterRobertsUpBorder_8u_C1R
 - image_filtering_functions, [1072](#)
- nppiFilterRobertsUpBorder_8u_C3R
 - image_filtering_functions, [1072](#)
- nppiFilterRobertsUpBorder_8u_C4R
 - image_filtering_functions, [1072](#)
- nppiFilterRow32f_16s_AC4R
 - image_1D_linear_filter, [1150](#)
- nppiFilterRow32f_16s_C1R
 - image_1D_linear_filter, [1150](#)
- nppiFilterRow32f_16s_C3R
 - image_1D_linear_filter, [1151](#)
- nppiFilterRow32f_16s_C4R
 - image_1D_linear_filter, [1151](#)
- nppiFilterRow32f_16u_AC4R
 - image_1D_linear_filter, [1152](#)
- nppiFilterRow32f_16u_C1R
 - image_1D_linear_filter, [1152](#)
- nppiFilterRow32f_16u_C3R
 - image_1D_linear_filter, [1153](#)
- nppiFilterRow32f_16u_C4R
 - image_1D_linear_filter, [1153](#)
- nppiFilterRow32f_8u_AC4R
 - image_1D_linear_filter, [1153](#)
- nppiFilterRow32f_8u_C1R
 - image_1D_linear_filter, [1154](#)
- nppiFilterRow32f_8u_C3R
 - image_1D_linear_filter, [1154](#)
- nppiFilterRow32f_8u_C4R
 - image_1D_linear_filter, [1155](#)
- nppiFilterRow_16s_AC4R
 - image_1D_linear_filter, [1155](#)
- nppiFilterRow_16s_C1R
 - image_1D_linear_filter, [1156](#)
- nppiFilterRow_16s_C3R
 - image_1D_linear_filter, [1156](#)
- nppiFilterRow_16s_C4R
 - image_1D_linear_filter, [1157](#)
- nppiFilterRow_16u_AC4R
 - image_1D_linear_filter, [1157](#)
- nppiFilterRow_16u_C1R
 - image_1D_linear_filter, [1158](#)
- nppiFilterRow_16u_C3R
 - image_1D_linear_filter, [1158](#)
- nppiFilterRow_16u_C4R
 - image_1D_linear_filter, [1159](#)
- nppiFilterRow_32f_AC4R
 - image_1D_linear_filter, [1159](#)
- nppiFilterRow_32f_C1R
 - image_1D_linear_filter, [1160](#)
- nppiFilterRow_32f_C3R
 - image_1D_linear_filter, [1160](#)
- nppiFilterRow_32f_C4R
 - image_1D_linear_filter, [1161](#)
- nppiFilterRow_64f_C1R
 - image_1D_linear_filter, [1161](#)
- nppiFilterRow_8u_AC4R
 - image_1D_linear_filter, [1162](#)
- nppiFilterRow_8u_C1R
 - image_1D_linear_filter, [1162](#)
- nppiFilterRow_8u_C3R
 - image_1D_linear_filter, [1163](#)
- nppiFilterRow_8u_C4R
 - image_1D_linear_filter, [1163](#)
- nppiFilterRowBorder32f_16s_AC4R
 - image_1D_linear_filter, [1164](#)
- nppiFilterRowBorder32f_16s_C1R
 - image_1D_linear_filter, [1164](#)
- nppiFilterRowBorder32f_16s_C3R
 - image_1D_linear_filter, [1165](#)
- nppiFilterRowBorder32f_16s_C4R
 - image_1D_linear_filter, [1165](#)
- nppiFilterRowBorder32f_16u_AC4R
 - image_1D_linear_filter, [1166](#)
- nppiFilterRowBorder32f_16u_C1R
 - image_1D_linear_filter, [1166](#)
- nppiFilterRowBorder32f_16u_C3R
 - image_1D_linear_filter, [1167](#)
- nppiFilterRowBorder32f_16u_C4R
 - image_1D_linear_filter, [1167](#)
- nppiFilterRowBorder32f_8u_AC4R
 - image_1D_linear_filter, [1168](#)

- [nppiFilterRowBorder32f_8u_C1R](#)
 - [image_1D_linear_filter, 1168](#)
- [nppiFilterRowBorder32f_8u_C3R](#)
 - [image_1D_linear_filter, 1169](#)
- [nppiFilterRowBorder32f_8u_C4R](#)
 - [image_1D_linear_filter, 1169](#)
- [nppiFilterRowBorder_16s_AC4R](#)
 - [image_1D_linear_filter, 1170](#)
- [nppiFilterRowBorder_16s_C1R](#)
 - [image_1D_linear_filter, 1171](#)
- [nppiFilterRowBorder_16s_C3R](#)
 - [image_1D_linear_filter, 1171](#)
- [nppiFilterRowBorder_16s_C4R](#)
 - [image_1D_linear_filter, 1172](#)
- [nppiFilterRowBorder_16u_AC4R](#)
 - [image_1D_linear_filter, 1172](#)
- [nppiFilterRowBorder_16u_C1R](#)
 - [image_1D_linear_filter, 1173](#)
- [nppiFilterRowBorder_16u_C3R](#)
 - [image_1D_linear_filter, 1174](#)
- [nppiFilterRowBorder_16u_C4R](#)
 - [image_1D_linear_filter, 1174](#)
- [nppiFilterRowBorder_32f_AC4R](#)
 - [image_1D_linear_filter, 1175](#)
- [nppiFilterRowBorder_32f_C1R](#)
 - [image_1D_linear_filter, 1175](#)
- [nppiFilterRowBorder_32f_C3R](#)
 - [image_1D_linear_filter, 1176](#)
- [nppiFilterRowBorder_32f_C4R](#)
 - [image_1D_linear_filter, 1176](#)
- [nppiFilterRowBorder_8u_AC4R](#)
 - [image_1D_linear_filter, 1177](#)
- [nppiFilterRowBorder_8u_C1R](#)
 - [image_1D_linear_filter, 1178](#)
- [nppiFilterRowBorder_8u_C3R](#)
 - [image_1D_linear_filter, 1178](#)
- [nppiFilterRowBorder_8u_C4R](#)
 - [image_1D_linear_filter, 1179](#)
- [nppiFilterScharrHoriz_32f_C1R](#)
 - [fixed_filters, 1377](#)
- [nppiFilterScharrHoriz_8s16s_C1R](#)
 - [fixed_filters, 1377](#)
- [nppiFilterScharrHoriz_8u16s_C1R](#)
 - [fixed_filters, 1377](#)
- [nppiFilterScharrHorizBorder_32f_C1R](#)
 - [fixed_filters, 1378](#)
- [nppiFilterScharrHorizBorder_8s16s_C1R](#)
 - [fixed_filters, 1378](#)
- [nppiFilterScharrHorizBorder_8u16s_C1R](#)
 - [fixed_filters, 1379](#)
- [nppiFilterScharrVert_32f_C1R](#)
 - [fixed_filters, 1379](#)
- [nppiFilterScharrVert_8s16s_C1R](#)
 - [fixed_filters, 1379](#)
- [nppiFilterScharrVert_8u16s_C1R](#)
 - [fixed_filters, 1380](#)
- [nppiFilterScharrVertBorder_32f_C1R](#)
 - [fixed_filters, 1380](#)
- [nppiFilterScharrVertBorder_8s16s_C1R](#)
 - [fixed_filters, 1381](#)
- [nppiFilterScharrVertBorder_8u16s_C1R](#)
 - [fixed_filters, 1381](#)
- [nppiFilterSharpen_16s_AC4R](#)
 - [image_filtering_functions, 1073](#)
- [nppiFilterSharpen_16s_C1R](#)
 - [image_filtering_functions, 1073](#)
- [nppiFilterSharpen_16s_C3R](#)
 - [image_filtering_functions, 1074](#)
- [nppiFilterSharpen_16s_C4R](#)
 - [image_filtering_functions, 1074](#)
- [nppiFilterSharpen_16u_AC4R](#)
 - [image_filtering_functions, 1074](#)
- [nppiFilterSharpen_16u_C1R](#)
 - [image_filtering_functions, 1075](#)
- [nppiFilterSharpen_16u_C3R](#)
 - [image_filtering_functions, 1075](#)
- [nppiFilterSharpen_16u_C4R](#)
 - [image_filtering_functions, 1075](#)
- [nppiFilterSharpen_32f_AC4R](#)
 - [image_filtering_functions, 1076](#)
- [nppiFilterSharpen_32f_C1R](#)
 - [image_filtering_functions, 1076](#)
- [nppiFilterSharpen_32f_C3R](#)
 - [image_filtering_functions, 1076](#)
- [nppiFilterSharpen_32f_C4R](#)
 - [image_filtering_functions, 1077](#)
- [nppiFilterSharpen_8u_AC4R](#)
 - [image_filtering_functions, 1077](#)
- [nppiFilterSharpen_8u_C1R](#)
 - [image_filtering_functions, 1077](#)
- [nppiFilterSharpen_8u_C3R](#)
 - [image_filtering_functions, 1078](#)
- [nppiFilterSharpen_8u_C4R](#)
 - [image_filtering_functions, 1078](#)
- [nppiFilterSharpenBorder_16s_AC4R](#)
 - [image_filtering_functions, 1078](#)
- [nppiFilterSharpenBorder_16s_C1R](#)
 - [image_filtering_functions, 1079](#)
- [nppiFilterSharpenBorder_16s_C3R](#)
 - [image_filtering_functions, 1079](#)
- [nppiFilterSharpenBorder_16s_C4R](#)
 - [image_filtering_functions, 1080](#)
- [nppiFilterSharpenBorder_16u_AC4R](#)
 - [image_filtering_functions, 1080](#)
- [nppiFilterSharpenBorder_16u_C1R](#)
 - [image_filtering_functions, 1081](#)
- [nppiFilterSharpenBorder_16u_C3R](#)
 - [image_filtering_functions, 1081](#)

- nppiFilterSharpenBorder_16u_C4R
image_filtering_functions, 1081
- nppiFilterSharpenBorder_32f_AC4R
image_filtering_functions, 1082
- nppiFilterSharpenBorder_32f_C1R
image_filtering_functions, 1082
- nppiFilterSharpenBorder_32f_C3R
image_filtering_functions, 1083
- nppiFilterSharpenBorder_32f_C4R
image_filtering_functions, 1083
- nppiFilterSharpenBorder_8u_AC4R
image_filtering_functions, 1084
- nppiFilterSharpenBorder_8u_C1R
image_filtering_functions, 1084
- nppiFilterSharpenBorder_8u_C3R
image_filtering_functions, 1084
- nppiFilterSharpenBorder_8u_C4R
image_filtering_functions, 1085
- nppiFilterSobelCross_32f_C1R
image_1D_linear_filter, 1179
- nppiFilterSobelCross_8s16s_C1R
image_1D_linear_filter, 1180
- nppiFilterSobelCross_8u16s_C1R
image_1D_linear_filter, 1180
- nppiFilterSobelCrossBorder_32f_C1R
image_filtering_functions, 1085
- nppiFilterSobelCrossBorder_8s16s_C1R
image_filtering_functions, 1086
- nppiFilterSobelCrossBorder_8u16s_C1R
image_filtering_functions, 1086
- nppiFilterSobelHoriz_16s_AC4R
fixed_filters, 1381
- nppiFilterSobelHoriz_16s_C1R
fixed_filters, 1382
- nppiFilterSobelHoriz_16s_C3R
fixed_filters, 1382
- nppiFilterSobelHoriz_16s_C4R
fixed_filters, 1382
- nppiFilterSobelHoriz_32f_AC4R
fixed_filters, 1383
- nppiFilterSobelHoriz_32f_C1R
fixed_filters, 1383
- nppiFilterSobelHoriz_32f_C3R
fixed_filters, 1383
- nppiFilterSobelHoriz_32f_C4R
fixed_filters, 1384
- nppiFilterSobelHoriz_8s16s_C1R
fixed_filters, 1384
- nppiFilterSobelHoriz_8u16s_C1R
fixed_filters, 1384
- nppiFilterSobelHoriz_8u_AC4R
fixed_filters, 1385
- nppiFilterSobelHoriz_8u_C1R
fixed_filters, 1385
- nppiFilterSobelHoriz_8u_C3R
fixed_filters, 1385
- nppiFilterSobelHoriz_8u_C4R
fixed_filters, 1386
- nppiFilterSobelHorizBorder_16s_AC4R
image_1D_linear_filter, 1180
- nppiFilterSobelHorizBorder_16s_C1R
image_1D_linear_filter, 1181
- nppiFilterSobelHorizBorder_16s_C3R
image_1D_linear_filter, 1181
- nppiFilterSobelHorizBorder_16s_C4R
image_1D_linear_filter, 1182
- nppiFilterSobelHorizBorder_32f_AC4R
image_1D_linear_filter, 1182
- nppiFilterSobelHorizBorder_32f_C1R
image_1D_linear_filter, 1183
- nppiFilterSobelHorizBorder_32f_C3R
image_1D_linear_filter, 1183
- nppiFilterSobelHorizBorder_32f_C4R
image_1D_linear_filter, 1183
- nppiFilterSobelHorizBorder_8s16s_C1R
image_1D_linear_filter, 1184
- nppiFilterSobelHorizBorder_8u16s_C1R
image_1D_linear_filter, 1184
- nppiFilterSobelHorizBorder_8u_AC4R
image_1D_linear_filter, 1185
- nppiFilterSobelHorizBorder_8u_C1R
image_1D_linear_filter, 1185
- nppiFilterSobelHorizBorder_8u_C3R
image_1D_linear_filter, 1186
- nppiFilterSobelHorizBorder_8u_C4R
image_1D_linear_filter, 1186
- nppiFilterSobelHorizMask_32f_C1R
fixed_filters, 1386
- nppiFilterSobelHorizMaskBorder_32f_C1R
image_1D_linear_filter, 1187
- nppiFilterSobelHorizSecond_32f_C1R
fixed_filters, 1386
- nppiFilterSobelHorizSecond_8s16s_C1R
fixed_filters, 1387
- nppiFilterSobelHorizSecond_8u16s_C1R
fixed_filters, 1387
- nppiFilterSobelHorizSecondBorder_32f_C1R
image_1D_linear_filter, 1187
- nppiFilterSobelHorizSecondBorder_8s16s_C1R
image_1D_linear_filter, 1188
- nppiFilterSobelHorizSecondBorder_8u16s_C1R
image_1D_linear_filter, 1188
- nppiFilterSobelVert_16s_AC4R
fixed_filters, 1388
- nppiFilterSobelVert_16s_C1R
fixed_filters, 1388
- nppiFilterSobelVert_16s_C3R
fixed_filters, 1388

- [nppiFilterSobelVert_16s_C4R](#)
 - [fixed_filters, 1389](#)
- [nppiFilterSobelVert_32f_AC4R](#)
 - [fixed_filters, 1389](#)
- [nppiFilterSobelVert_32f_C1R](#)
 - [fixed_filters, 1389](#)
- [nppiFilterSobelVert_32f_C3R](#)
 - [fixed_filters, 1390](#)
- [nppiFilterSobelVert_32f_C4R](#)
 - [fixed_filters, 1390](#)
- [nppiFilterSobelVert_8s16s_C1R](#)
 - [fixed_filters, 1390](#)
- [nppiFilterSobelVert_8u16s_C1R](#)
 - [fixed_filters, 1391](#)
- [nppiFilterSobelVert_8u_AC4R](#)
 - [fixed_filters, 1391](#)
- [nppiFilterSobelVert_8u_C1R](#)
 - [fixed_filters, 1391](#)
- [nppiFilterSobelVert_8u_C3R](#)
 - [fixed_filters, 1392](#)
- [nppiFilterSobelVert_8u_C4R](#)
 - [fixed_filters, 1392](#)
- [nppiFilterSobelVertBorder_16s_AC4R](#)
 - [image_1D_linear_filter, 1189](#)
- [nppiFilterSobelVertBorder_16s_C1R](#)
 - [image_1D_linear_filter, 1189](#)
- [nppiFilterSobelVertBorder_16s_C3R](#)
 - [image_1D_linear_filter, 1189](#)
- [nppiFilterSobelVertBorder_16s_C4R](#)
 - [image_1D_linear_filter, 1190](#)
- [nppiFilterSobelVertBorder_32f_AC4R](#)
 - [image_1D_linear_filter, 1190](#)
- [nppiFilterSobelVertBorder_32f_C1R](#)
 - [image_1D_linear_filter, 1191](#)
- [nppiFilterSobelVertBorder_32f_C3R](#)
 - [image_1D_linear_filter, 1191](#)
- [nppiFilterSobelVertBorder_32f_C4R](#)
 - [image_1D_linear_filter, 1192](#)
- [nppiFilterSobelVertBorder_8s16s_C1R](#)
 - [image_1D_linear_filter, 1192](#)
- [nppiFilterSobelVertBorder_8u16s_C1R](#)
 - [image_1D_linear_filter, 1192](#)
- [nppiFilterSobelVertBorder_8u_AC4R](#)
 - [image_1D_linear_filter, 1193](#)
- [nppiFilterSobelVertBorder_8u_C1R](#)
 - [image_1D_linear_filter, 1193](#)
- [nppiFilterSobelVertBorder_8u_C3R](#)
 - [image_1D_linear_filter, 1194](#)
- [nppiFilterSobelVertBorder_8u_C4R](#)
 - [image_1D_linear_filter, 1194](#)
- [nppiFilterSobelVertMask_32f_C1R](#)
 - [fixed_filters, 1392](#)
- [nppiFilterSobelVertMaskBorder_32f_C1R](#)
 - [image_1D_linear_filter, 1195](#)
- [nppiFilterSobelVertSecond_32f_C1R](#)
 - [image_1D_linear_filter, 1195](#)
- [nppiFilterSobelVertSecond_8s16s_C1R](#)
 - [image_1D_linear_filter, 1196](#)
- [nppiFilterSobelVertSecond_8u16s_C1R](#)
 - [image_1D_linear_filter, 1196](#)
- [nppiFilterSobelVertSecondBorder_32f_C1R](#)
 - [image_filtering_functions, 1087](#)
- [nppiFilterSobelVertSecondBorder_8s16s_C1R](#)
 - [image_filtering_functions, 1087](#)
- [nppiFilterSobelVertSecondBorder_8u16s_C1R](#)
 - [image_filtering_functions, 1088](#)
- [nppiFilterUnsharpBorder_16s_AC4R](#)
 - [image_filtering_functions, 1088](#)
- [nppiFilterUnsharpBorder_16s_C1R](#)
 - [image_filtering_functions, 1089](#)
- [nppiFilterUnsharpBorder_16s_C3R](#)
 - [image_filtering_functions, 1089](#)
- [nppiFilterUnsharpBorder_16s_C4R](#)
 - [image_filtering_functions, 1090](#)
- [nppiFilterUnsharpBorder_16u_AC4R](#)
 - [image_filtering_functions, 1090](#)
- [nppiFilterUnsharpBorder_16u_C1R](#)
 - [image_filtering_functions, 1091](#)
- [nppiFilterUnsharpBorder_16u_C3R](#)
 - [image_filtering_functions, 1092](#)
- [nppiFilterUnsharpBorder_16u_C4R](#)
 - [image_filtering_functions, 1092](#)
- [nppiFilterUnsharpBorder_32f_AC4R](#)
 - [image_filtering_functions, 1093](#)
- [nppiFilterUnsharpBorder_32f_C1R](#)
 - [image_filtering_functions, 1093](#)
- [nppiFilterUnsharpBorder_32f_C3R](#)
 - [image_filtering_functions, 1094](#)
- [nppiFilterUnsharpBorder_32f_C4R](#)
 - [image_filtering_functions, 1094](#)
- [nppiFilterUnsharpBorder_8u_AC4R](#)
 - [image_filtering_functions, 1095](#)
- [nppiFilterUnsharpBorder_8u_C1R](#)
 - [image_filtering_functions, 1096](#)
- [nppiFilterUnsharpBorder_8u_C3R](#)
 - [image_filtering_functions, 1096](#)
- [nppiFilterUnsharpBorder_8u_C4R](#)
 - [image_filtering_functions, 1097](#)
- [nppiFilterUnsharpGetBufferSize_16s_AC4R](#)
 - [image_filtering_functions, 1097](#)
- [nppiFilterUnsharpGetBufferSize_16s_C1R](#)
 - [image_filtering_functions, 1098](#)
- [nppiFilterUnsharpGetBufferSize_16s_C3R](#)
 - [image_filtering_functions, 1098](#)
- [nppiFilterUnsharpGetBufferSize_16s_C4R](#)
 - [image_filtering_functions, 1098](#)
- [nppiFilterUnsharpGetBufferSize_16u_AC4R](#)
 - [image_filtering_functions, 1098](#)

- nppiFilterUnsharpGetBufferSize_16u_C1R
image_filtering_functions, [1099](#)
- nppiFilterUnsharpGetBufferSize_16u_C3R
image_filtering_functions, [1099](#)
- nppiFilterUnsharpGetBufferSize_16u_C4R
image_filtering_functions, [1099](#)
- nppiFilterUnsharpGetBufferSize_32f_AC4R
image_filtering_functions, [1100](#)
- nppiFilterUnsharpGetBufferSize_32f_C1R
image_filtering_functions, [1100](#)
- nppiFilterUnsharpGetBufferSize_32f_C3R
image_filtering_functions, [1100](#)
- nppiFilterUnsharpGetBufferSize_32f_C4R
image_filtering_functions, [1100](#)
- nppiFilterUnsharpGetBufferSize_8u_AC4R
image_filtering_functions, [1101](#)
- nppiFilterUnsharpGetBufferSize_8u_C1R
image_filtering_functions, [1101](#)
- nppiFilterUnsharpGetBufferSize_8u_C3R
image_filtering_functions, [1101](#)
- nppiFilterUnsharpGetBufferSize_8u_C4R
image_filtering_functions, [1102](#)
- nppiFree
image_memory_management, [2362](#)
- nppiFullNormLevelGetBufferHostSize_16u32f_-
AC4R
crosscorrfullnormlevel, [2210](#)
- nppiFullNormLevelGetBufferHostSize_16u32f_-
C1R
crosscorrfullnormlevel, [2211](#)
- nppiFullNormLevelGetBufferHostSize_16u32f_-
C3R
crosscorrfullnormlevel, [2211](#)
- nppiFullNormLevelGetBufferHostSize_16u32f_-
C4R
crosscorrfullnormlevel, [2211](#)
- nppiFullNormLevelGetBufferHostSize_32f_AC4R
crosscorrfullnormlevel, [2212](#)
- nppiFullNormLevelGetBufferHostSize_32f_C1R
crosscorrfullnormlevel, [2212](#)
- nppiFullNormLevelGetBufferHostSize_32f_C3R
crosscorrfullnormlevel, [2212](#)
- nppiFullNormLevelGetBufferHostSize_32f_C4R
crosscorrfullnormlevel, [2212](#)
- nppiFullNormLevelGetBufferHostSize_8s32f_-
AC4R
crosscorrfullnormlevel, [2213](#)
- nppiFullNormLevelGetBufferHostSize_8s32f_C1R
crosscorrfullnormlevel, [2213](#)
- nppiFullNormLevelGetBufferHostSize_8s32f_C3R
crosscorrfullnormlevel, [2213](#)
- nppiFullNormLevelGetBufferHostSize_8s32f_C4R
crosscorrfullnormlevel, [2214](#)
- nppiFullNormLevelGetBufferHostSize_8u32f_-
AC4R
crosscorrfullnormlevel, [2214](#)
- nppiFullNormLevelGetBufferHostSize_8u32f_-
C1R
crosscorrfullnormlevel, [2214](#)
- nppiFullNormLevelGetBufferHostSize_8u32f_-
C3R
crosscorrfullnormlevel, [2214](#)
- nppiFullNormLevelGetBufferHostSize_8u32f_-
C4R
crosscorrfullnormlevel, [2215](#)
- nppiFullNormLevelGetBufferHostSize_8u_-
AC4RSfs
crosscorrfullnormlevel, [2215](#)
- nppiFullNormLevelGetBufferHostSize_8u_C1RSfs
crosscorrfullnormlevel, [2215](#)
- nppiFullNormLevelGetBufferHostSize_8u_C3RSfs
crosscorrfullnormlevel, [2216](#)
- nppiFullNormLevelGetBufferHostSize_8u_C4RSfs
crosscorrfullnormlevel, [2216](#)
- nppiGammaFwd_8u_AC4IR
image_color_gamma_correction, [615](#)
- nppiGammaFwd_8u_AC4R
image_color_gamma_correction, [615](#)
- nppiGammaFwd_8u_C3IR
image_color_gamma_correction, [615](#)
- nppiGammaFwd_8u_C3R
image_color_gamma_correction, [616](#)
- nppiGammaFwd_8u_IP3R
image_color_gamma_correction, [616](#)
- nppiGammaFwd_8u_P3R
image_color_gamma_correction, [616](#)
- nppiGammaInv_8u_AC4IR
image_color_gamma_correction, [617](#)
- nppiGammaInv_8u_AC4R
image_color_gamma_correction, [617](#)
- nppiGammaInv_8u_C3IR
image_color_gamma_correction, [617](#)
- nppiGammaInv_8u_C3R
image_color_gamma_correction, [618](#)
- nppiGammaInv_8u_IP3R
image_color_gamma_correction, [618](#)
- nppiGammaInv_8u_P3R
image_color_gamma_correction, [618](#)
- nppiGetAffineBound
image_affine_transform, [1488](#)
- nppiGetAffineQuad
image_affine_transform, [1488](#)
- nppiGetAffineTransform
image_affine_transform, [1489](#)
- nppiGetPerspectiveBound
image_perspective_transforms, [1537](#)
- nppiGetPerspectiveQuad

- image_perspective_transforms, 1537
- nppiGetPerspectiveTransform
 - image_perspective_transforms, 1538
- nppiGetResizeRect
 - image_resize_square_pixel, 1400
- nppiGetRotateBound
 - image_rotate, 1454
- nppiGetRotateQuad
 - image_rotate, 1455
- nppiGraphcut8_32f8u
 - image_graphcut, 732
- nppiGraphcut8_32s8u
 - image_graphcut, 732
- nppiGraphcut8GetSize
 - image_graphcut, 733
- nppiGraphcut8InitAlloc
 - image_graphcut, 734
- nppiGraphcut_32f8u
 - image_graphcut, 734
- nppiGraphcut_32s8u
 - image_graphcut, 735
- nppiGraphcutFree
 - image_graphcut, 736
- nppiGraphcutGetSize
 - image_graphcut, 736
- nppiGraphcutInitAlloc
 - image_graphcut, 737
- NppiGraphcutState
 - image_labeling_and_segmentation, 730
- NppiHaarBuffer, 2871
 - haarBuffer, 2871
 - haarBufferSize, 2871
- NppiHaarClassifier_32f, 2872
 - classifiers, 2872
 - classifierSize, 2872
 - classifierStep, 2872
 - counterDevice, 2872
 - numClassifiers, 2872
- nppiHistogramEven_16s_AC4R
 - image_histogrameven, 2099
- nppiHistogramEven_16s_C1R
 - image_histogrameven, 2099
- nppiHistogramEven_16s_C3R
 - image_histogrameven, 2099
- nppiHistogramEven_16s_C4R
 - image_histogrameven, 2100
- nppiHistogramEven_16u_AC4R
 - image_histogrameven, 2100
- nppiHistogramEven_16u_C1R
 - image_histogrameven, 2101
- nppiHistogramEven_16u_C3R
 - image_histogrameven, 2101
- nppiHistogramEven_16u_C4R
 - image_histogrameven, 2102
- nppiHistogramEven_8u_AC4R
 - image_histogrameven, 2102
- nppiHistogramEven_8u_C1R
 - image_histogrameven, 2103
- nppiHistogramEven_8u_C3R
 - image_histogrameven, 2103
- nppiHistogramEven_8u_C4R
 - image_histogrameven, 2104
- nppiHistogramEvenGetBufferSize_16s_AC4R
 - image_histogrameven, 2104
- nppiHistogramEvenGetBufferSize_16s_C1R
 - image_histogrameven, 2104
- nppiHistogramEvenGetBufferSize_16s_C3R
 - image_histogrameven, 2105
- nppiHistogramEvenGetBufferSize_16s_C4R
 - image_histogrameven, 2105
- nppiHistogramEvenGetBufferSize_16u_AC4R
 - image_histogrameven, 2105
- nppiHistogramEvenGetBufferSize_16u_C1R
 - image_histogrameven, 2106
- nppiHistogramEvenGetBufferSize_16u_C3R
 - image_histogrameven, 2106
- nppiHistogramEvenGetBufferSize_16u_C4R
 - image_histogrameven, 2106
- nppiHistogramEvenGetBufferSize_8u_AC4R
 - image_histogrameven, 2107
- nppiHistogramEvenGetBufferSize_8u_C1R
 - image_histogrameven, 2107
- nppiHistogramEvenGetBufferSize_8u_C3R
 - image_histogrameven, 2107
- nppiHistogramEvenGetBufferSize_8u_C4R
 - image_histogrameven, 2108
- nppiHistogramRange_16s_AC4R
 - image_histogramrange, 2112
- nppiHistogramRange_16s_C1R
 - image_histogramrange, 2112
- nppiHistogramRange_16s_C3R
 - image_histogramrange, 2112
- nppiHistogramRange_16s_C4R
 - image_histogramrange, 2113
- nppiHistogramRange_16u_AC4R
 - image_histogramrange, 2113
- nppiHistogramRange_16u_C1R
 - image_histogramrange, 2114
- nppiHistogramRange_16u_C3R
 - image_histogramrange, 2114
- nppiHistogramRange_16u_C4R
 - image_histogramrange, 2115
- nppiHistogramRange_32f_AC4R
 - image_histogramrange, 2115
- nppiHistogramRange_32f_C1R
 - image_histogramrange, 2116
- nppiHistogramRange_32f_C3R
 - image_histogramrange, 2116

- nppiHistogramRange_32f_C4R
 - image_histogramrange, [2116](#)
- nppiHistogramRange_8u_AC4R
 - image_histogramrange, [2117](#)
- nppiHistogramRange_8u_C1R
 - image_histogramrange, [2117](#)
- nppiHistogramRange_8u_C3R
 - image_histogramrange, [2118](#)
- nppiHistogramRange_8u_C4R
 - image_histogramrange, [2118](#)
- nppiHistogramRangeGetBufferSize_16s_AC4R
 - image_histogramrange, [2119](#)
- nppiHistogramRangeGetBufferSize_16s_C1R
 - image_histogramrange, [2119](#)
- nppiHistogramRangeGetBufferSize_16s_C3R
 - image_histogramrange, [2119](#)
- nppiHistogramRangeGetBufferSize_16s_C4R
 - image_histogramrange, [2120](#)
- nppiHistogramRangeGetBufferSize_16u_AC4R
 - image_histogramrange, [2120](#)
- nppiHistogramRangeGetBufferSize_16u_C1R
 - image_histogramrange, [2120](#)
- nppiHistogramRangeGetBufferSize_16u_C3R
 - image_histogramrange, [2121](#)
- nppiHistogramRangeGetBufferSize_16u_C4R
 - image_histogramrange, [2121](#)
- nppiHistogramRangeGetBufferSize_32f_AC4R
 - image_histogramrange, [2121](#)
- nppiHistogramRangeGetBufferSize_32f_C1R
 - image_histogramrange, [2122](#)
- nppiHistogramRangeGetBufferSize_32f_C3R
 - image_histogramrange, [2122](#)
- nppiHistogramRangeGetBufferSize_32f_C4R
 - image_histogramrange, [2122](#)
- nppiHistogramRangeGetBufferSize_8u_AC4R
 - image_histogramrange, [2123](#)
- nppiHistogramRangeGetBufferSize_8u_C1R
 - image_histogramrange, [2123](#)
- nppiHistogramRangeGetBufferSize_8u_C3R
 - image_histogramrange, [2123](#)
- nppiHistogramRangeGetBufferSize_8u_C4R
 - image_histogramrange, [2124](#)
- nppiHLSToBGR_8u_AC4P4R
 - image_color_model_conversion, [548](#)
- nppiHLSToBGR_8u_AC4R
 - image_color_model_conversion, [549](#)
- nppiHLSToBGR_8u_AP4C4R
 - image_color_model_conversion, [549](#)
- nppiHLSToBGR_8u_AP4R
 - image_color_model_conversion, [549](#)
- nppiHLSToBGR_8u_C3P3R
 - image_color_model_conversion, [550](#)
- nppiHLSToBGR_8u_P3C3R
 - image_color_model_conversion, [550](#)
- nppiHLSToBGR_8u_P3R
 - image_color_model_conversion, [550](#)
- nppiHLSToRGB_8u_AC4R
 - image_color_model_conversion, [551](#)
- nppiHLSToRGB_8u_C3R
 - image_color_model_conversion, [551](#)
- nppiHSVToRGB_8u_AC4R
 - image_color_model_conversion, [551](#)
- nppiHSVToRGB_8u_C3R
 - image_color_model_conversion, [552](#)
- NppiHuffmanTableType
 - typedefs_npp, [43](#)
- nppiIntegral_8u32f_C1R
 - image_integral, [2088](#)
- nppiIntegral_8u32s_C1R
 - image_integral, [2088](#)
- NppiInterpolationMode
 - typedefs_npp, [43](#)
- nppiLabToBGR_8u_C3R
 - image_color_model_conversion, [552](#)
- nppiLn_16s_C1RSfs
 - image_ln, [358](#)
- nppiLn_16s_C1RSfs
 - image_ln, [358](#)
- nppiLn_16s_C3RSfs
 - image_ln, [359](#)
- nppiLn_16s_C3RSfs
 - image_ln, [359](#)
- nppiLn_16u_C1RSfs
 - image_ln, [359](#)
- nppiLn_16u_C1RSfs
 - image_ln, [360](#)
- nppiLn_16u_C3RSfs
 - image_ln, [360](#)
- nppiLn_16u_C3RSfs
 - image_ln, [360](#)
- nppiLn_32f_C1IR
 - image_ln, [361](#)
- nppiLn_32f_C1R
 - image_ln, [361](#)
- nppiLn_32f_C3IR
 - image_ln, [361](#)
- nppiLn_32f_C3R
 - image_ln, [362](#)
- nppiLn_8u_C1RSfs
 - image_ln, [362](#)
- nppiLn_8u_C1RSfs
 - image_ln, [362](#)
- nppiLn_8u_C3RSfs
 - image_ln, [363](#)
- nppiLn_8u_C3RSfs
 - image_ln, [363](#)
- nppiLShiftC_16u_AC4IR
 - image_lshiftc, [424](#)

- nppiLShiftC_16u_AC4R
 - image_lshiftc, [424](#)
- nppiLShiftC_16u_C1IR
 - image_lshiftc, [424](#)
- nppiLShiftC_16u_C1R
 - image_lshiftc, [425](#)
- nppiLShiftC_16u_C3IR
 - image_lshiftc, [425](#)
- nppiLShiftC_16u_C3R
 - image_lshiftc, [425](#)
- nppiLShiftC_16u_C4IR
 - image_lshiftc, [426](#)
- nppiLShiftC_16u_C4R
 - image_lshiftc, [426](#)
- nppiLShiftC_32s_AC4IR
 - image_lshiftc, [426](#)
- nppiLShiftC_32s_AC4R
 - image_lshiftc, [427](#)
- nppiLShiftC_32s_C1IR
 - image_lshiftc, [427](#)
- nppiLShiftC_32s_C1R
 - image_lshiftc, [427](#)
- nppiLShiftC_32s_C3IR
 - image_lshiftc, [428](#)
- nppiLShiftC_32s_C3R
 - image_lshiftc, [428](#)
- nppiLShiftC_32s_C4IR
 - image_lshiftc, [428](#)
- nppiLShiftC_32s_C4R
 - image_lshiftc, [429](#)
- nppiLShiftC_8u_AC4IR
 - image_lshiftc, [429](#)
- nppiLShiftC_8u_AC4R
 - image_lshiftc, [429](#)
- nppiLShiftC_8u_C1IR
 - image_lshiftc, [430](#)
- nppiLShiftC_8u_C1R
 - image_lshiftc, [430](#)
- nppiLShiftC_8u_C3IR
 - image_lshiftc, [430](#)
- nppiLShiftC_8u_C3R
 - image_lshiftc, [431](#)
- nppiLShiftC_8u_C4IR
 - image_lshiftc, [431](#)
- nppiLShiftC_8u_C4R
 - image_lshiftc, [431](#)
- nppiLUT_16s_AC4IR
 - image_color_processing, [661](#)
- nppiLUT_16s_AC4R
 - image_color_processing, [661](#)
- nppiLUT_16s_C1IR
 - image_color_processing, [662](#)
- nppiLUT_16s_C1R
 - image_color_processing, [662](#)
- nppiLUT_16s_C3IR
 - image_color_processing, [663](#)
- nppiLUT_16s_C3R
 - image_color_processing, [663](#)
- nppiLUT_16s_C4IR
 - image_color_processing, [664](#)
- nppiLUT_16s_C4R
 - image_color_processing, [664](#)
- nppiLUT_16u_AC4IR
 - image_color_processing, [665](#)
- nppiLUT_16u_AC4R
 - image_color_processing, [665](#)
- nppiLUT_16u_C1IR
 - image_color_processing, [666](#)
- nppiLUT_16u_C1R
 - image_color_processing, [666](#)
- nppiLUT_16u_C3IR
 - image_color_processing, [667](#)
- nppiLUT_16u_C3R
 - image_color_processing, [667](#)
- nppiLUT_16u_C4IR
 - image_color_processing, [668](#)
- nppiLUT_16u_C4R
 - image_color_processing, [668](#)
- nppiLUT_32f_AC4IR
 - image_color_processing, [669](#)
- nppiLUT_32f_AC4R
 - image_color_processing, [669](#)
- nppiLUT_32f_C1IR
 - image_color_processing, [670](#)
- nppiLUT_32f_C1R
 - image_color_processing, [670](#)
- nppiLUT_32f_C3IR
 - image_color_processing, [671](#)
- nppiLUT_32f_C3R
 - image_color_processing, [671](#)
- nppiLUT_32f_C4IR
 - image_color_processing, [672](#)
- nppiLUT_32f_C4R
 - image_color_processing, [672](#)
- nppiLUT_8u_AC4IR
 - image_color_processing, [673](#)
- nppiLUT_8u_AC4R
 - image_color_processing, [673](#)
- nppiLUT_8u_C1IR
 - image_color_processing, [674](#)
- nppiLUT_8u_C1R
 - image_color_processing, [674](#)
- nppiLUT_8u_C3IR
 - image_color_processing, [675](#)
- nppiLUT_8u_C3R
 - image_color_processing, [675](#)
- nppiLUT_8u_C4IR
 - image_color_processing, [676](#)

- nppiLUT_8u_C4R
 - image_color_processing, [676](#)
- nppiLUT_Cubic_16s_AC4IR
 - image_color_processing, [677](#)
- nppiLUT_Cubic_16s_AC4R
 - image_color_processing, [677](#)
- nppiLUT_Cubic_16s_C1IR
 - image_color_processing, [678](#)
- nppiLUT_Cubic_16s_C1R
 - image_color_processing, [678](#)
- nppiLUT_Cubic_16s_C3IR
 - image_color_processing, [679](#)
- nppiLUT_Cubic_16s_C3R
 - image_color_processing, [679](#)
- nppiLUT_Cubic_16s_C4IR
 - image_color_processing, [680](#)
- nppiLUT_Cubic_16s_C4R
 - image_color_processing, [680](#)
- nppiLUT_Cubic_16u_AC4IR
 - image_color_processing, [681](#)
- nppiLUT_Cubic_16u_AC4R
 - image_color_processing, [681](#)
- nppiLUT_Cubic_16u_C1IR
 - image_color_processing, [682](#)
- nppiLUT_Cubic_16u_C1R
 - image_color_processing, [682](#)
- nppiLUT_Cubic_16u_C3IR
 - image_color_processing, [683](#)
- nppiLUT_Cubic_16u_C3R
 - image_color_processing, [683](#)
- nppiLUT_Cubic_16u_C4IR
 - image_color_processing, [684](#)
- nppiLUT_Cubic_16u_C4R
 - image_color_processing, [684](#)
- nppiLUT_Cubic_32f_AC4IR
 - image_color_processing, [685](#)
- nppiLUT_Cubic_32f_AC4R
 - image_color_processing, [685](#)
- nppiLUT_Cubic_32f_C1IR
 - image_color_processing, [686](#)
- nppiLUT_Cubic_32f_C1R
 - image_color_processing, [686](#)
- nppiLUT_Cubic_32f_C3IR
 - image_color_processing, [687](#)
- nppiLUT_Cubic_32f_C3R
 - image_color_processing, [687](#)
- nppiLUT_Cubic_32f_C4IR
 - image_color_processing, [688](#)
- nppiLUT_Cubic_32f_C4R
 - image_color_processing, [688](#)
- nppiLUT_Cubic_8u_AC4IR
 - image_color_processing, [689](#)
- nppiLUT_Cubic_8u_AC4R
 - image_color_processing, [689](#)
- nppiLUT_Cubic_8u_C1IR
 - image_color_processing, [690](#)
- nppiLUT_Cubic_8u_C1R
 - image_color_processing, [690](#)
- nppiLUT_Cubic_8u_C3IR
 - image_color_processing, [691](#)
- nppiLUT_Cubic_8u_C3R
 - image_color_processing, [691](#)
- nppiLUT_Cubic_8u_C4IR
 - image_color_processing, [692](#)
- nppiLUT_Cubic_8u_C4R
 - image_color_processing, [692](#)
- nppiLUT_Linear_16s_AC4IR
 - image_color_processing, [693](#)
- nppiLUT_Linear_16s_AC4R
 - image_color_processing, [693](#)
- nppiLUT_Linear_16s_C1IR
 - image_color_processing, [694](#)
- nppiLUT_Linear_16s_C1R
 - image_color_processing, [694](#)
- nppiLUT_Linear_16s_C3IR
 - image_color_processing, [695](#)
- nppiLUT_Linear_16s_C3R
 - image_color_processing, [695](#)
- nppiLUT_Linear_16s_C4IR
 - image_color_processing, [696](#)
- nppiLUT_Linear_16s_C4R
 - image_color_processing, [696](#)
- nppiLUT_Linear_16u_AC4IR
 - image_color_processing, [697](#)
- nppiLUT_Linear_16u_AC4R
 - image_color_processing, [697](#)
- nppiLUT_Linear_16u_C1IR
 - image_color_processing, [698](#)
- nppiLUT_Linear_16u_C1R
 - image_color_processing, [698](#)
- nppiLUT_Linear_16u_C3IR
 - image_color_processing, [699](#)
- nppiLUT_Linear_16u_C3R
 - image_color_processing, [699](#)
- nppiLUT_Linear_16u_C4IR
 - image_color_processing, [700](#)
- nppiLUT_Linear_16u_C4R
 - image_color_processing, [700](#)
- nppiLUT_Linear_32f_AC4IR
 - image_color_processing, [701](#)
- nppiLUT_Linear_32f_AC4R
 - image_color_processing, [701](#)
- nppiLUT_Linear_32f_C1IR
 - image_color_processing, [702](#)
- nppiLUT_Linear_32f_C1R
 - image_color_processing, [702](#)
- nppiLUT_Linear_32f_C3IR
 - image_color_processing, [703](#)

- nppiLUT_Linear_32f_C3R
 - image_color_processing, [703](#)
- nppiLUT_Linear_32f_C4IR
 - image_color_processing, [704](#)
- nppiLUT_Linear_32f_C4R
 - image_color_processing, [704](#)
- nppiLUT_Linear_8u_AC4IR
 - image_color_processing, [705](#)
- nppiLUT_Linear_8u_AC4R
 - image_color_processing, [705](#)
- nppiLUT_Linear_8u_C1IR
 - image_color_processing, [706](#)
- nppiLUT_Linear_8u_C1R
 - image_color_processing, [707](#)
- nppiLUT_Linear_8u_C3IR
 - image_color_processing, [707](#)
- nppiLUT_Linear_8u_C3R
 - image_color_processing, [708](#)
- nppiLUT_Linear_8u_C4IR
 - image_color_processing, [708](#)
- nppiLUT_Linear_8u_C4R
 - image_color_processing, [709](#)
- nppiLUT_Trilinear_8u_AC4IR
 - image_color_processing, [709](#)
- nppiLUT_Trilinear_8u_AC4R
 - image_color_processing, [710](#)
- nppiLUT_Trilinear_8u_C4R
 - image_color_processing, [711](#)
- nppiLUTPalette_16u24u_C1R
 - image_color_processing, [711](#)
- nppiLUTPalette_16u32u_C1R
 - image_color_processing, [712](#)
- nppiLUTPalette_16u8u_C1R
 - image_color_processing, [712](#)
- nppiLUTPalette_16u_AC4R
 - image_color_processing, [713](#)
- nppiLUTPalette_16u_C1R
 - image_color_processing, [713](#)
- nppiLUTPalette_16u_C3R
 - image_color_processing, [714](#)
- nppiLUTPalette_16u_C4R
 - image_color_processing, [714](#)
- nppiLUTPalette_8u24u_C1R
 - image_color_processing, [715](#)
- nppiLUTPalette_8u32u_C1R
 - image_color_processing, [715](#)
- nppiLUTPalette_8u_AC4R
 - image_color_processing, [716](#)
- nppiLUTPalette_8u_C1R
 - image_color_processing, [716](#)
- nppiLUTPalette_8u_C3R
 - image_color_processing, [717](#)
- nppiLUTPalette_8u_C4R
 - image_color_processing, [717](#)
- nppiLUTPaletteSwap_16u_C3A0C4R
 - image_color_processing, [718](#)
- nppiLUTPaletteSwap_8u_C3A0C4R
 - image_color_processing, [718](#)
- nppiLUVToRGB_8u_AC4R
 - image_color_model_conversion, [552](#)
- nppiLUVToRGB_8u_C3R
 - image_color_model_conversion, [553](#)
- nppiMagnitude_32fc32f_C1R
 - image_fourier_transforms, [1576](#)
- nppiMagnitudeSqr_32fc32f_C1R
 - image_fourier_transforms, [1576](#)
- nppiMalloc_16s_C1
 - image_memory_management, [2362](#)
- nppiMalloc_16s_C2
 - image_memory_management, [2362](#)
- nppiMalloc_16s_C4
 - image_memory_management, [2363](#)
- nppiMalloc_16sc_C1
 - image_memory_management, [2363](#)
- nppiMalloc_16sc_C2
 - image_memory_management, [2363](#)
- nppiMalloc_16sc_C3
 - image_memory_management, [2364](#)
- nppiMalloc_16sc_C4
 - image_memory_management, [2364](#)
- nppiMalloc_16u_C1
 - image_memory_management, [2364](#)
- nppiMalloc_16u_C2
 - image_memory_management, [2364](#)
- nppiMalloc_16u_C3
 - image_memory_management, [2365](#)
- nppiMalloc_16u_C4
 - image_memory_management, [2365](#)
- nppiMalloc_32f_C1
 - image_memory_management, [2365](#)
- nppiMalloc_32f_C2
 - image_memory_management, [2366](#)
- nppiMalloc_32f_C3
 - image_memory_management, [2366](#)
- nppiMalloc_32f_C4
 - image_memory_management, [2366](#)
- nppiMalloc_32fc_C1
 - image_memory_management, [2366](#)
- nppiMalloc_32fc_C2
 - image_memory_management, [2367](#)
- nppiMalloc_32fc_C3
 - image_memory_management, [2367](#)
- nppiMalloc_32fc_C4
 - image_memory_management, [2367](#)
- nppiMalloc_32s_C1
 - image_memory_management, [2368](#)
- nppiMalloc_32s_C3
 - image_memory_management, [2368](#)

- nppiMalloc_32s_C4
 - image_memory_management, [2368](#)
- nppiMalloc_32sc_C1
 - image_memory_management, [2368](#)
- nppiMalloc_32sc_C2
 - image_memory_management, [2369](#)
- nppiMalloc_32sc_C3
 - image_memory_management, [2369](#)
- nppiMalloc_32sc_C4
 - image_memory_management, [2369](#)
- nppiMalloc_8u_C1
 - image_memory_management, [2370](#)
- nppiMalloc_8u_C2
 - image_memory_management, [2370](#)
- nppiMalloc_8u_C3
 - image_memory_management, [2370](#)
- nppiMalloc_8u_C4
 - image_memory_management, [2370](#)
- NppiMaskSize
 - typedefs_npp, [44](#)
- nppiMax_16s_AC4R
 - image_max, [1746](#)
- nppiMax_16s_C1R
 - image_max, [1746](#)
- nppiMax_16s_C3R
 - image_max, [1747](#)
- nppiMax_16s_C4R
 - image_max, [1747](#)
- nppiMax_16u_AC4R
 - image_max, [1747](#)
- nppiMax_16u_C1R
 - image_max, [1748](#)
- nppiMax_16u_C3R
 - image_max, [1748](#)
- nppiMax_16u_C4R
 - image_max, [1749](#)
- nppiMax_32f_AC4R
 - image_max, [1749](#)
- nppiMax_32f_C1R
 - image_max, [1749](#)
- nppiMax_32f_C3R
 - image_max, [1750](#)
- nppiMax_32f_C4R
 - image_max, [1750](#)
- nppiMax_8u_AC4R
 - image_max, [1750](#)
- nppiMax_8u_C1R
 - image_max, [1751](#)
- nppiMax_8u_C3R
 - image_max, [1751](#)
- nppiMax_8u_C4R
 - image_max, [1752](#)
- nppiMaxEvery_16s_AC4IR
 - image_maxevery, [2075](#)
- nppiMaxEvery_16s_C1IR
 - image_maxevery, [2075](#)
- nppiMaxEvery_16s_C3IR
 - image_maxevery, [2076](#)
- nppiMaxEvery_16s_C4IR
 - image_maxevery, [2076](#)
- nppiMaxEvery_16u_AC4IR
 - image_maxevery, [2076](#)
- nppiMaxEvery_16u_C1IR
 - image_maxevery, [2077](#)
- nppiMaxEvery_16u_C3IR
 - image_maxevery, [2077](#)
- nppiMaxEvery_16u_C4IR
 - image_maxevery, [2077](#)
- nppiMaxEvery_32f_AC4IR
 - image_maxevery, [2078](#)
- nppiMaxEvery_32f_C1IR
 - image_maxevery, [2078](#)
- nppiMaxEvery_32f_C3IR
 - image_maxevery, [2078](#)
- nppiMaxEvery_32f_C4IR
 - image_maxevery, [2079](#)
- nppiMaxEvery_8u_AC4IR
 - image_maxevery, [2079](#)
- nppiMaxEvery_8u_C1IR
 - image_maxevery, [2079](#)
- nppiMaxEvery_8u_C3IR
 - image_maxevery, [2080](#)
- nppiMaxEvery_8u_C4IR
 - image_maxevery, [2080](#)
- nppiMaxGetBufferHostSize_16s_AC4R
 - image_max, [1752](#)
- nppiMaxGetBufferHostSize_16s_C1R
 - image_max, [1752](#)
- nppiMaxGetBufferHostSize_16s_C3R
 - image_max, [1752](#)
- nppiMaxGetBufferHostSize_16s_C4R
 - image_max, [1753](#)
- nppiMaxGetBufferHostSize_16u_AC4R
 - image_max, [1753](#)
- nppiMaxGetBufferHostSize_16u_C1R
 - image_max, [1753](#)
- nppiMaxGetBufferHostSize_16u_C3R
 - image_max, [1754](#)
- nppiMaxGetBufferHostSize_16u_C4R
 - image_max, [1754](#)
- nppiMaxGetBufferHostSize_32f_AC4R
 - image_max, [1754](#)
- nppiMaxGetBufferHostSize_32f_C1R
 - image_max, [1754](#)
- nppiMaxGetBufferHostSize_32f_C3R
 - image_max, [1755](#)
- nppiMaxGetBufferHostSize_32f_C4R
 - image_max, [1755](#)

- nppiMaxGetBufferHostSize_8u_AC4R
 - image_max, [1755](#)
- nppiMaxGetBufferHostSize_8u_C1R
 - image_max, [1756](#)
- nppiMaxGetBufferHostSize_8u_C3R
 - image_max, [1756](#)
- nppiMaxGetBufferHostSize_8u_C4R
 - image_max, [1756](#)
- nppiMaximumError_16s_C1R
 - image_maximum_error, [2269](#)
- nppiMaximumError_16s_C2R
 - image_maximum_error, [2270](#)
- nppiMaximumError_16s_C3R
 - image_maximum_error, [2270](#)
- nppiMaximumError_16s_C4R
 - image_maximum_error, [2270](#)
- nppiMaximumError_16sc_C1R
 - image_maximum_error, [2271](#)
- nppiMaximumError_16sc_C2R
 - image_maximum_error, [2271](#)
- nppiMaximumError_16sc_C3R
 - image_maximum_error, [2272](#)
- nppiMaximumError_16sc_C4R
 - image_maximum_error, [2272](#)
- nppiMaximumError_16u_C1R
 - image_maximum_error, [2273](#)
- nppiMaximumError_16u_C2R
 - image_maximum_error, [2273](#)
- nppiMaximumError_16u_C3R
 - image_maximum_error, [2273](#)
- nppiMaximumError_16u_C4R
 - image_maximum_error, [2274](#)
- nppiMaximumError_32f_C1R
 - image_maximum_error, [2274](#)
- nppiMaximumError_32f_C2R
 - image_maximum_error, [2275](#)
- nppiMaximumError_32f_C3R
 - image_maximum_error, [2275](#)
- nppiMaximumError_32f_C4R
 - image_maximum_error, [2276](#)
- nppiMaximumError_32fc_C1R
 - image_maximum_error, [2276](#)
- nppiMaximumError_32fc_C2R
 - image_maximum_error, [2277](#)
- nppiMaximumError_32fc_C3R
 - image_maximum_error, [2277](#)
- nppiMaximumError_32fc_C4R
 - image_maximum_error, [2277](#)
- nppiMaximumError_32s_C1R
 - image_maximum_error, [2278](#)
- nppiMaximumError_32s_C2R
 - image_maximum_error, [2278](#)
- nppiMaximumError_32s_C3R
 - image_maximum_error, [2279](#)
- nppiMaximumError_32s_C4R
 - image_maximum_error, [2279](#)
- nppiMaximumError_32sc_C1R
 - image_maximum_error, [2280](#)
- nppiMaximumError_32sc_C2R
 - image_maximum_error, [2280](#)
- nppiMaximumError_32sc_C3R
 - image_maximum_error, [2280](#)
- nppiMaximumError_32sc_C4R
 - image_maximum_error, [2281](#)
- nppiMaximumError_32u_C1R
 - image_maximum_error, [2281](#)
- nppiMaximumError_32u_C2R
 - image_maximum_error, [2282](#)
- nppiMaximumError_32u_C3R
 - image_maximum_error, [2282](#)
- nppiMaximumError_32u_C4R
 - image_maximum_error, [2283](#)
- nppiMaximumError_64f_C1R
 - image_maximum_error, [2283](#)
- nppiMaximumError_64f_C2R
 - image_maximum_error, [2283](#)
- nppiMaximumError_64f_C3R
 - image_maximum_error, [2284](#)
- nppiMaximumError_64f_C4R
 - image_maximum_error, [2284](#)
- nppiMaximumError_8s_C1R
 - image_maximum_error, [2285](#)
- nppiMaximumError_8s_C2R
 - image_maximum_error, [2285](#)
- nppiMaximumError_8s_C3R
 - image_maximum_error, [2286](#)
- nppiMaximumError_8s_C4R
 - image_maximum_error, [2286](#)
- nppiMaximumError_8u_C1R
 - image_maximum_error, [2286](#)
- nppiMaximumError_8u_C2R
 - image_maximum_error, [2287](#)
- nppiMaximumError_8u_C3R
 - image_maximum_error, [2287](#)
- nppiMaximumError_8u_C4R
 - image_maximum_error, [2288](#)
- nppiMaximumErrorGetBufferHostSize_16s_C1R
 - image_statistics_functions, [1676](#)
- nppiMaximumErrorGetBufferHostSize_16s_C2R
 - image_statistics_functions, [1676](#)
- nppiMaximumErrorGetBufferHostSize_16s_C3R
 - image_statistics_functions, [1676](#)
- nppiMaximumErrorGetBufferHostSize_16s_C4R
 - image_statistics_functions, [1677](#)
- nppiMaximumErrorGetBufferHostSize_16sc_C1R
 - image_statistics_functions, [1677](#)
- nppiMaximumErrorGetBufferHostSize_16sc_C2R
 - image_statistics_functions, [1677](#)

- nppiMaximumErrorGetBufferHostSize_16sc_C3R
image_statistics_functions, [1677](#)
- nppiMaximumErrorGetBufferHostSize_16sc_C4R
image_statistics_functions, [1678](#)
- nppiMaximumErrorGetBufferHostSize_16u_C1R
image_statistics_functions, [1678](#)
- nppiMaximumErrorGetBufferHostSize_16u_C2R
image_statistics_functions, [1678](#)
- nppiMaximumErrorGetBufferHostSize_16u_C3R
image_statistics_functions, [1679](#)
- nppiMaximumErrorGetBufferHostSize_16u_C4R
image_statistics_functions, [1679](#)
- nppiMaximumErrorGetBufferHostSize_32f_C1R
image_statistics_functions, [1679](#)
- nppiMaximumErrorGetBufferHostSize_32f_C2R
image_statistics_functions, [1679](#)
- nppiMaximumErrorGetBufferHostSize_32f_C3R
image_statistics_functions, [1680](#)
- nppiMaximumErrorGetBufferHostSize_32f_C4R
image_statistics_functions, [1680](#)
- nppiMaximumErrorGetBufferHostSize_32fc_C1R
image_statistics_functions, [1680](#)
- nppiMaximumErrorGetBufferHostSize_32fc_C2R
image_statistics_functions, [1681](#)
- nppiMaximumErrorGetBufferHostSize_32fc_C3R
image_statistics_functions, [1681](#)
- nppiMaximumErrorGetBufferHostSize_32fc_C4R
image_statistics_functions, [1681](#)
- nppiMaximumErrorGetBufferHostSize_32s_C1R
image_statistics_functions, [1681](#)
- nppiMaximumErrorGetBufferHostSize_32s_C2R
image_statistics_functions, [1682](#)
- nppiMaximumErrorGetBufferHostSize_32s_C3R
image_statistics_functions, [1682](#)
- nppiMaximumErrorGetBufferHostSize_32s_C4R
image_statistics_functions, [1682](#)
- nppiMaximumErrorGetBufferHostSize_32sc_C1R
image_statistics_functions, [1683](#)
- nppiMaximumErrorGetBufferHostSize_32sc_C2R
image_statistics_functions, [1683](#)
- nppiMaximumErrorGetBufferHostSize_32sc_C3R
image_statistics_functions, [1683](#)
- nppiMaximumErrorGetBufferHostSize_32sc_C4R
image_statistics_functions, [1683](#)
- nppiMaximumErrorGetBufferHostSize_32u_C1R
image_statistics_functions, [1684](#)
- nppiMaximumErrorGetBufferHostSize_32u_C2R
image_statistics_functions, [1684](#)
- nppiMaximumErrorGetBufferHostSize_32u_C3R
image_statistics_functions, [1684](#)
- nppiMaximumErrorGetBufferHostSize_32u_C4R
image_statistics_functions, [1685](#)
- nppiMaximumErrorGetBufferHostSize_64f_C1R
image_statistics_functions, [1685](#)
- nppiMaximumErrorGetBufferHostSize_64f_C2R
image_statistics_functions, [1685](#)
- nppiMaximumErrorGetBufferHostSize_64f_C3R
image_statistics_functions, [1685](#)
- nppiMaximumErrorGetBufferHostSize_64f_C4R
image_statistics_functions, [1686](#)
- nppiMaximumErrorGetBufferHostSize_8s_C1R
image_statistics_functions, [1686](#)
- nppiMaximumErrorGetBufferHostSize_8s_C2R
image_statistics_functions, [1686](#)
- nppiMaximumErrorGetBufferHostSize_8s_C3R
image_statistics_functions, [1687](#)
- nppiMaximumErrorGetBufferHostSize_8s_C4R
image_statistics_functions, [1687](#)
- nppiMaximumErrorGetBufferHostSize_8u_C1R
image_statistics_functions, [1687](#)
- nppiMaximumErrorGetBufferHostSize_8u_C2R
image_statistics_functions, [1687](#)
- nppiMaximumErrorGetBufferHostSize_8u_C3R
image_statistics_functions, [1688](#)
- nppiMaximumErrorGetBufferHostSize_8u_C4R
image_statistics_functions, [1688](#)
- nppiMaximumRelativeError_16s_C1R
image_maximum_relative_error, [2315](#)
- nppiMaximumRelativeError_16s_C2R
image_maximum_relative_error, [2316](#)
- nppiMaximumRelativeError_16s_C3R
image_maximum_relative_error, [2316](#)
- nppiMaximumRelativeError_16s_C4R
image_maximum_relative_error, [2317](#)
- nppiMaximumRelativeError_16sc_C1R
image_maximum_relative_error, [2317](#)
- nppiMaximumRelativeError_16sc_C2R
image_maximum_relative_error, [2318](#)
- nppiMaximumRelativeError_16sc_C3R
image_maximum_relative_error, [2318](#)
- nppiMaximumRelativeError_16sc_C4R
image_maximum_relative_error, [2318](#)
- nppiMaximumRelativeError_16u_C1R
image_maximum_relative_error, [2319](#)
- nppiMaximumRelativeError_16u_C2R
image_maximum_relative_error, [2319](#)
- nppiMaximumRelativeError_16u_C3R
image_maximum_relative_error, [2320](#)
- nppiMaximumRelativeError_16u_C4R
image_maximum_relative_error, [2320](#)
- nppiMaximumRelativeError_32f_C1R
image_maximum_relative_error, [2321](#)
- nppiMaximumRelativeError_32f_C2R
image_maximum_relative_error, [2321](#)
- nppiMaximumRelativeError_32f_C3R
image_maximum_relative_error, [2322](#)
- nppiMaximumRelativeError_32f_C4R
image_maximum_relative_error, [2322](#)

- nppiMaximumRelativeError_32fc_C1R
 - image_maximum_relative_error, [2323](#)
- nppiMaximumRelativeError_32fc_C2R
 - image_maximum_relative_error, [2323](#)
- nppiMaximumRelativeError_32fc_C3R
 - image_maximum_relative_error, [2323](#)
- nppiMaximumRelativeError_32fc_C4R
 - image_maximum_relative_error, [2324](#)
- nppiMaximumRelativeError_32s_C1R
 - image_maximum_relative_error, [2324](#)
- nppiMaximumRelativeError_32s_C2R
 - image_maximum_relative_error, [2325](#)
- nppiMaximumRelativeError_32s_C3R
 - image_maximum_relative_error, [2325](#)
- nppiMaximumRelativeError_32s_C4R
 - image_maximum_relative_error, [2326](#)
- nppiMaximumRelativeError_32sc_C1R
 - image_maximum_relative_error, [2326](#)
- nppiMaximumRelativeError_32sc_C2R
 - image_maximum_relative_error, [2327](#)
- nppiMaximumRelativeError_32sc_C3R
 - image_maximum_relative_error, [2327](#)
- nppiMaximumRelativeError_32sc_C4R
 - image_maximum_relative_error, [2328](#)
- nppiMaximumRelativeError_32u_C1R
 - image_maximum_relative_error, [2328](#)
- nppiMaximumRelativeError_32u_C2R
 - image_maximum_relative_error, [2328](#)
- nppiMaximumRelativeError_32u_C3R
 - image_maximum_relative_error, [2329](#)
- nppiMaximumRelativeError_32u_C4R
 - image_maximum_relative_error, [2329](#)
- nppiMaximumRelativeError_64f_C1R
 - image_maximum_relative_error, [2330](#)
- nppiMaximumRelativeError_64f_C2R
 - image_maximum_relative_error, [2330](#)
- nppiMaximumRelativeError_64f_C3R
 - image_maximum_relative_error, [2331](#)
- nppiMaximumRelativeError_64f_C4R
 - image_maximum_relative_error, [2331](#)
- nppiMaximumRelativeError_8s_C1R
 - image_maximum_relative_error, [2332](#)
- nppiMaximumRelativeError_8s_C2R
 - image_maximum_relative_error, [2332](#)
- nppiMaximumRelativeError_8s_C3R
 - image_maximum_relative_error, [2333](#)
- nppiMaximumRelativeError_8s_C4R
 - image_maximum_relative_error, [2333](#)
- nppiMaximumRelativeError_8u_C1R
 - image_maximum_relative_error, [2333](#)
- nppiMaximumRelativeError_8u_C2R
 - image_maximum_relative_error, [2334](#)
- nppiMaximumRelativeError_8u_C3R
 - image_maximum_relative_error, [2334](#)
- nppiMaximumRelativeError_8u_C4R
 - image_maximum_relative_error, [2335](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16s_C1R
 - image_statistics_functions, [1688](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16s_C2R
 - image_statistics_functions, [1689](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16s_C3R
 - image_statistics_functions, [1689](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16s_C4R
 - image_statistics_functions, [1689](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16sc_C1R
 - image_statistics_functions, [1689](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16sc_C2R
 - image_statistics_functions, [1690](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16sc_C3R
 - image_statistics_functions, [1690](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16sc_C4R
 - image_statistics_functions, [1690](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16u_C1R
 - image_statistics_functions, [1691](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16u_C2R
 - image_statistics_functions, [1691](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16u_C3R
 - image_statistics_functions, [1691](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-16u_C4R
 - image_statistics_functions, [1691](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32f_C1R
 - image_statistics_functions, [1692](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32f_C2R
 - image_statistics_functions, [1692](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32f_C3R
 - image_statistics_functions, [1692](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32f_C4R
 - image_statistics_functions, [1693](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32fc_C1R
 - image_statistics_functions, [1693](#)

- nppiMaximumRelativeErrorGetBufferHostSize_-32fc_C2R
image_statistics_functions, [1693](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32fc_C3R
image_statistics_functions, [1693](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32fc_C4R
image_statistics_functions, [1694](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32s_C1R
image_statistics_functions, [1694](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32s_C2R
image_statistics_functions, [1694](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32s_C3R
image_statistics_functions, [1695](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32s_C4R
image_statistics_functions, [1695](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C1R
image_statistics_functions, [1695](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C2R
image_statistics_functions, [1695](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C3R
image_statistics_functions, [1696](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32sc_C4R
image_statistics_functions, [1696](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C1R
image_statistics_functions, [1696](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C2R
image_statistics_functions, [1697](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C3R
image_statistics_functions, [1697](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-32u_C4R
image_statistics_functions, [1697](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C1R
image_statistics_functions, [1697](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C2R
image_statistics_functions, [1698](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C3R
image_statistics_functions, [1698](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-64f_C4R
image_statistics_functions, [1698](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C1R
image_statistics_functions, [1699](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C2R
image_statistics_functions, [1699](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C3R
image_statistics_functions, [1699](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8s_C4R
image_statistics_functions, [1699](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C1R
image_statistics_functions, [1700](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C2R
image_statistics_functions, [1700](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C3R
image_statistics_functions, [1700](#)
- nppiMaximumRelativeErrorGetBufferHostSize_-8u_C4R
image_statistics_functions, [1701](#)
- nppiMaxIdx_16s_AC4R
image_max_index, [1759](#)
- nppiMaxIdx_16s_C1R
image_max_index, [1760](#)
- nppiMaxIdx_16s_C3R
image_max_index, [1760](#)
- nppiMaxIdx_16s_C4R
image_max_index, [1760](#)
- nppiMaxIdx_16u_AC4R
image_max_index, [1761](#)
- nppiMaxIdx_16u_C1R
image_max_index, [1761](#)
- nppiMaxIdx_16u_C3R
image_max_index, [1762](#)
- nppiMaxIdx_16u_C4R
image_max_index, [1762](#)
- nppiMaxIdx_32f_AC4R
image_max_index, [1762](#)
- nppiMaxIdx_32f_C1R
image_max_index, [1763](#)
- nppiMaxIdx_32f_C3R
image_max_index, [1763](#)
- nppiMaxIdx_32f_C4R
image_max_index, [1764](#)
- nppiMaxIdx_8u_AC4R
image_max_index, [1764](#)
- nppiMaxIdx_8u_C1R

- image_max_index, [1764](#)
- nppiMaxIndx_8u_C3R
 - image_max_index, [1765](#)
- nppiMaxIndx_8u_C4R
 - image_max_index, [1765](#)
- nppiMaxIndxGetBufferHostSize_16s_AC4R
 - image_max_index, [1766](#)
- nppiMaxIndxGetBufferHostSize_16s_C1R
 - image_max_index, [1766](#)
- nppiMaxIndxGetBufferHostSize_16s_C3R
 - image_max_index, [1766](#)
- nppiMaxIndxGetBufferHostSize_16s_C4R
 - image_max_index, [1767](#)
- nppiMaxIndxGetBufferHostSize_16u_AC4R
 - image_max_index, [1767](#)
- nppiMaxIndxGetBufferHostSize_16u_C1R
 - image_max_index, [1767](#)
- nppiMaxIndxGetBufferHostSize_16u_C3R
 - image_max_index, [1767](#)
- nppiMaxIndxGetBufferHostSize_16u_C4R
 - image_max_index, [1768](#)
- nppiMaxIndxGetBufferHostSize_32f_AC4R
 - image_max_index, [1768](#)
- nppiMaxIndxGetBufferHostSize_32f_C1R
 - image_max_index, [1768](#)
- nppiMaxIndxGetBufferHostSize_32f_C3R
 - image_max_index, [1769](#)
- nppiMaxIndxGetBufferHostSize_32f_C4R
 - image_max_index, [1769](#)
- nppiMaxIndxGetBufferHostSize_8u_AC4R
 - image_max_index, [1769](#)
- nppiMaxIndxGetBufferHostSize_8u_C1R
 - image_max_index, [1769](#)
- nppiMaxIndxGetBufferHostSize_8u_C3R
 - image_max_index, [1770](#)
- nppiMaxIndxGetBufferHostSize_8u_C4R
 - image_max_index, [1770](#)
- nppiMean_16s_AC4R
 - image_mean, [1806](#)
- nppiMean_16s_C1R
 - image_mean, [1806](#)
- nppiMean_16s_C3R
 - image_mean, [1806](#)
- nppiMean_16s_C4R
 - image_mean, [1807](#)
- nppiMean_16u_AC4R
 - image_mean, [1807](#)
- nppiMean_16u_C1MR
 - image_mean, [1807](#)
- nppiMean_16u_C1R
 - image_mean, [1808](#)
- nppiMean_16u_C3CMR
 - image_mean, [1808](#)
- nppiMean_16u_C3R
 - image_mean, [1809](#)
- nppiMean_32f_AC4R
 - image_mean, [1809](#)
- nppiMean_32f_C1MR
 - image_mean, [1810](#)
- nppiMean_32f_C1R
 - image_mean, [1810](#)
- nppiMean_32f_C3CMR
 - image_mean, [1810](#)
- nppiMean_32f_C3R
 - image_mean, [1811](#)
- nppiMean_32f_C4R
 - image_mean, [1811](#)
- nppiMean_8s_C1MR
 - image_mean, [1812](#)
- nppiMean_8s_C3CMR
 - image_mean, [1812](#)
- nppiMean_8u_AC4R
 - image_mean, [1813](#)
- nppiMean_8u_C1MR
 - image_mean, [1813](#)
- nppiMean_8u_C1R
 - image_mean, [1813](#)
- nppiMean_8u_C3CMR
 - image_mean, [1814](#)
- nppiMean_8u_C3R
 - image_mean, [1814](#)
- nppiMean_8u_C4R
 - image_mean, [1815](#)
- nppiMean_StdDev_16u_C1MR
 - image_mean_stddev, [1826](#)
- nppiMean_StdDev_16u_C1R
 - image_mean_stddev, [1826](#)
- nppiMean_StdDev_16u_C3CMR
 - image_mean_stddev, [1827](#)
- nppiMean_StdDev_16u_C3CR
 - image_mean_stddev, [1827](#)
- nppiMean_StdDev_32f_C1MR
 - image_mean_stddev, [1828](#)
- nppiMean_StdDev_32f_C1R
 - image_mean_stddev, [1828](#)
- nppiMean_StdDev_32f_C3CMR
 - image_mean_stddev, [1829](#)
- nppiMean_StdDev_32f_C3CR
 - image_mean_stddev, [1829](#)
- nppiMean_StdDev_8s_C1MR
 - image_mean_stddev, [1830](#)
- nppiMean_StdDev_8s_C1R
 - image_mean_stddev, [1830](#)
- nppiMean_StdDev_8s_C3CMR
 - image_mean_stddev, [1831](#)
- nppiMean_StdDev_8s_C3CR

- image_mean_stddev, [1831](#)
- nppiMean_StdDev_8u_C1MR
 - image_mean_stddev, [1832](#)
- nppiMean_StdDev_8u_C1R
 - image_mean_stddev, [1832](#)
- nppiMean_StdDev_8u_C3CMR
 - image_mean_stddev, [1833](#)
- nppiMean_StdDev_8u_C3CR
 - image_mean_stddev, [1833](#)
- nppiMeanGetBufferHostSize_16s_AC4R
 - image_mean, [1815](#)
- nppiMeanGetBufferHostSize_16s_C1R
 - image_mean, [1815](#)
- nppiMeanGetBufferHostSize_16s_C3R
 - image_mean, [1816](#)
- nppiMeanGetBufferHostSize_16s_C4R
 - image_mean, [1816](#)
- nppiMeanGetBufferHostSize_16u_AC4R
 - image_mean, [1816](#)
- nppiMeanGetBufferHostSize_16u_C1MR
 - image_mean, [1816](#)
- nppiMeanGetBufferHostSize_16u_C1R
 - image_mean, [1817](#)
- nppiMeanGetBufferHostSize_16u_C3CMR
 - image_mean, [1817](#)
- nppiMeanGetBufferHostSize_16u_C3R
 - image_mean, [1817](#)
- nppiMeanGetBufferHostSize_16u_C4R
 - image_mean, [1818](#)
- nppiMeanGetBufferHostSize_32f_AC4R
 - image_mean, [1818](#)
- nppiMeanGetBufferHostSize_32f_C1MR
 - image_mean, [1818](#)
- nppiMeanGetBufferHostSize_32f_C1R
 - image_mean, [1818](#)
- nppiMeanGetBufferHostSize_32f_C3CMR
 - image_mean, [1819](#)
- nppiMeanGetBufferHostSize_32f_C3R
 - image_mean, [1819](#)
- nppiMeanGetBufferHostSize_32f_C4R
 - image_mean, [1819](#)
- nppiMeanGetBufferHostSize_8s_C1MR
 - image_mean, [1820](#)
- nppiMeanGetBufferHostSize_8s_C3CMR
 - image_mean, [1820](#)
- nppiMeanGetBufferHostSize_8u_AC4R
 - image_mean, [1820](#)
- nppiMeanGetBufferHostSize_8u_C1MR
 - image_mean, [1820](#)
- nppiMeanGetBufferHostSize_8u_C1R
 - image_mean, [1821](#)
- nppiMeanGetBufferHostSize_8u_C3CMR
 - image_mean, [1821](#)
- nppiMeanGetBufferHostSize_8u_C3R
 - image_mean, [1821](#)
- nppiMeanGetBufferHostSize_8u_C4R
 - image_mean, [1822](#)
- nppiMeanStdDevGetBufferHostSize_16u_C1MR
 - image_mean_stddev, [1834](#)
- nppiMeanStdDevGetBufferHostSize_16u_C1R
 - image_mean_stddev, [1834](#)
- nppiMeanStdDevGetBufferHostSize_16u_C3CMR
 - image_mean_stddev, [1834](#)
- nppiMeanStdDevGetBufferHostSize_16u_C3CR
 - image_mean_stddev, [1835](#)
- nppiMeanStdDevGetBufferHostSize_32f_C1MR
 - image_mean_stddev, [1835](#)
- nppiMeanStdDevGetBufferHostSize_32f_C1R
 - image_mean_stddev, [1835](#)
- nppiMeanStdDevGetBufferHostSize_32f_C3CMR
 - image_mean_stddev, [1836](#)
- nppiMeanStdDevGetBufferHostSize_32f_C3CR
 - image_mean_stddev, [1836](#)
- nppiMeanStdDevGetBufferHostSize_8s_C1MR
 - image_mean_stddev, [1836](#)
- nppiMeanStdDevGetBufferHostSize_8s_C1R
 - image_mean_stddev, [1836](#)
- nppiMeanStdDevGetBufferHostSize_8s_C3CMR
 - image_mean_stddev, [1837](#)
- nppiMeanStdDevGetBufferHostSize_8s_C3CR
 - image_mean_stddev, [1837](#)
- nppiMeanStdDevGetBufferHostSize_8u_C1MR
 - image_mean_stddev, [1837](#)
- nppiMeanStdDevGetBufferHostSize_8u_C1R
 - image_mean_stddev, [1838](#)
- nppiMeanStdDevGetBufferHostSize_8u_C3CMR
 - image_mean_stddev, [1838](#)
- nppiMeanStdDevGetBufferHostSize_8u_C3CR
 - image_mean_stddev, [1838](#)
- nppiMin_16s_AC4R
 - image_min, [1719](#)
- nppiMin_16s_C1R
 - image_min, [1719](#)
- nppiMin_16s_C3R
 - image_min, [1720](#)
- nppiMin_16s_C4R
 - image_min, [1720](#)
- nppiMin_16u_AC4R
 - image_min, [1720](#)
- nppiMin_16u_C1R
 - image_min, [1721](#)
- nppiMin_16u_C3R
 - image_min, [1721](#)
- nppiMin_16u_C4R
 - image_min, [1722](#)
- nppiMin_32f_AC4R
 - image_min, [1722](#)
- nppiMin_32f_C1R
 - image_min, [1722](#)

- image_min, [1722](#)
- nppiMin_32f_C3R
 - image_min, [1723](#)
- nppiMin_32f_C4R
 - image_min, [1723](#)
- nppiMin_8u_AC4R
 - image_min, [1723](#)
- nppiMin_8u_C1R
 - image_min, [1724](#)
- nppiMin_8u_C3R
 - image_min, [1724](#)
- nppiMin_8u_C4R
 - image_min, [1725](#)
- nppiMinEvery_16s_AC4IR
 - image_minevery, [2082](#)
- nppiMinEvery_16s_C1IR
 - image_minevery, [2082](#)
- nppiMinEvery_16s_C3IR
 - image_minevery, [2083](#)
- nppiMinEvery_16s_C4IR
 - image_minevery, [2083](#)
- nppiMinEvery_16u_AC4IR
 - image_minevery, [2083](#)
- nppiMinEvery_16u_C1IR
 - image_minevery, [2084](#)
- nppiMinEvery_16u_C3IR
 - image_minevery, [2084](#)
- nppiMinEvery_16u_C4IR
 - image_minevery, [2084](#)
- nppiMinEvery_32f_AC4IR
 - image_minevery, [2085](#)
- nppiMinEvery_32f_C1IR
 - image_minevery, [2085](#)
- nppiMinEvery_32f_C3IR
 - image_minevery, [2085](#)
- nppiMinEvery_32f_C4IR
 - image_minevery, [2086](#)
- nppiMinEvery_8u_AC4IR
 - image_minevery, [2086](#)
- nppiMinEvery_8u_C1IR
 - image_minevery, [2086](#)
- nppiMinEvery_8u_C3IR
 - image_minevery, [2087](#)
- nppiMinEvery_8u_C4IR
 - image_minevery, [2087](#)
- nppiMinGetBufferHostSize_16s_AC4R
 - image_min, [1725](#)
- nppiMinGetBufferHostSize_16s_C1R
 - image_min, [1725](#)
- nppiMinGetBufferHostSize_16s_C3R
 - image_min, [1725](#)
- nppiMinGetBufferHostSize_16s_C4R
 - image_min, [1726](#)
- nppiMinGetBufferHostSize_16u_AC4R
 - image_min, [1726](#)
- nppiMinGetBufferHostSize_16u_C1R
 - image_min, [1726](#)
- nppiMinGetBufferHostSize_16u_C3R
 - image_min, [1727](#)
- nppiMinGetBufferHostSize_16u_C4R
 - image_min, [1727](#)
- nppiMinGetBufferHostSize_32f_AC4R
 - image_min, [1727](#)
- nppiMinGetBufferHostSize_32f_C1R
 - image_min, [1727](#)
- nppiMinGetBufferHostSize_32f_C3R
 - image_min, [1728](#)
- nppiMinGetBufferHostSize_32f_C4R
 - image_min, [1728](#)
- nppiMinGetBufferHostSize_8u_AC4R
 - image_min, [1728](#)
- nppiMinGetBufferHostSize_8u_C1R
 - image_min, [1729](#)
- nppiMinGetBufferHostSize_8u_C3R
 - image_min, [1729](#)
- nppiMinGetBufferHostSize_8u_C4R
 - image_min, [1729](#)
- nppiMinIndx_16s_AC4R
 - image_min_index, [1732](#)
- nppiMinIndx_16s_C1R
 - image_min_index, [1733](#)
- nppiMinIndx_16s_C3R
 - image_min_index, [1733](#)
- nppiMinIndx_16s_C4R
 - image_min_index, [1733](#)
- nppiMinIndx_16u_AC4R
 - image_min_index, [1734](#)
- nppiMinIndx_16u_C1R
 - image_min_index, [1734](#)
- nppiMinIndx_16u_C3R
 - image_min_index, [1735](#)
- nppiMinIndx_16u_C4R
 - image_min_index, [1735](#)
- nppiMinIndx_32f_AC4R
 - image_min_index, [1735](#)
- nppiMinIndx_32f_C1R
 - image_min_index, [1736](#)
- nppiMinIndx_32f_C3R
 - image_min_index, [1736](#)
- nppiMinIndx_32f_C4R
 - image_min_index, [1737](#)
- nppiMinIndx_8u_AC4R
 - image_min_index, [1737](#)
- nppiMinIndx_8u_C1R
 - image_min_index, [1737](#)
- nppiMinIndx_8u_C3R
 - image_min_index, [1738](#)
- nppiMinIndx_8u_C4R

- image_min_index, [1738](#)
- nppiMinIdxGetBufferHostSize_16s_AC4R
 - image_min_index, [1739](#)
- nppiMinIdxGetBufferHostSize_16s_C1R
 - image_min_index, [1739](#)
- nppiMinIdxGetBufferHostSize_16s_C3R
 - image_min_index, [1739](#)
- nppiMinIdxGetBufferHostSize_16s_C4R
 - image_min_index, [1740](#)
- nppiMinIdxGetBufferHostSize_16u_AC4R
 - image_min_index, [1740](#)
- nppiMinIdxGetBufferHostSize_16u_C1R
 - image_min_index, [1740](#)
- nppiMinIdxGetBufferHostSize_16u_C3R
 - image_min_index, [1740](#)
- nppiMinIdxGetBufferHostSize_16u_C4R
 - image_min_index, [1741](#)
- nppiMinIdxGetBufferHostSize_32f_AC4R
 - image_min_index, [1741](#)
- nppiMinIdxGetBufferHostSize_32f_C1R
 - image_min_index, [1741](#)
- nppiMinIdxGetBufferHostSize_32f_C3R
 - image_min_index, [1742](#)
- nppiMinIdxGetBufferHostSize_32f_C4R
 - image_min_index, [1742](#)
- nppiMinIdxGetBufferHostSize_8u_AC4R
 - image_min_index, [1742](#)
- nppiMinIdxGetBufferHostSize_8u_C1R
 - image_min_index, [1742](#)
- nppiMinIdxGetBufferHostSize_8u_C3R
 - image_min_index, [1743](#)
- nppiMinIdxGetBufferHostSize_8u_C4R
 - image_min_index, [1743](#)
- nppiMinMax_16s_AC4R
 - image_min_max, [1773](#)
- nppiMinMax_16s_C1R
 - image_min_max, [1773](#)
- nppiMinMax_16s_C3R
 - image_min_max, [1774](#)
- nppiMinMax_16s_C4R
 - image_min_max, [1774](#)
- nppiMinMax_16u_AC4R
 - image_min_max, [1775](#)
- nppiMinMax_16u_C1R
 - image_min_max, [1775](#)
- nppiMinMax_16u_C3R
 - image_min_max, [1775](#)
- nppiMinMax_16u_C4R
 - image_min_max, [1776](#)
- nppiMinMax_32f_AC4R
 - image_min_max, [1776](#)
- nppiMinMax_32f_C1R
 - image_min_max, [1777](#)
- nppiMinMax_32f_C3R
 - image_min_max, [1777](#)
- nppiMinMax_32f_C4R
 - image_min_max, [1777](#)
- nppiMinMax_8u_AC4R
 - image_min_max, [1778](#)
- nppiMinMax_8u_C1R
 - image_min_max, [1778](#)
- nppiMinMax_8u_C3R
 - image_min_max, [1779](#)
- nppiMinMax_8u_C4R
 - image_min_max, [1779](#)
- nppiMinMaxGetBufferHostSize_16s_AC4R
 - image_min_max, [1779](#)
- nppiMinMaxGetBufferHostSize_16s_C1R
 - image_min_max, [1780](#)
- nppiMinMaxGetBufferHostSize_16s_C3R
 - image_min_max, [1780](#)
- nppiMinMaxGetBufferHostSize_16s_C4R
 - image_min_max, [1780](#)
- nppiMinMaxGetBufferHostSize_16u_AC4R
 - image_min_max, [1781](#)
- nppiMinMaxGetBufferHostSize_16u_C1R
 - image_min_max, [1781](#)
- nppiMinMaxGetBufferHostSize_16u_C3R
 - image_min_max, [1781](#)
- nppiMinMaxGetBufferHostSize_16u_C4R
 - image_min_max, [1781](#)
- nppiMinMaxGetBufferHostSize_32f_AC4R
 - image_min_max, [1782](#)
- nppiMinMaxGetBufferHostSize_32f_C1R
 - image_min_max, [1782](#)
- nppiMinMaxGetBufferHostSize_32f_C3R
 - image_min_max, [1782](#)
- nppiMinMaxGetBufferHostSize_32f_C4R
 - image_min_max, [1783](#)
- nppiMinMaxGetBufferHostSize_8u_AC4R
 - image_min_max, [1783](#)
- nppiMinMaxGetBufferHostSize_8u_C1R
 - image_min_max, [1783](#)
- nppiMinMaxGetBufferHostSize_8u_C3R
 - image_min_max, [1783](#)
- nppiMinMaxGetBufferHostSize_8u_C4R
 - image_min_max, [1784](#)
- nppiMinMaxIdx_16u_C1MR
 - image_min_max_index, [1788](#)
- nppiMinMaxIdx_16u_C1R
 - image_min_max_index, [1789](#)
- nppiMinMaxIdx_16u_C3CMR
 - image_min_max_index, [1789](#)
- nppiMinMaxIdx_16u_C3CR
 - image_min_max_index, [1790](#)
- nppiMinMaxIdx_32f_C1MR
 - image_min_max_index, [1790](#)
- nppiMinMaxIdx_32f_C1R

- image_min_max_index, [1791](#)
- nppiMinMaxIdx_32f_C3CMR
 - image_min_max_index, [1791](#)
- nppiMinMaxIdx_32f_C3CR
 - image_min_max_index, [1792](#)
- nppiMinMaxIdx_8s_C1MR
 - image_min_max_index, [1793](#)
- nppiMinMaxIdx_8s_C1R
 - image_min_max_index, [1793](#)
- nppiMinMaxIdx_8s_C3CMR
 - image_min_max_index, [1794](#)
- nppiMinMaxIdx_8s_C3CR
 - image_min_max_index, [1794](#)
- nppiMinMaxIdx_8u_C1MR
 - image_min_max_index, [1795](#)
- nppiMinMaxIdx_8u_C1R
 - image_min_max_index, [1795](#)
- nppiMinMaxIdx_8u_C3CMR
 - image_min_max_index, [1796](#)
- nppiMinMaxIdx_8u_C3CR
 - image_min_max_index, [1796](#)
- nppiMinMaxIdxGetBufferHostSize_16u_C1MR
 - image_min_max_index, [1797](#)
- nppiMinMaxIdxGetBufferHostSize_16u_C1R
 - image_min_max_index, [1797](#)
- nppiMinMaxIdxGetBufferHostSize_16u_C3CMR
 - image_min_max_index, [1797](#)
- nppiMinMaxIdxGetBufferHostSize_16u_C3CR
 - image_min_max_index, [1798](#)
- nppiMinMaxIdxGetBufferHostSize_32f_C1MR
 - image_min_max_index, [1798](#)
- nppiMinMaxIdxGetBufferHostSize_32f_C1R
 - image_min_max_index, [1798](#)
- nppiMinMaxIdxGetBufferHostSize_32f_C3CMR
 - image_min_max_index, [1799](#)
- nppiMinMaxIdxGetBufferHostSize_32f_C3CR
 - image_min_max_index, [1799](#)
- nppiMinMaxIdxGetBufferHostSize_8s_C1MR
 - image_min_max_index, [1799](#)
- nppiMinMaxIdxGetBufferHostSize_8s_C1R
 - image_min_max_index, [1799](#)
- nppiMinMaxIdxGetBufferHostSize_8s_C3CMR
 - image_min_max_index, [1800](#)
- nppiMinMaxIdxGetBufferHostSize_8s_C3CR
 - image_min_max_index, [1800](#)
- nppiMinMaxIdxGetBufferHostSize_8u_C1MR
 - image_min_max_index, [1800](#)
- nppiMinMaxIdxGetBufferHostSize_8u_C1R
 - image_min_max_index, [1801](#)
- nppiMinMaxIdxGetBufferHostSize_8u_C3CMR
 - image_min_max_index, [1801](#)
- nppiMinMaxIdxGetBufferHostSize_8u_C3CR
 - image_min_max_index, [1801](#)
- nppiMirror_16s_AC4R
 - image_mirror, [1465](#)
- nppiMirror_16s_AC4R
 - image_mirror, [1465](#)
- nppiMirror_16s_C1IR
 - image_mirror, [1466](#)
- nppiMirror_16s_C1R
 - image_mirror, [1466](#)
- nppiMirror_16s_C3IR
 - image_mirror, [1466](#)
- nppiMirror_16s_C3R
 - image_mirror, [1467](#)
- nppiMirror_16s_C4IR
 - image_mirror, [1467](#)
- nppiMirror_16s_C4R
 - image_mirror, [1467](#)
- nppiMirror_16u_AC4IR
 - image_mirror, [1468](#)
- nppiMirror_16u_AC4R
 - image_mirror, [1468](#)
- nppiMirror_16u_C1IR
 - image_mirror, [1468](#)
- nppiMirror_16u_C1R
 - image_mirror, [1469](#)
- nppiMirror_16u_C3IR
 - image_mirror, [1469](#)
- nppiMirror_16u_C3R
 - image_mirror, [1469](#)
- nppiMirror_16u_C4IR
 - image_mirror, [1470](#)
- nppiMirror_16u_C4R
 - image_mirror, [1470](#)
- nppiMirror_32f_AC4IR
 - image_mirror, [1470](#)
- nppiMirror_32f_AC4R
 - image_mirror, [1471](#)
- nppiMirror_32f_C1IR
 - image_mirror, [1471](#)
- nppiMirror_32f_C1R
 - image_mirror, [1471](#)
- nppiMirror_32f_C3IR
 - image_mirror, [1472](#)
- nppiMirror_32f_C3R
 - image_mirror, [1472](#)
- nppiMirror_32f_C4IR
 - image_mirror, [1472](#)
- nppiMirror_32f_C4R
 - image_mirror, [1473](#)
- nppiMirror_32s_AC4IR
 - image_mirror, [1473](#)
- nppiMirror_32s_AC4R
 - image_mirror, [1473](#)
- nppiMirror_32s_C1IR
 - image_mirror, [1474](#)
- nppiMirror_32s_C1R
 - image_mirror, [1474](#)

- image_mirror, [1474](#)
- nppiMirror_32s_C3IR
 - image_mirror, [1474](#)
- nppiMirror_32s_C3R
 - image_mirror, [1475](#)
- nppiMirror_32s_C4IR
 - image_mirror, [1475](#)
- nppiMirror_32s_C4R
 - image_mirror, [1475](#)
- nppiMirror_8u_AC4IR
 - image_mirror, [1476](#)
- nppiMirror_8u_AC4R
 - image_mirror, [1476](#)
- nppiMirror_8u_C1IR
 - image_mirror, [1476](#)
- nppiMirror_8u_C1R
 - image_mirror, [1477](#)
- nppiMirror_8u_C3IR
 - image_mirror, [1477](#)
- nppiMirror_8u_C3R
 - image_mirror, [1477](#)
- nppiMirror_8u_C4IR
 - image_mirror, [1478](#)
- nppiMirror_8u_C4R
 - image_mirror, [1478](#)
- nppiMul_16s_AC4IRSfs
 - image_mul, [214](#)
- nppiMul_16s_AC4RSfs
 - image_mul, [214](#)
- nppiMul_16s_C1IRSfs
 - image_mul, [215](#)
- nppiMul_16s_C1RSfs
 - image_mul, [215](#)
- nppiMul_16s_C3IRSfs
 - image_mul, [216](#)
- nppiMul_16s_C3RSfs
 - image_mul, [216](#)
- nppiMul_16s_C4IRSfs
 - image_mul, [216](#)
- nppiMul_16s_C4RSfs
 - image_mul, [217](#)
- nppiMul_16sc_AC4IRSfs
 - image_mul, [217](#)
- nppiMul_16sc_AC4RSfs
 - image_mul, [218](#)
- nppiMul_16sc_C1IRSfs
 - image_mul, [218](#)
- nppiMul_16sc_C1RSfs
 - image_mul, [218](#)
- nppiMul_16sc_C3IRSfs
 - image_mul, [219](#)
- nppiMul_16sc_C3RSfs
 - image_mul, [219](#)
- nppiMul_16u_AC4IRSfs
 - image_mul, [220](#)
- nppiMul_16u_AC4RSfs
 - image_mul, [220](#)
- nppiMul_16u_C1IRSfs
 - image_mul, [221](#)
- nppiMul_16u_C1RSfs
 - image_mul, [221](#)
- nppiMul_16u_C3IRSfs
 - image_mul, [221](#)
- nppiMul_16u_C3RSfs
 - image_mul, [222](#)
- nppiMul_16u_C4IRSfs
 - image_mul, [222](#)
- nppiMul_16u_C4RSfs
 - image_mul, [223](#)
- nppiMul_32f_AC4IR
 - image_mul, [223](#)
- nppiMul_32f_AC4R
 - image_mul, [223](#)
- nppiMul_32f_C1IR
 - image_mul, [224](#)
- nppiMul_32f_C1R
 - image_mul, [224](#)
- nppiMul_32f_C3IR
 - image_mul, [225](#)
- nppiMul_32f_C3R
 - image_mul, [225](#)
- nppiMul_32f_C4IR
 - image_mul, [225](#)
- nppiMul_32f_C4R
 - image_mul, [226](#)
- nppiMul_32fc_AC4IR
 - image_mul, [226](#)
- nppiMul_32fc_AC4R
 - image_mul, [226](#)
- nppiMul_32fc_C1IR
 - image_mul, [227](#)
- nppiMul_32fc_C1R
 - image_mul, [227](#)
- nppiMul_32fc_C3IR
 - image_mul, [228](#)
- nppiMul_32fc_C3R
 - image_mul, [228](#)
- nppiMul_32fc_C4IR
 - image_mul, [228](#)
- nppiMul_32fc_C4R
 - image_mul, [229](#)
- nppiMul_32s_C1IRSfs
 - image_mul, [229](#)
- nppiMul_32s_C1R
 - image_mul, [230](#)
- nppiMul_32s_C1RSfs
 - image_mul, [230](#)
- nppiMul_32s_C3IRSfs

image_mul, 230
nppiMul_32s_C3RSfs
image_mul, 231
nppiMul_32sc_AC4IRSfs
image_mul, 231
nppiMul_32sc_AC4RSfs
image_mul, 232
nppiMul_32sc_C1IRSfs
image_mul, 232
nppiMul_32sc_C1RSfs
image_mul, 232
nppiMul_32sc_C3IRSfs
image_mul, 233
nppiMul_32sc_C3RSfs
image_mul, 233
nppiMul_8u_AC4IRSfs
image_mul, 234
nppiMul_8u_AC4RSfs
image_mul, 234
nppiMul_8u_C1IRSfs
image_mul, 235
nppiMul_8u_C1RSfs
image_mul, 235
nppiMul_8u_C3IRSfs
image_mul, 235
nppiMul_8u_C3RSfs
image_mul, 236
nppiMul_8u_C4IRSfs
image_mul, 236
nppiMul_8u_C4RSfs
image_mul, 237
nppiMulC_16s_AC4IRSfs
image_mulc, 87
nppiMulC_16s_AC4RSfs
image_mulc, 87
nppiMulC_16s_C1IRSfs
image_mulc, 87
nppiMulC_16s_C1RSfs
image_mulc, 88
nppiMulC_16s_C3IRSfs
image_mulc, 88
nppiMulC_16s_C3RSfs
image_mulc, 88
nppiMulC_16s_C4IRSfs
image_mulc, 89
nppiMulC_16s_C4RSfs
image_mulc, 89
nppiMulC_16sc_AC4IRSfs
image_mulc, 90
nppiMulC_16sc_AC4RSfs
image_mulc, 90
nppiMulC_16sc_C1IRSfs
image_mulc, 90
nppiMulC_16sc_C1RSfs
image_mulc, 91
nppiMulC_16sc_C3IRSfs
image_mulc, 91
nppiMulC_16sc_C3RSfs
image_mulc, 92
nppiMulC_16u_AC4IRSfs
image_mulc, 92
nppiMulC_16u_AC4RSfs
image_mulc, 92
nppiMulC_16u_C1IRSfs
image_mulc, 93
nppiMulC_16u_C1RSfs
image_mulc, 93
nppiMulC_16u_C3IRSfs
image_mulc, 94
nppiMulC_16u_C3RSfs
image_mulc, 94
nppiMulC_16u_C4IRSfs
image_mulc, 94
nppiMulC_16u_C4RSfs
image_mulc, 95
nppiMulC_32f_AC4IR
image_mulc, 95
nppiMulC_32f_AC4R
image_mulc, 95
nppiMulC_32f_C1IR
image_mulc, 96
nppiMulC_32f_C1R
image_mulc, 96
nppiMulC_32f_C3IR
image_mulc, 96
nppiMulC_32f_C3R
image_mulc, 97
nppiMulC_32f_C4IR
image_mulc, 97
nppiMulC_32f_C4R
image_mulc, 97
nppiMulC_32fc_AC4IR
image_mulc, 98
nppiMulC_32fc_AC4R
image_mulc, 98
nppiMulC_32fc_C1IR
image_mulc, 98
nppiMulC_32fc_C1R
image_mulc, 99
nppiMulC_32fc_C3IR
image_mulc, 99
nppiMulC_32fc_C3R
image_mulc, 99
nppiMulC_32fc_C4IR
image_mulc, 100
nppiMulC_32fc_C4R
image_mulc, 100
nppiMulC_32s_C1IRSfs

image_mulc, [101](#)
 nppiMulC_32s_C1RSfs
 image_mulc, [101](#)
 nppiMulC_32s_C3IRSfs
 image_mulc, [101](#)
 nppiMulC_32s_C3RSfs
 image_mulc, [102](#)
 nppiMulC_32sc_AC4IRSfs
 image_mulc, [102](#)
 nppiMulC_32sc_AC4RSfs
 image_mulc, [102](#)
 nppiMulC_32sc_C1IRSfs
 image_mulc, [103](#)
 nppiMulC_32sc_C1RSfs
 image_mulc, [103](#)
 nppiMulC_32sc_C3IRSfs
 image_mulc, [104](#)
 nppiMulC_32sc_C3RSfs
 image_mulc, [104](#)
 nppiMulC_8u_AC4IRSfs
 image_mulc, [104](#)
 nppiMulC_8u_AC4RSfs
 image_mulc, [105](#)
 nppiMulC_8u_C1IRSfs
 image_mulc, [105](#)
 nppiMulC_8u_C1RSfs
 image_mulc, [106](#)
 nppiMulC_8u_C3IRSfs
 image_mulc, [106](#)
 nppiMulC_8u_C3RSfs
 image_mulc, [106](#)
 nppiMulC_8u_C4IRSfs
 image_mulc, [107](#)
 nppiMulC_8u_C4RSfs
 image_mulc, [107](#)
 nppiMulCScale_16u_AC4IR
 image_mulcscale, [109](#)
 nppiMulCScale_16u_AC4R
 image_mulcscale, [109](#)
 nppiMulCScale_16u_C1IR
 image_mulcscale, [110](#)
 nppiMulScale_16u_C1R
 image_mulcscale, [110](#)
 nppiMulScale_16u_C3IR
 image_mulcscale, [110](#)
 nppiMulScale_16u_C3R
 image_mulcscale, [111](#)
 nppiMulScale_16u_C4IR
 image_mulcscale, [111](#)
 nppiMulScale_16u_C4R
 image_mulcscale, [111](#)
 nppiMulScale_8u_AC4IR
 image_mulcscale, [112](#)
 nppiMulScale_8u_AC4R

image_mulcscale, [112](#)
 nppiMulCScale_8u_C1IR
 image_mulcscale, [112](#)
 nppiMulCScale_8u_C1R
 image_mulcscale, [113](#)
 nppiMulCScale_8u_C3IR
 image_mulcscale, [113](#)
 nppiMulCScale_8u_C3R
 image_mulcscale, [113](#)
 nppiMulCScale_8u_C4IR
 image_mulcscale, [114](#)
 nppiMulCScale_8u_C4R
 image_mulcscale, [114](#)
 nppiMulScale_16u_AC4IR
 image_mulscale, [239](#)
 nppiMulScale_16u_AC4R
 image_mulscale, [240](#)
 nppiMulScale_16u_C1IR
 image_mulscale, [240](#)
 nppiMulScale_16u_C1R
 image_mulscale, [240](#)
 nppiMulScale_16u_C3IR
 image_mulscale, [241](#)
 nppiMulScale_16u_C3R
 image_mulscale, [241](#)
 nppiMulScale_16u_C4IR
 image_mulscale, [242](#)
 nppiMulScale_16u_C4R
 image_mulscale, [242](#)
 nppiMulScale_8u_AC4IR
 image_mulscale, [242](#)
 nppiMulScale_8u_AC4R
 image_mulscale, [243](#)
 nppiMulScale_8u_C1IR
 image_mulscale, [243](#)
 nppiMulScale_8u_C1R
 image_mulscale, [244](#)
 nppiMulScale_8u_C3IR
 image_mulscale, [244](#)
 nppiMulScale_8u_C3R
 image_mulscale, [244](#)
 nppiMulScale_8u_C4IR
 image_mulscale, [245](#)
 nppiMulScale_8u_C4R
 image_mulscale, [245](#)
 nppiNorm_Inf_16s_AC4R
 image_inf_norm, [1845](#)
 nppiNorm_Inf_16s_C1R
 image_inf_norm, [1845](#)
 nppiNorm_Inf_16s_C3R
 image_inf_norm, [1845](#)
 nppiNorm_Inf_16s_C4R
 image_inf_norm, [1846](#)
 nppiNorm_Inf_16u_AC4R

image_inf_norm, [1846](#)
nppiNorm_Inf_16u_C1MR
image_inf_norm, [1846](#)
nppiNorm_Inf_16u_C1R
image_inf_norm, [1847](#)
nppiNorm_Inf_16u_C3CMR
image_inf_norm, [1847](#)
nppiNorm_Inf_16u_C3R
image_inf_norm, [1848](#)
nppiNorm_Inf_16u_C4R
image_inf_norm, [1848](#)
nppiNorm_Inf_32f_AC4R
image_inf_norm, [1848](#)
nppiNorm_Inf_32f_C1MR
image_inf_norm, [1849](#)
nppiNorm_Inf_32f_C1R
image_inf_norm, [1849](#)
nppiNorm_Inf_32f_C3CMR
image_inf_norm, [1850](#)
nppiNorm_Inf_32f_C3R
image_inf_norm, [1850](#)
nppiNorm_Inf_32f_C4R
image_inf_norm, [1850](#)
nppiNorm_Inf_32s_C1R
image_inf_norm, [1851](#)
nppiNorm_Inf_8s_C1MR
image_inf_norm, [1851](#)
nppiNorm_Inf_8s_C3CMR
image_inf_norm, [1852](#)
nppiNorm_Inf_8u_AC4R
image_inf_norm, [1852](#)
nppiNorm_Inf_8u_C1MR
image_inf_norm, [1852](#)
nppiNorm_Inf_8u_C1R
image_inf_norm, [1853](#)
nppiNorm_Inf_8u_C3CMR
image_inf_norm, [1853](#)
nppiNorm_Inf_8u_C3R
image_inf_norm, [1854](#)
nppiNorm_Inf_8u_C4R
image_inf_norm, [1854](#)
nppiNorm_L1_16s_AC4R
image_L1_norm, [1867](#)
nppiNorm_L1_16s_C1R
image_L1_norm, [1867](#)
nppiNorm_L1_16s_C3R
image_L1_norm, [1867](#)
nppiNorm_L1_16s_C4R
image_L1_norm, [1868](#)
nppiNorm_L1_16u_AC4R
image_L1_norm, [1868](#)
nppiNorm_L1_16u_C1MR
image_L1_norm, [1868](#)
nppiNorm_L1_16u_C1R
image_L1_norm, [1869](#)
nppiNorm_L1_16u_C3CMR
image_L1_norm, [1869](#)
nppiNorm_L1_16u_C3R
image_L1_norm, [1870](#)
nppiNorm_L1_16u_C4R
image_L1_norm, [1870](#)
nppiNorm_L1_32f_AC4R
image_L1_norm, [1870](#)
nppiNorm_L1_32f_C1MR
image_L1_norm, [1871](#)
nppiNorm_L1_32f_C1R
image_L1_norm, [1871](#)
nppiNorm_L1_32f_C3CMR
image_L1_norm, [1871](#)
nppiNorm_L1_32f_C3R
image_L1_norm, [1872](#)
nppiNorm_L1_32f_C4R
image_L1_norm, [1872](#)
nppiNorm_L1_8s_C1MR
image_L1_norm, [1873](#)
nppiNorm_L1_8s_C3CMR
image_L1_norm, [1873](#)
nppiNorm_L1_8u_AC4R
image_L1_norm, [1873](#)
nppiNorm_L1_8u_C1MR
image_L1_norm, [1874](#)
nppiNorm_L1_8u_C1R
image_L1_norm, [1874](#)
nppiNorm_L1_8u_C3CMR
image_L1_norm, [1875](#)
nppiNorm_L1_8u_C3R
image_L1_norm, [1875](#)
nppiNorm_L1_8u_C4R
image_L1_norm, [1875](#)
nppiNorm_L2_16s_AC4R
image_L2_norm, [1888](#)
nppiNorm_L2_16s_C1R
image_L2_norm, [1888](#)
nppiNorm_L2_16s_C3R
image_L2_norm, [1888](#)
nppiNorm_L2_16s_C4R
image_L2_norm, [1889](#)
nppiNorm_L2_16u_AC4R
image_L2_norm, [1889](#)
nppiNorm_L2_16u_C1MR
image_L2_norm, [1889](#)
nppiNorm_L2_16u_C1R
image_L2_norm, [1890](#)
nppiNorm_L2_16u_C3CMR
image_L2_norm, [1890](#)
nppiNorm_L2_16u_C3R
image_L2_norm, [1891](#)
nppiNorm_L2_16u_C4R

- image_L2_norm, [1891](#)
- nppiNorm_L2_32f_AC4R
 - image_L2_norm, [1891](#)
- nppiNorm_L2_32f_C1MR
 - image_L2_norm, [1892](#)
- nppiNorm_L2_32f_C1R
 - image_L2_norm, [1892](#)
- nppiNorm_L2_32f_C3CMR
 - image_L2_norm, [1892](#)
- nppiNorm_L2_32f_C3R
 - image_L2_norm, [1893](#)
- nppiNorm_L2_32f_C4R
 - image_L2_norm, [1893](#)
- nppiNorm_L2_8s_C1MR
 - image_L2_norm, [1894](#)
- nppiNorm_L2_8s_C3CMR
 - image_L2_norm, [1894](#)
- nppiNorm_L2_8u_AC4R
 - image_L2_norm, [1894](#)
- nppiNorm_L2_8u_C1MR
 - image_L2_norm, [1895](#)
- nppiNorm_L2_8u_C1R
 - image_L2_norm, [1895](#)
- nppiNorm_L2_8u_C3CMR
 - image_L2_norm, [1896](#)
- nppiNorm_L2_8u_C3R
 - image_L2_norm, [1896](#)
- nppiNorm_L2_8u_C4R
 - image_L2_norm, [1896](#)
- nppiNormDiff_Inf_16s_AC4R
 - image_inf_normdiff, [1909](#)
- nppiNormDiff_Inf_16s_C1R
 - image_inf_normdiff, [1909](#)
- nppiNormDiff_Inf_16s_C3R
 - image_inf_normdiff, [1910](#)
- nppiNormDiff_Inf_16s_C4R
 - image_inf_normdiff, [1910](#)
- nppiNormDiff_Inf_16u_AC4R
 - image_inf_normdiff, [1911](#)
- nppiNormDiff_Inf_16u_C1MR
 - image_inf_normdiff, [1911](#)
- nppiNormDiff_Inf_16u_C1R
 - image_inf_normdiff, [1912](#)
- nppiNormDiff_Inf_16u_C3CMR
 - image_inf_normdiff, [1912](#)
- nppiNormDiff_Inf_16u_C3R
 - image_inf_normdiff, [1913](#)
- nppiNormDiff_Inf_16u_C4R
 - image_inf_normdiff, [1913](#)
- nppiNormDiff_Inf_32f_AC4R
 - image_inf_normdiff, [1913](#)
- nppiNormDiff_Inf_32f_C1MR
 - image_inf_normdiff, [1914](#)
- nppiNormDiff_Inf_32f_C1R
 - image_inf_normdiff, [1914](#)
- nppiNormDiff_Inf_32f_C3CMR
 - image_inf_normdiff, [1915](#)
- nppiNormDiff_Inf_32f_C3R
 - image_inf_normdiff, [1915](#)
- nppiNormDiff_Inf_32f_C4R
 - image_inf_normdiff, [1916](#)
- nppiNormDiff_Inf_8s_C1MR
 - image_inf_normdiff, [1916](#)
- nppiNormDiff_Inf_8s_C3CMR
 - image_inf_normdiff, [1917](#)
- nppiNormDiff_Inf_8u_AC4R
 - image_inf_normdiff, [1917](#)
- nppiNormDiff_Inf_8u_C1MR
 - image_inf_normdiff, [1918](#)
- nppiNormDiff_Inf_8u_C1R
 - image_inf_normdiff, [1918](#)
- nppiNormDiff_Inf_8u_C3CMR
 - image_inf_normdiff, [1919](#)
- nppiNormDiff_Inf_8u_C3R
 - image_inf_normdiff, [1919](#)
- nppiNormDiff_Inf_8u_C4R
 - image_inf_normdiff, [1920](#)
- nppiNormDiff_L1_16s_AC4R
 - image_L1_normdiff, [1932](#)
- nppiNormDiff_L1_16s_C1R
 - image_L1_normdiff, [1932](#)
- nppiNormDiff_L1_16s_C3R
 - image_L1_normdiff, [1933](#)
- nppiNormDiff_L1_16s_C4R
 - image_L1_normdiff, [1933](#)
- nppiNormDiff_L1_16u_AC4R
 - image_L1_normdiff, [1934](#)
- nppiNormDiff_L1_16u_C1MR
 - image_L1_normdiff, [1934](#)
- nppiNormDiff_L1_16u_C1R
 - image_L1_normdiff, [1934](#)
- nppiNormDiff_L1_16u_C3CMR
 - image_L1_normdiff, [1935](#)
- nppiNormDiff_L1_16u_C3R
 - image_L1_normdiff, [1935](#)
- nppiNormDiff_L1_16u_C4R
 - image_L1_normdiff, [1936](#)
- nppiNormDiff_L1_32f_AC4R
 - image_L1_normdiff, [1936](#)
- nppiNormDiff_L1_32f_C1MR
 - image_L1_normdiff, [1937](#)
- nppiNormDiff_L1_32f_C1R
 - image_L1_normdiff, [1937](#)
- nppiNormDiff_L1_32f_C3CMR
 - image_L1_normdiff, [1938](#)
- nppiNormDiff_L1_32f_C3R
 - image_L1_normdiff, [1938](#)
- nppiNormDiff_L1_32f_C4R

- image_L1_normdiff, [1939](#)
- nppiNormDiff_L1_8s_C1MR
 - image_L1_normdiff, [1939](#)
- nppiNormDiff_L1_8s_C3CMR
 - image_L1_normdiff, [1940](#)
- nppiNormDiff_L1_8u_AC4R
 - image_L1_normdiff, [1940](#)
- nppiNormDiff_L1_8u_C1MR
 - image_L1_normdiff, [1941](#)
- nppiNormDiff_L1_8u_C1R
 - image_L1_normdiff, [1941](#)
- nppiNormDiff_L1_8u_C3CMR
 - image_L1_normdiff, [1941](#)
- nppiNormDiff_L1_8u_C3R
 - image_L1_normdiff, [1942](#)
- nppiNormDiff_L1_8u_C4R
 - image_L1_normdiff, [1942](#)
- nppiNormDiff_L2_16s_AC4R
 - image_L2_normdiff, [1955](#)
- nppiNormDiff_L2_16s_C1R
 - image_L2_normdiff, [1955](#)
- nppiNormDiff_L2_16s_C3R
 - image_L2_normdiff, [1956](#)
- nppiNormDiff_L2_16s_C4R
 - image_L2_normdiff, [1956](#)
- nppiNormDiff_L2_16u_AC4R
 - image_L2_normdiff, [1957](#)
- nppiNormDiff_L2_16u_C1MR
 - image_L2_normdiff, [1957](#)
- nppiNormDiff_L2_16u_C1R
 - image_L2_normdiff, [1957](#)
- nppiNormDiff_L2_16u_C3CMR
 - image_L2_normdiff, [1958](#)
- nppiNormDiff_L2_16u_C3R
 - image_L2_normdiff, [1958](#)
- nppiNormDiff_L2_16u_C4R
 - image_L2_normdiff, [1959](#)
- nppiNormDiff_L2_32f_AC4R
 - image_L2_normdiff, [1959](#)
- nppiNormDiff_L2_32f_C1MR
 - image_L2_normdiff, [1960](#)
- nppiNormDiff_L2_32f_C1R
 - image_L2_normdiff, [1960](#)
- nppiNormDiff_L2_32f_C3CMR
 - image_L2_normdiff, [1961](#)
- nppiNormDiff_L2_32f_C3R
 - image_L2_normdiff, [1961](#)
- nppiNormDiff_L2_32f_C4R
 - image_L2_normdiff, [1962](#)
- nppiNormDiff_L2_8s_C1MR
 - image_L2_normdiff, [1962](#)
- nppiNormDiff_L2_8s_C3CMR
 - image_L2_normdiff, [1963](#)
- nppiNormDiff_L2_8u_AC4R
 - image_L2_normdiff, [1963](#)
- nppiNormDiff_L2_8u_C1MR
 - image_L2_normdiff, [1964](#)
- nppiNormDiff_L2_8u_C1R
 - image_L2_normdiff, [1964](#)
- nppiNormDiff_L2_8u_C3CMR
 - image_L2_normdiff, [1964](#)
- nppiNormDiff_L2_8u_C3R
 - image_L2_normdiff, [1965](#)
- nppiNormDiff_L2_8u_C4R
 - image_L2_normdiff, [1965](#)
- nppiNormDiffInfGetBufferHostSize_16s_AC4R
 - image_inf_normdiff, [1920](#)
- nppiNormDiffInfGetBufferHostSize_16s_C1R
 - image_inf_normdiff, [1920](#)
- nppiNormDiffInfGetBufferHostSize_16s_C3R
 - image_inf_normdiff, [1921](#)
- nppiNormDiffInfGetBufferHostSize_16s_C4R
 - image_inf_normdiff, [1921](#)
- nppiNormDiffInfGetBufferHostSize_16u_AC4R
 - image_inf_normdiff, [1921](#)
- nppiNormDiffInfGetBufferHostSize_16u_C1MR
 - image_inf_normdiff, [1922](#)
- nppiNormDiffInfGetBufferHostSize_16u_C1R
 - image_inf_normdiff, [1922](#)
- nppiNormDiffInfGetBufferHostSize_16u_C3CMR
 - image_inf_normdiff, [1922](#)
- nppiNormDiffInfGetBufferHostSize_16u_C3R
 - image_inf_normdiff, [1922](#)
- nppiNormDiffInfGetBufferHostSize_16u_C4R
 - image_inf_normdiff, [1923](#)
- nppiNormDiffInfGetBufferHostSize_32f_AC4R
 - image_inf_normdiff, [1923](#)
- nppiNormDiffInfGetBufferHostSize_32f_C1MR
 - image_inf_normdiff, [1923](#)
- nppiNormDiffInfGetBufferHostSize_32f_C1R
 - image_inf_normdiff, [1924](#)
- nppiNormDiffInfGetBufferHostSize_32f_C3CMR
 - image_inf_normdiff, [1924](#)
- nppiNormDiffInfGetBufferHostSize_32f_C3R
 - image_inf_normdiff, [1924](#)
- nppiNormDiffInfGetBufferHostSize_32f_C4R
 - image_inf_normdiff, [1924](#)
- nppiNormDiffInfGetBufferHostSize_8s_C1MR
 - image_inf_normdiff, [1925](#)
- nppiNormDiffInfGetBufferHostSize_8s_C3CMR
 - image_inf_normdiff, [1925](#)
- nppiNormDiffInfGetBufferHostSize_8u_AC4R
 - image_inf_normdiff, [1925](#)
- nppiNormDiffInfGetBufferHostSize_8u_C1MR
 - image_inf_normdiff, [1926](#)
- nppiNormDiffInfGetBufferHostSize_8u_C1R
 - image_inf_normdiff, [1926](#)
- nppiNormDiffInfGetBufferHostSize_8u_C3CMR

- image_inf_normdiff, [1926](#)
- nppiNormDiffInfGetBufferHostSize_8u_C3R
 - image_inf_normdiff, [1926](#)
- nppiNormDiffInfGetBufferHostSize_8u_C4R
 - image_inf_normdiff, [1927](#)
- nppiNormDiffL1GetBufferHostSize_16s_AC4R
 - image_L1_normdiff, [1943](#)
- nppiNormDiffL1GetBufferHostSize_16s_C1R
 - image_L1_normdiff, [1943](#)
- nppiNormDiffL1GetBufferHostSize_16s_C3R
 - image_L1_normdiff, [1943](#)
- nppiNormDiffL1GetBufferHostSize_16s_C4R
 - image_L1_normdiff, [1944](#)
- nppiNormDiffL1GetBufferHostSize_16u_AC4R
 - image_L1_normdiff, [1944](#)
- nppiNormDiffL1GetBufferHostSize_16u_C1MR
 - image_L1_normdiff, [1944](#)
- nppiNormDiffL1GetBufferHostSize_16u_C1R
 - image_L1_normdiff, [1945](#)
- nppiNormDiffL1GetBufferHostSize_16u_C3CMR
 - image_L1_normdiff, [1945](#)
- nppiNormDiffL1GetBufferHostSize_16u_C3R
 - image_L1_normdiff, [1945](#)
- nppiNormDiffL1GetBufferHostSize_16u_C4R
 - image_L1_normdiff, [1945](#)
- nppiNormDiffL1GetBufferHostSize_32f_AC4R
 - image_L1_normdiff, [1946](#)
- nppiNormDiffL1GetBufferHostSize_32f_C1MR
 - image_L1_normdiff, [1946](#)
- nppiNormDiffL1GetBufferHostSize_32f_C1R
 - image_L1_normdiff, [1946](#)
- nppiNormDiffL1GetBufferHostSize_32f_C3CMR
 - image_L1_normdiff, [1947](#)
- nppiNormDiffL1GetBufferHostSize_32f_C3R
 - image_L1_normdiff, [1947](#)
- nppiNormDiffL1GetBufferHostSize_32f_C4R
 - image_L1_normdiff, [1947](#)
- nppiNormDiffL1GetBufferHostSize_8s_C1MR
 - image_L1_normdiff, [1947](#)
- nppiNormDiffL1GetBufferHostSize_8s_C3CMR
 - image_L1_normdiff, [1948](#)
- nppiNormDiffL1GetBufferHostSize_8u_AC4R
 - image_L1_normdiff, [1948](#)
- nppiNormDiffL1GetBufferHostSize_8u_C1MR
 - image_L1_normdiff, [1948](#)
- nppiNormDiffL1GetBufferHostSize_8u_C1R
 - image_L1_normdiff, [1949](#)
- nppiNormDiffL1GetBufferHostSize_8u_C3CMR
 - image_L1_normdiff, [1949](#)
- nppiNormDiffL1GetBufferHostSize_8u_C3R
 - image_L1_normdiff, [1949](#)
- nppiNormDiffL1GetBufferHostSize_8u_C4R
 - image_L1_normdiff, [1949](#)
- nppiNormDiffL2GetBufferHostSize_16s_AC4R
 - image_L2_normdiff, [1966](#)
- nppiNormDiffL2GetBufferHostSize_16s_C1R
 - image_L2_normdiff, [1966](#)
- nppiNormDiffL2GetBufferHostSize_16s_C3R
 - image_L2_normdiff, [1966](#)
- nppiNormDiffL2GetBufferHostSize_16s_C4R
 - image_L2_normdiff, [1967](#)
- nppiNormDiffL2GetBufferHostSize_16u_AC4R
 - image_L2_normdiff, [1967](#)
- nppiNormDiffL2GetBufferHostSize_16u_C1MR
 - image_L2_normdiff, [1967](#)
- nppiNormDiffL2GetBufferHostSize_16u_C1R
 - image_L2_normdiff, [1968](#)
- nppiNormDiffL2GetBufferHostSize_16u_C3CMR
 - image_L2_normdiff, [1968](#)
- nppiNormDiffL2GetBufferHostSize_16u_C3R
 - image_L2_normdiff, [1968](#)
- nppiNormDiffL2GetBufferHostSize_16u_C4R
 - image_L2_normdiff, [1968](#)
- nppiNormDiffL2GetBufferHostSize_32f_AC4R
 - image_L2_normdiff, [1969](#)
- nppiNormDiffL2GetBufferHostSize_32f_C1MR
 - image_L2_normdiff, [1969](#)
- nppiNormDiffL2GetBufferHostSize_32f_C1R
 - image_L2_normdiff, [1969](#)
- nppiNormDiffL2GetBufferHostSize_32f_C3CMR
 - image_L2_normdiff, [1970](#)
- nppiNormDiffL2GetBufferHostSize_32f_C3R
 - image_L2_normdiff, [1970](#)
- nppiNormDiffL2GetBufferHostSize_32f_C4R
 - image_L2_normdiff, [1970](#)
- nppiNormDiffL2GetBufferHostSize_8s_C1MR
 - image_L2_normdiff, [1970](#)
- nppiNormDiffL2GetBufferHostSize_8s_C3CMR
 - image_L2_normdiff, [1971](#)
- nppiNormDiffL2GetBufferHostSize_8u_AC4R
 - image_L2_normdiff, [1971](#)
- nppiNormDiffL2GetBufferHostSize_8u_C1MR
 - image_L2_normdiff, [1971](#)
- nppiNormDiffL2GetBufferHostSize_8u_C1R
 - image_L2_normdiff, [1972](#)
- nppiNormDiffL2GetBufferHostSize_8u_C3CMR
 - image_L2_normdiff, [1972](#)
- nppiNormDiffL2GetBufferHostSize_8u_C3R
 - image_L2_normdiff, [1972](#)
- nppiNormDiffL2GetBufferHostSize_8u_C4R
 - image_L2_normdiff, [1972](#)
- nppiNormInfGetBufferHostSize_16s_AC4R
 - image_inf_norm, [1854](#)
- nppiNormInfGetBufferHostSize_16s_C1R
 - image_inf_norm, [1855](#)
- nppiNormInfGetBufferHostSize_16s_C3R
 - image_inf_norm, [1855](#)
- nppiNormInfGetBufferHostSize_16s_C4R

- image_inf_norm, [1855](#)
- nppiNormInfGetBufferHostSize_16u_AC4R
 - image_inf_norm, [1856](#)
- nppiNormInfGetBufferHostSize_16u_C1MR
 - image_inf_norm, [1856](#)
- nppiNormInfGetBufferHostSize_16u_C1R
 - image_inf_norm, [1856](#)
- nppiNormInfGetBufferHostSize_16u_C3CMR
 - image_inf_norm, [1856](#)
- nppiNormInfGetBufferHostSize_16u_C3R
 - image_inf_norm, [1857](#)
- nppiNormInfGetBufferHostSize_16u_C4R
 - image_inf_norm, [1857](#)
- nppiNormInfGetBufferHostSize_32f_AC4R
 - image_inf_norm, [1857](#)
- nppiNormInfGetBufferHostSize_32f_C1MR
 - image_inf_norm, [1858](#)
- nppiNormInfGetBufferHostSize_32f_C1R
 - image_inf_norm, [1858](#)
- nppiNormInfGetBufferHostSize_32f_C3CMR
 - image_inf_norm, [1858](#)
- nppiNormInfGetBufferHostSize_32f_C3R
 - image_inf_norm, [1858](#)
- nppiNormInfGetBufferHostSize_32f_C4R
 - image_inf_norm, [1859](#)
- nppiNormInfGetBufferHostSize_32s_C1R
 - image_inf_norm, [1859](#)
- nppiNormInfGetBufferHostSize_8s_C1MR
 - image_inf_norm, [1859](#)
- nppiNormInfGetBufferHostSize_8s_C3CMR
 - image_inf_norm, [1860](#)
- nppiNormInfGetBufferHostSize_8u_AC4R
 - image_inf_norm, [1860](#)
- nppiNormInfGetBufferHostSize_8u_C1MR
 - image_inf_norm, [1860](#)
- nppiNormInfGetBufferHostSize_8u_C1R
 - image_inf_norm, [1860](#)
- nppiNormInfGetBufferHostSize_8u_C3CMR
 - image_inf_norm, [1861](#)
- nppiNormInfGetBufferHostSize_8u_C3R
 - image_inf_norm, [1861](#)
- nppiNormInfGetBufferHostSize_8u_C4R
 - image_inf_norm, [1861](#)
- nppiNormL1GetBufferHostSize_16s_AC4R
 - image_L1_norm, [1876](#)
- nppiNormL1GetBufferHostSize_16s_C1R
 - image_L1_norm, [1876](#)
- nppiNormL1GetBufferHostSize_16s_C3R
 - image_L1_norm, [1876](#)
- nppiNormL1GetBufferHostSize_16s_C4R
 - image_L1_norm, [1877](#)
- nppiNormL1GetBufferHostSize_16u_AC4R
 - image_L1_norm, [1877](#)
- nppiNormL1GetBufferHostSize_16u_C1MR
 - image_L1_norm, [1877](#)
- nppiNormL1GetBufferHostSize_16u_C1R
 - image_L1_norm, [1878](#)
- nppiNormL1GetBufferHostSize_16u_C3CMR
 - image_L1_norm, [1878](#)
- nppiNormL1GetBufferHostSize_16u_C3R
 - image_L1_norm, [1878](#)
- nppiNormL1GetBufferHostSize_16u_C4R
 - image_L1_norm, [1878](#)
- nppiNormL1GetBufferHostSize_32f_AC4R
 - image_L1_norm, [1879](#)
- nppiNormL1GetBufferHostSize_32f_C1MR
 - image_L1_norm, [1879](#)
- nppiNormL1GetBufferHostSize_32f_C1R
 - image_L1_norm, [1879](#)
- nppiNormL1GetBufferHostSize_32f_C3CMR
 - image_L1_norm, [1880](#)
- nppiNormL1GetBufferHostSize_32f_C3R
 - image_L1_norm, [1880](#)
- nppiNormL1GetBufferHostSize_32f_C4R
 - image_L1_norm, [1880](#)
- nppiNormL1GetBufferHostSize_8s_C1MR
 - image_L1_norm, [1880](#)
- nppiNormL1GetBufferHostSize_8s_C3CMR
 - image_L1_norm, [1881](#)
- nppiNormL1GetBufferHostSize_8u_AC4R
 - image_L1_norm, [1881](#)
- nppiNormL1GetBufferHostSize_8u_C1MR
 - image_L1_norm, [1881](#)
- nppiNormL1GetBufferHostSize_8u_C1R
 - image_L1_norm, [1882](#)
- nppiNormL1GetBufferHostSize_8u_C3CMR
 - image_L1_norm, [1882](#)
- nppiNormL1GetBufferHostSize_8u_C3R
 - image_L1_norm, [1882](#)
- nppiNormL1GetBufferHostSize_8u_C4R
 - image_L1_norm, [1882](#)
- nppiNormL2GetBufferHostSize_16s_AC4R
 - image_L2_norm, [1897](#)
- nppiNormL2GetBufferHostSize_16s_C1R
 - image_L2_norm, [1897](#)
- nppiNormL2GetBufferHostSize_16s_C3R
 - image_L2_norm, [1897](#)
- nppiNormL2GetBufferHostSize_16s_C4R
 - image_L2_norm, [1898](#)
- nppiNormL2GetBufferHostSize_16u_AC4R
 - image_L2_norm, [1898](#)
- nppiNormL2GetBufferHostSize_16u_C1MR
 - image_L2_norm, [1898](#)
- nppiNormL2GetBufferHostSize_16u_C1R
 - image_L2_norm, [1899](#)
- nppiNormL2GetBufferHostSize_16u_C3CMR
 - image_L2_norm, [1899](#)
- nppiNormL2GetBufferHostSize_16u_C3R

- image_L2_norm, [1899](#)
- nppiNormL2GetBufferHostSize_16u_C4R
 - image_L2_norm, [1899](#)
- nppiNormL2GetBufferHostSize_32f_AC4R
 - image_L2_norm, [1900](#)
- nppiNormL2GetBufferHostSize_32f_C1MR
 - image_L2_norm, [1900](#)
- nppiNormL2GetBufferHostSize_32f_C1R
 - image_L2_norm, [1900](#)
- nppiNormL2GetBufferHostSize_32f_C3CMR
 - image_L2_norm, [1901](#)
- nppiNormL2GetBufferHostSize_32f_C3R
 - image_L2_norm, [1901](#)
- nppiNormL2GetBufferHostSize_32f_C4R
 - image_L2_norm, [1901](#)
- nppiNormL2GetBufferHostSize_8s_C1MR
 - image_L2_norm, [1901](#)
- nppiNormL2GetBufferHostSize_8s_C3CMR
 - image_L2_norm, [1902](#)
- nppiNormL2GetBufferHostSize_8u_AC4R
 - image_L2_norm, [1902](#)
- nppiNormL2GetBufferHostSize_8u_C1MR
 - image_L2_norm, [1902](#)
- nppiNormL2GetBufferHostSize_8u_C1R
 - image_L2_norm, [1903](#)
- nppiNormL2GetBufferHostSize_8u_C3CMR
 - image_L2_norm, [1903](#)
- nppiNormL2GetBufferHostSize_8u_C3R
 - image_L2_norm, [1903](#)
- nppiNormL2GetBufferHostSize_8u_C4R
 - image_L2_norm, [1903](#)
- nppiNormRel_Inf_16s_AC4R
 - image_inf_normrel, [1978](#)
- nppiNormRel_Inf_16s_C1R
 - image_inf_normrel, [1978](#)
- nppiNormRel_Inf_16s_C3R
 - image_inf_normrel, [1979](#)
- nppiNormRel_Inf_16s_C4R
 - image_inf_normrel, [1979](#)
- nppiNormRel_Inf_16u_AC4R
 - image_inf_normrel, [1980](#)
- nppiNormRel_Inf_16u_C1MR
 - image_inf_normrel, [1980](#)
- nppiNormRel_Inf_16u_C1R
 - image_inf_normrel, [1981](#)
- nppiNormRel_Inf_16u_C3CMR
 - image_inf_normrel, [1981](#)
- nppiNormRel_Inf_16u_C3R
 - image_inf_normrel, [1982](#)
- nppiNormRel_Inf_16u_C4R
 - image_inf_normrel, [1982](#)
- nppiNormRel_Inf_32f_AC4R
 - image_inf_normrel, [1982](#)
- nppiNormRel_Inf_32f_C1MR
 - image_inf_normrel, [1983](#)
- nppiNormRel_Inf_32f_C1R
 - image_inf_normrel, [1983](#)
- nppiNormRel_Inf_32f_C3CMR
 - image_inf_normrel, [1984](#)
- nppiNormRel_Inf_32f_C3R
 - image_inf_normrel, [1984](#)
- nppiNormRel_Inf_32f_C4R
 - image_inf_normrel, [1985](#)
- nppiNormRel_Inf_8s_C1MR
 - image_inf_normrel, [1985](#)
- nppiNormRel_Inf_8s_C3CMR
 - image_inf_normrel, [1986](#)
- nppiNormRel_Inf_8u_AC4R
 - image_inf_normrel, [1986](#)
- nppiNormRel_Inf_8u_C1MR
 - image_inf_normrel, [1987](#)
- nppiNormRel_Inf_8u_C1R
 - image_inf_normrel, [1987](#)
- nppiNormRel_Inf_8u_C3CMR
 - image_inf_normrel, [1988](#)
- nppiNormRel_Inf_8u_C3R
 - image_inf_normrel, [1988](#)
- nppiNormRel_Inf_8u_C4R
 - image_inf_normrel, [1989](#)
- nppiNormRel_L1_16s_AC4R
 - image_L1_normrel, [2001](#)
- nppiNormRel_L1_16s_C1R
 - image_L1_normrel, [2001](#)
- nppiNormRel_L1_16s_C3R
 - image_L1_normrel, [2002](#)
- nppiNormRel_L1_16s_C4R
 - image_L1_normrel, [2002](#)
- nppiNormRel_L1_16u_AC4R
 - image_L1_normrel, [2003](#)
- nppiNormRel_L1_16u_C1MR
 - image_L1_normrel, [2003](#)
- nppiNormRel_L1_16u_C1R
 - image_L1_normrel, [2004](#)
- nppiNormRel_L1_16u_C3CMR
 - image_L1_normrel, [2004](#)
- nppiNormRel_L1_16u_C3R
 - image_L1_normrel, [2004](#)
- nppiNormRel_L1_16u_C4R
 - image_L1_normrel, [2005](#)
- nppiNormRel_L1_32f_AC4R
 - image_L1_normrel, [2005](#)
- nppiNormRel_L1_32f_C1MR
 - image_L1_normrel, [2006](#)
- nppiNormRel_L1_32f_C1R
 - image_L1_normrel, [2006](#)
- nppiNormRel_L1_32f_C3CMR
 - image_L1_normrel, [2007](#)
- nppiNormRel_L1_32f_C3R

- image_L1_normrel, [2007](#)
- nppiNormRel_L1_32f_C4R
 - image_L1_normrel, [2008](#)
- nppiNormRel_L1_8s_C1MR
 - image_L1_normrel, [2008](#)
- nppiNormRel_L1_8s_C3CMR
 - image_L1_normrel, [2009](#)
- nppiNormRel_L1_8u_AC4R
 - image_L1_normrel, [2009](#)
- nppiNormRel_L1_8u_C1MR
 - image_L1_normrel, [2010](#)
- nppiNormRel_L1_8u_C1R
 - image_L1_normrel, [2010](#)
- nppiNormRel_L1_8u_C3CMR
 - image_L1_normrel, [2011](#)
- nppiNormRel_L1_8u_C3R
 - image_L1_normrel, [2011](#)
- nppiNormRel_L1_8u_C4R
 - image_L1_normrel, [2012](#)
- nppiNormRel_L2_16s_AC4R
 - image_L2_normrel, [2024](#)
- nppiNormRel_L2_16s_C1R
 - image_L2_normrel, [2024](#)
- nppiNormRel_L2_16s_C3R
 - image_L2_normrel, [2025](#)
- nppiNormRel_L2_16s_C4R
 - image_L2_normrel, [2025](#)
- nppiNormRel_L2_16u_AC4R
 - image_L2_normrel, [2026](#)
- nppiNormRel_L2_16u_C1MR
 - image_L2_normrel, [2026](#)
- nppiNormRel_L2_16u_C1R
 - image_L2_normrel, [2027](#)
- nppiNormRel_L2_16u_C3CMR
 - image_L2_normrel, [2027](#)
- nppiNormRel_L2_16u_C3R
 - image_L2_normrel, [2027](#)
- nppiNormRel_L2_16u_C4R
 - image_L2_normrel, [2028](#)
- nppiNormRel_L2_32f_AC4R
 - image_L2_normrel, [2028](#)
- nppiNormRel_L2_32f_C1MR
 - image_L2_normrel, [2029](#)
- nppiNormRel_L2_32f_C1R
 - image_L2_normrel, [2029](#)
- nppiNormRel_L2_32f_C3CMR
 - image_L2_normrel, [2030](#)
- nppiNormRel_L2_32f_C3R
 - image_L2_normrel, [2030](#)
- nppiNormRel_L2_32f_C4R
 - image_L2_normrel, [2031](#)
- nppiNormRel_L2_8s_C1MR
 - image_L2_normrel, [2031](#)
- nppiNormRel_L2_8s_C3CMR
 - image_L2_normrel, [2032](#)
- nppiNormRel_L2_8u_AC4R
 - image_L2_normrel, [2032](#)
- nppiNormRel_L2_8u_C1MR
 - image_L2_normrel, [2033](#)
- nppiNormRel_L2_8u_C1R
 - image_L2_normrel, [2033](#)
- nppiNormRel_L2_8u_C3CMR
 - image_L2_normrel, [2034](#)
- nppiNormRel_L2_8u_C3R
 - image_L2_normrel, [2034](#)
- nppiNormRel_L2_8u_C4R
 - image_L2_normrel, [2035](#)
- nppiNormRelInfGetBufferHostSize_16s_AC4R
 - image_inf_normrel, [1989](#)
- nppiNormRelInfGetBufferHostSize_16s_C1R
 - image_inf_normrel, [1990](#)
- nppiNormRelInfGetBufferHostSize_16s_C3R
 - image_inf_normrel, [1990](#)
- nppiNormRelInfGetBufferHostSize_16s_C4R
 - image_inf_normrel, [1990](#)
- nppiNormRelInfGetBufferHostSize_16u_AC4R
 - image_inf_normrel, [1990](#)
- nppiNormRelInfGetBufferHostSize_16u_C1MR
 - image_inf_normrel, [1991](#)
- nppiNormRelInfGetBufferHostSize_16u_C1R
 - image_inf_normrel, [1991](#)
- nppiNormRelInfGetBufferHostSize_16u_C3CMR
 - image_inf_normrel, [1991](#)
- nppiNormRelInfGetBufferHostSize_16u_C3R
 - image_inf_normrel, [1992](#)
- nppiNormRelInfGetBufferHostSize_16u_C4R
 - image_inf_normrel, [1992](#)
- nppiNormRelInfGetBufferHostSize_32f_AC4R
 - image_inf_normrel, [1992](#)
- nppiNormRelInfGetBufferHostSize_32f_C1MR
 - image_inf_normrel, [1992](#)
- nppiNormRelInfGetBufferHostSize_32f_C1R
 - image_inf_normrel, [1993](#)
- nppiNormRelInfGetBufferHostSize_32f_C3CMR
 - image_inf_normrel, [1993](#)
- nppiNormRelInfGetBufferHostSize_32f_C3R
 - image_inf_normrel, [1993](#)
- nppiNormRelInfGetBufferHostSize_32f_C4R
 - image_inf_normrel, [1994](#)
- nppiNormRelInfGetBufferHostSize_32s_C1R
 - image_inf_normrel, [1994](#)
- nppiNormRelInfGetBufferHostSize_8s_C1MR
 - image_inf_normrel, [1994](#)
- nppiNormRelInfGetBufferHostSize_8s_C3CMR
 - image_inf_normrel, [1994](#)
- nppiNormRelInfGetBufferHostSize_8u_AC4R
 - image_inf_normrel, [1995](#)
- nppiNormRelInfGetBufferHostSize_8u_C1MR

- image_inf_normrel, [1995](#)
- nppiNormRelInfGetBufferHostSize_8u_C1R
 - image_inf_normrel, [1995](#)
- nppiNormRelInfGetBufferHostSize_8u_C3CMR
 - image_inf_normrel, [1996](#)
- nppiNormRelInfGetBufferHostSize_8u_C3R
 - image_inf_normrel, [1996](#)
- nppiNormRelInfGetBufferHostSize_8u_C4R
 - image_inf_normrel, [1996](#)
- nppiNormRelL1GetBufferHostSize_16s_AC4R
 - image_L1_normrel, [2012](#)
- nppiNormRelL1GetBufferHostSize_16s_C1R
 - image_L1_normrel, [2012](#)
- nppiNormRelL1GetBufferHostSize_16s_C3R
 - image_L1_normrel, [2013](#)
- nppiNormRelL1GetBufferHostSize_16s_C4R
 - image_L1_normrel, [2013](#)
- nppiNormRelL1GetBufferHostSize_16u_AC4R
 - image_L1_normrel, [2013](#)
- nppiNormRelL1GetBufferHostSize_16u_C1MR
 - image_L1_normrel, [2014](#)
- nppiNormRelL1GetBufferHostSize_16u_C1R
 - image_L1_normrel, [2014](#)
- nppiNormRelL1GetBufferHostSize_16u_C3CMR
 - image_L1_normrel, [2014](#)
- nppiNormRelL1GetBufferHostSize_16u_C3R
 - image_L1_normrel, [2014](#)
- nppiNormRelL1GetBufferHostSize_16u_C4R
 - image_L1_normrel, [2015](#)
- nppiNormRelL1GetBufferHostSize_32f_AC4R
 - image_L1_normrel, [2015](#)
- nppiNormRelL1GetBufferHostSize_32f_C1MR
 - image_L1_normrel, [2015](#)
- nppiNormRelL1GetBufferHostSize_32f_C1R
 - image_L1_normrel, [2016](#)
- nppiNormRelL1GetBufferHostSize_32f_C3CMR
 - image_L1_normrel, [2016](#)
- nppiNormRelL1GetBufferHostSize_32f_C3R
 - image_L1_normrel, [2016](#)
- nppiNormRelL1GetBufferHostSize_32f_C4R
 - image_L1_normrel, [2016](#)
- nppiNormRelL1GetBufferHostSize_8s_C1MR
 - image_L1_normrel, [2017](#)
- nppiNormRelL1GetBufferHostSize_8s_C3CMR
 - image_L1_normrel, [2017](#)
- nppiNormRelL1GetBufferHostSize_8u_AC4R
 - image_L1_normrel, [2017](#)
- nppiNormRelL1GetBufferHostSize_8u_C1MR
 - image_L1_normrel, [2018](#)
- nppiNormRelL1GetBufferHostSize_8u_C1R
 - image_L1_normrel, [2018](#)
- nppiNormRelL1GetBufferHostSize_8u_C3CMR
 - image_L1_normrel, [2018](#)
- nppiNormRelL1GetBufferHostSize_8u_C3R
 - image_L1_normrel, [2018](#)
- image_L1_normrel, [2018](#)
- nppiNormRelL1GetBufferHostSize_8u_C4R
 - image_L1_normrel, [2019](#)
- nppiNormRelL2GetBufferHostSize_16s_AC4R
 - image_L2_normrel, [2035](#)
- nppiNormRelL2GetBufferHostSize_16s_C1R
 - image_L2_normrel, [2035](#)
- nppiNormRelL2GetBufferHostSize_16s_C3R
 - image_L2_normrel, [2036](#)
- nppiNormRelL2GetBufferHostSize_16s_C4R
 - image_L2_normrel, [2036](#)
- nppiNormRelL2GetBufferHostSize_16u_AC4R
 - image_L2_normrel, [2036](#)
- nppiNormRelL2GetBufferHostSize_16u_C1MR
 - image_L2_normrel, [2037](#)
- nppiNormRelL2GetBufferHostSize_16u_C1R
 - image_L2_normrel, [2037](#)
- nppiNormRelL2GetBufferHostSize_16u_C3CMR
 - image_L2_normrel, [2037](#)
- nppiNormRelL2GetBufferHostSize_16u_C3R
 - image_L2_normrel, [2037](#)
- nppiNormRelL2GetBufferHostSize_16u_C4R
 - image_L2_normrel, [2038](#)
- nppiNormRelL2GetBufferHostSize_32f_AC4R
 - image_L2_normrel, [2038](#)
- nppiNormRelL2GetBufferHostSize_32f_C1MR
 - image_L2_normrel, [2038](#)
- nppiNormRelL2GetBufferHostSize_32f_C1R
 - image_L2_normrel, [2039](#)
- nppiNormRelL2GetBufferHostSize_32f_C3CMR
 - image_L2_normrel, [2039](#)
- nppiNormRelL2GetBufferHostSize_32f_C3R
 - image_L2_normrel, [2039](#)
- nppiNormRelL2GetBufferHostSize_32f_C4R
 - image_L2_normrel, [2039](#)
- nppiNormRelL2GetBufferHostSize_8s_C1MR
 - image_L2_normrel, [2040](#)
- nppiNormRelL2GetBufferHostSize_8s_C3CMR
 - image_L2_normrel, [2040](#)
- nppiNormRelL2GetBufferHostSize_8u_AC4R
 - image_L2_normrel, [2040](#)
- nppiNormRelL2GetBufferHostSize_8u_C1MR
 - image_L2_normrel, [2041](#)
- nppiNormRelL2GetBufferHostSize_8u_C1R
 - image_L2_normrel, [2041](#)
- nppiNormRelL2GetBufferHostSize_8u_C3CMR
 - image_L2_normrel, [2041](#)
- nppiNormRelL2GetBufferHostSize_8u_C3R
 - image_L2_normrel, [2041](#)
- nppiNormRelL2GetBufferHostSize_8u_C4R
 - image_L2_normrel, [2042](#)
- nppiNot_8u_AC4IR
 - image_not, [469](#)
- nppiNot_8u_AC4R

- image_not, [470](#)
- nppiNot_8u_C1IR
 - image_not, [470](#)
- nppiNot_8u_C1R
 - image_not, [470](#)
- nppiNot_8u_C3IR
 - image_not, [470](#)
- nppiNot_8u_C3R
 - image_not, [471](#)
- nppiNot_8u_C4IR
 - image_not, [471](#)
- nppiNot_8u_C4R
 - image_not, [471](#)
- nppiNV21ToBGR_8u_P2C4R
 - image_color_model_conversion, [553](#)
- nppiNV21ToRGB_8u_P2C4R
 - image_color_model_conversion, [553](#)
- nppiOr_16u_AC4IR
 - image_or, [447](#)
- nppiOr_16u_AC4R
 - image_or, [447](#)
- nppiOr_16u_C1IR
 - image_or, [447](#)
- nppiOr_16u_C1R
 - image_or, [448](#)
- nppiOr_16u_C3IR
 - image_or, [448](#)
- nppiOr_16u_C3R
 - image_or, [448](#)
- nppiOr_16u_C4IR
 - image_or, [449](#)
- nppiOr_16u_C4R
 - image_or, [449](#)
- nppiOr_32s_AC4IR
 - image_or, [450](#)
- nppiOr_32s_AC4R
 - image_or, [450](#)
- nppiOr_32s_C1IR
 - image_or, [450](#)
- nppiOr_32s_C1R
 - image_or, [451](#)
- nppiOr_32s_C3IR
 - image_or, [451](#)
- nppiOr_32s_C3R
 - image_or, [451](#)
- nppiOr_32s_C4IR
 - image_or, [452](#)
- nppiOr_32s_C4R
 - image_or, [452](#)
- nppiOr_8u_AC4IR
 - image_or, [453](#)
- nppiOr_8u_AC4R
 - image_or, [453](#)
- nppiOr_8u_C1IR
 - image_or, [453](#)
- nppiOr_8u_C1R
 - image_or, [454](#)
- nppiOr_8u_C3IR
 - image_or, [454](#)
- nppiOr_8u_C3R
 - image_or, [454](#)
- nppiOr_8u_C4IR
 - image_or, [455](#)
- nppiOr_8u_C4R
 - image_or, [455](#)
- nppiOrC_16u_AC4IR
 - image_orc, [385](#)
- nppiOrC_16u_AC4R
 - image_orc, [385](#)
- nppiOrC_16u_C1IR
 - image_orc, [385](#)
- nppiOrC_16u_C1R
 - image_orc, [386](#)
- nppiOrC_16u_C3IR
 - image_orc, [386](#)
- nppiOrC_16u_C3R
 - image_orc, [386](#)
- nppiOrC_16u_C4IR
 - image_orc, [387](#)
- nppiOrC_16u_C4R
 - image_orc, [387](#)
- nppiOrC_32s_AC4IR
 - image_orc, [387](#)
- nppiOrC_32s_AC4R
 - image_orc, [388](#)
- nppiOrC_32s_C1IR
 - image_orc, [388](#)
- nppiOrC_32s_C1R
 - image_orc, [388](#)
- nppiOrC_32s_C3IR
 - image_orc, [389](#)
- nppiOrC_32s_C3R
 - image_orc, [389](#)
- nppiOrC_32s_C4IR
 - image_orc, [389](#)
- nppiOrC_32s_C4R
 - image_orc, [390](#)
- nppiOrC_8u_AC4IR
 - image_orc, [390](#)
- nppiOrC_8u_AC4R
 - image_orc, [390](#)
- nppiOrC_8u_C1IR
 - image_orc, [391](#)
- nppiOrC_8u_C1R
 - image_orc, [391](#)
- nppiOrC_8u_C3IR
 - image_orc, [391](#)
- nppiOrC_8u_C3R

- image_orc, [392](#)
- npapiOrC_8u_C4IR
 - image_orc, [392](#)
- npapiOrC_8u_C4R
 - image_orc, [392](#)
- NppiPoint, [2873](#)
 - x, [2873](#)
 - y, [2873](#)
- npapiQualityIndex_16u32f_AC4R
 - image_quality_index, [2259](#)
- npapiQualityIndex_16u32f_C1R
 - image_quality_index, [2259](#)
- npapiQualityIndex_16u32f_C3R
 - image_quality_index, [2260](#)
- npapiQualityIndex_32f_AC4R
 - image_quality_index, [2260](#)
- npapiQualityIndex_32f_C1R
 - image_quality_index, [2261](#)
- npapiQualityIndex_32f_C3R
 - image_quality_index, [2261](#)
- npapiQualityIndex_8u32f_AC4R
 - image_quality_index, [2261](#)
- npapiQualityIndex_8u32f_C1R
 - image_quality_index, [2262](#)
- npapiQualityIndex_8u32f_C3R
 - image_quality_index, [2262](#)
- npapiQualityIndexGetBufferHostSize_16u32f_-AC4R
 - image_quality_index, [2263](#)
- npapiQualityIndexGetBufferHostSize_16u32f_C1R
 - image_quality_index, [2263](#)
- npapiQualityIndexGetBufferHostSize_16u32f_C3R
 - image_quality_index, [2263](#)
- npapiQualityIndexGetBufferHostSize_32f_AC4R
 - image_quality_index, [2264](#)
- npapiQualityIndexGetBufferHostSize_32f_C1R
 - image_quality_index, [2264](#)
- npapiQualityIndexGetBufferHostSize_32f_C3R
 - image_quality_index, [2264](#)
- npapiQualityIndexGetBufferHostSize_8u32f_AC4R
 - image_quality_index, [2265](#)
- npapiQualityIndexGetBufferHostSize_8u32f_C1R
 - image_quality_index, [2265](#)
- npapiQualityIndexGetBufferHostSize_8u32f_C3R
 - image_quality_index, [2265](#)
- npapiQuantFwdRawTableInit_JPEG_8u
 - image_quantization, [727](#)
- npapiQuantFwdTableInit_JPEG_8u16u
 - image_quantization, [728](#)
- npapiQuantInvTableInit_JPEG_8u16u
 - image_quantization, [728](#)
- NppiRect, [2874](#)
 - height, [2874](#)
 - width, [2874](#)
 - x, [2874](#)
 - y, [2874](#)
- npapiRectStdDev_32f_C1R
 - image_rectstddev, [2093](#)
- npapiRectStdDev_32s32f_C1R
 - image_rectstddev, [2094](#)
- npapiRectStdDev_32s_C1RSfs
 - image_rectstddev, [2094](#)
- npapiRemap_16s_AC4R
 - image_remap, [1434](#)
- npapiRemap_16s_C1R
 - image_remap, [1435](#)
- npapiRemap_16s_C3R
 - image_remap, [1435](#)
- npapiRemap_16s_C4R
 - image_remap, [1436](#)
- npapiRemap_16s_P3R
 - image_remap, [1437](#)
- npapiRemap_16s_P4R
 - image_remap, [1437](#)
- npapiRemap_16u_AC4R
 - image_remap, [1438](#)
- npapiRemap_16u_C1R
 - image_remap, [1438](#)
- npapiRemap_16u_C3R
 - image_remap, [1439](#)
- npapiRemap_16u_C4R
 - image_remap, [1440](#)
- npapiRemap_16u_P3R
 - image_remap, [1440](#)
- npapiRemap_16u_P4R
 - image_remap, [1441](#)
- npapiRemap_32f_AC4R
 - image_remap, [1441](#)
- npapiRemap_32f_C1R
 - image_remap, [1442](#)
- npapiRemap_32f_C3R
 - image_remap, [1443](#)
- npapiRemap_32f_C4R
 - image_remap, [1443](#)
- npapiRemap_32f_P3R
 - image_remap, [1444](#)
- npapiRemap_32f_P4R
 - image_remap, [1444](#)
- npapiRemap_64f_AC4R
 - image_remap, [1445](#)
- npapiRemap_64f_C1R
 - image_remap, [1446](#)
- npapiRemap_64f_C3R
 - image_remap, [1446](#)
- npapiRemap_64f_C4R
 - image_remap, [1447](#)
- npapiRemap_64f_P3R
 - image_remap, [1447](#)

- nppiRemap_64f_P4R
 - image_remap, [1448](#)
- nppiRemap_8u_AC4R
 - image_remap, [1449](#)
- nppiRemap_8u_C1R
 - image_remap, [1449](#)
- nppiRemap_8u_C3R
 - image_remap, [1450](#)
- nppiRemap_8u_C4R
 - image_remap, [1450](#)
- nppiRemap_8u_P3R
 - image_remap, [1451](#)
- nppiRemap_8u_P4R
 - image_remap, [1452](#)
- nppiResize_16u_AC4R
 - image_resize, [1421](#)
- nppiResize_16u_C1R
 - image_resize, [1422](#)
- nppiResize_16u_C3R
 - image_resize, [1422](#)
- nppiResize_16u_C4R
 - image_resize, [1423](#)
- nppiResize_16u_P3R
 - image_resize, [1423](#)
- nppiResize_16u_P4R
 - image_resize, [1424](#)
- nppiResize_32f_AC4R
 - image_resize, [1424](#)
- nppiResize_32f_C1R
 - image_resize, [1425](#)
- nppiResize_32f_C3R
 - image_resize, [1425](#)
- nppiResize_32f_C4R
 - image_resize, [1426](#)
- nppiResize_32f_P3R
 - image_resize, [1426](#)
- nppiResize_32f_P4R
 - image_resize, [1427](#)
- nppiResize_8u_AC4R
 - image_resize, [1427](#)
- nppiResize_8u_C1R
 - image_resize, [1428](#)
- nppiResize_8u_C3R
 - image_resize, [1428](#)
- nppiResize_8u_C4R
 - image_resize, [1429](#)
- nppiResize_8u_P3R
 - image_resize, [1429](#)
- nppiResize_8u_P4R
 - image_resize, [1430](#)
- nppiResizeAdvancedGetBufferHostSize_8u_C1R
 - image_resize_square_pixel, [1400](#)
- nppiResizeSqrPixel_16s_AC4R
 - image_resize_square_pixel, [1400](#)
- nppiResizeSqrPixel_16s_C1R
 - image_resize_square_pixel, [1401](#)
- nppiResizeSqrPixel_16s_C3R
 - image_resize_square_pixel, [1401](#)
- nppiResizeSqrPixel_16s_C4R
 - image_resize_square_pixel, [1402](#)
- nppiResizeSqrPixel_16s_P3R
 - image_resize_square_pixel, [1402](#)
- nppiResizeSqrPixel_16s_P4R
 - image_resize_square_pixel, [1403](#)
- nppiResizeSqrPixel_16u_AC4R
 - image_resize_square_pixel, [1404](#)
- nppiResizeSqrPixel_16u_C1R
 - image_resize_square_pixel, [1404](#)
- nppiResizeSqrPixel_16u_C3R
 - image_resize_square_pixel, [1405](#)
- nppiResizeSqrPixel_16u_C4R
 - image_resize_square_pixel, [1405](#)
- nppiResizeSqrPixel_16u_P3R
 - image_resize_square_pixel, [1406](#)
- nppiResizeSqrPixel_16u_P4R
 - image_resize_square_pixel, [1406](#)
- nppiResizeSqrPixel_32f_AC4R
 - image_resize_square_pixel, [1407](#)
- nppiResizeSqrPixel_32f_C1R
 - image_resize_square_pixel, [1408](#)
- nppiResizeSqrPixel_32f_C3R
 - image_resize_square_pixel, [1408](#)
- nppiResizeSqrPixel_32f_C4R
 - image_resize_square_pixel, [1409](#)
- nppiResizeSqrPixel_32f_P3R
 - image_resize_square_pixel, [1409](#)
- nppiResizeSqrPixel_32f_P4R
 - image_resize_square_pixel, [1410](#)
- nppiResizeSqrPixel_64f_AC4R
 - image_resize_square_pixel, [1410](#)
- nppiResizeSqrPixel_64f_C1R
 - image_resize_square_pixel, [1411](#)
- nppiResizeSqrPixel_64f_C3R
 - image_resize_square_pixel, [1411](#)
- nppiResizeSqrPixel_64f_C4R
 - image_resize_square_pixel, [1412](#)
- nppiResizeSqrPixel_64f_P3R
 - image_resize_square_pixel, [1412](#)
- nppiResizeSqrPixel_64f_P4R
 - image_resize_square_pixel, [1413](#)
- nppiResizeSqrPixel_8u_AC4R
 - image_resize_square_pixel, [1414](#)
- nppiResizeSqrPixel_8u_C1R
 - image_resize_square_pixel, [1414](#)
- nppiResizeSqrPixel_8u_C1R_Advanced
 - image_resize_square_pixel, [1415](#)
- nppiResizeSqrPixel_8u_C3R
 - image_resize_square_pixel, [1415](#)

- nppiResizeSqrPixel_8u_C4R
 - image_resize_square_pixel, [1416](#)
- nppiResizeSqrPixel_8u_P3R
 - image_resize_square_pixel, [1416](#)
- nppiResizeSqrPixel_8u_P4R
 - image_resize_square_pixel, [1417](#)
- nppiRGBToCbYCr422_8u_C3C2R
 - image_color_model_conversion, [554](#)
- nppiRGBToCbYCr422Gamma_8u_C3C2R
 - image_color_model_conversion, [554](#)
- nppiRGBToGray_16s_AC4C1R
 - image_color_model_conversion, [555](#)
- nppiRGBToGray_16s_C3C1R
 - image_color_model_conversion, [555](#)
- nppiRGBToGray_16u_AC4C1R
 - image_color_model_conversion, [555](#)
- nppiRGBToGray_16u_C3C1R
 - image_color_model_conversion, [556](#)
- nppiRGBToGray_32f_AC4C1R
 - image_color_model_conversion, [556](#)
- nppiRGBToGray_32f_C3C1R
 - image_color_model_conversion, [556](#)
- nppiRGBToGray_8u_AC4C1R
 - image_color_model_conversion, [557](#)
- nppiRGBToGray_8u_C3C1R
 - image_color_model_conversion, [557](#)
- nppiRGBToHLS_8u_AC4R
 - image_color_model_conversion, [557](#)
- nppiRGBToHLS_8u_C3R
 - image_color_model_conversion, [558](#)
- nppiRGBToHSV_8u_AC4R
 - image_color_model_conversion, [558](#)
- nppiRGBToHSV_8u_C3R
 - image_color_model_conversion, [558](#)
- nppiRGBToLUV_8u_AC4R
 - image_color_model_conversion, [559](#)
- nppiRGBToLUV_8u_C3R
 - image_color_model_conversion, [559](#)
- nppiRGBToXYZ_8u_AC4R
 - image_color_model_conversion, [559](#)
- nppiRGBToXYZ_8u_C3R
 - image_color_model_conversion, [560](#)
- nppiRGBToYCbCr420_8u_C3P3R
 - image_color_model_conversion, [560](#)
- nppiRGBToYCbCr422_8u_C3C2R
 - image_color_model_conversion, [560](#)
- nppiRGBToYCbCr422_8u_C3P3R
 - image_color_model_conversion, [561](#)
- nppiRGBToYCbCr422_8u_P3C2R
 - image_color_model_conversion, [561](#)
- nppiRGBToYCbCr_8u_AC4P3R
 - image_color_model_conversion, [562](#)
- nppiRGBToYCbCr_8u_AC4R
 - image_color_model_conversion, [562](#)
- nppiRGBToYCbCr_8u_C3P3R
 - image_color_model_conversion, [562](#)
- nppiRGBToYCbCr_8u_C3R
 - image_color_model_conversion, [563](#)
- nppiRGBToYCbCr_8u_P3R
 - image_color_model_conversion, [563](#)
- nppiRGBToYCC_8u_AC4R
 - image_color_model_conversion, [563](#)
- nppiRGBToYCC_8u_C3R
 - image_color_model_conversion, [564](#)
- nppiRGBToYCrCb420_8u_AC4P3R
 - image_color_model_conversion, [564](#)
- nppiRGBToYCrCb422_8u_C3C2R
 - image_color_model_conversion, [564](#)
- nppiRGBToYCrCb422_8u_P3C2R
 - image_color_model_conversion, [565](#)
- nppiRGBToYUV420_8u_C3P3R
 - image_color_model_conversion, [565](#)
- nppiRGBToYUV420_8u_P3R
 - image_color_model_conversion, [565](#)
- nppiRGBToYUV422_8u_C3C2R
 - image_color_model_conversion, [566](#)
- nppiRGBToYUV422_8u_C3P3R
 - image_color_model_conversion, [566](#)
- nppiRGBToYUV422_8u_P3R
 - image_color_model_conversion, [566](#)
- nppiRGBToYUV_8u_AC4P4R
 - image_color_model_conversion, [567](#)
- nppiRGBToYUV_8u_AC4R
 - image_color_model_conversion, [567](#)
- nppiRGBToYUV_8u_C3P3R
 - image_color_model_conversion, [568](#)
- nppiRGBToYUV_8u_C3R
 - image_color_model_conversion, [568](#)
- nppiRGBToYUV_8u_P3R
 - image_color_model_conversion, [568](#)
- nppiRotate_16u_AC4R
 - image_rotate, [1455](#)
- nppiRotate_16u_C1R
 - image_rotate, [1456](#)
- nppiRotate_16u_C3R
 - image_rotate, [1456](#)
- nppiRotate_16u_C4R
 - image_rotate, [1457](#)
- nppiRotate_32f_AC4R
 - image_rotate, [1457](#)
- nppiRotate_32f_C1R
 - image_rotate, [1458](#)
- nppiRotate_32f_C3R
 - image_rotate, [1458](#)
- nppiRotate_32f_C4R
 - image_rotate, [1459](#)
- nppiRotate_8u_AC4R
 - image_rotate, [1459](#)

- nppiRotate_8u_C1R
 - image_rotate, [1460](#)
- nppiRotate_8u_C3R
 - image_rotate, [1460](#)
- nppiRotate_8u_C4R
 - image_rotate, [1461](#)
- nppiRShiftC_16s_AC4IR
 - image_rshiftc, [408](#)
- nppiRShiftC_16s_AC4R
 - image_rshiftc, [408](#)
- nppiRShiftC_16s_C1IR
 - image_rshiftc, [409](#)
- nppiRShiftC_16s_C1R
 - image_rshiftc, [409](#)
- nppiRShiftC_16s_C3IR
 - image_rshiftc, [409](#)
- nppiRShiftC_16s_C3R
 - image_rshiftc, [410](#)
- nppiRShiftC_16s_C4IR
 - image_rshiftc, [410](#)
- nppiRShiftC_16s_C4R
 - image_rshiftc, [410](#)
- nppiRShiftC_16u_AC4IR
 - image_rshiftc, [411](#)
- nppiRShiftC_16u_AC4R
 - image_rshiftc, [411](#)
- nppiRShiftC_16u_C1IR
 - image_rshiftc, [411](#)
- nppiRShiftC_16u_C1R
 - image_rshiftc, [412](#)
- nppiRShiftC_16u_C3IR
 - image_rshiftc, [412](#)
- nppiRShiftC_16u_C3R
 - image_rshiftc, [412](#)
- nppiRShiftC_16u_C4IR
 - image_rshiftc, [413](#)
- nppiRShiftC_16u_C4R
 - image_rshiftc, [413](#)
- nppiRShiftC_32s_AC4IR
 - image_rshiftc, [413](#)
- nppiRShiftC_32s_AC4R
 - image_rshiftc, [414](#)
- nppiRShiftC_32s_C1IR
 - image_rshiftc, [414](#)
- nppiRShiftC_32s_C1R
 - image_rshiftc, [414](#)
- nppiRShiftC_32s_C3IR
 - image_rshiftc, [415](#)
- nppiRShiftC_32s_C3R
 - image_rshiftc, [415](#)
- nppiRShiftC_32s_C4IR
 - image_rshiftc, [415](#)
- nppiRShiftC_32s_C4R
 - image_rshiftc, [416](#)
- nppiRShiftC_8s_AC4IR
 - image_rshiftc, [416](#)
- nppiRShiftC_8s_AC4R
 - image_rshiftc, [416](#)
- nppiRShiftC_8s_C1IR
 - image_rshiftc, [417](#)
- nppiRShiftC_8s_C1R
 - image_rshiftc, [417](#)
- nppiRShiftC_8s_C3IR
 - image_rshiftc, [417](#)
- nppiRShiftC_8s_C3R
 - image_rshiftc, [418](#)
- nppiRShiftC_8s_C4IR
 - image_rshiftc, [418](#)
- nppiRShiftC_8s_C4R
 - image_rshiftc, [418](#)
- nppiRShiftC_8u_AC4IR
 - image_rshiftc, [419](#)
- nppiRShiftC_8u_AC4R
 - image_rshiftc, [419](#)
- nppiRShiftC_8u_C1IR
 - image_rshiftc, [419](#)
- nppiRShiftC_8u_C1R
 - image_rshiftc, [420](#)
- nppiRShiftC_8u_C3IR
 - image_rshiftc, [420](#)
- nppiRShiftC_8u_C3R
 - image_rshiftc, [420](#)
- nppiRShiftC_8u_C4IR
 - image_rshiftc, [421](#)
- nppiRShiftC_8u_C4R
 - image_rshiftc, [421](#)
- nppiSameNormLevelGetBufferHostSize_16u32f_-AC4R
 - crosscorrmenormlevel, [2230](#)
- nppiSameNormLevelGetBufferHostSize_16u32f_-C1R
 - crosscorrmenormlevel, [2231](#)
- nppiSameNormLevelGetBufferHostSize_16u32f_-C3R
 - crosscorrmenormlevel, [2231](#)
- nppiSameNormLevelGetBufferHostSize_16u32f_-C4R
 - crosscorrmenormlevel, [2231](#)
- nppiSameNormLevelGetBufferHostSize_32f_-AC4R
 - crosscorrmenormlevel, [2232](#)
- nppiSameNormLevelGetBufferHostSize_32f_-C1R
 - crosscorrmenormlevel, [2232](#)
- nppiSameNormLevelGetBufferHostSize_32f_-C3R
 - crosscorrmenormlevel, [2232](#)
- nppiSameNormLevelGetBufferHostSize_32f_-C4R
 - crosscorrmenormlevel, [2232](#)

- nppiSameNormLevelGetBufferHostSize_8s32f_-AC4R
 - crosscorrsmenormlevel, [2233](#)
- nppiSameNormLevelGetBufferHostSize_8s32f_-C1R
 - crosscorrsmenormlevel, [2233](#)
- nppiSameNormLevelGetBufferHostSize_8s32f_-C3R
 - crosscorrsmenormlevel, [2233](#)
- nppiSameNormLevelGetBufferHostSize_8s32f_-C4R
 - crosscorrsmenormlevel, [2234](#)
- nppiSameNormLevelGetBufferHostSize_8u32f_-AC4R
 - crosscorrsmenormlevel, [2234](#)
- nppiSameNormLevelGetBufferHostSize_8u32f_-C1R
 - crosscorrsmenormlevel, [2234](#)
- nppiSameNormLevelGetBufferHostSize_8u32f_-C3R
 - crosscorrsmenormlevel, [2234](#)
- nppiSameNormLevelGetBufferHostSize_8u32f_-C4R
 - crosscorrsmenormlevel, [2235](#)
- nppiSameNormLevelGetBufferHostSize_8u_-AC4RSfs
 - crosscorrsmenormlevel, [2235](#)
- nppiSameNormLevelGetBufferHostSize_8u_-C1RSfs
 - crosscorrsmenormlevel, [2235](#)
- nppiSameNormLevelGetBufferHostSize_8u_-C3RSfs
 - crosscorrsmenormlevel, [2236](#)
- nppiSameNormLevelGetBufferHostSize_8u_-C4RSfs
 - crosscorrsmenormlevel, [2236](#)
- nppiScale_16s8u_AC4R
 - image_scale, [867](#)
- nppiScale_16s8u_C1R
 - image_scale, [867](#)
- nppiScale_16s8u_C3R
 - image_scale, [867](#)
- nppiScale_16s8u_C4R
 - image_scale, [868](#)
- nppiScale_16u8u_AC4R
 - image_scale, [868](#)
- nppiScale_16u8u_C1R
 - image_scale, [868](#)
- nppiScale_16u8u_C3R
 - image_scale, [869](#)
- nppiScale_16u8u_C4R
 - image_scale, [869](#)
- nppiScale_32f8u_AC4R
 - image_scale, [869](#)
- nppiScale_32f8u_C1R
 - image_scale, [870](#)
- nppiScale_32f8u_C3R
 - image_scale, [870](#)
- nppiScale_32f8u_C4R
 - image_scale, [871](#)
- nppiScale_32s8u_AC4R
 - image_scale, [871](#)
- nppiScale_32s8u_C1R
 - image_scale, [871](#)
- nppiScale_32s8u_C3R
 - image_scale, [872](#)
- nppiScale_32s8u_C4R
 - image_scale, [872](#)
- nppiScale_8u16s_AC4R
 - image_scale, [872](#)
- nppiScale_8u16s_C1R
 - image_scale, [873](#)
- nppiScale_8u16s_C3R
 - image_scale, [873](#)
- nppiScale_8u16s_C4R
 - image_scale, [873](#)
- nppiScale_8u16u_AC4R
 - image_scale, [874](#)
- nppiScale_8u16u_C1R
 - image_scale, [874](#)
- nppiScale_8u16u_C3R
 - image_scale, [874](#)
- nppiScale_8u16u_C4R
 - image_scale, [875](#)
- nppiScale_8u32f_AC4R
 - image_scale, [875](#)
- nppiScale_8u32f_C1R
 - image_scale, [875](#)
- nppiScale_8u32f_C3R
 - image_scale, [876](#)
- nppiScale_8u32f_C4R
 - image_scale, [876](#)
- nppiScale_8u32s_AC4R
 - image_scale, [877](#)
- nppiScale_8u32s_C1R
 - image_scale, [877](#)
- nppiScale_8u32s_C3R
 - image_scale, [877](#)
- nppiScale_8u32s_C4R
 - image_scale, [878](#)
- nppiSet_16s_AC4MR
 - image_set, [745](#)
- nppiSet_16s_AC4R
 - image_set, [746](#)
- nppiSet_16s_C1MR
 - image_set, [746](#)
- nppiSet_16s_C1R
 - image_set, [746](#)

- nppiSet_16s_C2R
 - image_set, [747](#)
- nppiSet_16s_C3CR
 - image_set, [747](#)
- nppiSet_16s_C3MR
 - image_set, [747](#)
- nppiSet_16s_C3R
 - image_set, [748](#)
- nppiSet_16s_C4CR
 - image_set, [748](#)
- nppiSet_16s_C4MR
 - image_set, [748](#)
- nppiSet_16s_C4R
 - image_set, [749](#)
- nppiSet_16sc_AC4R
 - image_set, [749](#)
- nppiSet_16sc_C1R
 - image_set, [749](#)
- nppiSet_16sc_C2R
 - image_set, [750](#)
- nppiSet_16sc_C3R
 - image_set, [750](#)
- nppiSet_16sc_C4R
 - image_set, [750](#)
- nppiSet_16u_AC4MR
 - image_set, [751](#)
- nppiSet_16u_AC4R
 - image_set, [751](#)
- nppiSet_16u_C1MR
 - image_set, [751](#)
- nppiSet_16u_C1R
 - image_set, [752](#)
- nppiSet_16u_C2R
 - image_set, [752](#)
- nppiSet_16u_C3CR
 - image_set, [752](#)
- nppiSet_16u_C3MR
 - image_set, [753](#)
- nppiSet_16u_C3R
 - image_set, [753](#)
- nppiSet_16u_C4CR
 - image_set, [753](#)
- nppiSet_16u_C4MR
 - image_set, [754](#)
- nppiSet_16u_C4R
 - image_set, [754](#)
- nppiSet_32f_AC4MR
 - image_set, [754](#)
- nppiSet_32f_AC4R
 - image_set, [755](#)
- nppiSet_32f_C1MR
 - image_set, [755](#)
- nppiSet_32f_C1R
 - image_set, [755](#)
- nppiSet_32f_C2R
 - image_set, [756](#)
- nppiSet_32f_C3CR
 - image_set, [756](#)
- nppiSet_32f_C3MR
 - image_set, [756](#)
- nppiSet_32f_C3R
 - image_set, [757](#)
- nppiSet_32f_C4CR
 - image_set, [757](#)
- nppiSet_32f_C4MR
 - image_set, [757](#)
- nppiSet_32f_C4R
 - image_set, [758](#)
- nppiSet_32fc_AC4R
 - image_set, [758](#)
- nppiSet_32fc_C1R
 - image_set, [758](#)
- nppiSet_32fc_C2R
 - image_set, [759](#)
- nppiSet_32fc_C3R
 - image_set, [759](#)
- nppiSet_32fc_C4R
 - image_set, [759](#)
- nppiSet_32s_AC4MR
 - image_set, [760](#)
- nppiSet_32s_AC4R
 - image_set, [760](#)
- nppiSet_32s_C1MR
 - image_set, [760](#)
- nppiSet_32s_C1R
 - image_set, [761](#)
- nppiSet_32s_C2R
 - image_set, [761](#)
- nppiSet_32s_C3CR
 - image_set, [761](#)
- nppiSet_32s_C3MR
 - image_set, [762](#)
- nppiSet_32s_C3R
 - image_set, [762](#)
- nppiSet_32s_C4CR
 - image_set, [762](#)
- nppiSet_32s_C4MR
 - image_set, [763](#)
- nppiSet_32s_C4R
 - image_set, [763](#)
- nppiSet_32sc_AC4R
 - image_set, [763](#)
- nppiSet_32sc_C1R
 - image_set, [764](#)
- nppiSet_32sc_C2R
 - image_set, [764](#)
- nppiSet_32sc_C3R
 - image_set, [764](#)

nppiSet_32sc_C4R
 image_set, [765](#)
 nppiSet_32u_AC4R
 image_set, [765](#)
 nppiSet_32u_C1R
 image_set, [765](#)
 nppiSet_32u_C2R
 image_set, [766](#)
 nppiSet_32u_C3R
 image_set, [766](#)
 nppiSet_32u_C4R
 image_set, [766](#)
 nppiSet_8s_AC4R
 image_set, [767](#)
 nppiSet_8s_C1R
 image_set, [767](#)
 nppiSet_8s_C2R
 image_set, [767](#)
 nppiSet_8s_C3R
 image_set, [768](#)
 nppiSet_8s_C4R
 image_set, [768](#)
 nppiSet_8u_AC4MR
 image_set, [768](#)
 nppiSet_8u_AC4R
 image_set, [769](#)
 nppiSet_8u_C1MR
 image_set, [769](#)
 nppiSet_8u_C1R
 image_set, [769](#)
 nppiSet_8u_C2R
 image_set, [770](#)
 nppiSet_8u_C3CR
 image_set, [770](#)
 nppiSet_8u_C3MR
 image_set, [770](#)
 nppiSet_8u_C3R
 image_set, [771](#)
 nppiSet_8u_C4CR
 image_set, [771](#)
 nppiSet_8u_C4MR
 image_set, [771](#)
 nppiSet_8u_C4R
 image_set, [772](#)
 NppiSize, [2875](#)
 height, [2875](#)
 width, [2875](#)
 nppiSqr_16s_AC4IRSfs
 image_sqr, [334](#)
 nppiSqr_16s_AC4RSfs
 image_sqr, [334](#)
 nppiSqr_16s_C1IRSfs
 image_sqr, [334](#)
 nppiSqr_16s_C1RSfs
 image_sqr, [334](#)
 nppiSqr_16s_C3IRSfs
 image_sqr, [335](#)
 nppiSqr_16s_C3RSfs
 image_sqr, [335](#)
 nppiSqr_16s_C4IRSfs
 image_sqr, [335](#)
 nppiSqr_16s_C4RSfs
 image_sqr, [336](#)
 nppiSqr_16u_AC4IRSfs
 image_sqr, [336](#)
 nppiSqr_16u_AC4RSfs
 image_sqr, [336](#)
 nppiSqr_16u_C1IRSfs
 image_sqr, [337](#)
 nppiSqr_16u_C1RSfs
 image_sqr, [337](#)
 nppiSqr_16u_C3IRSfs
 image_sqr, [338](#)
 nppiSqr_16u_C3RSfs
 image_sqr, [338](#)
 nppiSqr_16u_C4IRSfs
 image_sqr, [338](#)
 nppiSqr_16u_C4RSfs
 image_sqr, [339](#)
 nppiSqr_32f_AC4IR
 image_sqr, [339](#)
 nppiSqr_32f_AC4R
 image_sqr, [339](#)
 nppiSqr_32f_C1IR
 image_sqr, [340](#)
 nppiSqr_32f_C1R
 image_sqr, [340](#)
 nppiSqr_32f_C3IR
 image_sqr, [340](#)
 nppiSqr_32f_C3R
 image_sqr, [340](#)
 nppiSqr_32f_C4IR
 image_sqr, [341](#)
 nppiSqr_32f_C4R
 image_sqr, [341](#)
 nppiSqr_8u_AC4IRSfs
 image_sqr, [341](#)
 nppiSqr_8u_AC4RSfs
 image_sqr, [342](#)
 nppiSqr_8u_C1IRSfs
 image_sqr, [342](#)
 nppiSqr_8u_C1RSfs
 image_sqr, [342](#)
 nppiSqr_8u_C3IRSfs
 image_sqr, [343](#)
 nppiSqr_8u_C3RSfs
 image_sqr, [343](#)
 nppiSqr_8u_C4IRSfs

- image_sqr, [343](#)
- nppiSqr_8u_C4RSfs
 - image_sqr, [344](#)
- nppiSqrDistanceFull_Norm_16u32f_AC4R
 - sqrdistancefullnorm, [2130](#)
- nppiSqrDistanceFull_Norm_16u32f_C1R
 - sqrdistancefullnorm, [2130](#)
- nppiSqrDistanceFull_Norm_16u32f_C3R
 - sqrdistancefullnorm, [2130](#)
- nppiSqrDistanceFull_Norm_16u32f_C4R
 - sqrdistancefullnorm, [2131](#)
- nppiSqrDistanceFull_Norm_32f_AC4R
 - sqrdistancefullnorm, [2131](#)
- nppiSqrDistanceFull_Norm_32f_C1R
 - sqrdistancefullnorm, [2132](#)
- nppiSqrDistanceFull_Norm_32f_C3R
 - sqrdistancefullnorm, [2132](#)
- nppiSqrDistanceFull_Norm_32f_C4R
 - sqrdistancefullnorm, [2133](#)
- nppiSqrDistanceFull_Norm_8s32f_AC4R
 - sqrdistancefullnorm, [2133](#)
- nppiSqrDistanceFull_Norm_8s32f_C1R
 - sqrdistancefullnorm, [2133](#)
- nppiSqrDistanceFull_Norm_8s32f_C3R
 - sqrdistancefullnorm, [2134](#)
- nppiSqrDistanceFull_Norm_8s32f_C4R
 - sqrdistancefullnorm, [2134](#)
- nppiSqrDistanceFull_Norm_8u32f_AC4R
 - sqrdistancefullnorm, [2135](#)
- nppiSqrDistanceFull_Norm_8u32f_C1R
 - sqrdistancefullnorm, [2135](#)
- nppiSqrDistanceFull_Norm_8u32f_C3R
 - sqrdistancefullnorm, [2136](#)
- nppiSqrDistanceFull_Norm_8u32f_C4R
 - sqrdistancefullnorm, [2136](#)
- nppiSqrDistanceFull_Norm_8u_AC4RSfs
 - sqrdistancefullnorm, [2136](#)
- nppiSqrDistanceFull_Norm_8u_C1RSfs
 - sqrdistancefullnorm, [2137](#)
- nppiSqrDistanceFull_Norm_8u_C3RSfs
 - sqrdistancefullnorm, [2137](#)
- nppiSqrDistanceFull_Norm_8u_C4RSfs
 - sqrdistancefullnorm, [2138](#)
- nppiSqrDistanceSame_Norm_16u32f_AC4R
 - sqrdistancesamenorm, [2141](#)
- nppiSqrDistanceSame_Norm_16u32f_C1R
 - sqrdistancesamenorm, [2141](#)
- nppiSqrDistanceSame_Norm_16u32f_C3R
 - sqrdistancesamenorm, [2142](#)
- nppiSqrDistanceSame_Norm_16u32f_C4R
 - sqrdistancesamenorm, [2142](#)
- nppiSqrDistanceSame_Norm_32f_AC4R
 - sqrdistancesamenorm, [2142](#)
- nppiSqrDistanceSame_Norm_32f_C1R
 - sqrdistancesamenorm, [2143](#)
- nppiSqrDistanceSame_Norm_32f_C3R
 - sqrdistancesamenorm, [2143](#)
- nppiSqrDistanceSame_Norm_32f_C4R
 - sqrdistancesamenorm, [2144](#)
- nppiSqrDistanceSame_Norm_8s32f_AC4R
 - sqrdistancesamenorm, [2144](#)
- nppiSqrDistanceSame_Norm_8s32f_C1R
 - sqrdistancesamenorm, [2145](#)
- nppiSqrDistanceSame_Norm_8s32f_C3R
 - sqrdistancesamenorm, [2145](#)
- nppiSqrDistanceSame_Norm_8s32f_C4R
 - sqrdistancesamenorm, [2145](#)
- nppiSqrDistanceSame_Norm_8u32f_AC4R
 - sqrdistancesamenorm, [2146](#)
- nppiSqrDistanceSame_Norm_8u32f_C1R
 - sqrdistancesamenorm, [2146](#)
- nppiSqrDistanceSame_Norm_8u32f_C3R
 - sqrdistancesamenorm, [2147](#)
- nppiSqrDistanceSame_Norm_8u32f_C4R
 - sqrdistancesamenorm, [2147](#)
- nppiSqrDistanceSame_Norm_8u_AC4RSfs
 - sqrdistancesamenorm, [2148](#)
- nppiSqrDistanceSame_Norm_8u_C1RSfs
 - sqrdistancesamenorm, [2148](#)
- nppiSqrDistanceSame_Norm_8u_C3RSfs
 - sqrdistancesamenorm, [2149](#)
- nppiSqrDistanceSame_Norm_8u_C4RSfs
 - sqrdistancesamenorm, [2149](#)
- nppiSqrDistanceValid_Norm_16u32f_AC4R
 - sqrdistancevalidnorm, [2152](#)
- nppiSqrDistanceValid_Norm_16u32f_C1R
 - sqrdistancevalidnorm, [2152](#)
- nppiSqrDistanceValid_Norm_16u32f_C3R
 - sqrdistancevalidnorm, [2153](#)
- nppiSqrDistanceValid_Norm_16u32f_C4R
 - sqrdistancevalidnorm, [2153](#)
- nppiSqrDistanceValid_Norm_32f_AC4R
 - sqrdistancevalidnorm, [2153](#)
- nppiSqrDistanceValid_Norm_32f_C1R
 - sqrdistancevalidnorm, [2154](#)
- nppiSqrDistanceValid_Norm_32f_C3R
 - sqrdistancevalidnorm, [2154](#)
- nppiSqrDistanceValid_Norm_32f_C4R
 - sqrdistancevalidnorm, [2155](#)
- nppiSqrDistanceValid_Norm_8s32f_AC4R
 - sqrdistancevalidnorm, [2155](#)
- nppiSqrDistanceValid_Norm_8s32f_C1R
 - sqrdistancevalidnorm, [2156](#)
- nppiSqrDistanceValid_Norm_8s32f_C3R
 - sqrdistancevalidnorm, [2156](#)
- nppiSqrDistanceValid_Norm_8s32f_C4R
 - sqrdistancevalidnorm, [2156](#)
- nppiSqrDistanceValid_Norm_8u32f_AC4R

- sqrdistancevalidnorm, [2157](#)
- nppiSqrDistanceValid_Norm_8u32f_C1R
 - sqrdistancevalidnorm, [2157](#)
- nppiSqrDistanceValid_Norm_8u32f_C3R
 - sqrdistancevalidnorm, [2158](#)
- nppiSqrDistanceValid_Norm_8u32f_C4R
 - sqrdistancevalidnorm, [2158](#)
- nppiSqrDistanceValid_Norm_8u_AC4RSfs
 - sqrdistancevalidnorm, [2159](#)
- nppiSqrDistanceValid_Norm_8u_C1RSfs
 - sqrdistancevalidnorm, [2159](#)
- nppiSqrDistanceValid_Norm_8u_C3RSfs
 - sqrdistancevalidnorm, [2160](#)
- nppiSqrDistanceValid_Norm_8u_C4RSfs
 - sqrdistancevalidnorm, [2160](#)
- nppiSqrIntegral_8u32f64f_C1R
 - image_sqrintegral, [2090](#)
- nppiSqrIntegral_8u32s64f_C1R
 - image_sqrintegral, [2091](#)
- nppiSqrIntegral_8u32s_C1R
 - image_sqrintegral, [2091](#)
- nppiSqrt_16s_AC4IRSfs
 - image_sqrt, [347](#)
- nppiSqrt_16s_AC4RSfs
 - image_sqrt, [347](#)
- nppiSqrt_16s_C1IRSfs
 - image_sqrt, [348](#)
- nppiSqrt_16s_C1RSfs
 - image_sqrt, [348](#)
- nppiSqrt_16s_C3IRSfs
 - image_sqrt, [349](#)
- nppiSqrt_16s_C3RSfs
 - image_sqrt, [349](#)
- nppiSqrt_16u_AC4IRSfs
 - image_sqrt, [349](#)
- nppiSqrt_16u_AC4RSfs
 - image_sqrt, [350](#)
- nppiSqrt_16u_C1IRSfs
 - image_sqrt, [350](#)
- nppiSqrt_16u_C1RSfs
 - image_sqrt, [350](#)
- nppiSqrt_16u_C3IRSfs
 - image_sqrt, [351](#)
- nppiSqrt_16u_C3RSfs
 - image_sqrt, [351](#)
- nppiSqrt_32f_AC4IR
 - image_sqrt, [351](#)
- nppiSqrt_32f_AC4R
 - image_sqrt, [352](#)
- nppiSqrt_32f_C1IR
 - image_sqrt, [352](#)
- nppiSqrt_32f_C1R
 - image_sqrt, [352](#)
- nppiSqrt_32f_C3IR
 - image_sqrt, [353](#)
- nppiSqrt_32f_C3R
 - image_sqrt, [353](#)
- nppiSqrt_32f_C4IR
 - image_sqrt, [353](#)
- nppiSqrt_32f_C4R
 - image_sqrt, [354](#)
- nppiSqrt_8u_AC4IRSfs
 - image_sqrt, [354](#)
- nppiSqrt_8u_AC4RSfs
 - image_sqrt, [354](#)
- nppiSqrt_8u_C1IRSfs
 - image_sqrt, [355](#)
- nppiSqrt_8u_C1RSfs
 - image_sqrt, [355](#)
- nppiSqrt_8u_C3IRSfs
 - image_sqrt, [356](#)
- nppiSqrt_8u_C3RSfs
 - image_sqrt, [356](#)
- nppiSub_16s_AC4IRSfs
 - image_sub, [252](#)
- nppiSub_16s_AC4RSfs
 - image_sub, [253](#)
- nppiSub_16s_C1IRSfs
 - image_sub, [253](#)
- nppiSub_16s_C1RSfs
 - image_sub, [253](#)
- nppiSub_16s_C3IRSfs
 - image_sub, [254](#)
- nppiSub_16s_C3RSfs
 - image_sub, [254](#)
- nppiSub_16s_C4IRSfs
 - image_sub, [255](#)
- nppiSub_16s_C4RSfs
 - image_sub, [255](#)
- nppiSub_16sc_AC4IRSfs
 - image_sub, [255](#)
- nppiSub_16sc_AC4RSfs
 - image_sub, [256](#)
- nppiSub_16sc_C1IRSfs
 - image_sub, [256](#)
- nppiSub_16sc_C1RSfs
 - image_sub, [257](#)
- nppiSub_16sc_C3IRSfs
 - image_sub, [257](#)
- nppiSub_16sc_C3RSfs
 - image_sub, [257](#)
- nppiSub_16u_AC4IRSfs
 - image_sub, [258](#)
- nppiSub_16u_AC4RSfs
 - image_sub, [258](#)
- nppiSub_16u_C1IRSfs
 - image_sub, [259](#)
- nppiSub_16u_C1RSfs

- image_sub, [259](#)
- nppiSub_16u_C3IRSfs
 - image_sub, [260](#)
- nppiSub_16u_C3RSfs
 - image_sub, [260](#)
- nppiSub_16u_C4IRSfs
 - image_sub, [260](#)
- nppiSub_16u_C4RSfs
 - image_sub, [261](#)
- nppiSub_32f_AC4IR
 - image_sub, [261](#)
- nppiSub_32f_AC4R
 - image_sub, [262](#)
- nppiSub_32f_C1IR
 - image_sub, [262](#)
- nppiSub_32f_C1R
 - image_sub, [262](#)
- nppiSub_32f_C3IR
 - image_sub, [263](#)
- nppiSub_32f_C3R
 - image_sub, [263](#)
- nppiSub_32f_C4IR
 - image_sub, [264](#)
- nppiSub_32f_C4R
 - image_sub, [264](#)
- nppiSub_32fc_AC4IR
 - image_sub, [264](#)
- nppiSub_32fc_AC4R
 - image_sub, [265](#)
- nppiSub_32fc_C1IR
 - image_sub, [265](#)
- nppiSub_32fc_C1R
 - image_sub, [266](#)
- nppiSub_32fc_C3IR
 - image_sub, [266](#)
- nppiSub_32fc_C3R
 - image_sub, [266](#)
- nppiSub_32fc_C4IR
 - image_sub, [267](#)
- nppiSub_32fc_C4R
 - image_sub, [267](#)
- nppiSub_32s_C1IRSfs
 - image_sub, [268](#)
- nppiSub_32s_C1R
 - image_sub, [268](#)
- nppiSub_32s_C1RSfs
 - image_sub, [268](#)
- nppiSub_32s_C3IRSfs
 - image_sub, [269](#)
- nppiSub_32s_C3RSfs
 - image_sub, [269](#)
- nppiSub_32s_C4IRSfs
 - image_sub, [270](#)
- nppiSub_32s_C4RSfs
 - image_sub, [270](#)
- nppiSub_32sc_AC4IRSfs
 - image_sub, [271](#)
- nppiSub_32sc_AC4RSfs
 - image_sub, [271](#)
- nppiSub_32sc_C1IRSfs
 - image_sub, [271](#)
- nppiSub_32sc_C1RSfs
 - image_sub, [272](#)
- nppiSub_32sc_C3IRSfs
 - image_sub, [272](#)
- nppiSub_32sc_C3RSfs
 - image_sub, [273](#)
- nppiSub_8u_AC4IRSfs
 - image_sub, [273](#)
- nppiSub_8u_AC4RSfs
 - image_sub, [273](#)
- nppiSub_8u_C1IRSfs
 - image_sub, [274](#)
- nppiSub_8u_C1RSfs
 - image_sub, [274](#)
- nppiSub_8u_C3IRSfs
 - image_sub, [275](#)
- nppiSub_8u_C3RSfs
 - image_sub, [275](#)
- nppiSub_8u_C4IRSfs
 - image_sub, [275](#)
- nppiSub_8u_C4RSfs
 - image_sub, [276](#)
- nppiSubC_16s_AC4IRSfs
 - image_subc, [120](#)
- nppiSubC_16s_AC4RSfs
 - image_subc, [120](#)
- nppiSubC_16s_C1IRSfs
 - image_subc, [120](#)
- nppiSubC_16s_C1RSfs
 - image_subc, [121](#)
- nppiSubC_16s_C3IRSfs
 - image_subc, [121](#)
- nppiSubC_16s_C3RSfs
 - image_subc, [121](#)
- nppiSubC_16s_C4IRSfs
 - image_subc, [122](#)
- nppiSubC_16s_C4RSfs
 - image_subc, [122](#)
- nppiSubC_16sc_AC4IRSfs
 - image_subc, [123](#)
- nppiSubC_16sc_AC4RSfs
 - image_subc, [123](#)
- nppiSubC_16sc_C1IRSfs
 - image_subc, [123](#)
- nppiSubC_16sc_C1RSfs
 - image_subc, [124](#)
- nppiSubC_16sc_C3IRSfs

image_subc, [124](#)
 nppiSubC_16sc_C3RSfs
 image_subc, [125](#)
 nppiSubC_16u_AC4IRSfs
 image_subc, [125](#)
 nppiSubC_16u_AC4RSfs
 image_subc, [125](#)
 nppiSubC_16u_C1IRSfs
 image_subc, [126](#)
 nppiSubC_16u_C1RSfs
 image_subc, [126](#)
 nppiSubC_16u_C3IRSfs
 image_subc, [127](#)
 nppiSubC_16u_C3RSfs
 image_subc, [127](#)
 nppiSubC_16u_C4IRSfs
 image_subc, [127](#)
 nppiSubC_16u_C4RSfs
 image_subc, [128](#)
 nppiSubC_32f_AC4IR
 image_subc, [128](#)
 nppiSubC_32f_AC4R
 image_subc, [128](#)
 nppiSubC_32f_C1IR
 image_subc, [129](#)
 nppiSubC_32f_C1R
 image_subc, [129](#)
 nppiSubC_32f_C3IR
 image_subc, [129](#)
 nppiSubC_32f_C3R
 image_subc, [130](#)
 nppiSubC_32f_C4IR
 image_subc, [130](#)
 nppiSubC_32f_C4R
 image_subc, [130](#)
 nppiSubC_32fc_AC4IR
 image_subc, [131](#)
 nppiSubC_32fc_AC4R
 image_subc, [131](#)
 nppiSubC_32fc_C1IR
 image_subc, [131](#)
 nppiSubC_32fc_C1R
 image_subc, [132](#)
 nppiSubC_32fc_C3IR
 image_subc, [132](#)
 nppiSubC_32fc_C3R
 image_subc, [132](#)
 nppiSubC_32fc_C4IR
 image_subc, [133](#)
 nppiSubC_32fc_C4R
 image_subc, [133](#)
 nppiSubC_32s_C1IRSfs
 image_subc, [134](#)
 nppiSubC_32s_C1RSfs

image_subc, [134](#)
 nppiSubC_32s_C3IRSfs
 image_subc, [134](#)
 nppiSubC_32s_C3RSfs
 image_subc, [135](#)
 nppiSubC_32sc_AC4IRSfs
 image_subc, [135](#)
 nppiSubC_32sc_AC4RSfs
 image_subc, [135](#)
 nppiSubC_32sc_C1IRSfs
 image_subc, [136](#)
 nppiSubC_32sc_C1RSfs
 image_subc, [136](#)
 nppiSubC_32sc_C3IRSfs
 image_subc, [137](#)
 nppiSubC_32sc_C3RSfs
 image_subc, [137](#)
 nppiSubC_8u_AC4IRSfs
 image_subc, [137](#)
 nppiSubC_8u_AC4RSfs
 image_subc, [138](#)
 nppiSubC_8u_C1IRSfs
 image_subc, [138](#)
 nppiSubC_8u_C1RSfs
 image_subc, [139](#)
 nppiSubC_8u_C3IRSfs
 image_subc, [139](#)
 nppiSubC_8u_C3RSfs
 image_subc, [139](#)
 nppiSubC_8u_C4IRSfs
 image_subc, [140](#)
 nppiSubC_8u_C4RSfs
 image_subc, [140](#)
 nppiSum_16s_AC4R
 image_sum, [1705](#)
 nppiSum_16s_C1R
 image_sum, [1705](#)
 nppiSum_16s_C3R
 image_sum, [1705](#)
 nppiSum_16s_C4R
 image_sum, [1706](#)
 nppiSum_16u_AC4R
 image_sum, [1706](#)
 nppiSum_16u_C1R
 image_sum, [1706](#)
 nppiSum_16u_C3R
 image_sum, [1707](#)
 nppiSum_16u_C4R
 image_sum, [1707](#)
 nppiSum_32f_AC4R
 image_sum, [1707](#)
 nppiSum_32f_C1R
 image_sum, [1708](#)
 nppiSum_32f_C3R

- image_sum, [1708](#)
- nppiSum_32f_C4R
 - image_sum, [1708](#)
- nppiSum_8u64s_C1R
 - image_sum, [1709](#)
- nppiSum_8u64s_C4R
 - image_sum, [1709](#)
- nppiSum_8u_AC4R
 - image_sum, [1710](#)
- nppiSum_8u_C1R
 - image_sum, [1710](#)
- nppiSum_8u_C3R
 - image_sum, [1710](#)
- nppiSum_8u_C4R
 - image_sum, [1711](#)
- nppiSumGetBufferHostSize_16s_AC4R
 - image_sum, [1711](#)
- nppiSumGetBufferHostSize_16s_C1R
 - image_sum, [1711](#)
- nppiSumGetBufferHostSize_16s_C3R
 - image_sum, [1712](#)
- nppiSumGetBufferHostSize_16s_C4R
 - image_sum, [1712](#)
- nppiSumGetBufferHostSize_16u_AC4R
 - image_sum, [1712](#)
- nppiSumGetBufferHostSize_16u_C1R
 - image_sum, [1713](#)
- nppiSumGetBufferHostSize_16u_C3R
 - image_sum, [1713](#)
- nppiSumGetBufferHostSize_16u_C4R
 - image_sum, [1713](#)
- nppiSumGetBufferHostSize_32f_AC4R
 - image_sum, [1713](#)
- nppiSumGetBufferHostSize_32f_C1R
 - image_sum, [1714](#)
- nppiSumGetBufferHostSize_32f_C3R
 - image_sum, [1714](#)
- nppiSumGetBufferHostSize_32f_C4R
 - image_sum, [1714](#)
- nppiSumGetBufferHostSize_8u64s_C1R
 - image_sum, [1715](#)
- nppiSumGetBufferHostSize_8u64s_C4R
 - image_sum, [1715](#)
- nppiSumGetBufferHostSize_8u_AC4R
 - image_sum, [1715](#)
- nppiSumGetBufferHostSize_8u_C1R
 - image_sum, [1715](#)
- nppiSumGetBufferHostSize_8u_C3R
 - image_sum, [1716](#)
- nppiSumGetBufferHostSize_8u_C4R
 - image_sum, [1716](#)
- nppiSumWindowColumn_16s32f_C1R
 - image_1D_window_sum, [1198](#)
- nppiSumWindowColumn_16s32f_C3R
 - image_1D_window_sum, [1199](#)
- nppiSumWindowColumn_16s32f_C4R
 - image_1D_window_sum, [1199](#)
- nppiSumWindowColumn_16u32f_C1R
 - image_1D_window_sum, [1200](#)
- nppiSumWindowColumn_16u32f_C3R
 - image_1D_window_sum, [1200](#)
- nppiSumWindowColumn_16u32f_C4R
 - image_1D_window_sum, [1201](#)
- nppiSumWindowColumn_8u32f_C1R
 - image_1D_window_sum, [1201](#)
- nppiSumWindowColumn_8u32f_C3R
 - image_1D_window_sum, [1201](#)
- nppiSumWindowColumn_8u32f_C4R
 - image_1D_window_sum, [1202](#)
- nppiSumWindowColumnBorder_16s32f_C1R
 - image_1D_window_sum_border, [1210](#)
- nppiSumWindowColumnBorder_16s32f_C3R
 - image_1D_window_sum_border, [1210](#)
- nppiSumWindowColumnBorder_16s32f_C4R
 - image_1D_window_sum_border, [1211](#)
- nppiSumWindowColumnBorder_16u32f_C1R
 - image_1D_window_sum_border, [1211](#)
- nppiSumWindowColumnBorder_16u32f_C3R
 - image_1D_window_sum_border, [1212](#)
- nppiSumWindowColumnBorder_16u32f_C4R
 - image_1D_window_sum_border, [1212](#)
- nppiSumWindowColumnBorder_8u32f_C1R
 - image_1D_window_sum_border, [1213](#)
- nppiSumWindowColumnBorder_8u32f_C3R
 - image_1D_window_sum_border, [1214](#)
- nppiSumWindowColumnBorder_8u32f_C4R
 - image_1D_window_sum_border, [1214](#)
- nppiSumWindowRow_16s32f_C1R
 - image_1D_window_sum, [1202](#)
- nppiSumWindowRow_16s32f_C3R
 - image_1D_window_sum, [1203](#)
- nppiSumWindowRow_16s32f_C4R
 - image_1D_window_sum, [1203](#)
- nppiSumWindowRow_16u32f_C1R
 - image_1D_window_sum, [1204](#)
- nppiSumWindowRow_16u32f_C3R
 - image_1D_window_sum, [1204](#)
- nppiSumWindowRow_16u32f_C4R
 - image_1D_window_sum, [1205](#)
- nppiSumWindowRow_8u32f_C1R
 - image_1D_window_sum, [1205](#)
- nppiSumWindowRow_8u32f_C3R
 - image_1D_window_sum, [1206](#)
- nppiSumWindowRow_8u32f_C4R
 - image_1D_window_sum, [1206](#)
- nppiSumWindowRowBorder_16s32f_C1R
 - image_1D_window_sum_border, [1215](#)
- nppiSumWindowRowBorder_16s32f_C3R

- image_1D_window_sum_border, [1215](#)
- nppiSumWindowRowBorder_16s32f_C4R
 - image_1D_window_sum_border, [1216](#)
- nppiSumWindowRowBorder_16u32f_C1R
 - image_1D_window_sum_border, [1216](#)
- nppiSumWindowRowBorder_16u32f_C3R
 - image_1D_window_sum_border, [1217](#)
- nppiSumWindowRowBorder_16u32f_C4R
 - image_1D_window_sum_border, [1218](#)
- nppiSumWindowRowBorder_8u32f_C1R
 - image_1D_window_sum_border, [1218](#)
- nppiSumWindowRowBorder_8u32f_C3R
 - image_1D_window_sum_border, [1219](#)
- nppiSumWindowRowBorder_8u32f_C4R
 - image_1D_window_sum_border, [1219](#)
- nppiSwapChannels_16s_AC4R
 - image_swap_channels, [945](#)
- nppiSwapChannels_16s_C3C4R
 - image_swap_channels, [945](#)
- nppiSwapChannels_16s_C3IR
 - image_swap_channels, [945](#)
- nppiSwapChannels_16s_C3R
 - image_swap_channels, [946](#)
- nppiSwapChannels_16s_C4C3R
 - image_swap_channels, [946](#)
- nppiSwapChannels_16s_C4IR
 - image_swap_channels, [947](#)
- nppiSwapChannels_16s_C4R
 - image_swap_channels, [947](#)
- nppiSwapChannels_16u_AC4R
 - image_swap_channels, [947](#)
- nppiSwapChannels_16u_C3C4R
 - image_swap_channels, [948](#)
- nppiSwapChannels_16u_C3IR
 - image_swap_channels, [948](#)
- nppiSwapChannels_16u_C3R
 - image_swap_channels, [949](#)
- nppiSwapChannels_16u_C4C3R
 - image_swap_channels, [949](#)
- nppiSwapChannels_16u_C4IR
 - image_swap_channels, [950](#)
- nppiSwapChannels_16u_C4R
 - image_swap_channels, [950](#)
- nppiSwapChannels_32f_AC4R
 - image_swap_channels, [950](#)
- nppiSwapChannels_32f_C3C4R
 - image_swap_channels, [951](#)
- nppiSwapChannels_32f_C3IR
 - image_swap_channels, [951](#)
- nppiSwapChannels_32f_C3R
 - image_swap_channels, [952](#)
- nppiSwapChannels_32f_C4C3R
 - image_swap_channels, [952](#)
- nppiSwapChannels_32f_C4IR
 - image_swap_channels, [953](#)
- nppiSwapChannels_32f_C4R
 - image_swap_channels, [953](#)
- nppiSwapChannels_32s_AC4R
 - image_swap_channels, [953](#)
- nppiSwapChannels_32s_C3C4R
 - image_swap_channels, [954](#)
- nppiSwapChannels_32s_C3IR
 - image_swap_channels, [954](#)
- nppiSwapChannels_32s_C3R
 - image_swap_channels, [955](#)
- nppiSwapChannels_32s_C4C3R
 - image_swap_channels, [955](#)
- nppiSwapChannels_32s_C4IR
 - image_swap_channels, [956](#)
- nppiSwapChannels_32s_C4R
 - image_swap_channels, [956](#)
- nppiSwapChannels_8u_AC4R
 - image_swap_channels, [956](#)
- nppiSwapChannels_8u_C3C4R
 - image_swap_channels, [957](#)
- nppiSwapChannels_8u_C3IR
 - image_swap_channels, [957](#)
- nppiSwapChannels_8u_C3R
 - image_swap_channels, [958](#)
- nppiSwapChannels_8u_C4C3R
 - image_swap_channels, [958](#)
- nppiSwapChannels_8u_C4IR
 - image_swap_channels, [959](#)
- nppiSwapChannels_8u_C4R
 - image_swap_channels, [959](#)
- nppiThreshold_16s_AC4IR
 - image_threshold_operations, [2387](#)
- nppiThreshold_16s_AC4R
 - image_threshold_operations, [2387](#)
- nppiThreshold_16s_C1IR
 - image_threshold_operations, [2388](#)
- nppiThreshold_16s_C1R
 - image_threshold_operations, [2388](#)
- nppiThreshold_16s_C3IR
 - image_threshold_operations, [2389](#)
- nppiThreshold_16s_C3R
 - image_threshold_operations, [2389](#)
- nppiThreshold_16u_AC4IR
 - image_threshold_operations, [2390](#)
- nppiThreshold_16u_AC4R
 - image_threshold_operations, [2390](#)
- nppiThreshold_16u_C1IR
 - image_threshold_operations, [2390](#)
- nppiThreshold_16u_C1R
 - image_threshold_operations, [2391](#)
- nppiThreshold_16u_C3IR
 - image_threshold_operations, [2391](#)
- nppiThreshold_16u_C3R

- image_threshold_operations, 2392
- npptThreshold_32f_AC4IR
 - image_threshold_operations, 2392
- npptThreshold_32f_AC4R
 - image_threshold_operations, 2393
- npptThreshold_32f_C1IR
 - image_threshold_operations, 2393
- npptThreshold_32f_C1R
 - image_threshold_operations, 2394
- npptThreshold_32f_C3IR
 - image_threshold_operations, 2394
- npptThreshold_32f_C3R
 - image_threshold_operations, 2394
- npptThreshold_8u_AC4IR
 - image_threshold_operations, 2395
- npptThreshold_8u_AC4R
 - image_threshold_operations, 2395
- npptThreshold_8u_C1IR
 - image_threshold_operations, 2396
- npptThreshold_8u_C1R
 - image_threshold_operations, 2396
- npptThreshold_8u_C3IR
 - image_threshold_operations, 2397
- npptThreshold_8u_C3R
 - image_threshold_operations, 2397
- npptThreshold_GT_16s_AC4IR
 - image_threshold_operations, 2398
- npptThreshold_GT_16s_AC4R
 - image_threshold_operations, 2398
- npptThreshold_GT_16s_C1IR
 - image_threshold_operations, 2399
- npptThreshold_GT_16s_C1R
 - image_threshold_operations, 2399
- npptThreshold_GT_16s_C3IR
 - image_threshold_operations, 2399
- npptThreshold_GT_16s_C3R
 - image_threshold_operations, 2400
- npptThreshold_GT_16u_AC4IR
 - image_threshold_operations, 2400
- npptThreshold_GT_16u_AC4R
 - image_threshold_operations, 2401
- npptThreshold_GT_16u_C1IR
 - image_threshold_operations, 2401
- npptThreshold_GT_16u_C1R
 - image_threshold_operations, 2401
- npptThreshold_GT_16u_C3IR
 - image_threshold_operations, 2402
- npptThreshold_GT_16u_C3R
 - image_threshold_operations, 2402
- npptThreshold_GT_32f_AC4IR
 - image_threshold_operations, 2403
- npptThreshold_GT_32f_AC4R
 - image_threshold_operations, 2403
- npptThreshold_GT_32f_C1IR
 - image_threshold_operations, 2403
- npptThreshold_GT_32f_C1R
 - image_threshold_operations, 2403
- npptThreshold_GT_32f_C3IR
 - image_threshold_operations, 2404
- npptThreshold_GT_32f_C3R
 - image_threshold_operations, 2404
- npptThreshold_GT_8u_AC4IR
 - image_threshold_operations, 2405
- npptThreshold_GT_8u_AC4R
 - image_threshold_operations, 2405
- npptThreshold_GT_8u_C1IR
 - image_threshold_operations, 2406
- npptThreshold_GT_8u_C1R
 - image_threshold_operations, 2406
- npptThreshold_GT_8u_C3IR
 - image_threshold_operations, 2407
- npptThreshold_GT_8u_C3R
 - image_threshold_operations, 2407
- npptThreshold_GTVAl_16s_AC4IR
 - image_threshold_operations, 2407
- npptThreshold_GTVAl_16s_AC4R
 - image_threshold_operations, 2408
- npptThreshold_GTVAl_16s_C1IR
 - image_threshold_operations, 2408
- npptThreshold_GTVAl_16s_C1R
 - image_threshold_operations, 2409
- npptThreshold_GTVAl_16s_C3IR
 - image_threshold_operations, 2409
- npptThreshold_GTVAl_16s_C3R
 - image_threshold_operations, 2409
- npptThreshold_GTVAl_16u_AC4IR
 - image_threshold_operations, 2410
- npptThreshold_GTVAl_16u_AC4R
 - image_threshold_operations, 2410
- npptThreshold_GTVAl_16u_C1IR
 - image_threshold_operations, 2411
- npptThreshold_GTVAl_16u_C1R
 - image_threshold_operations, 2411
- npptThreshold_GTVAl_16u_C3IR
 - image_threshold_operations, 2412
- npptThreshold_GTVAl_16u_C3R
 - image_threshold_operations, 2412
- npptThreshold_GTVAl_32f_AC4IR
 - image_threshold_operations, 2412
- npptThreshold_GTVAl_32f_AC4R
 - image_threshold_operations, 2413
- npptThreshold_GTVAl_32f_C1IR
 - image_threshold_operations, 2413
- npptThreshold_GTVAl_32f_C1R
 - image_threshold_operations, 2414
- npptThreshold_GTVAl_32f_C3IR
 - image_threshold_operations, 2414
- npptThreshold_GTVAl_32f_C3R
 - image_threshold_operations, 2414

- image_threshold_operations, [2414](#)
- npptThreshold_GTVal_8u_AC4IR
 - image_threshold_operations, [2415](#)
- npptThreshold_GTVal_8u_AC4R
 - image_threshold_operations, [2415](#)
- npptThreshold_GTVal_8u_C1IR
 - image_threshold_operations, [2416](#)
- npptThreshold_GTVal_8u_C1R
 - image_threshold_operations, [2416](#)
- npptThreshold_GTVal_8u_C3IR
 - image_threshold_operations, [2417](#)
- npptThreshold_GTVal_8u_C3R
 - image_threshold_operations, [2417](#)
- npptThreshold_LT_16s_AC4IR
 - image_threshold_operations, [2417](#)
- npptThreshold_LT_16s_AC4R
 - image_threshold_operations, [2418](#)
- npptThreshold_LT_16s_C1IR
 - image_threshold_operations, [2418](#)
- npptThreshold_LT_16s_C1R
 - image_threshold_operations, [2419](#)
- npptThreshold_LT_16s_C3IR
 - image_threshold_operations, [2419](#)
- npptThreshold_LT_16s_C3R
 - image_threshold_operations, [2419](#)
- npptThreshold_LT_16u_AC4IR
 - image_threshold_operations, [2420](#)
- npptThreshold_LT_16u_AC4R
 - image_threshold_operations, [2420](#)
- npptThreshold_LT_16u_C1IR
 - image_threshold_operations, [2421](#)
- npptThreshold_LT_16u_C1R
 - image_threshold_operations, [2421](#)
- npptThreshold_LT_16u_C3IR
 - image_threshold_operations, [2421](#)
- npptThreshold_LT_16u_C3R
 - image_threshold_operations, [2422](#)
- npptThreshold_LT_32f_AC4IR
 - image_threshold_operations, [2422](#)
- npptThreshold_LT_32f_AC4R
 - image_threshold_operations, [2423](#)
- npptThreshold_LT_32f_C1IR
 - image_threshold_operations, [2423](#)
- npptThreshold_LT_32f_C1R
 - image_threshold_operations, [2423](#)
- npptThreshold_LT_32f_C3IR
 - image_threshold_operations, [2424](#)
- npptThreshold_LT_32f_C3R
 - image_threshold_operations, [2424](#)
- npptThreshold_LT_8u_AC4IR
 - image_threshold_operations, [2425](#)
- npptThreshold_LT_8u_AC4R
 - image_threshold_operations, [2425](#)
- npptThreshold_LT_8u_C1IR
 - image_threshold_operations, [2425](#)
- npptThreshold_LT_8u_C1R
 - image_threshold_operations, [2426](#)
- npptThreshold_LT_8u_C3IR
 - image_threshold_operations, [2426](#)
- npptThreshold_LT_8u_C3R
 - image_threshold_operations, [2427](#)
- npptThreshold_LTVal_16s_AC4IR
 - image_threshold_operations, [2427](#)
- npptThreshold_LTVal_16s_AC4R
 - image_threshold_operations, [2427](#)
- npptThreshold_LTVal_16s_C1IR
 - image_threshold_operations, [2428](#)
- npptThreshold_LTVal_16s_C1R
 - image_threshold_operations, [2428](#)
- npptThreshold_LTVal_16s_C3IR
 - image_threshold_operations, [2429](#)
- npptThreshold_LTVal_16s_C3R
 - image_threshold_operations, [2429](#)
- npptThreshold_LTVal_16u_AC4IR
 - image_threshold_operations, [2430](#)
- npptThreshold_LTVal_16u_AC4R
 - image_threshold_operations, [2430](#)
- npptThreshold_LTVal_16u_C1IR
 - image_threshold_operations, [2430](#)
- npptThreshold_LTVal_16u_C1R
 - image_threshold_operations, [2431](#)
- npptThreshold_LTVal_16u_C3IR
 - image_threshold_operations, [2431](#)
- npptThreshold_LTVal_16u_C3R
 - image_threshold_operations, [2432](#)
- npptThreshold_LTVal_32f_AC4IR
 - image_threshold_operations, [2432](#)
- npptThreshold_LTVal_32f_AC4R
 - image_threshold_operations, [2432](#)
- npptThreshold_LTVal_32f_C1IR
 - image_threshold_operations, [2433](#)
- npptThreshold_LTVal_32f_C1R
 - image_threshold_operations, [2433](#)
- npptThreshold_LTVal_32f_C3IR
 - image_threshold_operations, [2434](#)
- npptThreshold_LTVal_32f_C3R
 - image_threshold_operations, [2434](#)
- npptThreshold_LTVal_8u_AC4IR
 - image_threshold_operations, [2435](#)
- npptThreshold_LTVal_8u_AC4R
 - image_threshold_operations, [2435](#)
- npptThreshold_LTVal_8u_C1IR
 - image_threshold_operations, [2435](#)
- npptThreshold_LTVal_8u_C1R
 - image_threshold_operations, [2436](#)
- npptThreshold_LTVal_8u_C3IR
 - image_threshold_operations, [2436](#)
- npptThreshold_LTVal_8u_C3R
 - image_threshold_operations, [2436](#)

- image_threshold_operations, [2437](#)
- nppiThreshold_LTVaGTVal_16s_AC4IR
 - image_threshold_operations, [2437](#)
- nppiThreshold_LTVaGTVal_16s_AC4R
 - image_threshold_operations, [2438](#)
- nppiThreshold_LTVaGTVal_16s_C1IR
 - image_threshold_operations, [2438](#)
- nppiThreshold_LTVaGTVal_16s_C1R
 - image_threshold_operations, [2439](#)
- nppiThreshold_LTVaGTVal_16s_C3IR
 - image_threshold_operations, [2439](#)
- nppiThreshold_LTVaGTVal_16s_C3R
 - image_threshold_operations, [2440](#)
- nppiThreshold_LTVaGTVal_16u_AC4IR
 - image_threshold_operations, [2440](#)
- nppiThreshold_LTVaGTVal_16u_AC4R
 - image_threshold_operations, [2441](#)
- nppiThreshold_LTVaGTVal_16u_C1IR
 - image_threshold_operations, [2441](#)
- nppiThreshold_LTVaGTVal_16u_C1R
 - image_threshold_operations, [2442](#)
- nppiThreshold_LTVaGTVal_16u_C3IR
 - image_threshold_operations, [2442](#)
- nppiThreshold_LTVaGTVal_16u_C3R
 - image_threshold_operations, [2443](#)
- nppiThreshold_LTVaGTVal_32f_AC4IR
 - image_threshold_operations, [2443](#)
- nppiThreshold_LTVaGTVal_32f_AC4R
 - image_threshold_operations, [2444](#)
- nppiThreshold_LTVaGTVal_32f_C1IR
 - image_threshold_operations, [2444](#)
- nppiThreshold_LTVaGTVal_32f_C1R
 - image_threshold_operations, [2445](#)
- nppiThreshold_LTVaGTVal_32f_C3IR
 - image_threshold_operations, [2445](#)
- nppiThreshold_LTVaGTVal_32f_C3R
 - image_threshold_operations, [2446](#)
- nppiThreshold_LTVaGTVal_8u_AC4IR
 - image_threshold_operations, [2446](#)
- nppiThreshold_LTVaGTVal_8u_AC4R
 - image_threshold_operations, [2447](#)
- nppiThreshold_LTVaGTVal_8u_C1IR
 - image_threshold_operations, [2447](#)
- nppiThreshold_LTVaGTVal_8u_C1R
 - image_threshold_operations, [2448](#)
- nppiThreshold_LTVaGTVal_8u_C3IR
 - image_threshold_operations, [2448](#)
- nppiThreshold_LTVaGTVal_8u_C3R
 - image_threshold_operations, [2449](#)
- nppiThreshold_Val_16s_AC4IR
 - image_threshold_operations, [2449](#)
- nppiThreshold_Val_16s_AC4R
 - image_threshold_operations, [2450](#)
- nppiThreshold_Val_16s_C1IR
 - image_threshold_operations, [2450](#)
- nppiThreshold_Val_16s_C1R
 - image_threshold_operations, [2451](#)
- nppiThreshold_Val_16s_C3IR
 - image_threshold_operations, [2451](#)
- nppiThreshold_Val_16s_C3R
 - image_threshold_operations, [2452](#)
- nppiThreshold_Val_16u_AC4IR
 - image_threshold_operations, [2452](#)
- nppiThreshold_Val_16u_AC4R
 - image_threshold_operations, [2453](#)
- nppiThreshold_Val_16u_C1IR
 - image_threshold_operations, [2453](#)
- nppiThreshold_Val_16u_C1R
 - image_threshold_operations, [2454](#)
- nppiThreshold_Val_16u_C3IR
 - image_threshold_operations, [2454](#)
- nppiThreshold_Val_16u_C3R
 - image_threshold_operations, [2455](#)
- nppiThreshold_Val_32f_AC4IR
 - image_threshold_operations, [2455](#)
- nppiThreshold_Val_32f_AC4R
 - image_threshold_operations, [2456](#)
- nppiThreshold_Val_32f_C1IR
 - image_threshold_operations, [2456](#)
- nppiThreshold_Val_32f_C1R
 - image_threshold_operations, [2457](#)
- nppiThreshold_Val_32f_C3IR
 - image_threshold_operations, [2457](#)
- nppiThreshold_Val_32f_C3R
 - image_threshold_operations, [2458](#)
- nppiThreshold_Val_8u_AC4IR
 - image_threshold_operations, [2458](#)
- nppiThreshold_Val_8u_AC4R
 - image_threshold_operations, [2459](#)
- nppiThreshold_Val_8u_C1IR
 - image_threshold_operations, [2459](#)
- nppiThreshold_Val_8u_C1R
 - image_threshold_operations, [2460](#)
- nppiThreshold_Val_8u_C3IR
 - image_threshold_operations, [2460](#)
- nppiThreshold_Val_8u_C3R
 - image_threshold_operations, [2461](#)
- nppiTranspose_16s_C1R
 - image_transpose, [936](#)
- nppiTranspose_16s_C3R
 - image_transpose, [936](#)
- nppiTranspose_16s_C4R
 - image_transpose, [937](#)
- nppiTranspose_16u_C1R
 - image_transpose, [937](#)
- nppiTranspose_16u_C3R
 - image_transpose, [937](#)
- nppiTranspose_16u_C4R

- image_transpose, [938](#)
- npPiTranspose_32f_C1R
 - image_transpose, [938](#)
- npPiTranspose_32f_C3R
 - image_transpose, [938](#)
- npPiTranspose_32f_C4R
 - image_transpose, [939](#)
- npPiTranspose_32s_C1R
 - image_transpose, [939](#)
- npPiTranspose_32s_C3R
 - image_transpose, [939](#)
- npPiTranspose_32s_C4R
 - image_transpose, [940](#)
- npPiTranspose_8u_C1R
 - image_transpose, [940](#)
- npPiTranspose_8u_C3R
 - image_transpose, [940](#)
- npPiTranspose_8u_C4R
 - image_transpose, [941](#)
- npPiValidNormLevelGetBufferHostSize_16u32f_-AC4R
 - crosscorrvalidnormlevel, [2250](#)
- npPiValidNormLevelGetBufferHostSize_16u32f_-C1R
 - crosscorrvalidnormlevel, [2251](#)
- npPiValidNormLevelGetBufferHostSize_16u32f_-C3R
 - crosscorrvalidnormlevel, [2251](#)
- npPiValidNormLevelGetBufferHostSize_16u32f_-C4R
 - crosscorrvalidnormlevel, [2251](#)
- npPiValidNormLevelGetBufferHostSize_32f_-AC4R
 - crosscorrvalidnormlevel, [2252](#)
- npPiValidNormLevelGetBufferHostSize_32f_C1R
 - crosscorrvalidnormlevel, [2252](#)
- npPiValidNormLevelGetBufferHostSize_32f_C3R
 - crosscorrvalidnormlevel, [2252](#)
- npPiValidNormLevelGetBufferHostSize_32f_C4R
 - crosscorrvalidnormlevel, [2252](#)
- npPiValidNormLevelGetBufferHostSize_8s32f_-AC4R
 - crosscorrvalidnormlevel, [2253](#)
- npPiValidNormLevelGetBufferHostSize_8s32f_-C1R
 - crosscorrvalidnormlevel, [2253](#)
- npPiValidNormLevelGetBufferHostSize_8s32f_-C3R
 - crosscorrvalidnormlevel, [2253](#)
- npPiValidNormLevelGetBufferHostSize_8s32f_-C4R
 - crosscorrvalidnormlevel, [2254](#)
- npPiValidNormLevelGetBufferHostSize_8u32f_-AC4R
 - crosscorrvalidnormlevel, [2254](#)
- npPiValidNormLevelGetBufferHostSize_8u32f_-C1R
 - crosscorrvalidnormlevel, [2254](#)
- npPiValidNormLevelGetBufferHostSize_8u32f_-C3R
 - crosscorrvalidnormlevel, [2254](#)
- npPiValidNormLevelGetBufferHostSize_8u32f_-C4R
 - crosscorrvalidnormlevel, [2255](#)
- npPiValidNormLevelGetBufferHostSize_8u_-AC4RSfs
 - crosscorrvalidnormlevel, [2255](#)
- npPiValidNormLevelGetBufferHostSize_8u_-C1RSfs
 - crosscorrvalidnormlevel, [2255](#)
- npPiValidNormLevelGetBufferHostSize_8u_-C3RSfs
 - crosscorrvalidnormlevel, [2256](#)
- npPiValidNormLevelGetBufferHostSize_8u_-C4RSfs
 - crosscorrvalidnormlevel, [2256](#)
- npPiWarpAffine_16u_AC4R
 - image_affine_transform, [1489](#)
- npPiWarpAffine_16u_C1R
 - image_affine_transform, [1490](#)
- npPiWarpAffine_16u_C3R
 - image_affine_transform, [1490](#)
- npPiWarpAffine_16u_C4R
 - image_affine_transform, [1491](#)
- npPiWarpAffine_16u_P3R
 - image_affine_transform, [1491](#)
- npPiWarpAffine_16u_P4R
 - image_affine_transform, [1492](#)
- npPiWarpAffine_32f_AC4R
 - image_affine_transform, [1492](#)
- npPiWarpAffine_32f_C1R
 - image_affine_transform, [1493](#)
- npPiWarpAffine_32f_C3R
 - image_affine_transform, [1493](#)
- npPiWarpAffine_32f_C4R
 - image_affine_transform, [1494](#)
- npPiWarpAffine_32f_P3R
 - image_affine_transform, [1494](#)
- npPiWarpAffine_32f_P4R
 - image_affine_transform, [1495](#)
- npPiWarpAffine_32s_AC4R
 - image_affine_transform, [1495](#)
- npPiWarpAffine_32s_C1R
 - image_affine_transform, [1496](#)
- npPiWarpAffine_32s_C3R
 - image_affine_transform, [1496](#)
- npPiWarpAffine_32s_C4R
 - image_affine_transform, [1497](#)

- npapiWarpAffine_32s_P3R
 - image_affine_transform, [1497](#)
- npapiWarpAffine_32s_P4R
 - image_affine_transform, [1498](#)
- npapiWarpAffine_64f_AC4R
 - image_affine_transform, [1498](#)
- npapiWarpAffine_64f_C1R
 - image_affine_transform, [1499](#)
- npapiWarpAffine_64f_C3R
 - image_affine_transform, [1499](#)
- npapiWarpAffine_64f_C4R
 - image_affine_transform, [1500](#)
- npapiWarpAffine_64f_P3R
 - image_affine_transform, [1500](#)
- npapiWarpAffine_64f_P4R
 - image_affine_transform, [1501](#)
- npapiWarpAffine_8u_AC4R
 - image_affine_transform, [1501](#)
- npapiWarpAffine_8u_C1R
 - image_affine_transform, [1502](#)
- npapiWarpAffine_8u_C3R
 - image_affine_transform, [1502](#)
- npapiWarpAffine_8u_C4R
 - image_affine_transform, [1503](#)
- npapiWarpAffine_8u_P3R
 - image_affine_transform, [1503](#)
- npapiWarpAffine_8u_P4R
 - image_affine_transform, [1504](#)
- npapiWarpAffineBack_16u_AC4R
 - image_affine_transform, [1504](#)
- npapiWarpAffineBack_16u_C1R
 - image_affine_transform, [1505](#)
- npapiWarpAffineBack_16u_C3R
 - image_affine_transform, [1505](#)
- npapiWarpAffineBack_16u_C4R
 - image_affine_transform, [1506](#)
- npapiWarpAffineBack_16u_P3R
 - image_affine_transform, [1506](#)
- npapiWarpAffineBack_16u_P4R
 - image_affine_transform, [1507](#)
- npapiWarpAffineBack_32f_AC4R
 - image_affine_transform, [1507](#)
- npapiWarpAffineBack_32f_C1R
 - image_affine_transform, [1508](#)
- npapiWarpAffineBack_32f_C3R
 - image_affine_transform, [1508](#)
- npapiWarpAffineBack_32f_C4R
 - image_affine_transform, [1509](#)
- npapiWarpAffineBack_32f_P3R
 - image_affine_transform, [1509](#)
- npapiWarpAffineBack_32f_P4R
 - image_affine_transform, [1510](#)
- npapiWarpAffineBack_32s_AC4R
 - image_affine_transform, [1510](#)
- npapiWarpAffineBack_32s_C1R
 - image_affine_transform, [1511](#)
- npapiWarpAffineBack_32s_C3R
 - image_affine_transform, [1511](#)
- npapiWarpAffineBack_32s_C4R
 - image_affine_transform, [1512](#)
- npapiWarpAffineBack_32s_P3R
 - image_affine_transform, [1512](#)
- npapiWarpAffineBack_32s_P4R
 - image_affine_transform, [1513](#)
- npapiWarpAffineBack_8u_AC4R
 - image_affine_transform, [1513](#)
- npapiWarpAffineBack_8u_C1R
 - image_affine_transform, [1514](#)
- npapiWarpAffineBack_8u_C3R
 - image_affine_transform, [1514](#)
- npapiWarpAffineBack_8u_C4R
 - image_affine_transform, [1515](#)
- npapiWarpAffineBack_8u_P3R
 - image_affine_transform, [1515](#)
- npapiWarpAffineBack_8u_P4R
 - image_affine_transform, [1516](#)
- npapiWarpAffineQuad_16u_AC4R
 - image_affine_transform, [1516](#)
- npapiWarpAffineQuad_16u_C1R
 - image_affine_transform, [1517](#)
- npapiWarpAffineQuad_16u_C3R
 - image_affine_transform, [1517](#)
- npapiWarpAffineQuad_16u_C4R
 - image_affine_transform, [1518](#)
- npapiWarpAffineQuad_16u_P3R
 - image_affine_transform, [1518](#)
- npapiWarpAffineQuad_16u_P4R
 - image_affine_transform, [1519](#)
- npapiWarpAffineQuad_32f_AC4R
 - image_affine_transform, [1519](#)
- npapiWarpAffineQuad_32f_C1R
 - image_affine_transform, [1520](#)
- npapiWarpAffineQuad_32f_C3R
 - image_affine_transform, [1520](#)
- npapiWarpAffineQuad_32f_C4R
 - image_affine_transform, [1521](#)
- npapiWarpAffineQuad_32f_P3R
 - image_affine_transform, [1521](#)
- npapiWarpAffineQuad_32f_P4R
 - image_affine_transform, [1522](#)
- npapiWarpAffineQuad_32s_AC4R
 - image_affine_transform, [1522](#)
- npapiWarpAffineQuad_32s_C1R
 - image_affine_transform, [1523](#)
- npapiWarpAffineQuad_32s_C3R
 - image_affine_transform, [1523](#)
- npapiWarpAffineQuad_32s_C4R
 - image_affine_transform, [1524](#)

- nppiWarpAffineQuad_32s_P3R
 - image_affine_transform, [1524](#)
- nppiWarpAffineQuad_32s_P4R
 - image_affine_transform, [1525](#)
- nppiWarpAffineQuad_8u_AC4R
 - image_affine_transform, [1525](#)
- nppiWarpAffineQuad_8u_C1R
 - image_affine_transform, [1526](#)
- nppiWarpAffineQuad_8u_C3R
 - image_affine_transform, [1526](#)
- nppiWarpAffineQuad_8u_C4R
 - image_affine_transform, [1527](#)
- nppiWarpAffineQuad_8u_P3R
 - image_affine_transform, [1527](#)
- nppiWarpAffineQuad_8u_P4R
 - image_affine_transform, [1528](#)
- nppiWarpPerspective_16u_AC4R
 - image_perspective_transforms, [1538](#)
- nppiWarpPerspective_16u_C1R
 - image_perspective_transforms, [1539](#)
- nppiWarpPerspective_16u_C3R
 - image_perspective_transforms, [1539](#)
- nppiWarpPerspective_16u_C4R
 - image_perspective_transforms, [1540](#)
- nppiWarpPerspective_16u_P3R
 - image_perspective_transforms, [1540](#)
- nppiWarpPerspective_16u_P4R
 - image_perspective_transforms, [1541](#)
- nppiWarpPerspective_32f_AC4R
 - image_perspective_transforms, [1541](#)
- nppiWarpPerspective_32f_C1R
 - image_perspective_transforms, [1542](#)
- nppiWarpPerspective_32f_C3R
 - image_perspective_transforms, [1542](#)
- nppiWarpPerspective_32f_C4R
 - image_perspective_transforms, [1543](#)
- nppiWarpPerspective_32f_P3R
 - image_perspective_transforms, [1543](#)
- nppiWarpPerspective_32f_P4R
 - image_perspective_transforms, [1544](#)
- nppiWarpPerspective_32s_AC4R
 - image_perspective_transforms, [1544](#)
- nppiWarpPerspective_32s_C1R
 - image_perspective_transforms, [1545](#)
- nppiWarpPerspective_32s_C3R
 - image_perspective_transforms, [1545](#)
- nppiWarpPerspective_32s_C4R
 - image_perspective_transforms, [1546](#)
- nppiWarpPerspective_32s_P3R
 - image_perspective_transforms, [1546](#)
- nppiWarpPerspective_32s_P4R
 - image_perspective_transforms, [1546](#)
- nppiWarpPerspective_8u_AC4R
 - image_perspective_transforms, [1547](#)
- nppiWarpPerspective_8u_C1R
 - image_perspective_transforms, [1547](#)
- nppiWarpPerspective_8u_C3R
 - image_perspective_transforms, [1548](#)
- nppiWarpPerspective_8u_C4R
 - image_perspective_transforms, [1548](#)
- nppiWarpPerspective_8u_P3R
 - image_perspective_transforms, [1549](#)
- nppiWarpPerspective_8u_P4R
 - image_perspective_transforms, [1549](#)
- nppiWarpPerspectiveBack_16u_AC4R
 - image_perspective_transforms, [1550](#)
- nppiWarpPerspectiveBack_16u_C1R
 - image_perspective_transforms, [1550](#)
- nppiWarpPerspectiveBack_16u_C3R
 - image_perspective_transforms, [1551](#)
- nppiWarpPerspectiveBack_16u_C4R
 - image_perspective_transforms, [1551](#)
- nppiWarpPerspectiveBack_16u_P3R
 - image_perspective_transforms, [1552](#)
- nppiWarpPerspectiveBack_16u_P4R
 - image_perspective_transforms, [1552](#)
- nppiWarpPerspectiveBack_32f_AC4R
 - image_perspective_transforms, [1553](#)
- nppiWarpPerspectiveBack_32f_C1R
 - image_perspective_transforms, [1553](#)
- nppiWarpPerspectiveBack_32f_C3R
 - image_perspective_transforms, [1554](#)
- nppiWarpPerspectiveBack_32f_C4R
 - image_perspective_transforms, [1554](#)
- nppiWarpPerspectiveBack_32f_P3R
 - image_perspective_transforms, [1555](#)
- nppiWarpPerspectiveBack_32f_P4R
 - image_perspective_transforms, [1555](#)
- nppiWarpPerspectiveBack_32s_AC4R
 - image_perspective_transforms, [1556](#)
- nppiWarpPerspectiveBack_32s_C1R
 - image_perspective_transforms, [1556](#)
- nppiWarpPerspectiveBack_32s_C3R
 - image_perspective_transforms, [1557](#)
- nppiWarpPerspectiveBack_32s_C4R
 - image_perspective_transforms, [1557](#)
- nppiWarpPerspectiveBack_32s_P3R
 - image_perspective_transforms, [1558](#)
- nppiWarpPerspectiveBack_32s_P4R
 - image_perspective_transforms, [1558](#)
- nppiWarpPerspectiveBack_8u_AC4R
 - image_perspective_transforms, [1559](#)
- nppiWarpPerspectiveBack_8u_C1R
 - image_perspective_transforms, [1559](#)
- nppiWarpPerspectiveBack_8u_C3R
 - image_perspective_transforms, [1560](#)
- nppiWarpPerspectiveBack_8u_C4R
 - image_perspective_transforms, [1560](#)

- nppiWarpPerspectiveBack_8u_P3R
 - image_perspective_transforms, [1561](#)
- nppiWarpPerspectiveBack_8u_P4R
 - image_perspective_transforms, [1561](#)
- nppiWarpPerspectiveQuad_16u_AC4R
 - image_perspective_transforms, [1562](#)
- nppiWarpPerspectiveQuad_16u_C1R
 - image_perspective_transforms, [1562](#)
- nppiWarpPerspectiveQuad_16u_C3R
 - image_perspective_transforms, [1563](#)
- nppiWarpPerspectiveQuad_16u_C4R
 - image_perspective_transforms, [1563](#)
- nppiWarpPerspectiveQuad_16u_P3R
 - image_perspective_transforms, [1564](#)
- nppiWarpPerspectiveQuad_16u_P4R
 - image_perspective_transforms, [1564](#)
- nppiWarpPerspectiveQuad_32f_AC4R
 - image_perspective_transforms, [1565](#)
- nppiWarpPerspectiveQuad_32f_C1R
 - image_perspective_transforms, [1565](#)
- nppiWarpPerspectiveQuad_32f_C3R
 - image_perspective_transforms, [1566](#)
- nppiWarpPerspectiveQuad_32f_C4R
 - image_perspective_transforms, [1566](#)
- nppiWarpPerspectiveQuad_32f_P3R
 - image_perspective_transforms, [1567](#)
- nppiWarpPerspectiveQuad_32f_P4R
 - image_perspective_transforms, [1567](#)
- nppiWarpPerspectiveQuad_32s_AC4R
 - image_perspective_transforms, [1568](#)
- nppiWarpPerspectiveQuad_32s_C1R
 - image_perspective_transforms, [1568](#)
- nppiWarpPerspectiveQuad_32s_C3R
 - image_perspective_transforms, [1569](#)
- nppiWarpPerspectiveQuad_32s_C4R
 - image_perspective_transforms, [1569](#)
- nppiWarpPerspectiveQuad_32s_P3R
 - image_perspective_transforms, [1570](#)
- nppiWarpPerspectiveQuad_32s_P4R
 - image_perspective_transforms, [1570](#)
- nppiWarpPerspectiveQuad_8u_AC4R
 - image_perspective_transforms, [1571](#)
- nppiWarpPerspectiveQuad_8u_C1R
 - image_perspective_transforms, [1571](#)
- nppiWarpPerspectiveQuad_8u_C3R
 - image_perspective_transforms, [1572](#)
- nppiWarpPerspectiveQuad_8u_C4R
 - image_perspective_transforms, [1572](#)
- nppiWarpPerspectiveQuad_8u_P3R
 - image_perspective_transforms, [1573](#)
- nppiWarpPerspectiveQuad_8u_P4R
 - image_perspective_transforms, [1573](#)
- nppiXor_16u_AC4IR
 - image_xor, [459](#)
- nppiXor_16u_AC4R
 - image_xor, [459](#)
- nppiXor_16u_C1IR
 - image_xor, [459](#)
- nppiXor_16u_C1R
 - image_xor, [460](#)
- nppiXor_16u_C3IR
 - image_xor, [460](#)
- nppiXor_16u_C3R
 - image_xor, [460](#)
- nppiXor_16u_C4IR
 - image_xor, [461](#)
- nppiXor_16u_C4R
 - image_xor, [461](#)
- nppiXor_32s_AC4IR
 - image_xor, [462](#)
- nppiXor_32s_AC4R
 - image_xor, [462](#)
- nppiXor_32s_C1IR
 - image_xor, [462](#)
- nppiXor_32s_C1R
 - image_xor, [463](#)
- nppiXor_32s_C3IR
 - image_xor, [463](#)
- nppiXor_32s_C3R
 - image_xor, [463](#)
- nppiXor_32s_C4IR
 - image_xor, [464](#)
- nppiXor_32s_C4R
 - image_xor, [464](#)
- nppiXor_8u_AC4IR
 - image_xor, [465](#)
- nppiXor_8u_AC4R
 - image_xor, [465](#)
- nppiXor_8u_C1IR
 - image_xor, [465](#)
- nppiXor_8u_C1R
 - image_xor, [466](#)
- nppiXor_8u_C3IR
 - image_xor, [466](#)
- nppiXor_8u_C3R
 - image_xor, [466](#)
- nppiXor_8u_C4IR
 - image_xor, [467](#)
- nppiXor_8u_C4R
 - image_xor, [467](#)
- nppiXorC_16u_AC4IR
 - image_xorc, [396](#)
- nppiXorC_16u_AC4R
 - image_xorc, [396](#)
- nppiXorC_16u_C1IR
 - image_xorc, [396](#)
- nppiXorC_16u_C1R
 - image_xorc, [397](#)

- nppiXorC_16u_C3IR
 - image_xorc, [397](#)
- nppiXorC_16u_C3R
 - image_xorc, [397](#)
- nppiXorC_16u_C4IR
 - image_xorc, [398](#)
- nppiXorC_16u_C4R
 - image_xorc, [398](#)
- nppiXorC_32s_AC4IR
 - image_xorc, [398](#)
- nppiXorC_32s_AC4R
 - image_xorc, [399](#)
- nppiXorC_32s_C1IR
 - image_xorc, [399](#)
- nppiXorC_32s_C1R
 - image_xorc, [399](#)
- nppiXorC_32s_C3IR
 - image_xorc, [400](#)
- nppiXorC_32s_C3R
 - image_xorc, [400](#)
- nppiXorC_32s_C4IR
 - image_xorc, [400](#)
- nppiXorC_32s_C4R
 - image_xorc, [401](#)
- nppiXorC_8u_AC4IR
 - image_xorc, [401](#)
- nppiXorC_8u_AC4R
 - image_xorc, [401](#)
- nppiXorC_8u_C1IR
 - image_xorc, [402](#)
- nppiXorC_8u_C1R
 - image_xorc, [402](#)
- nppiXorC_8u_C3IR
 - image_xorc, [402](#)
- nppiXorC_8u_C3R
 - image_xorc, [403](#)
- nppiXorC_8u_C4IR
 - image_xorc, [403](#)
- nppiXorC_8u_C4R
 - image_xorc, [403](#)
- nppiXYZToRGB_8u_AC4R
 - image_color_model_conversion, [569](#)
- nppiXYZToRGB_8u_C3R
 - image_color_model_conversion, [569](#)
- nppiYCbCr411_8u_P2P3R
 - image_color_sampling_format_conversion, [596](#)
- nppiYCbCr411_8u_P3P2R
 - image_color_sampling_format_conversion, [596](#)
- nppiYCbCr411ToBGR_8u_P3C3R
 - image_color_model_conversion, [569](#)
- nppiYCbCr411ToBGR_8u_P3C4R
 - image_color_model_conversion, [570](#)
- nppiYCbCr411ToYCbCr420_8u_P2P3R
 - image_color_sampling_format_conversion, [596](#)
- nppiYCbCr411ToYCbCr420_8u_P3P2R
 - image_color_sampling_format_conversion, [597](#)
- nppiYCbCr411ToYCbCr420_8u_P3R
 - image_color_sampling_format_conversion, [597](#)
- nppiYCbCr411ToYCbCr422_8u_P2C2R
 - image_color_sampling_format_conversion, [598](#)
- nppiYCbCr411ToYCbCr422_8u_P2P3R
 - image_color_sampling_format_conversion, [598](#)
- nppiYCbCr411ToYCbCr422_8u_P3C2R
 - image_color_sampling_format_conversion, [598](#)
- nppiYCbCr411ToYCbCr422_8u_P3R
 - image_color_sampling_format_conversion, [599](#)
- nppiYCbCr411ToYCrCb420_8u_P2P3R
 - image_color_sampling_format_conversion, [599](#)
- nppiYCbCr411ToYCrCb422_8u_P3C2R
 - image_color_sampling_format_conversion, [600](#)
- nppiYCbCr411ToYCrCb422_8u_P3R
 - image_color_sampling_format_conversion, [600](#)
- nppiYCbCr420_8u_P2P3R
 - image_color_sampling_format_conversion, [600](#)
- nppiYCbCr420_8u_P3P2R
 - image_color_sampling_format_conversion, [601](#)
- nppiYCbCr420ToBGR_709CSC_8u_P3C3R
 - image_color_model_conversion, [570](#)
- nppiYCbCr420ToBGR_709HDTV_8u_P3C4R
 - image_color_model_conversion, [570](#)
- nppiYCbCr420ToBGR_8u_P3C3R
 - image_color_model_conversion, [571](#)
- nppiYCbCr420ToBGR_8u_P3C4R
 - image_color_model_conversion, [571](#)
- nppiYCbCr420ToCbYCr422_8u_P2C2R
 - image_color_sampling_format_conversion, [601](#)
- nppiYCbCr420ToRGB_8u_P3C3R
 - image_color_model_conversion, [572](#)
- nppiYCbCr420ToYCbCr411_8u_P2P3R
 - image_color_sampling_format_conversion, [602](#)
- nppiYCbCr420ToYCbCr411_8u_P3P2R

- image_color_sampling_format_conversion, 602
- nppiYCbCr420ToYCbCr422_8u_P2C2R
 - image_color_sampling_format_conversion, 603
- nppiYCbCr420ToYCbCr422_8u_P2P3R
 - image_color_sampling_format_conversion, 603
- nppiYCbCr420ToYCbCr422_8u_P3R
 - image_color_sampling_format_conversion, 603
- nppiYCbCr420ToYCrCb420_8u_P2P3R
 - image_color_sampling_format_conversion, 604
- nppiYCbCr422_8u_C2P3R
 - image_color_sampling_format_conversion, 604
- nppiYCbCr422_8u_P3C2R
 - image_color_sampling_format_conversion, 605
- nppiYCbCr422ToBGR_8u_C2C3R
 - image_color_model_conversion, 572
- nppiYCbCr422ToBGR_8u_C2C4R
 - image_color_model_conversion, 572
- nppiYCbCr422ToBGR_8u_P3C3R
 - image_color_model_conversion, 573
- nppiYCbCr422ToCbYCr422_8u_C2R
 - image_color_sampling_format_conversion, 605
- nppiYCbCr422ToRGB_8u_C2C3R
 - image_color_model_conversion, 573
- nppiYCbCr422ToRGB_8u_C2P3R
 - image_color_model_conversion, 573
- nppiYCbCr422ToRGB_8u_P3C3R
 - image_color_model_conversion, 574
- nppiYCbCr422ToYCbCr411_8u_C2P2R
 - image_color_sampling_format_conversion, 605
- nppiYCbCr422ToYCbCr411_8u_C2P3R
 - image_color_sampling_format_conversion, 606
- nppiYCbCr422ToYCbCr411_8u_P3P2R
 - image_color_sampling_format_conversion, 606
- nppiYCbCr422ToYCbCr411_8u_P3R
 - image_color_sampling_format_conversion, 607
- nppiYCbCr422ToYCbCr420_8u_C2P2R
 - image_color_sampling_format_conversion, 607
- nppiYCbCr422ToYCbCr420_8u_C2P3R
 - image_color_sampling_format_conversion, 608
- nppiYCbCr422ToYCbCr420_8u_P3P2R
 - image_color_sampling_format_conversion, 608
- nppiYCbCr422ToYCrCb420_8u_P3R
 - image_color_sampling_format_conversion, 608
- nppiYCbCr422ToYCrCb420_8u_C2P3R
 - image_color_sampling_format_conversion, 609
- nppiYCbCr422ToYCrCb422_8u_C2R
 - image_color_sampling_format_conversion, 609
- nppiYCbCr422ToYCrCb422_8u_P3C2R
 - image_color_sampling_format_conversion, 610
- nppiYCbCrToBGR_709CSC_8u_P3C3R
 - image_color_model_conversion, 574
- nppiYCbCrToBGR_709CSC_8u_P3C4R
 - image_color_model_conversion, 574
- nppiYCbCrToBGR_8u_P3C3R
 - image_color_model_conversion, 575
- nppiYCbCrToBGR_8u_P3C4R
 - image_color_model_conversion, 575
- nppiYCbCrToRGB_8u_AC4R
 - image_color_model_conversion, 576
- nppiYCbCrToRGB_8u_C3R
 - image_color_model_conversion, 576
- nppiYCbCrToRGB_8u_P3C3R
 - image_color_model_conversion, 576
- nppiYCbCrToRGB_8u_P3C4R
 - image_color_model_conversion, 577
- nppiYCbCrToRGB_8u_P3R
 - image_color_model_conversion, 577
- nppiYCCToRGB_8u_AC4R
 - image_color_model_conversion, 577
- nppiYCCToRGB_8u_C3R
 - image_color_model_conversion, 578
- nppiYCrCb420ToCbYCr422_8u_P3C2R
 - image_color_sampling_format_conversion, 610
- nppiYCrCb420ToRGB_8u_P3C4R
 - image_color_model_conversion, 578
- nppiYCrCb420ToYCbCr411_8u_P3P2R
 - image_color_sampling_format_conversion, 610
- nppiYCrCb420ToYCbCr420_8u_P3P2R
 - image_color_sampling_format_conversion, 611
- nppiYCrCb420ToYCbCr422_8u_P3C2R
 - image_color_sampling_format_conversion, 611
- nppiYCrCb420ToYCbCr422_8u_P3R
 - image_color_sampling_format_conversion, 612
- nppiYCrCb422ToRGB_8u_C2C3R

- image_color_model_conversion, 578
- nppiYCrCb422ToRGB_8u_C2P3R
 - image_color_model_conversion, 579
- nppiYCrCb422ToYCbCr411_8u_C2P3R
 - image_color_sampling_format_conversion, 612
- nppiYCrCb422ToYCbCr420_8u_C2P3R
 - image_color_sampling_format_conversion, 613
- nppiYCrCb422ToYCbCr422_8u_C2P3R
 - image_color_sampling_format_conversion, 613
- nppiYUV420ToBGR_8u_P3C3R
 - image_color_model_conversion, 579
- nppiYUV420ToBGR_8u_P3C4R
 - image_color_model_conversion, 579
- nppiYUV420ToRGB_8u_P3AC4R
 - image_color_model_conversion, 580
- nppiYUV420ToRGB_8u_P3C3R
 - image_color_model_conversion, 580
- nppiYUV420ToRGB_8u_P3C4R
 - image_color_model_conversion, 580
- nppiYUV420ToRGB_8u_P3R
 - image_color_model_conversion, 581
- nppiYUV422ToRGB_8u_C2C3R
 - image_color_model_conversion, 581
- nppiYUV422ToRGB_8u_P3AC4R
 - image_color_model_conversion, 581
- nppiYUV422ToRGB_8u_P3C3R
 - image_color_model_conversion, 582
- nppiYUV422ToRGB_8u_P3R
 - image_color_model_conversion, 582
- nppiYUVToBGR_8u_AC4R
 - image_color_model_conversion, 582
- nppiYUVToBGR_8u_C3R
 - image_color_model_conversion, 583
- nppiYUVToBGR_8u_P3C3R
 - image_color_model_conversion, 583
- nppiYUVToBGR_8u_P3R
 - image_color_model_conversion, 583
- nppiYUVToRGB_8u_AC4R
 - image_color_model_conversion, 584
- nppiYUVToRGB_8u_C3R
 - image_color_model_conversion, 584
- nppiYUVToRGB_8u_P3C3R
 - image_color_model_conversion, 584
- nppiYUVToRGB_8u_P3R
 - image_color_model_conversion, 585
- NppLibraryVersion, 2876
 - build, 2876
 - major, 2876
 - minor, 2876
- NppRoundMode
 - typedefs_npp, 44
- npps10Log10_32s_ISfs
 - signal_10log10, 2612
- npps10Log10_32s_Sfs
 - signal_10log10, 2612
- nppsAbs_16s
 - signal_abs, 2586
- nppsAbs_16s_I
 - signal_abs, 2586
- nppsAbs_32f
 - signal_abs, 2587
- nppsAbs_32f_I
 - signal_abs, 2587
- nppsAbs_32s
 - signal_abs, 2587
- nppsAbs_32s_I
 - signal_abs, 2587
- nppsAbs_64f
 - signal_abs, 2588
- nppsAbs_64f_I
 - signal_abs, 2588
- nppsAdd_16s
 - signal_add, 2538
- nppsAdd_16s32f
 - signal_add, 2538
- nppsAdd_16s32s_I
 - signal_add, 2538
- nppsAdd_16s_I
 - signal_add, 2539
- nppsAdd_16s_ISfs
 - signal_add, 2539
- nppsAdd_16s_Sfs
 - signal_add, 2539
- nppsAdd_16sc_ISfs
 - signal_add, 2540
- nppsAdd_16sc_Sfs
 - signal_add, 2540
- nppsAdd_16u
 - signal_add, 2540
- nppsAdd_16u_ISfs
 - signal_add, 2541
- nppsAdd_16u_Sfs
 - signal_add, 2541
- nppsAdd_32f
 - signal_add, 2541
- nppsAdd_32f_I
 - signal_add, 2542
- nppsAdd_32fc
 - signal_add, 2542
- nppsAdd_32fc_I
 - signal_add, 2542
- nppsAdd_32s_ISfs
 - signal_add, 2543
- nppsAdd_32s_Sfs
 - signal_add, 2543

- npopsAdd_32sc_ISfs
 - signal_add, [2543](#)
- npopsAdd_32sc_Sfs
 - signal_add, [2544](#)
- npopsAdd_32u
 - signal_add, [2544](#)
- npopsAdd_64f
 - signal_add, [2544](#)
- npopsAdd_64f_I
 - signal_add, [2545](#)
- npopsAdd_64fc
 - signal_add, [2545](#)
- npopsAdd_64fc_I
 - signal_add, [2545](#)
- npopsAdd_64s_Sfs
 - signal_add, [2546](#)
- npopsAdd_8u16u
 - signal_add, [2546](#)
- npopsAdd_8u_ISfs
 - signal_add, [2546](#)
- npopsAdd_8u_Sfs
 - signal_add, [2547](#)
- npopsAddC_16s_ISfs
 - signal_addc, [2490](#)
- npopsAddC_16s_Sfs
 - signal_addc, [2490](#)
- npopsAddC_16sc_ISfs
 - signal_addc, [2491](#)
- npopsAddC_16sc_Sfs
 - signal_addc, [2491](#)
- npopsAddC_16u_ISfs
 - signal_addc, [2491](#)
- npopsAddC_16u_Sfs
 - signal_addc, [2492](#)
- npopsAddC_32f
 - signal_addc, [2492](#)
- npopsAddC_32f_I
 - signal_addc, [2492](#)
- npopsAddC_32fc
 - signal_addc, [2493](#)
- npopsAddC_32fc_I
 - signal_addc, [2493](#)
- npopsAddC_32s_ISfs
 - signal_addc, [2493](#)
- npopsAddC_32s_Sfs
 - signal_addc, [2494](#)
- npopsAddC_32sc_ISfs
 - signal_addc, [2494](#)
- npopsAddC_32sc_Sfs
 - signal_addc, [2494](#)
- npopsAddC_64f
 - signal_addc, [2495](#)
- npopsAddC_64f_I
 - signal_addc, [2495](#)
- npopsAddC_64fc
 - signal_addc, [2495](#)
- npopsAddC_64fc_I
 - signal_addc, [2496](#)
- npopsAddC_8u_ISfs
 - signal_addc, [2496](#)
- npopsAddC_8u_Sfs
 - signal_addc, [2496](#)
- npopsAddProduct_16s32s_Sfs
 - signal_addproduct, [2549](#)
- npopsAddProduct_16s_Sfs
 - signal_addproduct, [2549](#)
- npopsAddProduct_32f
 - signal_addproduct, [2549](#)
- npopsAddProduct_32fc
 - signal_addproduct, [2550](#)
- npopsAddProduct_32s_Sfs
 - signal_addproduct, [2550](#)
- npopsAddProduct_64f
 - signal_addproduct, [2550](#)
- npopsAddProduct_64fc
 - signal_addproduct, [2551](#)
- npopsAddProductC_32f
 - signal_addproductc, [2498](#)
- npopsAnd_16u
 - signal_and, [2628](#)
- npopsAnd_16u_I
 - signal_and, [2628](#)
- npopsAnd_32u
 - signal_and, [2629](#)
- npopsAnd_32u_I
 - signal_and, [2629](#)
- npopsAnd_8u
 - signal_and, [2629](#)
- npopsAnd_8u_I
 - signal_and, [2630](#)
- npopsAndC_16u
 - signal_andc, [2625](#)
- npopsAndC_16u_I
 - signal_andc, [2625](#)
- npopsAndC_32u
 - signal_andc, [2626](#)
- npopsAndC_32u_I
 - signal_andc, [2626](#)
- npopsAndC_8u
 - signal_andc, [2626](#)
- npopsAndC_8u_I
 - signal_andc, [2627](#)
- npopsArctan_32f
 - signal_inversetan, [2617](#)
- npopsArctan_32f_I
 - signal_inversetan, [2617](#)
- npopsArctan_64f
 - signal_inversetan, [2617](#)

- npPsArctan_64f_I
 - signal_inversetan, [2618](#)
- npPsAverageError_16s
 - signal_average_error, [2826](#)
- npPsAverageError_16sc
 - signal_average_error, [2826](#)
- npPsAverageError_16u
 - signal_average_error, [2826](#)
- npPsAverageError_32f
 - signal_average_error, [2827](#)
- npPsAverageError_32fc
 - signal_average_error, [2827](#)
- npPsAverageError_32s
 - signal_average_error, [2827](#)
- npPsAverageError_32sc
 - signal_average_error, [2828](#)
- npPsAverageError_32u
 - signal_average_error, [2828](#)
- npPsAverageError_64f
 - signal_average_error, [2828](#)
- npPsAverageError_64fc
 - signal_average_error, [2829](#)
- npPsAverageError_64s
 - signal_average_error, [2829](#)
- npPsAverageError_64sc
 - signal_average_error, [2829](#)
- npPsAverageError_8s
 - signal_average_error, [2830](#)
- npPsAverageError_8u
 - signal_average_error, [2830](#)
- npPsAverageErrorGetBufferSize_16s
 - signal_average_error, [2830](#)
- npPsAverageErrorGetBufferSize_16sc
 - signal_average_error, [2831](#)
- npPsAverageErrorGetBufferSize_16u
 - signal_average_error, [2831](#)
- npPsAverageErrorGetBufferSize_32f
 - signal_average_error, [2831](#)
- npPsAverageErrorGetBufferSize_32fc
 - signal_average_error, [2831](#)
- npPsAverageErrorGetBufferSize_32s
 - signal_average_error, [2832](#)
- npPsAverageErrorGetBufferSize_32sc
 - signal_average_error, [2832](#)
- npPsAverageErrorGetBufferSize_32u
 - signal_average_error, [2832](#)
- npPsAverageErrorGetBufferSize_64f
 - signal_average_error, [2832](#)
- npPsAverageErrorGetBufferSize_64fc
 - signal_average_error, [2833](#)
- npPsAverageErrorGetBufferSize_64s
 - signal_average_error, [2833](#)
- npPsAverageErrorGetBufferSize_64sc
 - signal_average_error, [2833](#)
- npPsAverageErrorGetBufferSize_8s
 - signal_average_error, [2833](#)
- npPsAverageErrorGetBufferSize_8u
 - signal_average_error, [2834](#)
- npPsAverageRelativeError_16s
 - signal_average_relative_error, [2849](#)
- npPsAverageRelativeError_16sc
 - signal_average_relative_error, [2849](#)
- npPsAverageRelativeError_16u
 - signal_average_relative_error, [2850](#)
- npPsAverageRelativeError_32f
 - signal_average_relative_error, [2850](#)
- npPsAverageRelativeError_32fc
 - signal_average_relative_error, [2850](#)
- npPsAverageRelativeError_32s
 - signal_average_relative_error, [2851](#)
- npPsAverageRelativeError_32sc
 - signal_average_relative_error, [2851](#)
- npPsAverageRelativeError_32u
 - signal_average_relative_error, [2852](#)
- npPsAverageRelativeError_64f
 - signal_average_relative_error, [2852](#)
- npPsAverageRelativeError_64fc
 - signal_average_relative_error, [2852](#)
- npPsAverageRelativeError_64s
 - signal_average_relative_error, [2853](#)
- npPsAverageRelativeError_64sc
 - signal_average_relative_error, [2853](#)
- npPsAverageRelativeError_8s
 - signal_average_relative_error, [2854](#)
- npPsAverageRelativeError_8u
 - signal_average_relative_error, [2854](#)
- npPsAverageRelativeErrorGetBufferSize_16s
 - signal_average_relative_error, [2854](#)
- npPsAverageRelativeErrorGetBufferSize_16sc
 - signal_average_relative_error, [2855](#)
- npPsAverageRelativeErrorGetBufferSize_16u
 - signal_average_relative_error, [2855](#)
- npPsAverageRelativeErrorGetBufferSize_32f
 - signal_average_relative_error, [2855](#)
- npPsAverageRelativeErrorGetBufferSize_32fc
 - signal_average_relative_error, [2855](#)
- npPsAverageRelativeErrorGetBufferSize_32s
 - signal_average_relative_error, [2856](#)
- npPsAverageRelativeErrorGetBufferSize_32sc
 - signal_average_relative_error, [2856](#)
- npPsAverageRelativeErrorGetBufferSize_32u
 - signal_average_relative_error, [2856](#)
- npPsAverageRelativeErrorGetBufferSize_64f
 - signal_average_relative_error, [2856](#)
- npPsAverageRelativeErrorGetBufferSize_64fc
 - signal_average_relative_error, [2857](#)
- npPsAverageRelativeErrorGetBufferSize_64s
 - signal_average_relative_error, [2857](#)

- nppsAverageRelativeErrorGetBufferSize_64sc
 - signal_average_relative_error, [2857](#)
- nppsAverageRelativeErrorGetBufferSize_8s
 - signal_average_relative_error, [2857](#)
- nppsAverageRelativeErrorGetBufferSize_8u
 - signal_average_relative_error, [2858](#)
- nppsCauchy_32f_I
 - signal_cauchy, [2622](#)
- nppsCauchyD_32f_I
 - signal_cauchy, [2622](#)
- nppsCauchyDD2_32f_I
 - signal_cauchy, [2622](#)
- nppsConvert_16s32f
 - signal_convert, [2657](#)
- nppsConvert_16s32f_Sfs
 - signal_convert, [2657](#)
- nppsConvert_16s32s
 - signal_convert, [2657](#)
- nppsConvert_16s64f_Sfs
 - signal_convert, [2657](#)
- nppsConvert_16s8s_Sfs
 - signal_convert, [2657](#)
- nppsConvert_16u32f
 - signal_convert, [2657](#)
- nppsConvert_32f16s_Sfs
 - signal_convert, [2657](#)
- nppsConvert_32f16u_Sfs
 - signal_convert, [2657](#)
- nppsConvert_32f32s_Sfs
 - signal_convert, [2657](#)
- nppsConvert_32f64f
 - signal_convert, [2657](#)
- nppsConvert_32f8s_Sfs
 - signal_convert, [2657](#)
- nppsConvert_32f8u_Sfs
 - signal_convert, [2657](#)
- nppsConvert_32s16s
 - signal_convert, [2657](#)
- nppsConvert_32s16s_Sfs
 - signal_convert, [2657](#)
- nppsConvert_32s32f
 - signal_convert, [2657](#)
- nppsConvert_32s32f_Sfs
 - signal_convert, [2657](#)
- nppsConvert_32s64f
 - signal_convert, [2657](#)
- nppsConvert_32s64f_Sfs
 - signal_convert, [2657](#)
- nppsConvert_64f16s_Sfs
 - signal_convert, [2657](#)
- nppsConvert_64f32f
 - signal_convert, [2657](#)
- nppsConvert_64f32s_Sfs
 - signal_convert, [2657](#)
- nppsConvert_64f64s_Sfs
 - signal_convert, [2657](#)
- nppsConvert_64s32s_Sfs
 - signal_convert, [2657](#)
- nppsConvert_64s64f
 - signal_convert, [2657](#)
- nppsConvert_8s16s
 - signal_convert, [2657](#)
- nppsConvert_8s32f
 - signal_convert, [2657](#)
- nppsConvert_8u32f
 - signal_convert, [2657](#)
- nppsCopy_16s
 - signal_copy, [2695](#)
- nppsCopy_16sc
 - signal_copy, [2696](#)
- nppsCopy_32f
 - signal_copy, [2696](#)
- nppsCopy_32fc
 - signal_copy, [2696](#)
- nppsCopy_32s
 - signal_copy, [2696](#)
- nppsCopy_32sc
 - signal_copy, [2697](#)
- nppsCopy_64fc
 - signal_copy, [2697](#)
- nppsCopy_64s
 - signal_copy, [2697](#)
- nppsCopy_64sc
 - signal_copy, [2698](#)
- nppsCopy_8u
 - signal_copy, [2698](#)
- nppsCountInRange_32s
 - signal_count_in_range, [2810](#)
- nppsCountInRangeGetBufferSize_32s
 - signal_count_in_range, [2810](#)
- nppsCubrt_32f
 - signal_cuberoot, [2603](#)
- nppsCubrt_32s16s_Sfs
 - signal_cuberoot, [2603](#)
- nppsDiv_16s_ISfs
 - signal_div, [2576](#)
- nppsDiv_16s_Sfs
 - signal_div, [2576](#)
- nppsDiv_16sc_ISfs
 - signal_div, [2577](#)
- nppsDiv_16sc_Sfs
 - signal_div, [2577](#)
- nppsDiv_16u_ISfs
 - signal_div, [2577](#)
- nppsDiv_16u_Sfs
 - signal_div, [2578](#)
- nppsDiv_32f
 - signal_div, [2578](#)

- npopsDiv_32f_I
 - signal_div, [2578](#)
- npopsDiv_32fc
 - signal_div, [2579](#)
- npopsDiv_32fc_I
 - signal_div, [2579](#)
- npopsDiv_32s16s_Sfs
 - signal_div, [2579](#)
- npopsDiv_32s_ISfs
 - signal_div, [2580](#)
- npopsDiv_32s_Sfs
 - signal_div, [2580](#)
- npopsDiv_64f
 - signal_div, [2580](#)
- npopsDiv_64f_I
 - signal_div, [2581](#)
- npopsDiv_64fc
 - signal_div, [2581](#)
- npopsDiv_64fc_I
 - signal_div, [2581](#)
- npopsDiv_8u_ISfs
 - signal_div, [2582](#)
- npopsDiv_8u_Sfs
 - signal_div, [2582](#)
- npopsDiv_Round_16s_ISfs
 - signal_divround, [2583](#)
- npopsDiv_Round_16s_Sfs
 - signal_divround, [2584](#)
- npopsDiv_Round_16u_ISfs
 - signal_divround, [2584](#)
- npopsDiv_Round_16u_Sfs
 - signal_divround, [2584](#)
- npopsDiv_Round_8u_ISfs
 - signal_divround, [2585](#)
- npopsDiv_Round_8u_Sfs
 - signal_divround, [2585](#)
- npopsDivC_16s_ISfs
 - signal_divc, [2528](#)
- npopsDivC_16s_Sfs
 - signal_divc, [2528](#)
- npopsDivC_16sc_ISfs
 - signal_divc, [2528](#)
- npopsDivC_16sc_Sfs
 - signal_divc, [2529](#)
- npopsDivC_16u_ISfs
 - signal_divc, [2529](#)
- npopsDivC_16u_Sfs
 - signal_divc, [2529](#)
- npopsDivC_32f
 - signal_divc, [2530](#)
- npopsDivC_32f_I
 - signal_divc, [2530](#)
- npopsDivC_32fc
 - signal_divc, [2530](#)
- npopsDivC_32fc_I
 - signal_divc, [2531](#)
- npopsDivC_64f
 - signal_divc, [2531](#)
- npopsDivC_64f_I
 - signal_divc, [2531](#)
- npopsDivC_64fc
 - signal_divc, [2532](#)
- npopsDivC_64fc_I
 - signal_divc, [2532](#)
- npopsDivC_8u_ISfs
 - signal_divc, [2532](#)
- npopsDivC_8u_Sfs
 - signal_divc, [2533](#)
- npopsDivCRev_16u
 - signal_divcrev, [2534](#)
- npopsDivCRev_16u_I
 - signal_divcrev, [2534](#)
- npopsDivCRev_32f
 - signal_divcrev, [2535](#)
- npopsDivCRev_32f_I
 - signal_divcrev, [2535](#)
- npopsDotProd_16s16sc32fc
 - signal_dot_product, [2793](#)
- npopsDotProd_16s16sc32sc_Sfs
 - signal_dot_product, [2794](#)
- npopsDotProd_16s16sc64sc
 - signal_dot_product, [2794](#)
- npopsDotProd_16s16sc_Sfs
 - signal_dot_product, [2794](#)
- npopsDotProd_16s32f
 - signal_dot_product, [2795](#)
- npopsDotProd_16s32s32s_Sfs
 - signal_dot_product, [2795](#)
- npopsDotProd_16s32s_Sfs
 - signal_dot_product, [2796](#)
- npopsDotProd_16s64s
 - signal_dot_product, [2796](#)
- npopsDotProd_16s_Sfs
 - signal_dot_product, [2796](#)
- npopsDotProd_16sc32fc
 - signal_dot_product, [2797](#)
- npopsDotProd_16sc32sc_Sfs
 - signal_dot_product, [2797](#)
- npopsDotProd_16sc64sc
 - signal_dot_product, [2798](#)
- npopsDotProd_16sc_Sfs
 - signal_dot_product, [2798](#)
- npopsDotProd_32f
 - signal_dot_product, [2798](#)
- npopsDotProd_32f32fc
 - signal_dot_product, [2799](#)
- npopsDotProd_32f32fc64fc
 - signal_dot_product, [2799](#)

- nppsDotProd_32f64f
 - signal_dot_product, [2799](#)
- nppsDotProd_32fc
 - signal_dot_product, [2800](#)
- nppsDotProd_32fc64fc
 - signal_dot_product, [2800](#)
- nppsDotProd_32s32sc_Sfs
 - signal_dot_product, [2800](#)
- nppsDotProd_32s_Sfs
 - signal_dot_product, [2801](#)
- nppsDotProd_32sc_Sfs
 - signal_dot_product, [2801](#)
- nppsDotProd_64f
 - signal_dot_product, [2802](#)
- nppsDotProd_64f64fc
 - signal_dot_product, [2802](#)
- nppsDotProd_64fc
 - signal_dot_product, [2802](#)
- nppsDotProdGetBufferSize_16s16sc32fc
 - signal_dot_product, [2803](#)
- nppsDotProdGetBufferSize_16s16sc32sc_Sfs
 - signal_dot_product, [2803](#)
- nppsDotProdGetBufferSize_16s16sc64sc
 - signal_dot_product, [2803](#)
- nppsDotProdGetBufferSize_16s16sc_Sfs
 - signal_dot_product, [2804](#)
- nppsDotProdGetBufferSize_16s32f
 - signal_dot_product, [2804](#)
- nppsDotProdGetBufferSize_16s32s32s_Sfs
 - signal_dot_product, [2804](#)
- nppsDotProdGetBufferSize_16s32s_Sfs
 - signal_dot_product, [2804](#)
- nppsDotProdGetBufferSize_16s64s
 - signal_dot_product, [2805](#)
- nppsDotProdGetBufferSize_16s_Sfs
 - signal_dot_product, [2805](#)
- nppsDotProdGetBufferSize_16sc32fc
 - signal_dot_product, [2805](#)
- nppsDotProdGetBufferSize_16sc32sc_Sfs
 - signal_dot_product, [2805](#)
- nppsDotProdGetBufferSize_16sc64sc
 - signal_dot_product, [2806](#)
- nppsDotProdGetBufferSize_16sc_Sfs
 - signal_dot_product, [2806](#)
- nppsDotProdGetBufferSize_32f
 - signal_dot_product, [2806](#)
- nppsDotProdGetBufferSize_32f32fc
 - signal_dot_product, [2806](#)
- nppsDotProdGetBufferSize_32f32fc64fc
 - signal_dot_product, [2807](#)
- nppsDotProdGetBufferSize_32f64f
 - signal_dot_product, [2807](#)
- nppsDotProdGetBufferSize_32fc
 - signal_dot_product, [2807](#)
- nppsDotProdGetBufferSize_32fc64fc
 - signal_dot_product, [2807](#)
- nppsDotProdGetBufferSize_32s32sc_Sfs
 - signal_dot_product, [2808](#)
- nppsDotProdGetBufferSize_32s_Sfs
 - signal_dot_product, [2808](#)
- nppsDotProdGetBufferSize_32sc_Sfs
 - signal_dot_product, [2808](#)
- nppsDotProdGetBufferSize_64f
 - signal_dot_product, [2808](#)
- nppsDotProdGetBufferSize_64f64fc
 - signal_dot_product, [2809](#)
- nppsDotProdGetBufferSize_64fc
 - signal_dot_product, [2809](#)
- nppSetStream
 - core_npp, [33](#)
- nppsExp_16s_ISfs
 - signal_exp, [2604](#)
- nppsExp_16s_Sfs
 - signal_exp, [2605](#)
- nppsExp_32f
 - signal_exp, [2605](#)
- nppsExp_32f64f
 - signal_exp, [2605](#)
- nppsExp_32f_I
 - signal_exp, [2605](#)
- nppsExp_32s_ISfs
 - signal_exp, [2606](#)
- nppsExp_32s_Sfs
 - signal_exp, [2606](#)
- nppsExp_64f
 - signal_exp, [2606](#)
- nppsExp_64f_I
 - signal_exp, [2607](#)
- nppsExp_64s_ISfs
 - signal_exp, [2607](#)
- nppsExp_64s_Sfs
 - signal_exp, [2607](#)
- nppsFree
 - signal_free, [2865](#)
- nppsIntegral_32s
 - signal_integral, [2684](#)
- nppsIntegralGetBufferSize_32s
 - signal_integral, [2684](#)
- nppsLn_16s_ISfs
 - signal_ln, [2608](#)
- nppsLn_16s_Sfs
 - signal_ln, [2609](#)
- nppsLn_32f
 - signal_ln, [2609](#)
- nppsLn_32f_I
 - signal_ln, [2609](#)
- nppsLn_32s16s_Sfs
 - signal_ln, [2609](#)

- nppsLn_32s_ISfs
 - signal_ln, [2610](#)
- nppsLn_32s_Sfs
 - signal_ln, [2610](#)
- nppsLn_64f
 - signal_ln, [2610](#)
- nppsLn_64f32f
 - signal_ln, [2611](#)
- nppsLn_64f_I
 - signal_ln, [2611](#)
- nppsLShiftC_16s
 - signal_lshiftc, [2646](#)
- nppsLShiftC_16s_I
 - signal_lshiftc, [2647](#)
- nppsLShiftC_16u
 - signal_lshiftc, [2647](#)
- nppsLShiftC_16u_I
 - signal_lshiftc, [2647](#)
- nppsLShiftC_32s
 - signal_lshiftc, [2647](#)
- nppsLShiftC_32s_I
 - signal_lshiftc, [2648](#)
- nppsLShiftC_32u
 - signal_lshiftc, [2648](#)
- nppsLShiftC_32u_I
 - signal_lshiftc, [2648](#)
- nppsLShiftC_8u
 - signal_lshiftc, [2649](#)
- nppsLShiftC_8u_I
 - signal_lshiftc, [2649](#)
- nppsMalloc_16s
 - signal_malloc, [2861](#)
- nppsMalloc_16sc
 - signal_malloc, [2861](#)
- nppsMalloc_16u
 - signal_malloc, [2861](#)
- nppsMalloc_32f
 - signal_malloc, [2861](#)
- nppsMalloc_32fc
 - signal_malloc, [2862](#)
- nppsMalloc_32s
 - signal_malloc, [2862](#)
- nppsMalloc_32sc
 - signal_malloc, [2862](#)
- nppsMalloc_32u
 - signal_malloc, [2862](#)
- nppsMalloc_64f
 - signal_malloc, [2863](#)
- nppsMalloc_64fc
 - signal_malloc, [2863](#)
- nppsMalloc_64s
 - signal_malloc, [2863](#)
- nppsMalloc_64sc
 - signal_malloc, [2863](#)
- nppsMalloc_8s
 - signal_malloc, [2864](#)
- nppsMalloc_8u
 - signal_malloc, [2864](#)
- nppsMax_16s
 - signal_max, [2712](#)
- nppsMax_32f
 - signal_max, [2713](#)
- nppsMax_32s
 - signal_max, [2713](#)
- nppsMax_64f
 - signal_max, [2713](#)
- nppsMaxAbs_16s
 - signal_max, [2714](#)
- nppsMaxAbs_32s
 - signal_max, [2714](#)
- nppsMaxAbsGetBufferSize_16s
 - signal_max, [2714](#)
- nppsMaxAbsGetBufferSize_32s
 - signal_max, [2715](#)
- nppsMaxAbsIndx_16s
 - signal_max, [2715](#)
- nppsMaxAbsIndx_32s
 - signal_max, [2715](#)
- nppsMaxAbsIndxGetBufferSize_16s
 - signal_max, [2716](#)
- nppsMaxAbsIndxGetBufferSize_32s
 - signal_max, [2716](#)
- nppsMaxEvery_16s_I
 - signal_min_every_or_max_every, [2700](#)
- nppsMaxEvery_16u_I
 - signal_min_every_or_max_every, [2701](#)
- nppsMaxEvery_32f_I
 - signal_min_every_or_max_every, [2701](#)
- nppsMaxEvery_32s_I
 - signal_min_every_or_max_every, [2701](#)
- nppsMaxEvery_8u_I
 - signal_min_every_or_max_every, [2701](#)
- nppsMaxGetBufferSize_16s
 - signal_max, [2716](#)
- nppsMaxGetBufferSize_32f
 - signal_max, [2716](#)
- nppsMaxGetBufferSize_32s
 - signal_max, [2717](#)
- nppsMaxGetBufferSize_64f
 - signal_max, [2717](#)
- nppsMaximumError_16s
 - signal_maximum_error, [2815](#)
- nppsMaximumError_16sc
 - signal_maximum_error, [2815](#)
- nppsMaximumError_16u
 - signal_maximum_error, [2815](#)
- nppsMaximumError_32f
 - signal_maximum_error, [2816](#)

- nppsMaximumError_32fc
 - signal_maximum_error, [2816](#)
- nppsMaximumError_32s
 - signal_maximum_error, [2816](#)
- nppsMaximumError_32sc
 - signal_maximum_error, [2817](#)
- nppsMaximumError_32u
 - signal_maximum_error, [2817](#)
- nppsMaximumError_64f
 - signal_maximum_error, [2817](#)
- nppsMaximumError_64fc
 - signal_maximum_error, [2818](#)
- nppsMaximumError_64s
 - signal_maximum_error, [2818](#)
- nppsMaximumError_64sc
 - signal_maximum_error, [2818](#)
- nppsMaximumError_8s
 - signal_maximum_error, [2819](#)
- nppsMaximumError_8u
 - signal_maximum_error, [2819](#)
- nppsMaximumErrorGetBufferSize_16s
 - signal_maximum_error, [2819](#)
- nppsMaximumErrorGetBufferSize_16sc
 - signal_maximum_error, [2820](#)
- nppsMaximumErrorGetBufferSize_16u
 - signal_maximum_error, [2820](#)
- nppsMaximumErrorGetBufferSize_32f
 - signal_maximum_error, [2820](#)
- nppsMaximumErrorGetBufferSize_32fc
 - signal_maximum_error, [2820](#)
- nppsMaximumErrorGetBufferSize_32s
 - signal_maximum_error, [2821](#)
- nppsMaximumErrorGetBufferSize_32sc
 - signal_maximum_error, [2821](#)
- nppsMaximumErrorGetBufferSize_32u
 - signal_maximum_error, [2821](#)
- nppsMaximumErrorGetBufferSize_64f
 - signal_maximum_error, [2821](#)
- nppsMaximumErrorGetBufferSize_64fc
 - signal_maximum_error, [2822](#)
- nppsMaximumErrorGetBufferSize_64s
 - signal_maximum_error, [2822](#)
- nppsMaximumErrorGetBufferSize_64sc
 - signal_maximum_error, [2822](#)
- nppsMaximumErrorGetBufferSize_8s
 - signal_maximum_error, [2822](#)
- nppsMaximumErrorGetBufferSize_8u
 - signal_maximum_error, [2823](#)
- nppsMaximumRelativeError_16s
 - signal_maximum_relative_error, [2837](#)
- nppsMaximumRelativeError_16sc
 - signal_maximum_relative_error, [2837](#)
- nppsMaximumRelativeError_16u
 - signal_maximum_relative_error, [2838](#)
- nppsMaximumRelativeError_32f
 - signal_maximum_relative_error, [2838](#)
- nppsMaximumRelativeError_32fc
 - signal_maximum_relative_error, [2838](#)
- nppsMaximumRelativeError_32s
 - signal_maximum_relative_error, [2839](#)
- nppsMaximumRelativeError_32sc
 - signal_maximum_relative_error, [2839](#)
- nppsMaximumRelativeError_32u
 - signal_maximum_relative_error, [2840](#)
- nppsMaximumRelativeError_64f
 - signal_maximum_relative_error, [2840](#)
- nppsMaximumRelativeError_64fc
 - signal_maximum_relative_error, [2840](#)
- nppsMaximumRelativeError_64s
 - signal_maximum_relative_error, [2841](#)
- nppsMaximumRelativeError_64sc
 - signal_maximum_relative_error, [2841](#)
- nppsMaximumRelativeError_8s
 - signal_maximum_relative_error, [2842](#)
- nppsMaximumRelativeError_8u
 - signal_maximum_relative_error, [2842](#)
- nppsMaximumRelativeErrorGetBufferSize_16s
 - signal_maximum_relative_error, [2842](#)
- nppsMaximumRelativeErrorGetBufferSize_16sc
 - signal_maximum_relative_error, [2843](#)
- nppsMaximumRelativeErrorGetBufferSize_16u
 - signal_maximum_relative_error, [2843](#)
- nppsMaximumRelativeErrorGetBufferSize_32f
 - signal_maximum_relative_error, [2843](#)
- nppsMaximumRelativeErrorGetBufferSize_32fc
 - signal_maximum_relative_error, [2843](#)
- nppsMaximumRelativeErrorGetBufferSize_32s
 - signal_maximum_relative_error, [2844](#)
- nppsMaximumRelativeErrorGetBufferSize_32sc
 - signal_maximum_relative_error, [2844](#)
- nppsMaximumRelativeErrorGetBufferSize_32u
 - signal_maximum_relative_error, [2844](#)
- nppsMaximumRelativeErrorGetBufferSize_64f
 - signal_maximum_relative_error, [2844](#)
- nppsMaximumRelativeErrorGetBufferSize_64fc
 - signal_maximum_relative_error, [2845](#)
- nppsMaximumRelativeErrorGetBufferSize_64s
 - signal_maximum_relative_error, [2845](#)
- nppsMaximumRelativeErrorGetBufferSize_64sc
 - signal_maximum_relative_error, [2845](#)
- nppsMaximumRelativeErrorGetBufferSize_8s
 - signal_maximum_relative_error, [2845](#)
- nppsMaximumRelativeErrorGetBufferSize_8u
 - signal_maximum_relative_error, [2846](#)
- nppsMaxIdx_16s
 - signal_max, [2717](#)
- nppsMaxIdx_32f
 - signal_max, [2718](#)

- npplsMaxIndx_32s
 - signal_max, [2718](#)
- npplsMaxIndx_64f
 - signal_max, [2718](#)
- npplsMaxIndxGetBufferSize_16s
 - signal_max, [2719](#)
- npplsMaxIndxGetBufferSize_32f
 - signal_max, [2719](#)
- npplsMaxIndxGetBufferSize_32s
 - signal_max, [2719](#)
- npplsMaxIndxGetBufferSize_64f
 - signal_max, [2720](#)
- npplsMean_16s_Sfs
 - signal_mean, [2732](#)
- npplsMean_16sc_Sfs
 - signal_mean, [2732](#)
- npplsMean_32f
 - signal_mean, [2732](#)
- npplsMean_32fc
 - signal_mean, [2733](#)
- npplsMean_32s_Sfs
 - signal_mean, [2733](#)
- npplsMean_64f
 - signal_mean, [2733](#)
- npplsMean_64fc
 - signal_mean, [2734](#)
- npplsMeanGetBufferSize_16s_Sfs
 - signal_mean, [2734](#)
- npplsMeanGetBufferSize_16sc_Sfs
 - signal_mean, [2734](#)
- npplsMeanGetBufferSize_32f
 - signal_mean, [2735](#)
- npplsMeanGetBufferSize_32fc
 - signal_mean, [2735](#)
- npplsMeanGetBufferSize_32s_Sfs
 - signal_mean, [2735](#)
- npplsMeanGetBufferSize_64f
 - signal_mean, [2735](#)
- npplsMeanGetBufferSize_64fc
 - signal_mean, [2736](#)
- npplsMeanStdDev_16s32s_Sfs
 - signal_mean_and_standard_deviation, [2740](#)
- npplsMeanStdDev_16s_Sfs
 - signal_mean_and_standard_deviation, [2741](#)
- npplsMeanStdDev_32f
 - signal_mean_and_standard_deviation, [2741](#)
- npplsMeanStdDev_64f
 - signal_mean_and_standard_deviation, [2741](#)
- npplsMeanStdDevGetBufferSize_16s32s_Sfs
 - signal_mean_and_standard_deviation, [2742](#)
- npplsMeanStdDevGetBufferSize_16s_Sfs
 - signal_mean_and_standard_deviation, [2742](#)
- npplsMeanStdDevGetBufferSize_32f
 - signal_mean_and_standard_deviation, [2742](#)
- npplsMeanStdDevGetBufferSize_64f
 - signal_mean_and_standard_deviation, [2742](#)
- npplsMin_16s
 - signal_min, [2722](#)
- npplsMin_32f
 - signal_min, [2723](#)
- npplsMin_32s
 - signal_min, [2723](#)
- npplsMin_64f
 - signal_min, [2723](#)
- npplsMinAbs_16s
 - signal_min, [2724](#)
- npplsMinAbs_32s
 - signal_min, [2724](#)
- npplsMinAbsGetBufferSize_16s
 - signal_min, [2724](#)
- npplsMinAbsGetBufferSize_32s
 - signal_min, [2725](#)
- npplsMinAbsIndx_16s
 - signal_min, [2725](#)
- npplsMinAbsIndx_32s
 - signal_min, [2725](#)
- npplsMinAbsIndxGetBufferSize_16s
 - signal_min, [2726](#)
- npplsMinAbsIndxGetBufferSize_32s
 - signal_min, [2726](#)
- npplsMinEvery_16s_I
 - signal_min_every_or_max_every, [2702](#)
- npplsMinEvery_16u_I
 - signal_min_every_or_max_every, [2702](#)
- npplsMinEvery_32f_I
 - signal_min_every_or_max_every, [2702](#)
- npplsMinEvery_32s_I
 - signal_min_every_or_max_every, [2703](#)
- npplsMinEvery_64f_I
 - signal_min_every_or_max_every, [2703](#)
- npplsMinEvery_8u_I
 - signal_min_every_or_max_every, [2703](#)
- npplsMinGetBufferSize_16s
 - signal_min, [2726](#)
- npplsMinGetBufferSize_32f
 - signal_min, [2726](#)
- npplsMinGetBufferSize_32s
 - signal_min, [2727](#)
- npplsMinGetBufferSize_64f
 - signal_min, [2727](#)
- npplsMinIndx_16s
 - signal_min, [2727](#)
- npplsMinIndx_32f
 - signal_min, [2728](#)
- npplsMinIndx_32s
 - signal_min, [2728](#)
- npplsMinIndx_64f
 - signal_min, [2728](#)

- npplsMinIndxGetBufferSize_16s
 - signal_min, [2729](#)
- npplsMinIndxGetBufferSize_32f
 - signal_min, [2729](#)
- npplsMinIndxGetBufferSize_32s
 - signal_min, [2729](#)
- npplsMinIndxGetBufferSize_64f
 - signal_min, [2730](#)
- npplsMinMax_16s
 - signal_min_max, [2746](#)
- npplsMinMax_16u
 - signal_min_max, [2746](#)
- npplsMinMax_32f
 - signal_min_max, [2746](#)
- npplsMinMax_32s
 - signal_min_max, [2747](#)
- npplsMinMax_32u
 - signal_min_max, [2747](#)
- npplsMinMax_64f
 - signal_min_max, [2747](#)
- npplsMinMax_8u
 - signal_min_max, [2748](#)
- npplsMinMaxGetBufferSize_16s
 - signal_min_max, [2748](#)
- npplsMinMaxGetBufferSize_16u
 - signal_min_max, [2748](#)
- npplsMinMaxGetBufferSize_32f
 - signal_min_max, [2749](#)
- npplsMinMaxGetBufferSize_32s
 - signal_min_max, [2749](#)
- npplsMinMaxGetBufferSize_32u
 - signal_min_max, [2749](#)
- npplsMinMaxGetBufferSize_64f
 - signal_min_max, [2749](#)
- npplsMinMaxGetBufferSize_8u
 - signal_min_max, [2750](#)
- npplsMinMaxIndx_16s
 - signal_min_max, [2750](#)
- npplsMinMaxIndx_16u
 - signal_min_max, [2750](#)
- npplsMinMaxIndx_32f
 - signal_min_max, [2751](#)
- npplsMinMaxIndx_32s
 - signal_min_max, [2751](#)
- npplsMinMaxIndx_32u
 - signal_min_max, [2752](#)
- npplsMinMaxIndx_64f
 - signal_min_max, [2752](#)
- npplsMinMaxIndx_8u
 - signal_min_max, [2752](#)
- npplsMinMaxIndxGetBufferSize_16s
 - signal_min_max, [2753](#)
- npplsMinMaxIndxGetBufferSize_16u
 - signal_min_max, [2753](#)
- npplsMinMaxIndxGetBufferSize_32f
 - signal_min_max, [2753](#)
- npplsMinMaxIndxGetBufferSize_32s
 - signal_min_max, [2754](#)
- npplsMinMaxIndxGetBufferSize_32u
 - signal_min_max, [2754](#)
- npplsMinMaxIndxGetBufferSize_64f
 - signal_min_max, [2754](#)
- npplsMinMaxIndxGetBufferSize_8u
 - signal_min_max, [2754](#)
- npplsMul_16s
 - signal_mul, [2554](#)
- npplsMul_16s32f
 - signal_mul, [2554](#)
- npplsMul_16s32s_Sfs
 - signal_mul, [2555](#)
- npplsMul_16s_I
 - signal_mul, [2555](#)
- npplsMul_16s_ISfs
 - signal_mul, [2555](#)
- npplsMul_16s_Sfs
 - signal_mul, [2556](#)
- npplsMul_16sc_ISfs
 - signal_mul, [2556](#)
- npplsMul_16sc_Sfs
 - signal_mul, [2556](#)
- npplsMul_16u16s_Sfs
 - signal_mul, [2557](#)
- npplsMul_16u_ISfs
 - signal_mul, [2557](#)
- npplsMul_16u_Sfs
 - signal_mul, [2557](#)
- npplsMul_32f
 - signal_mul, [2558](#)
- npplsMul_32f32fc
 - signal_mul, [2558](#)
- npplsMul_32f32fc_I
 - signal_mul, [2558](#)
- npplsMul_32f_I
 - signal_mul, [2559](#)
- npplsMul_32fc
 - signal_mul, [2559](#)
- npplsMul_32fc_I
 - signal_mul, [2559](#)
- npplsMul_32s32sc_ISfs
 - signal_mul, [2560](#)
- npplsMul_32s32sc_Sfs
 - signal_mul, [2560](#)
- npplsMul_32s_ISfs
 - signal_mul, [2560](#)
- npplsMul_32s_Sfs
 - signal_mul, [2561](#)
- npplsMul_32sc_ISfs
 - signal_mul, [2561](#)

- signal_mul, [2561](#)
- nppsMul_64f
 - signal_mul, [2562](#)
- nppsMul_64f_I
 - signal_mul, [2562](#)
- nppsMul_64fc
 - signal_mul, [2562](#)
- nppsMul_64fc_I
 - signal_mul, [2563](#)
- nppsMul_8u16u
 - signal_mul, [2563](#)
- nppsMul_8u_ISfs
 - signal_mul, [2563](#)
- nppsMul_8u_Sfs
 - signal_mul, [2564](#)
- nppsMul_Low_32s_Sfs
 - signal_mul, [2564](#)
- nppsMulC_16s_ISfs
 - signal_mulc, [2500](#)
- nppsMulC_16s_Sfs
 - signal_mulc, [2501](#)
- nppsMulC_16sc_ISfs
 - signal_mulc, [2501](#)
- nppsMulC_16sc_Sfs
 - signal_mulc, [2501](#)
- nppsMulC_16u_ISfs
 - signal_mulc, [2502](#)
- nppsMulC_16u_Sfs
 - signal_mulc, [2502](#)
- nppsMulC_32f
 - signal_mulc, [2502](#)
- nppsMulC_32f16s_Sfs
 - signal_mulc, [2503](#)
- nppsMulC_32f_I
 - signal_mulc, [2503](#)
- nppsMulC_32fc
 - signal_mulc, [2503](#)
- nppsMulC_32fc_I
 - signal_mulc, [2504](#)
- nppsMulC_32s_ISfs
 - signal_mulc, [2504](#)
- nppsMulC_32s_Sfs
 - signal_mulc, [2504](#)
- nppsMulC_32sc_ISfs
 - signal_mulc, [2505](#)
- nppsMulC_32sc_Sfs
 - signal_mulc, [2505](#)
- nppsMulC_64f
 - signal_mulc, [2505](#)
- nppsMulC_64f64s_ISfs
 - signal_mulc, [2506](#)
- nppsMulC_64f_I
 - signal_mulc, [2506](#)
- nppsMulC_64fc
 - signal_mulc, [2506](#)
- nppsMulC_64fc_I
 - signal_mulc, [2507](#)
- nppsMulC_8u_ISfs
 - signal_mulc, [2507](#)
- nppsMulC_8u_Sfs
 - signal_mulc, [2507](#)
- nppsMulC_Low_32f16s
 - signal_mulc, [2508](#)
- nppsNorm_Inf_16s32f
 - signal_infinity_norm, [2757](#)
- nppsNorm_Inf_16s32s_Sfs
 - signal_infinity_norm, [2757](#)
- nppsNorm_Inf_32f
 - signal_infinity_norm, [2757](#)
- nppsNorm_Inf_32fc32f
 - signal_infinity_norm, [2757](#)
- nppsNorm_Inf_64f
 - signal_infinity_norm, [2758](#)
- nppsNorm_Inf_64fc64f
 - signal_infinity_norm, [2758](#)
- nppsNorm_L1_16s32f
 - signal_L1_norm, [2762](#)
- nppsNorm_L1_16s32s_Sfs
 - signal_L1_norm, [2762](#)
- nppsNorm_L1_16s64s_Sfs
 - signal_L1_norm, [2762](#)
- nppsNorm_L1_32f
 - signal_L1_norm, [2763](#)
- nppsNorm_L1_32fc64f
 - signal_L1_norm, [2763](#)
- nppsNorm_L1_64f
 - signal_L1_norm, [2763](#)
- nppsNorm_L1_64fc64f
 - signal_L1_norm, [2764](#)
- nppsNorm_L2_16s32f
 - signal_L2_norm, [2768](#)
- nppsNorm_L2_16s32s_Sfs
 - signal_L2_norm, [2768](#)
- nppsNorm_L2_32f
 - signal_L2_norm, [2768](#)
- nppsNorm_L2_32fc64f
 - signal_L2_norm, [2769](#)
- nppsNorm_L2_64f
 - signal_L2_norm, [2769](#)
- nppsNorm_L2_64fc64f
 - signal_L2_norm, [2769](#)
- nppsNorm_L2Sqr_16s64s_Sfs
 - signal_L2_norm, [2770](#)
- nppsNormalize_16s_Sfs
 - signal_normalize, [2619](#)
- nppsNormalize_16sc_Sfs
 - signal_normalize, [2620](#)

- nppsNormalize_32f
 - signal_normalize, [2620](#)
- nppsNormalize_32fc
 - signal_normalize, [2620](#)
- nppsNormalize_64f
 - signal_normalize, [2621](#)
- nppsNormalize_64fc
 - signal_normalize, [2621](#)
- nppsNormDiff_Inf_16s32f
 - signal_infinity_norm_diff, [2774](#)
- nppsNormDiff_Inf_16s32s_Sfs
 - signal_infinity_norm_diff, [2774](#)
- nppsNormDiff_Inf_32f
 - signal_infinity_norm_diff, [2774](#)
- nppsNormDiff_Inf_32fc32f
 - signal_infinity_norm_diff, [2775](#)
- nppsNormDiff_Inf_64f
 - signal_infinity_norm_diff, [2775](#)
- nppsNormDiff_Inf_64fc64f
 - signal_infinity_norm_diff, [2775](#)
- nppsNormDiff_L1_16s32f
 - signal_L1_norm_diff, [2779](#)
- nppsNormDiff_L1_16s32s_Sfs
 - signal_L1_norm_diff, [2779](#)
- nppsNormDiff_L1_16s64s_Sfs
 - signal_L1_norm_diff, [2779](#)
- nppsNormDiff_L1_32f
 - signal_L1_norm_diff, [2780](#)
- nppsNormDiff_L1_32fc64f
 - signal_L1_norm_diff, [2780](#)
- nppsNormDiff_L1_64f
 - signal_L1_norm_diff, [2780](#)
- nppsNormDiff_L1_64fc64f
 - signal_L1_norm_diff, [2781](#)
- nppsNormDiff_L2_16s32f
 - signal_L2_norm_diff, [2785](#)
- nppsNormDiff_L2_16s32s_Sfs
 - signal_L2_norm_diff, [2785](#)
- nppsNormDiff_L2_32f
 - signal_L2_norm_diff, [2785](#)
- nppsNormDiff_L2_32fc64f
 - signal_L2_norm_diff, [2786](#)
- nppsNormDiff_L2_64f
 - signal_L2_norm_diff, [2786](#)
- nppsNormDiff_L2_64fc64f
 - signal_L2_norm_diff, [2786](#)
- nppsNormDiff_L2Sqr_16s64s_Sfs
 - signal_L2_norm_diff, [2787](#)
- nppsNormDiffInfGetBufferSize_16s32f
 - signal_infinity_norm_diff, [2776](#)
- nppsNormDiffInfGetBufferSize_16s32s_Sfs
 - signal_infinity_norm_diff, [2776](#)
- nppsNormDiffInfGetBufferSize_32f
 - signal_infinity_norm_diff, [2776](#)
- nppsNormDiffInfGetBufferSize_32fc32f
 - signal_infinity_norm_diff, [2777](#)
- nppsNormDiffInfGetBufferSize_64f
 - signal_infinity_norm_diff, [2777](#)
- nppsNormDiffInfGetBufferSize_64fc64f
 - signal_infinity_norm_diff, [2777](#)
- nppsNormDiffL1GetBufferSize_16s32f
 - signal_L1_norm_diff, [2781](#)
- nppsNormDiffL1GetBufferSize_16s32s_Sfs
 - signal_L1_norm_diff, [2781](#)
- nppsNormDiffL1GetBufferSize_16s64s_Sfs
 - signal_L1_norm_diff, [2782](#)
- nppsNormDiffL1GetBufferSize_32f
 - signal_L1_norm_diff, [2782](#)
- nppsNormDiffL1GetBufferSize_32fc64f
 - signal_L1_norm_diff, [2782](#)
- nppsNormDiffL1GetBufferSize_64f
 - signal_L1_norm_diff, [2782](#)
- nppsNormDiffL1GetBufferSize_64fc64f
 - signal_L1_norm_diff, [2783](#)
- nppsNormDiffL2GetBufferSize_16s32f
 - signal_L2_norm_diff, [2787](#)
- nppsNormDiffL2GetBufferSize_16s32s_Sfs
 - signal_L2_norm_diff, [2787](#)
- nppsNormDiffL2GetBufferSize_32f
 - signal_L2_norm_diff, [2788](#)
- nppsNormDiffL2GetBufferSize_32fc64f
 - signal_L2_norm_diff, [2788](#)
- nppsNormDiffL2GetBufferSize_64f
 - signal_L2_norm_diff, [2788](#)
- nppsNormDiffL2GetBufferSize_64fc64f
 - signal_L2_norm_diff, [2788](#)
- nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs
 - signal_L2_norm_diff, [2789](#)
- nppsNormInfGetBufferSize_16s32f
 - signal_infinity_norm, [2758](#)
- nppsNormInfGetBufferSize_16s32s_Sfs
 - signal_infinity_norm, [2759](#)
- nppsNormInfGetBufferSize_32f
 - signal_infinity_norm, [2759](#)
- nppsNormInfGetBufferSize_32fc32f
 - signal_infinity_norm, [2759](#)
- nppsNormInfGetBufferSize_64f
 - signal_infinity_norm, [2759](#)
- nppsNormInfGetBufferSize_64fc64f
 - signal_infinity_norm, [2760](#)
- nppsNormL1GetBufferSize_16s32f
 - signal_L1_norm, [2764](#)
- nppsNormL1GetBufferSize_16s32s_Sfs
 - signal_L1_norm, [2764](#)
- nppsNormL1GetBufferSize_16s64s_Sfs
 - signal_L1_norm, [2764](#)
- nppsNormL1GetBufferSize_32f
 - signal_L1_norm, [2765](#)

- nppsNormL1GetBufferSize_32fc64f
 - signal_L1_norm, [2765](#)
- nppsNormL1GetBufferSize_64f
 - signal_L1_norm, [2765](#)
- nppsNormL1GetBufferSize_64fc64f
 - signal_L1_norm, [2765](#)
- nppsNormL2GetBufferSize_16s32f
 - signal_L2_norm, [2770](#)
- nppsNormL2GetBufferSize_16s32s_Sfs
 - signal_L2_norm, [2770](#)
- nppsNormL2GetBufferSize_32f
 - signal_L2_norm, [2770](#)
- nppsNormL2GetBufferSize_32fc64f
 - signal_L2_norm, [2771](#)
- nppsNormL2GetBufferSize_64f
 - signal_L2_norm, [2771](#)
- nppsNormL2GetBufferSize_64fc64f
 - signal_L2_norm, [2771](#)
- nppsNormL2SqrGetBufferSize_16s64s_Sfs
 - signal_L2_norm, [2771](#)
- nppsNot_16u
 - signal_not, [2643](#)
- nppsNot_16u_I
 - signal_not, [2643](#)
- nppsNot_32u
 - signal_not, [2644](#)
- nppsNot_32u_I
 - signal_not, [2644](#)
- nppsNot_8u
 - signal_not, [2644](#)
- nppsNot_8u_I
 - signal_not, [2644](#)
- nppsOr_16u
 - signal_or, [2634](#)
- nppsOr_16u_I
 - signal_or, [2634](#)
- nppsOr_32u
 - signal_or, [2635](#)
- nppsOr_32u_I
 - signal_or, [2635](#)
- nppsOr_8u
 - signal_or, [2635](#)
- nppsOr_8u_I
 - signal_or, [2636](#)
- nppsOrC_16u
 - signal_orc, [2631](#)
- nppsOrC_16u_I
 - signal_orc, [2631](#)
- nppsOrC_32u
 - signal_orc, [2632](#)
- nppsOrC_32u_I
 - signal_orc, [2632](#)
- nppsOrC_8u
 - signal_orc, [2632](#)
- nppsOrC_8u_I
 - signal_orc, [2633](#)
- nppsRShiftC_16s
 - signal_rshifc, [2650](#)
- nppsRShiftC_16s_I
 - signal_rshifc, [2651](#)
- nppsRShiftC_16u
 - signal_rshifc, [2651](#)
- nppsRShiftC_16u_I
 - signal_rshifc, [2651](#)
- nppsRShiftC_32s
 - signal_rshifc, [2651](#)
- nppsRShiftC_32s_I
 - signal_rshifc, [2652](#)
- nppsRShiftC_32u
 - signal_rshifc, [2652](#)
- nppsRShiftC_32u_I
 - signal_rshifc, [2652](#)
- nppsRShiftC_8u
 - signal_rshifc, [2653](#)
- nppsRShiftC_8u_I
 - signal_rshifc, [2653](#)
- nppsSet_16s
 - signal_set, [2687](#)
- nppsSet_16sc
 - signal_set, [2687](#)
- nppsSet_16u
 - signal_set, [2687](#)
- nppsSet_32f
 - signal_set, [2687](#)
- nppsSet_32fc
 - signal_set, [2688](#)
- nppsSet_32s
 - signal_set, [2688](#)
- nppsSet_32sc
 - signal_set, [2688](#)
- nppsSet_32u
 - signal_set, [2688](#)
- nppsSet_64f
 - signal_set, [2689](#)
- nppsSet_64fc
 - signal_set, [2689](#)
- nppsSet_64s
 - signal_set, [2689](#)
- nppsSet_64sc
 - signal_set, [2690](#)
- nppsSet_8s
 - signal_set, [2690](#)
- nppsSet_8u
 - signal_set, [2690](#)
- nppsSqr_16s_ISfs
 - signal_square, [2590](#)
- nppsSqr_16s_Sfs
 - signal_square, [2590](#)

- nppsSqr_16sc_ISfs
 - signal_square, [2590](#)
- nppsSqr_16sc_Sfs
 - signal_square, [2591](#)
- nppsSqr_16u_ISfs
 - signal_square, [2591](#)
- nppsSqr_16u_Sfs
 - signal_square, [2591](#)
- nppsSqr_32f
 - signal_square, [2591](#)
- nppsSqr_32f_I
 - signal_square, [2592](#)
- nppsSqr_32fc
 - signal_square, [2592](#)
- nppsSqr_32fc_I
 - signal_square, [2592](#)
- nppsSqr_64f
 - signal_square, [2592](#)
- nppsSqr_64f_I
 - signal_square, [2593](#)
- nppsSqr_64fc
 - signal_square, [2593](#)
- nppsSqr_64fc_I
 - signal_square, [2593](#)
- nppsSqr_8u_ISfs
 - signal_square, [2593](#)
- nppsSqr_8u_Sfs
 - signal_square, [2594](#)
- nppsSqrt_16s_ISfs
 - signal_sqrt, [2596](#)
- nppsSqrt_16s_Sfs
 - signal_sqrt, [2596](#)
- nppsSqrt_16sc_ISfs
 - signal_sqrt, [2597](#)
- nppsSqrt_16sc_Sfs
 - signal_sqrt, [2597](#)
- nppsSqrt_16u_ISfs
 - signal_sqrt, [2597](#)
- nppsSqrt_16u_Sfs
 - signal_sqrt, [2597](#)
- nppsSqrt_32f
 - signal_sqrt, [2598](#)
- nppsSqrt_32f_I
 - signal_sqrt, [2598](#)
- nppsSqrt_32fc
 - signal_sqrt, [2598](#)
- nppsSqrt_32fc_I
 - signal_sqrt, [2599](#)
- nppsSqrt_32s16s_Sfs
 - signal_sqrt, [2599](#)
- nppsSqrt_64f
 - signal_sqrt, [2599](#)
- nppsSqrt_64f_I
 - signal_sqrt, [2599](#)
- nppsSqrt_64fc
 - signal_sqrt, [2600](#)
- nppsSqrt_64fc_I
 - signal_sqrt, [2600](#)
- nppsSqrt_64s16s_Sfs
 - signal_sqrt, [2600](#)
- nppsSqrt_64s_ISfs
 - signal_sqrt, [2600](#)
- nppsSqrt_64s_Sfs
 - signal_sqrt, [2601](#)
- nppsSqrt_8u_ISfs
 - signal_sqrt, [2601](#)
- nppsSqrt_8u_Sfs
 - signal_sqrt, [2601](#)
- nppsStdDev_16s32s_Sfs
 - signal_standard_deviation, [2737](#)
- nppsStdDev_16s_Sfs
 - signal_standard_deviation, [2737](#)
- nppsStdDev_32f
 - signal_standard_deviation, [2738](#)
- nppsStdDev_64f
 - signal_standard_deviation, [2738](#)
- nppsStdDevGetBufferSize_16s32s_Sfs
 - signal_standard_deviation, [2738](#)
- nppsStdDevGetBufferSize_16s_Sfs
 - signal_standard_deviation, [2739](#)
- nppsStdDevGetBufferSize_32f
 - signal_standard_deviation, [2739](#)
- nppsStdDevGetBufferSize_64f
 - signal_standard_deviation, [2739](#)
- nppsSub_16s
 - signal_sub, [2566](#)
- nppsSub_16s32f
 - signal_sub, [2567](#)
- nppsSub_16s_I
 - signal_sub, [2567](#)
- nppsSub_16s_ISfs
 - signal_sub, [2567](#)
- nppsSub_16s_Sfs
 - signal_sub, [2568](#)
- nppsSub_16sc_ISfs
 - signal_sub, [2568](#)
- nppsSub_16sc_Sfs
 - signal_sub, [2568](#)
- nppsSub_16u_ISfs
 - signal_sub, [2569](#)
- nppsSub_16u_Sfs
 - signal_sub, [2569](#)
- nppsSub_32f
 - signal_sub, [2569](#)
- nppsSub_32f_I
 - signal_sub, [2570](#)
- nppsSub_32fc
 - signal_sub, [2570](#)

- npssSub_32fc_I
 - signal_sub, [2570](#)
- npssSub_32s_ISfs
 - signal_sub, [2570](#)
- npssSub_32s_Sfs
 - signal_sub, [2571](#)
- npssSub_32sc_ISfs
 - signal_sub, [2571](#)
- npssSub_32sc_Sfs
 - signal_sub, [2571](#)
- npssSub_64f
 - signal_sub, [2572](#)
- npssSub_64f_I
 - signal_sub, [2572](#)
- npssSub_64fc
 - signal_sub, [2572](#)
- npssSub_64fc_I
 - signal_sub, [2573](#)
- npssSub_8u_ISfs
 - signal_sub, [2573](#)
- npssSub_8u_Sfs
 - signal_sub, [2573](#)
- npssSubC_16s_ISfs
 - signal_subc, [2510](#)
- npssSubC_16s_Sfs
 - signal_subc, [2510](#)
- npssSubC_16sc_ISfs
 - signal_subc, [2511](#)
- npssSubC_16sc_Sfs
 - signal_subc, [2511](#)
- npssSubC_16u_ISfs
 - signal_subc, [2511](#)
- npssSubC_16u_Sfs
 - signal_subc, [2512](#)
- npssSubC_32f
 - signal_subc, [2512](#)
- npssSubC_32f_I
 - signal_subc, [2512](#)
- npssSubC_32fc
 - signal_subc, [2513](#)
- npssSubC_32fc_I
 - signal_subc, [2513](#)
- npssSubC_32s_ISfs
 - signal_subc, [2513](#)
- npssSubC_32s_Sfs
 - signal_subc, [2514](#)
- npssSubC_32sc_ISfs
 - signal_subc, [2514](#)
- npssSubC_32sc_Sfs
 - signal_subc, [2514](#)
- npssSubC_64f
 - signal_subc, [2515](#)
- npssSubC_64f_I
 - signal_subc, [2515](#)
- npssSubC_64fc
 - signal_subc, [2515](#)
- npssSubC_64fc_I
 - signal_subc, [2516](#)
- npssSubC_8u_ISfs
 - signal_subc, [2516](#)
- npssSubC_8u_Sfs
 - signal_subc, [2516](#)
- npssSubCRev_16s_ISfs
 - signal_subcrev, [2519](#)
- npssSubCRev_16s_Sfs
 - signal_subcrev, [2520](#)
- npssSubCRev_16sc_ISfs
 - signal_subcrev, [2520](#)
- npssSubCRev_16sc_Sfs
 - signal_subcrev, [2520](#)
- npssSubCRev_16u_ISfs
 - signal_subcrev, [2521](#)
- npssSubCRev_16u_Sfs
 - signal_subcrev, [2521](#)
- npssSubCRev_32f
 - signal_subcrev, [2521](#)
- npssSubCRev_32f_I
 - signal_subcrev, [2522](#)
- npssSubCRev_32fc
 - signal_subcrev, [2522](#)
- npssSubCRev_32fc_I
 - signal_subcrev, [2522](#)
- npssSubCRev_32s_ISfs
 - signal_subcrev, [2522](#)
- npssSubCRev_32s_Sfs
 - signal_subcrev, [2523](#)
- npssSubCRev_32sc_ISfs
 - signal_subcrev, [2523](#)
- npssSubCRev_32sc_Sfs
 - signal_subcrev, [2523](#)
- npssSubCRev_64f
 - signal_subcrev, [2524](#)
- npssSubCRev_64f_I
 - signal_subcrev, [2524](#)
- npssSubCRev_64fc
 - signal_subcrev, [2524](#)
- npssSubCRev_64fc_I
 - signal_subcrev, [2525](#)
- npssSubCRev_8u_ISfs
 - signal_subcrev, [2525](#)
- npssSubCRev_8u_Sfs
 - signal_subcrev, [2525](#)
- npssSum_16s32s_Sfs
 - signal_sum, [2705](#)
- npssSum_16s_Sfs
 - signal_sum, [2705](#)
- npssSum_16sc32sc_Sfs
 - signal_sum, [2706](#)

- nppsSum_16sc_Sfs
 - signal_sum, [2706](#)
- nppsSum_32f
 - signal_sum, [2706](#)
- nppsSum_32fc
 - signal_sum, [2707](#)
- nppsSum_32s_Sfs
 - signal_sum, [2707](#)
- nppsSum_64f
 - signal_sum, [2707](#)
- nppsSum_64fc
 - signal_sum, [2708](#)
- nppsSumGetBufferSize_16s32s_Sfs
 - signal_sum, [2708](#)
- nppsSumGetBufferSize_16s_Sfs
 - signal_sum, [2708](#)
- nppsSumGetBufferSize_16sc32sc_Sfs
 - signal_sum, [2709](#)
- nppsSumGetBufferSize_16sc_Sfs
 - signal_sum, [2709](#)
- nppsSumGetBufferSize_32f
 - signal_sum, [2709](#)
- nppsSumGetBufferSize_32fc
 - signal_sum, [2709](#)
- nppsSumGetBufferSize_32s_Sfs
 - signal_sum, [2710](#)
- nppsSumGetBufferSize_64f
 - signal_sum, [2710](#)
- nppsSumGetBufferSize_64fc
 - signal_sum, [2710](#)
- nppsSumLn_16s32f
 - signal_sumln, [2613](#)
- nppsSumLn_32f
 - signal_sumln, [2614](#)
- nppsSumLn_32f64f
 - signal_sumln, [2614](#)
- nppsSumLn_64f
 - signal_sumln, [2614](#)
- nppsSumLnGetBufferSize_16s32f
 - signal_sumln, [2615](#)
- nppsSumLnGetBufferSize_32f
 - signal_sumln, [2615](#)
- nppsSumLnGetBufferSize_32f64f
 - signal_sumln, [2615](#)
- nppsSumLnGetBufferSize_64f
 - signal_sumln, [2615](#)
- NppStatus
 - typedefs_npp, [45](#)
- nppsThreshold_16s
 - signal_threshold, [2662](#)
- nppsThreshold_16s_I
 - signal_threshold, [2663](#)
- nppsThreshold_16sc
 - signal_threshold, [2663](#)
- nppsThreshold_16sc_I
 - signal_threshold, [2663](#)
- nppsThreshold_32f
 - signal_threshold, [2664](#)
- nppsThreshold_32f_I
 - signal_threshold, [2664](#)
- nppsThreshold_32fc
 - signal_threshold, [2664](#)
- nppsThreshold_32fc_I
 - signal_threshold, [2665](#)
- nppsThreshold_64f
 - signal_threshold, [2665](#)
- nppsThreshold_64f_I
 - signal_threshold, [2665](#)
- nppsThreshold_64fc
 - signal_threshold, [2666](#)
- nppsThreshold_64fc_I
 - signal_threshold, [2666](#)
- nppsThreshold_GT_16s
 - signal_threshold, [2666](#)
- nppsThreshold_GT_16s_I
 - signal_threshold, [2667](#)
- nppsThreshold_GT_16sc
 - signal_threshold, [2667](#)
- nppsThreshold_GT_16sc_I
 - signal_threshold, [2667](#)
- nppsThreshold_GT_32f
 - signal_threshold, [2668](#)
- nppsThreshold_GT_32f_I
 - signal_threshold, [2668](#)
- nppsThreshold_GT_32fc
 - signal_threshold, [2668](#)
- nppsThreshold_GT_32fc_I
 - signal_threshold, [2669](#)
- nppsThreshold_GT_64f
 - signal_threshold, [2669](#)
- nppsThreshold_GT_64f_I
 - signal_threshold, [2669](#)
- nppsThreshold_GT_64fc
 - signal_threshold, [2670](#)
- nppsThreshold_GT_64fc_I
 - signal_threshold, [2670](#)
- nppsThreshold_GTVal_16s
 - signal_threshold, [2670](#)
- nppsThreshold_GTVal_16s_I
 - signal_threshold, [2671](#)
- nppsThreshold_GTVal_16sc
 - signal_threshold, [2671](#)
- nppsThreshold_GTVal_16sc_I
 - signal_threshold, [2671](#)
- nppsThreshold_GTVal_32f
 - signal_threshold, [2672](#)
- nppsThreshold_GTVal_32f_I
 - signal_threshold, [2672](#)

- nppsThreshold_GTVal_32fc
 - signal_threshold, [2672](#)
- nppsThreshold_GTVal_32fc_I
 - signal_threshold, [2673](#)
- nppsThreshold_GTVal_64f
 - signal_threshold, [2673](#)
- nppsThreshold_GTVal_64f_I
 - signal_threshold, [2673](#)
- nppsThreshold_GTVal_64fc
 - signal_threshold, [2674](#)
- nppsThreshold_GTVal_64fc_I
 - signal_threshold, [2674](#)
- nppsThreshold_LT_16s
 - signal_threshold, [2674](#)
- nppsThreshold_LT_16s_I
 - signal_threshold, [2675](#)
- nppsThreshold_LT_16sc
 - signal_threshold, [2675](#)
- nppsThreshold_LT_16sc_I
 - signal_threshold, [2675](#)
- nppsThreshold_LT_32f
 - signal_threshold, [2676](#)
- nppsThreshold_LT_32f_I
 - signal_threshold, [2676](#)
- nppsThreshold_LT_32fc
 - signal_threshold, [2676](#)
- nppsThreshold_LT_32fc_I
 - signal_threshold, [2677](#)
- nppsThreshold_LT_64f
 - signal_threshold, [2677](#)
- nppsThreshold_LT_64f_I
 - signal_threshold, [2677](#)
- nppsThreshold_LT_64fc
 - signal_threshold, [2678](#)
- nppsThreshold_LT_64fc_I
 - signal_threshold, [2678](#)
- nppsThreshold_LTVal_16s
 - signal_threshold, [2678](#)
- nppsThreshold_LTVal_16s_I
 - signal_threshold, [2679](#)
- nppsThreshold_LTVal_16sc
 - signal_threshold, [2679](#)
- nppsThreshold_LTVal_16sc_I
 - signal_threshold, [2679](#)
- nppsThreshold_LTVal_32f
 - signal_threshold, [2680](#)
- nppsThreshold_LTVal_32f_I
 - signal_threshold, [2680](#)
- nppsThreshold_LTVal_32fc
 - signal_threshold, [2680](#)
- nppsThreshold_LTVal_32fc_I
 - signal_threshold, [2681](#)
- nppsThreshold_LTVal_64f
 - signal_threshold, [2681](#)
- nppsThreshold_LTVal_64f_I
 - signal_threshold, [2681](#)
- nppsThreshold_LTVal_64fc
 - signal_threshold, [2682](#)
- nppsThreshold_LTVal_64fc_I
 - signal_threshold, [2682](#)
- nppsXor_16u
 - signal_xor, [2640](#)
- nppsXor_16u_I
 - signal_xor, [2640](#)
- nppsXor_32u
 - signal_xor, [2641](#)
- nppsXor_32u_I
 - signal_xor, [2641](#)
- nppsXor_8u
 - signal_xor, [2641](#)
- nppsXor_8u_I
 - signal_xor, [2642](#)
- nppsXorC_16u
 - signal_xor, [2637](#)
- nppsXorC_16u_I
 - signal_xor, [2637](#)
- nppsXorC_32u
 - signal_xor, [2638](#)
- nppsXorC_32u_I
 - signal_xor, [2638](#)
- nppsXorC_8u
 - signal_xor, [2638](#)
- nppsXorC_8u_I
 - signal_xor, [2639](#)
- NppsZCType
 - typedefs_npp, [47](#)
- nppsZero_16s
 - signal_zero, [2691](#)
- nppsZero_16sc
 - signal_zero, [2692](#)
- nppsZero_32f
 - signal_zero, [2692](#)
- nppsZero_32fc
 - signal_zero, [2692](#)
- nppsZero_32s
 - signal_zero, [2692](#)
- nppsZero_32sc
 - signal_zero, [2692](#)
- nppsZero_64f
 - signal_zero, [2693](#)
- nppsZero_64fc
 - signal_zero, [2693](#)
- nppsZero_64s
 - signal_zero, [2693](#)
- nppsZero_64sc
 - signal_zero, [2693](#)
- nppsZero_8u
 - signal_zero, [2694](#)

- nppsZeroCrossing_16s32f
 - signal_count_zero_crossings, [2811](#)
- nppsZeroCrossing_32f
 - signal_count_zero_crossings, [2811](#)
- nppsZeroCrossingGetBufferSize_16s32f
 - signal_count_zero_crossings, [2812](#)
- nppsZeroCrossingGetBufferSize_32f
 - signal_count_zero_crossings, [2812](#)
- nppZCC
 - typedefs_npp, [47](#)
- nppZCR
 - typedefs_npp, [47](#)
- nppZCxor
 - typedefs_npp, [47](#)
- numClassifiers
 - NppiHaarClassifier_32f, [2872](#)
- Or, [445](#), [2634](#)
- OrC, [383](#), [2631](#)
- Perspective Transform, [1529](#)
- Quantization Functions, [724](#)
- Rank Filters, [1298](#)
- re
 - NPP_ALIGN_16, [2868](#)
 - NPP_ALIGN_8, [2869](#), [2870](#)
- RectStdDev, [2093](#)
- Remap, [1431](#)
- Resize, [1419](#)
- ResizeSqrPixel, [1396](#)
- Rotate, [1453](#)
- RShiftC, [405](#), [2650](#)
- Scale, [864](#)
- Set, [739](#), [2686](#)
- signal_10log10
 - npps10Log10_32s_ISfs, [2612](#)
 - npps10Log10_32s_Sfs, [2612](#)
- signal_abs
 - nppsAbs_16s, [2586](#)
 - nppsAbs_16s_I, [2586](#)
 - nppsAbs_32f, [2587](#)
 - nppsAbs_32f_I, [2587](#)
 - nppsAbs_32s, [2587](#)
 - nppsAbs_32s_I, [2587](#)
 - nppsAbs_64f, [2588](#)
 - nppsAbs_64f_I, [2588](#)
- signal_add
 - nppsAdd_16s, [2538](#)
 - nppsAdd_16s32f, [2538](#)
 - nppsAdd_16s32s_I, [2538](#)
 - nppsAdd_16s_I, [2539](#)
 - nppsAdd_16s_ISfs, [2539](#)
 - nppsAdd_16s_Sfs, [2539](#)
 - nppsAdd_16sc_ISfs, [2540](#)
 - nppsAdd_16sc_Sfs, [2540](#)
 - nppsAdd_16u, [2540](#)
 - nppsAdd_16u_ISfs, [2541](#)
 - nppsAdd_16u_Sfs, [2541](#)
 - nppsAdd_32f, [2541](#)
 - nppsAdd_32f_I, [2542](#)
 - nppsAdd_32fc, [2542](#)
 - nppsAdd_32fc_I, [2542](#)
 - nppsAdd_32s_ISfs, [2543](#)
 - nppsAdd_32s_Sfs, [2543](#)
 - nppsAdd_32sc_ISfs, [2543](#)
 - nppsAdd_32sc_Sfs, [2544](#)
 - nppsAdd_32u, [2544](#)
 - nppsAdd_64f, [2544](#)
 - nppsAdd_64f_I, [2545](#)
 - nppsAdd_64fc, [2545](#)
 - nppsAdd_64fc_I, [2545](#)
 - nppsAdd_64s_Sfs, [2546](#)
 - nppsAdd_8u16u, [2546](#)
 - nppsAdd_8u_ISfs, [2546](#)
 - nppsAdd_8u_Sfs, [2547](#)
- signal_addc
 - nppsAddC_16s_ISfs, [2490](#)
 - nppsAddC_16s_Sfs, [2490](#)
 - nppsAddC_16sc_ISfs, [2491](#)
 - nppsAddC_16sc_Sfs, [2491](#)
 - nppsAddC_16u_ISfs, [2491](#)
 - nppsAddC_16u_Sfs, [2492](#)
 - nppsAddC_32f, [2492](#)
 - nppsAddC_32f_I, [2492](#)
 - nppsAddC_32fc, [2493](#)
 - nppsAddC_32fc_I, [2493](#)
 - nppsAddC_32s_ISfs, [2493](#)
 - nppsAddC_32s_Sfs, [2494](#)
 - nppsAddC_32sc_ISfs, [2494](#)
 - nppsAddC_32sc_Sfs, [2494](#)
 - nppsAddC_64f, [2495](#)
 - nppsAddC_64f_I, [2495](#)
 - nppsAddC_64fc, [2495](#)
 - nppsAddC_64fc_I, [2496](#)
 - nppsAddC_8u_ISfs, [2496](#)
 - nppsAddC_8u_Sfs, [2496](#)
- signal_addproduct
 - nppsAddProduct_16s32s_Sfs, [2549](#)
 - nppsAddProduct_16s_Sfs, [2549](#)
 - nppsAddProduct_32f, [2549](#)
 - nppsAddProduct_32fc, [2550](#)
 - nppsAddProduct_32s_Sfs, [2550](#)
 - nppsAddProduct_64f, [2550](#)
 - nppsAddProduct_64fc, [2551](#)
- signal_addproductc
 - nppsAddProductC_32f, [2498](#)

- signal_and
 - nppsAnd_16u, [2628](#)
 - nppsAnd_16u_I, [2628](#)
 - nppsAnd_32u, [2629](#)
 - nppsAnd_32u_I, [2629](#)
 - nppsAnd_8u, [2629](#)
 - nppsAnd_8u_I, [2630](#)
- signal_andc
 - nppsAndC_16u, [2625](#)
 - nppsAndC_16u_I, [2625](#)
 - nppsAndC_32u, [2626](#)
 - nppsAndC_32u_I, [2626](#)
 - nppsAndC_8u, [2626](#)
 - nppsAndC_8u_I, [2627](#)
- signal_average_error
 - nppsAverageError_16s, [2826](#)
 - nppsAverageError_16sc, [2826](#)
 - nppsAverageError_16u, [2826](#)
 - nppsAverageError_32f, [2827](#)
 - nppsAverageError_32fc, [2827](#)
 - nppsAverageError_32s, [2827](#)
 - nppsAverageError_32sc, [2828](#)
 - nppsAverageError_32u, [2828](#)
 - nppsAverageError_64f, [2828](#)
 - nppsAverageError_64fc, [2829](#)
 - nppsAverageError_64s, [2829](#)
 - nppsAverageError_64sc, [2829](#)
 - nppsAverageError_8s, [2830](#)
 - nppsAverageError_8u, [2830](#)
 - nppsAverageErrorGetBufferSize_16s, [2830](#)
 - nppsAverageErrorGetBufferSize_16sc, [2831](#)
 - nppsAverageErrorGetBufferSize_16u, [2831](#)
 - nppsAverageErrorGetBufferSize_32f, [2831](#)
 - nppsAverageErrorGetBufferSize_32fc, [2831](#)
 - nppsAverageErrorGetBufferSize_32s, [2832](#)
 - nppsAverageErrorGetBufferSize_32sc, [2832](#)
 - nppsAverageErrorGetBufferSize_32u, [2832](#)
 - nppsAverageErrorGetBufferSize_64f, [2832](#)
 - nppsAverageErrorGetBufferSize_64fc, [2833](#)
 - nppsAverageErrorGetBufferSize_64s, [2833](#)
 - nppsAverageErrorGetBufferSize_64sc, [2833](#)
 - nppsAverageErrorGetBufferSize_8s, [2833](#)
 - nppsAverageErrorGetBufferSize_8u, [2834](#)
- signal_average_relative_error
 - nppsAverageRelativeError_16s, [2849](#)
 - nppsAverageRelativeError_16sc, [2849](#)
 - nppsAverageRelativeError_16u, [2850](#)
 - nppsAverageRelativeError_32f, [2850](#)
 - nppsAverageRelativeError_32fc, [2850](#)
 - nppsAverageRelativeError_32s, [2851](#)
 - nppsAverageRelativeError_32sc, [2851](#)
 - nppsAverageRelativeError_32u, [2852](#)
 - nppsAverageRelativeError_64f, [2852](#)
 - nppsAverageRelativeError_64fc, [2852](#)
 - nppsAverageRelativeError_64s, [2853](#)
 - nppsAverageRelativeError_64sc, [2853](#)
 - nppsAverageRelativeError_8s, [2854](#)
 - nppsAverageRelativeError_8u, [2854](#)
 - nppsAverageRelativeErrorGetBufferSize_16s, [2854](#)
 - nppsAverageRelativeErrorGetBufferSize_16sc, [2855](#)
 - nppsAverageRelativeErrorGetBufferSize_16u, [2855](#)
 - nppsAverageRelativeErrorGetBufferSize_32f, [2855](#)
 - nppsAverageRelativeErrorGetBufferSize_32fc, [2855](#)
 - nppsAverageRelativeErrorGetBufferSize_32s, [2856](#)
 - nppsAverageRelativeErrorGetBufferSize_32sc, [2856](#)
 - nppsAverageRelativeErrorGetBufferSize_32u, [2856](#)
 - nppsAverageRelativeErrorGetBufferSize_64f, [2856](#)
 - nppsAverageRelativeErrorGetBufferSize_64fc, [2857](#)
 - nppsAverageRelativeErrorGetBufferSize_64s, [2857](#)
 - nppsAverageRelativeErrorGetBufferSize_64sc, [2857](#)
 - nppsAverageRelativeErrorGetBufferSize_8s, [2857](#)
 - nppsAverageRelativeErrorGetBufferSize_8u, [2858](#)
- signal_cauchy
 - nppsCauchy_32f_I, [2622](#)
 - nppsCauchyD_32f_I, [2622](#)
 - nppsCauchyDD2_32f_I, [2622](#)
- signal_convert
 - nppsConvert_16s32f, [2657](#)
 - nppsConvert_16s32f_Sfs, [2657](#)
 - nppsConvert_16s32s, [2657](#)
 - nppsConvert_16s64f_Sfs, [2657](#)
 - nppsConvert_16s8s_Sfs, [2657](#)
 - nppsConvert_16u32f, [2657](#)
 - nppsConvert_32f16s_Sfs, [2657](#)
 - nppsConvert_32f16u_Sfs, [2657](#)
 - nppsConvert_32f32s_Sfs, [2657](#)
 - nppsConvert_32f64f, [2657](#)
 - nppsConvert_32f8s_Sfs, [2657](#)
 - nppsConvert_32f8u_Sfs, [2657](#)
 - nppsConvert_32s16s, [2657](#)
 - nppsConvert_32s16s_Sfs, [2657](#)
 - nppsConvert_32s32f, [2657](#)
 - nppsConvert_32s32f_Sfs, [2657](#)
 - nppsConvert_32s64f, [2657](#)

- nppsConvert_32s64f_Sfs, 2657
- nppsConvert_64f16s_Sfs, 2657
- nppsConvert_64f32f, 2657
- nppsConvert_64f32s_Sfs, 2657
- nppsConvert_64f64s_Sfs, 2657
- nppsConvert_64s32s_Sfs, 2657
- nppsConvert_64s64f, 2657
- nppsConvert_8s16s, 2657
- nppsConvert_8s32f, 2657
- nppsConvert_8u32f, 2657
- signal_copy
 - nppsCopy_16s, 2695
 - nppsCopy_16sc, 2696
 - nppsCopy_32f, 2696
 - nppsCopy_32fc, 2696
 - nppsCopy_32s, 2696
 - nppsCopy_32sc, 2697
 - nppsCopy_64fc, 2697
 - nppsCopy_64s, 2697
 - nppsCopy_64sc, 2698
 - nppsCopy_8u, 2698
- signal_count_in_range
 - nppsCountInRange_32s, 2810
 - nppsCountInRangeGetBufferSize_32s, 2810
- signal_count_zero_crossings
 - nppsZeroCrossing_16s32f, 2811
 - nppsZeroCrossing_32f, 2811
 - nppsZeroCrossingGetBufferSize_16s32f, 2812
 - nppsZeroCrossingGetBufferSize_32f, 2812
- signal_cuberoot
 - nppsCubrt_32f, 2603
 - nppsCubrt_32s16s_Sfs, 2603
- signal_div
 - nppsDiv_16s_ISfs, 2576
 - nppsDiv_16s_Sfs, 2576
 - nppsDiv_16sc_ISfs, 2577
 - nppsDiv_16sc_Sfs, 2577
 - nppsDiv_16u_ISfs, 2577
 - nppsDiv_16u_Sfs, 2578
 - nppsDiv_32f, 2578
 - nppsDiv_32f_I, 2578
 - nppsDiv_32fc, 2579
 - nppsDiv_32fc_I, 2579
 - nppsDiv_32s16s_Sfs, 2579
 - nppsDiv_32s_ISfs, 2580
 - nppsDiv_32s_Sfs, 2580
 - nppsDiv_64f, 2580
 - nppsDiv_64f_I, 2581
 - nppsDiv_64fc, 2581
 - nppsDiv_64fc_I, 2581
 - nppsDiv_8u_ISfs, 2582
 - nppsDiv_8u_Sfs, 2582
- signal_divc
 - nppsDivC_16s_ISfs, 2528
 - nppsDivC_16s_Sfs, 2528
 - nppsDivC_16sc_ISfs, 2528
 - nppsDivC_16sc_Sfs, 2529
 - nppsDivC_16u_ISfs, 2529
 - nppsDivC_16u_Sfs, 2529
 - nppsDivC_32f, 2530
 - nppsDivC_32f_I, 2530
 - nppsDivC_32fc, 2530
 - nppsDivC_32fc_I, 2531
 - nppsDivC_64f, 2531
 - nppsDivC_64f_I, 2531
 - nppsDivC_64fc, 2532
 - nppsDivC_64fc_I, 2532
 - nppsDivC_8u_ISfs, 2532
 - nppsDivC_8u_Sfs, 2533
- signal_divcrev
 - nppsDivCRev_16u, 2534
 - nppsDivCRev_16u_I, 2534
 - nppsDivCRev_32f, 2535
 - nppsDivCRev_32f_I, 2535
- signal_divround
 - nppsDiv_Round_16s_ISfs, 2583
 - nppsDiv_Round_16s_Sfs, 2584
 - nppsDiv_Round_16u_ISfs, 2584
 - nppsDiv_Round_16u_Sfs, 2584
 - nppsDiv_Round_8u_ISfs, 2585
 - nppsDiv_Round_8u_Sfs, 2585
- signal_dot_product
 - nppsDotProd_16s16sc32fc, 2793
 - nppsDotProd_16s16sc32sc_Sfs, 2794
 - nppsDotProd_16s16sc64sc, 2794
 - nppsDotProd_16s16sc_Sfs, 2794
 - nppsDotProd_16s32f, 2795
 - nppsDotProd_16s32s32s_Sfs, 2795
 - nppsDotProd_16s32s_Sfs, 2796
 - nppsDotProd_16s64s, 2796
 - nppsDotProd_16s_Sfs, 2796
 - nppsDotProd_16sc32fc, 2797
 - nppsDotProd_16sc32sc_Sfs, 2797
 - nppsDotProd_16sc64sc, 2798
 - nppsDotProd_16sc_Sfs, 2798
 - nppsDotProd_32f, 2798
 - nppsDotProd_32f32fc, 2799
 - nppsDotProd_32f32fc64fc, 2799
 - nppsDotProd_32f64f, 2799
 - nppsDotProd_32fc, 2800
 - nppsDotProd_32fc64fc, 2800
 - nppsDotProd_32s32sc_Sfs, 2800
 - nppsDotProd_32s_Sfs, 2801
 - nppsDotProd_32sc_Sfs, 2801
 - nppsDotProd_64f, 2802
 - nppsDotProd_64f64fc, 2802
 - nppsDotProd_64fc, 2802

- nppsDotProdGetBufferSize_16s16sc32fc, 2803
- nppsDotProdGetBufferSize_16s16sc32sc_Sfs, 2803
- nppsDotProdGetBufferSize_16s16sc64sc, 2803
- nppsDotProdGetBufferSize_16s16sc_Sfs, 2804
- nppsDotProdGetBufferSize_16s32f, 2804
- nppsDotProdGetBufferSize_16s32s32s_Sfs, 2804
- nppsDotProdGetBufferSize_16s32s_Sfs, 2804
- nppsDotProdGetBufferSize_16s64s, 2805
- nppsDotProdGetBufferSize_16s_Sfs, 2805
- nppsDotProdGetBufferSize_16sc32fc, 2805
- nppsDotProdGetBufferSize_16sc32sc_Sfs, 2805
- nppsDotProdGetBufferSize_16sc64sc, 2806
- nppsDotProdGetBufferSize_16sc_Sfs, 2806
- nppsDotProdGetBufferSize_32f, 2806
- nppsDotProdGetBufferSize_32f32fc, 2806
- nppsDotProdGetBufferSize_32f32fc64fc, 2807
- nppsDotProdGetBufferSize_32f64f, 2807
- nppsDotProdGetBufferSize_32fc, 2807
- nppsDotProdGetBufferSize_32fc64fc, 2807
- nppsDotProdGetBufferSize_32s32sc_Sfs, 2808
- nppsDotProdGetBufferSize_32s_Sfs, 2808
- nppsDotProdGetBufferSize_32sc_Sfs, 2808
- nppsDotProdGetBufferSize_64f, 2808
- nppsDotProdGetBufferSize_64f64fc, 2809
- nppsDotProdGetBufferSize_64fc, 2809
- signal_exp
 - nppsExp_16s_ISfs, 2604
 - nppsExp_16s_Sfs, 2605
 - nppsExp_32f, 2605
 - nppsExp_32f64f, 2605
 - nppsExp_32f_I, 2605
 - nppsExp_32s_ISfs, 2606
 - nppsExp_32s_Sfs, 2606
 - nppsExp_64f, 2606
 - nppsExp_64f_I, 2607
 - nppsExp_64s_ISfs, 2607
 - nppsExp_64s_Sfs, 2607
- signal_free
 - nppsFree, 2865
- signal_infinity_norm
 - nppsNorm_Inf_16s32f, 2757
 - nppsNorm_Inf_16s32s_Sfs, 2757
 - nppsNorm_Inf_32f, 2757
 - nppsNorm_Inf_32fc32f, 2757
 - nppsNorm_Inf_64f, 2758
 - nppsNorm_Inf_64fc64f, 2758
- nppsNormInfGetBufferSize_16s32f, 2758
- nppsNormInfGetBufferSize_16s32s_Sfs, 2759
- nppsNormInfGetBufferSize_32f, 2759
- nppsNormInfGetBufferSize_32fc32f, 2759
- nppsNormInfGetBufferSize_64f, 2759
- nppsNormInfGetBufferSize_64fc64f, 2760
- signal_infinity_norm_diff
 - nppsNormDiff_Inf_16s32f, 2774
 - nppsNormDiff_Inf_16s32s_Sfs, 2774
 - nppsNormDiff_Inf_32f, 2774
 - nppsNormDiff_Inf_32fc32f, 2775
 - nppsNormDiff_Inf_64f, 2775
 - nppsNormDiff_Inf_64fc64f, 2775
 - nppsNormDiffInfGetBufferSize_16s32f, 2776
 - nppsNormDiffInfGetBufferSize_16s32s_Sfs, 2776
 - nppsNormDiffInfGetBufferSize_32f, 2776
 - nppsNormDiffInfGetBufferSize_32fc32f, 2777
 - nppsNormDiffInfGetBufferSize_64f, 2777
 - nppsNormDiffInfGetBufferSize_64fc64f, 2777
- signal_integral
 - nppsIntegral_32s, 2684
 - nppsIntegralGetBufferSize_32s, 2684
- signal_inversetan
 - nppsArctan_32f, 2617
 - nppsArctan_32f_I, 2617
 - nppsArctan_64f, 2617
 - nppsArctan_64f_I, 2618
- signal_L1_norm
 - nppsNorm_L1_16s32f, 2762
 - nppsNorm_L1_16s32s_Sfs, 2762
 - nppsNorm_L1_16s64s_Sfs, 2762
 - nppsNorm_L1_32f, 2763
 - nppsNorm_L1_32fc64f, 2763
 - nppsNorm_L1_64f, 2763
 - nppsNorm_L1_64fc64f, 2764
 - nppsNormL1GetBufferSize_16s32f, 2764
 - nppsNormL1GetBufferSize_16s32s_Sfs, 2764
 - nppsNormL1GetBufferSize_16s64s_Sfs, 2764
 - nppsNormL1GetBufferSize_32f, 2765
 - nppsNormL1GetBufferSize_32fc64f, 2765
 - nppsNormL1GetBufferSize_64f, 2765
 - nppsNormL1GetBufferSize_64fc64f, 2765
- signal_L1_norm_diff
 - nppsNormDiff_L1_16s32f, 2779
 - nppsNormDiff_L1_16s32s_Sfs, 2779
 - nppsNormDiff_L1_16s64s_Sfs, 2779
 - nppsNormDiff_L1_32f, 2780
 - nppsNormDiff_L1_32fc64f, 2780
 - nppsNormDiff_L1_64f, 2780
 - nppsNormDiff_L1_64fc64f, 2781
 - nppsNormDiffL1GetBufferSize_16s32f, 2781

- nppsNormDiffL1GetBufferSize_16s32s_Sfs, 2781
- nppsNormDiffL1GetBufferSize_16s64s_Sfs, 2782
- nppsNormDiffL1GetBufferSize_32f, 2782
- nppsNormDiffL1GetBufferSize_32fc64f, 2782
- nppsNormDiffL1GetBufferSize_64f, 2782
- nppsNormDiffL1GetBufferSize_64fc64f, 2783
- signal_L2_norm
 - nppsNorm_L2_16s32f, 2768
 - nppsNorm_L2_16s32s_Sfs, 2768
 - nppsNorm_L2_32f, 2768
 - nppsNorm_L2_32fc64f, 2769
 - nppsNorm_L2_64f, 2769
 - nppsNorm_L2_64fc64f, 2769
 - nppsNorm_L2Sqr_16s64s_Sfs, 2770
 - nppsNormL2GetBufferSize_16s32f, 2770
 - nppsNormL2GetBufferSize_16s32s_Sfs, 2770
 - nppsNormL2GetBufferSize_32f, 2770
 - nppsNormL2GetBufferSize_32fc64f, 2771
 - nppsNormL2GetBufferSize_64f, 2771
 - nppsNormL2GetBufferSize_64fc64f, 2771
 - nppsNormL2SqrGetBufferSize_16s64s_Sfs, 2771
- signal_L2_norm_diff
 - nppsNormDiff_L2_16s32f, 2785
 - nppsNormDiff_L2_16s32s_Sfs, 2785
 - nppsNormDiff_L2_32f, 2785
 - nppsNormDiff_L2_32fc64f, 2786
 - nppsNormDiff_L2_64f, 2786
 - nppsNormDiff_L2_64fc64f, 2786
 - nppsNormDiff_L2Sqr_16s64s_Sfs, 2787
 - nppsNormDiffL2GetBufferSize_16s32f, 2787
 - nppsNormDiffL2GetBufferSize_16s32s_Sfs, 2787
 - nppsNormDiffL2GetBufferSize_32f, 2788
 - nppsNormDiffL2GetBufferSize_32fc64f, 2788
 - nppsNormDiffL2GetBufferSize_64f, 2788
 - nppsNormDiffL2GetBufferSize_64fc64f, 2788
 - nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs, 2789
- signal_ln
 - nppsLn_16s_ISfs, 2608
 - nppsLn_16s_Sfs, 2609
 - nppsLn_32f, 2609
 - nppsLn_32f_I, 2609
 - nppsLn_32s16s_Sfs, 2609
 - nppsLn_32s_ISfs, 2610
 - nppsLn_32s_Sfs, 2610
 - nppsLn_64f, 2610
 - nppsLn_64f32f, 2611
 - nppsLn_64f_I, 2611
- signal_lshiftc
 - nppsLShiftC_16s, 2646
 - nppsLShiftC_16s_I, 2647
 - nppsLShiftC_16u, 2647
 - nppsLShiftC_16u_I, 2647
 - nppsLShiftC_32s, 2647
 - nppsLShiftC_32s_I, 2648
 - nppsLShiftC_32u, 2648
 - nppsLShiftC_32u_I, 2648
 - nppsLShiftC_8u, 2649
 - nppsLShiftC_8u_I, 2649
- signal_malloc
 - nppsMalloc_16s, 2861
 - nppsMalloc_16sc, 2861
 - nppsMalloc_16u, 2861
 - nppsMalloc_32f, 2861
 - nppsMalloc_32fc, 2862
 - nppsMalloc_32s, 2862
 - nppsMalloc_32sc, 2862
 - nppsMalloc_32u, 2862
 - nppsMalloc_64f, 2863
 - nppsMalloc_64fc, 2863
 - nppsMalloc_64s, 2863
 - nppsMalloc_64sc, 2863
 - nppsMalloc_8s, 2864
 - nppsMalloc_8u, 2864
- signal_max
 - nppsMax_16s, 2712
 - nppsMax_32f, 2713
 - nppsMax_32s, 2713
 - nppsMax_64f, 2713
 - nppsMaxAbs_16s, 2714
 - nppsMaxAbs_32s, 2714
 - nppsMaxAbsGetBufferSize_16s, 2714
 - nppsMaxAbsGetBufferSize_32s, 2715
 - nppsMaxAbsIndx_16s, 2715
 - nppsMaxAbsIndx_32s, 2715
 - nppsMaxAbsIndxGetBufferSize_16s, 2716
 - nppsMaxAbsIndxGetBufferSize_32s, 2716
 - nppsMaxGetBufferSize_16s, 2716
 - nppsMaxGetBufferSize_32f, 2716
 - nppsMaxGetBufferSize_32s, 2717
 - nppsMaxGetBufferSize_64f, 2717
 - nppsMaxIndx_16s, 2717
 - nppsMaxIndx_32f, 2718
 - nppsMaxIndx_32s, 2718
 - nppsMaxIndx_64f, 2718
 - nppsMaxIndxGetBufferSize_16s, 2719
 - nppsMaxIndxGetBufferSize_32f, 2719
 - nppsMaxIndxGetBufferSize_32s, 2719
 - nppsMaxIndxGetBufferSize_64f, 2720
- signal_maximum_error

- nppsMaximumError_16s, [2815](#)
- nppsMaximumError_16sc, [2815](#)
- nppsMaximumError_16u, [2815](#)
- nppsMaximumError_32f, [2816](#)
- nppsMaximumError_32fc, [2816](#)
- nppsMaximumError_32s, [2816](#)
- nppsMaximumError_32sc, [2817](#)
- nppsMaximumError_32u, [2817](#)
- nppsMaximumError_64f, [2817](#)
- nppsMaximumError_64fc, [2818](#)
- nppsMaximumError_64s, [2818](#)
- nppsMaximumError_64sc, [2818](#)
- nppsMaximumError_8s, [2819](#)
- nppsMaximumError_8u, [2819](#)
- nppsMaximumErrorGetBufferSize_16s, [2819](#)
- nppsMaximumErrorGetBufferSize_16sc, [2820](#)
- nppsMaximumErrorGetBufferSize_16u, [2820](#)
- nppsMaximumErrorGetBufferSize_32f, [2820](#)
- nppsMaximumErrorGetBufferSize_32fc, [2820](#)
- nppsMaximumErrorGetBufferSize_32s, [2821](#)
- nppsMaximumErrorGetBufferSize_32sc, [2821](#)
- nppsMaximumErrorGetBufferSize_32u, [2821](#)
- nppsMaximumErrorGetBufferSize_64f, [2821](#)
- nppsMaximumErrorGetBufferSize_64fc, [2822](#)
- nppsMaximumErrorGetBufferSize_64s, [2822](#)
- nppsMaximumErrorGetBufferSize_64sc, [2822](#)
- nppsMaximumErrorGetBufferSize_8s, [2822](#)
- nppsMaximumErrorGetBufferSize_8u, [2823](#)
- signal_maximum_relative_error
 - nppsMaximumRelativeError_16s, [2837](#)
 - nppsMaximumRelativeError_16sc, [2837](#)
 - nppsMaximumRelativeError_16u, [2838](#)
 - nppsMaximumRelativeError_32f, [2838](#)
 - nppsMaximumRelativeError_32fc, [2838](#)
 - nppsMaximumRelativeError_32s, [2839](#)
 - nppsMaximumRelativeError_32sc, [2839](#)
 - nppsMaximumRelativeError_32u, [2840](#)
 - nppsMaximumRelativeError_64f, [2840](#)
 - nppsMaximumRelativeError_64fc, [2840](#)
 - nppsMaximumRelativeError_64s, [2841](#)
 - nppsMaximumRelativeError_64sc, [2841](#)
 - nppsMaximumRelativeError_8s, [2842](#)
 - nppsMaximumRelativeError_8u, [2842](#)
 - nppsMaximumRelativeErrorGetBufferSize_-16s, [2842](#)
 - nppsMaximumRelativeErrorGetBufferSize_-16sc, [2843](#)
 - nppsMaximumRelativeErrorGetBufferSize_-16u, [2843](#)
 - nppsMaximumRelativeErrorGetBufferSize_-32f, [2843](#)
 - nppsMaximumRelativeErrorGetBufferSize_-32fc, [2843](#)
 - nppsMaximumRelativeErrorGetBufferSize_-32s, [2844](#)
 - nppsMaximumRelativeErrorGetBufferSize_-32sc, [2844](#)
 - nppsMaximumRelativeErrorGetBufferSize_-32u, [2844](#)
 - nppsMaximumRelativeErrorGetBufferSize_-64f, [2844](#)
 - nppsMaximumRelativeErrorGetBufferSize_-64fc, [2845](#)
 - nppsMaximumRelativeErrorGetBufferSize_-64s, [2845](#)
 - nppsMaximumRelativeErrorGetBufferSize_-64sc, [2845](#)
 - nppsMaximumRelativeErrorGetBufferSize_-8s, [2845](#)
 - nppsMaximumRelativeErrorGetBufferSize_-8u, [2846](#)
- signal_mean
 - nppsMean_16s_Sfs, [2732](#)
 - nppsMean_16sc_Sfs, [2732](#)
 - nppsMean_32f, [2732](#)
 - nppsMean_32fc, [2733](#)
 - nppsMean_32s_Sfs, [2733](#)
 - nppsMean_64f, [2733](#)
 - nppsMean_64fc, [2734](#)
 - nppsMeanGetBufferSize_16s_Sfs, [2734](#)
 - nppsMeanGetBufferSize_16sc_Sfs, [2734](#)
 - nppsMeanGetBufferSize_32f, [2735](#)
 - nppsMeanGetBufferSize_32fc, [2735](#)
 - nppsMeanGetBufferSize_32s_Sfs, [2735](#)
 - nppsMeanGetBufferSize_64f, [2735](#)
 - nppsMeanGetBufferSize_64fc, [2736](#)
- signal_mean_and_standard_deviation
 - nppsMeanStdDev_16s32s_Sfs, [2740](#)
 - nppsMeanStdDev_16s_Sfs, [2741](#)
 - nppsMeanStdDev_32f, [2741](#)
 - nppsMeanStdDev_64f, [2741](#)
 - nppsMeanStdDevGetBufferSize_16s32s_Sfs, [2742](#)
 - nppsMeanStdDevGetBufferSize_16s_Sfs, [2742](#)
 - nppsMeanStdDevGetBufferSize_32f, [2742](#)
 - nppsMeanStdDevGetBufferSize_64f, [2742](#)
- signal_min
 - nppsMin_16s, [2722](#)
 - nppsMin_32f, [2723](#)
 - nppsMin_32s, [2723](#)
 - nppsMin_64f, [2723](#)
 - nppsMinAbs_16s, [2724](#)
 - nppsMinAbs_32s, [2724](#)
 - nppsMinAbsGetBufferSize_16s, [2724](#)
 - nppsMinAbsGetBufferSize_32s, [2725](#)
 - nppsMinAbsIdx_16s, [2725](#)

- nppsMinAbsIdx_32s, [2725](#)
- nppsMinAbsIdxGetBufferSize_16s, [2726](#)
- nppsMinAbsIdxGetBufferSize_32s, [2726](#)
- nppsMinGetBufferSize_16s, [2726](#)
- nppsMinGetBufferSize_32f, [2726](#)
- nppsMinGetBufferSize_32s, [2727](#)
- nppsMinGetBufferSize_64f, [2727](#)
- nppsMinIdx_16s, [2727](#)
- nppsMinIdx_32f, [2728](#)
- nppsMinIdx_32s, [2728](#)
- nppsMinIdx_64f, [2728](#)
- nppsMinIdxGetBufferSize_16s, [2729](#)
- nppsMinIdxGetBufferSize_32f, [2729](#)
- nppsMinIdxGetBufferSize_32s, [2729](#)
- nppsMinIdxGetBufferSize_64f, [2730](#)
- signal_min_every_or_max_every
 - nppsMaxEvery_16s_I, [2700](#)
 - nppsMaxEvery_16u_I, [2701](#)
 - nppsMaxEvery_32f_I, [2701](#)
 - nppsMaxEvery_32s_I, [2701](#)
 - nppsMaxEvery_8u_I, [2701](#)
 - nppsMinEvery_16s_I, [2702](#)
 - nppsMinEvery_16u_I, [2702](#)
 - nppsMinEvery_32f_I, [2702](#)
 - nppsMinEvery_32s_I, [2703](#)
 - nppsMinEvery_64f_I, [2703](#)
 - nppsMinEvery_8u_I, [2703](#)
- signal_min_max
 - nppsMinMax_16s, [2746](#)
 - nppsMinMax_16u, [2746](#)
 - nppsMinMax_32f, [2746](#)
 - nppsMinMax_32s, [2747](#)
 - nppsMinMax_32u, [2747](#)
 - nppsMinMax_64f, [2747](#)
 - nppsMinMax_8u, [2748](#)
 - nppsMinMaxGetBufferSize_16s, [2748](#)
 - nppsMinMaxGetBufferSize_16u, [2748](#)
 - nppsMinMaxGetBufferSize_32f, [2749](#)
 - nppsMinMaxGetBufferSize_32s, [2749](#)
 - nppsMinMaxGetBufferSize_32u, [2749](#)
 - nppsMinMaxGetBufferSize_64f, [2749](#)
 - nppsMinMaxGetBufferSize_8u, [2750](#)
 - nppsMinMaxIdx_16s, [2750](#)
 - nppsMinMaxIdx_16u, [2750](#)
 - nppsMinMaxIdx_32f, [2751](#)
 - nppsMinMaxIdx_32s, [2751](#)
 - nppsMinMaxIdx_32u, [2752](#)
 - nppsMinMaxIdx_64f, [2752](#)
 - nppsMinMaxIdx_8u, [2752](#)
 - nppsMinMaxIdxGetBufferSize_16s, [2753](#)
 - nppsMinMaxIdxGetBufferSize_16u, [2753](#)
 - nppsMinMaxIdxGetBufferSize_32f, [2753](#)
 - nppsMinMaxIdxGetBufferSize_32s, [2754](#)
 - nppsMinMaxIdxGetBufferSize_32u, [2754](#)
 - nppsMinMaxIdxGetBufferSize_64f, [2754](#)
 - nppsMinMaxIdxGetBufferSize_8u, [2754](#)
- signal_mul
 - nppsMul_16s, [2554](#)
 - nppsMul_16s32f, [2554](#)
 - nppsMul_16s32s_Sfs, [2555](#)
 - nppsMul_16s_I, [2555](#)
 - nppsMul_16s_ISfs, [2555](#)
 - nppsMul_16s_Sfs, [2556](#)
 - nppsMul_16sc_ISfs, [2556](#)
 - nppsMul_16sc_Sfs, [2556](#)
 - nppsMul_16u16s_Sfs, [2557](#)
 - nppsMul_16u_ISfs, [2557](#)
 - nppsMul_16u_Sfs, [2557](#)
 - nppsMul_32f, [2558](#)
 - nppsMul_32f32fc, [2558](#)
 - nppsMul_32f32fc_I, [2558](#)
 - nppsMul_32f_I, [2559](#)
 - nppsMul_32fc, [2559](#)
 - nppsMul_32fc_I, [2559](#)
 - nppsMul_32s32sc_ISfs, [2560](#)
 - nppsMul_32s32sc_Sfs, [2560](#)
 - nppsMul_32s_ISfs, [2560](#)
 - nppsMul_32s_Sfs, [2561](#)
 - nppsMul_32sc_ISfs, [2561](#)
 - nppsMul_32sc_Sfs, [2561](#)
 - nppsMul_64f, [2562](#)
 - nppsMul_64f_I, [2562](#)
 - nppsMul_64fc, [2562](#)
 - nppsMul_64fc_I, [2563](#)
 - nppsMul_8u16u, [2563](#)
 - nppsMul_8u_ISfs, [2563](#)
 - nppsMul_8u_Sfs, [2564](#)
 - nppsMul_Low_32s_Sfs, [2564](#)
- signal_mulc
 - nppsMulC_16s_ISfs, [2500](#)
 - nppsMulC_16s_Sfs, [2501](#)
 - nppsMulC_16sc_ISfs, [2501](#)
 - nppsMulC_16sc_Sfs, [2501](#)
 - nppsMulC_16u_ISfs, [2502](#)
 - nppsMulC_16u_Sfs, [2502](#)
 - nppsMulC_32f, [2502](#)
 - nppsMulC_32f16s_Sfs, [2503](#)
 - nppsMulC_32f_I, [2503](#)
 - nppsMulC_32fc, [2503](#)
 - nppsMulC_32fc_I, [2504](#)
 - nppsMulC_32s_ISfs, [2504](#)
 - nppsMulC_32s_Sfs, [2504](#)
 - nppsMulC_32sc_ISfs, [2505](#)
 - nppsMulC_32sc_Sfs, [2505](#)
 - nppsMulC_64f, [2505](#)
 - nppsMulC_64f64s_ISfs, [2506](#)
 - nppsMulC_64f_I, [2506](#)
 - nppsMulC_64fc, [2506](#)

- nppsMulC_64fc_I, [2507](#)
- nppsMulC_8u_ISfs, [2507](#)
- nppsMulC_8u_Sfs, [2507](#)
- nppsMulC_Low_32f16s, [2508](#)
- signal_normalize
 - nppsNormalize_16s_Sfs, [2619](#)
 - nppsNormalize_16sc_Sfs, [2620](#)
 - nppsNormalize_32f, [2620](#)
 - nppsNormalize_32fc, [2620](#)
 - nppsNormalize_64f, [2621](#)
 - nppsNormalize_64fc, [2621](#)
- signal_not
 - nppsNot_16u, [2643](#)
 - nppsNot_16u_I, [2643](#)
 - nppsNot_32u, [2644](#)
 - nppsNot_32u_I, [2644](#)
 - nppsNot_8u, [2644](#)
 - nppsNot_8u_I, [2644](#)
- signal_or
 - nppsOr_16u, [2634](#)
 - nppsOr_16u_I, [2634](#)
 - nppsOr_32u, [2635](#)
 - nppsOr_32u_I, [2635](#)
 - nppsOr_8u, [2635](#)
 - nppsOr_8u_I, [2636](#)
- signal_orc
 - nppsOrC_16u, [2631](#)
 - nppsOrC_16u_I, [2631](#)
 - nppsOrC_32u, [2632](#)
 - nppsOrC_32u_I, [2632](#)
 - nppsOrC_8u, [2632](#)
 - nppsOrC_8u_I, [2633](#)
- signal_rshifc
 - nppsRShiftC_16s, [2650](#)
 - nppsRShiftC_16s_I, [2651](#)
 - nppsRShiftC_16u, [2651](#)
 - nppsRShiftC_16u_I, [2651](#)
 - nppsRShiftC_32s, [2651](#)
 - nppsRShiftC_32s_I, [2652](#)
 - nppsRShiftC_32u, [2652](#)
 - nppsRShiftC_32u_I, [2652](#)
 - nppsRShiftC_8u, [2653](#)
 - nppsRShiftC_8u_I, [2653](#)
- signal_set
 - nppsSet_16s, [2687](#)
 - nppsSet_16sc, [2687](#)
 - nppsSet_16u, [2687](#)
 - nppsSet_32f, [2687](#)
 - nppsSet_32fc, [2688](#)
 - nppsSet_32s, [2688](#)
 - nppsSet_32sc, [2688](#)
 - nppsSet_32u, [2688](#)
 - nppsSet_64f, [2689](#)
 - nppsSet_64fc, [2689](#)
 - nppsSet_64s, [2689](#)
 - nppsSet_64sc, [2690](#)
 - nppsSet_8s, [2690](#)
 - nppsSet_8u, [2690](#)
- signal_sqrt
 - nppsSqrt_16s_ISfs, [2596](#)
 - nppsSqrt_16s_Sfs, [2596](#)
 - nppsSqrt_16sc_ISfs, [2597](#)
 - nppsSqrt_16sc_Sfs, [2597](#)
 - nppsSqrt_16u_ISfs, [2597](#)
 - nppsSqrt_16u_Sfs, [2597](#)
 - nppsSqrt_32f, [2598](#)
 - nppsSqrt_32f_I, [2598](#)
 - nppsSqrt_32fc, [2598](#)
 - nppsSqrt_32fc_I, [2599](#)
 - nppsSqrt_32s16s_Sfs, [2599](#)
 - nppsSqrt_64f, [2599](#)
 - nppsSqrt_64f_I, [2599](#)
 - nppsSqrt_64fc, [2600](#)
 - nppsSqrt_64fc_I, [2600](#)
 - nppsSqrt_64s16s_Sfs, [2600](#)
 - nppsSqrt_64s_ISfs, [2600](#)
 - nppsSqrt_64s_Sfs, [2601](#)
 - nppsSqrt_8u_ISfs, [2601](#)
 - nppsSqrt_8u_Sfs, [2601](#)
- signal_square
 - nppsSqr_16s_ISfs, [2590](#)
 - nppsSqr_16s_Sfs, [2590](#)
 - nppsSqr_16sc_ISfs, [2590](#)
 - nppsSqr_16sc_Sfs, [2591](#)
 - nppsSqr_16u_ISfs, [2591](#)
 - nppsSqr_16u_Sfs, [2591](#)
 - nppsSqr_32f, [2591](#)
 - nppsSqr_32f_I, [2592](#)
 - nppsSqr_32fc, [2592](#)
 - nppsSqr_32fc_I, [2592](#)
 - nppsSqr_64f, [2592](#)
 - nppsSqr_64f_I, [2593](#)
 - nppsSqr_64fc, [2593](#)
 - nppsSqr_64fc_I, [2593](#)
 - nppsSqr_8u_ISfs, [2593](#)
 - nppsSqr_8u_Sfs, [2594](#)
- signal_standard_deviation
 - nppsStdDev_16s32s_Sfs, [2737](#)
 - nppsStdDev_16s_Sfs, [2737](#)
 - nppsStdDev_32f, [2738](#)
 - nppsStdDev_64f, [2738](#)
 - nppsStdDevGetBufferSize_16s32s_Sfs, [2738](#)
 - nppsStdDevGetBufferSize_16s_Sfs, [2739](#)
 - nppsStdDevGetBufferSize_32f, [2739](#)
 - nppsStdDevGetBufferSize_64f, [2739](#)
- signal_sub
 - nppsSub_16s, [2566](#)
 - nppsSub_16s32f, [2567](#)

- nppsSub_16s_I, [2567](#)
- nppsSub_16s_ISfs, [2567](#)
- nppsSub_16s_Sfs, [2568](#)
- nppsSub_16sc_ISfs, [2568](#)
- nppsSub_16sc_Sfs, [2568](#)
- nppsSub_16u_ISfs, [2569](#)
- nppsSub_16u_Sfs, [2569](#)
- nppsSub_32f, [2569](#)
- nppsSub_32f_I, [2570](#)
- nppsSub_32fc, [2570](#)
- nppsSub_32fc_I, [2570](#)
- nppsSub_32s_ISfs, [2570](#)
- nppsSub_32s_Sfs, [2571](#)
- nppsSub_32sc_ISfs, [2571](#)
- nppsSub_32sc_Sfs, [2571](#)
- nppsSub_64f, [2572](#)
- nppsSub_64f_I, [2572](#)
- nppsSub_64fc, [2572](#)
- nppsSub_64fc_I, [2573](#)
- nppsSub_8u_ISfs, [2573](#)
- nppsSub_8u_Sfs, [2573](#)
- signal_subc
 - nppsSubC_16s_ISfs, [2510](#)
 - nppsSubC_16s_Sfs, [2510](#)
 - nppsSubC_16sc_ISfs, [2511](#)
 - nppsSubC_16sc_Sfs, [2511](#)
 - nppsSubC_16u_ISfs, [2511](#)
 - nppsSubC_16u_Sfs, [2512](#)
 - nppsSubC_32f, [2512](#)
 - nppsSubC_32f_I, [2512](#)
 - nppsSubC_32fc, [2513](#)
 - nppsSubC_32fc_I, [2513](#)
 - nppsSubC_32s_ISfs, [2513](#)
 - nppsSubC_32s_Sfs, [2514](#)
 - nppsSubC_32sc_ISfs, [2514](#)
 - nppsSubC_32sc_Sfs, [2514](#)
 - nppsSubC_64f, [2515](#)
 - nppsSubC_64f_I, [2515](#)
 - nppsSubC_64fc, [2515](#)
 - nppsSubC_64fc_I, [2516](#)
 - nppsSubC_8u_ISfs, [2516](#)
 - nppsSubC_8u_Sfs, [2516](#)
- signal_subcrev
 - nppsSubCRev_16s_ISfs, [2519](#)
 - nppsSubCRev_16s_Sfs, [2520](#)
 - nppsSubCRev_16sc_ISfs, [2520](#)
 - nppsSubCRev_16sc_Sfs, [2520](#)
 - nppsSubCRev_16u_ISfs, [2521](#)
 - nppsSubCRev_16u_Sfs, [2521](#)
 - nppsSubCRev_32f, [2521](#)
 - nppsSubCRev_32f_I, [2522](#)
 - nppsSubCRev_32fc, [2522](#)
 - nppsSubCRev_32fc_I, [2522](#)
 - nppsSubCRev_32s_ISfs, [2522](#)
 - nppsSubCRev_32s_Sfs, [2523](#)
 - nppsSubCRev_32sc_ISfs, [2523](#)
 - nppsSubCRev_32sc_Sfs, [2523](#)
 - nppsSubCRev_64f, [2524](#)
 - nppsSubCRev_64f_I, [2524](#)
 - nppsSubCRev_64fc, [2524](#)
 - nppsSubCRev_64fc_I, [2525](#)
 - nppsSubCRev_8u_ISfs, [2525](#)
 - nppsSubCRev_8u_Sfs, [2525](#)
- signal_sum
 - nppsSum_16s32s_Sfs, [2705](#)
 - nppsSum_16s_Sfs, [2705](#)
 - nppsSum_16sc32sc_Sfs, [2706](#)
 - nppsSum_16sc_Sfs, [2706](#)
 - nppsSum_32f, [2706](#)
 - nppsSum_32fc, [2707](#)
 - nppsSum_32s_Sfs, [2707](#)
 - nppsSum_64f, [2707](#)
 - nppsSum_64fc, [2708](#)
 - nppsSumGetBufferSize_16s32s_Sfs, [2708](#)
 - nppsSumGetBufferSize_16s_Sfs, [2708](#)
 - nppsSumGetBufferSize_16sc32sc_Sfs, [2709](#)
 - nppsSumGetBufferSize_16sc_Sfs, [2709](#)
 - nppsSumGetBufferSize_32f, [2709](#)
 - nppsSumGetBufferSize_32fc, [2709](#)
 - nppsSumGetBufferSize_32s_Sfs, [2710](#)
 - nppsSumGetBufferSize_64f, [2710](#)
 - nppsSumGetBufferSize_64fc, [2710](#)
- signal_sumln
 - nppsSumLn_16s32f, [2613](#)
 - nppsSumLn_32f, [2614](#)
 - nppsSumLn_32f64f, [2614](#)
 - nppsSumLn_64f, [2614](#)
 - nppsSumLnGetBufferSize_16s32f, [2615](#)
 - nppsSumLnGetBufferSize_32f, [2615](#)
 - nppsSumLnGetBufferSize_32f64f, [2615](#)
 - nppsSumLnGetBufferSize_64f, [2615](#)
- signal_threshold
 - nppsThreshold_16s, [2662](#)
 - nppsThreshold_16s_I, [2663](#)
 - nppsThreshold_16sc, [2663](#)
 - nppsThreshold_16sc_I, [2663](#)
 - nppsThreshold_32f, [2664](#)
 - nppsThreshold_32f_I, [2664](#)
 - nppsThreshold_32fc, [2664](#)
 - nppsThreshold_32fc_I, [2665](#)
 - nppsThreshold_64f, [2665](#)
 - nppsThreshold_64f_I, [2665](#)
 - nppsThreshold_64fc, [2666](#)
 - nppsThreshold_64fc_I, [2666](#)
 - nppsThreshold_GT_16s, [2666](#)
 - nppsThreshold_GT_16s_I, [2667](#)
 - nppsThreshold_GT_16sc, [2667](#)
 - nppsThreshold_GT_16sc_I, [2667](#)

- nppsThreshold_GT_32f, [2668](#)
- nppsThreshold_GT_32f_I, [2668](#)
- nppsThreshold_GT_32fc, [2668](#)
- nppsThreshold_GT_32fc_I, [2669](#)
- nppsThreshold_GT_64f, [2669](#)
- nppsThreshold_GT_64f_I, [2669](#)
- nppsThreshold_GT_64fc, [2670](#)
- nppsThreshold_GT_64fc_I, [2670](#)
- nppsThreshold_GTVa1_16s, [2670](#)
- nppsThreshold_GTVa1_16s_I, [2671](#)
- nppsThreshold_GTVa1_16sc, [2671](#)
- nppsThreshold_GTVa1_16sc_I, [2671](#)
- nppsThreshold_GTVa1_32f, [2672](#)
- nppsThreshold_GTVa1_32f_I, [2672](#)
- nppsThreshold_GTVa1_32fc, [2672](#)
- nppsThreshold_GTVa1_32fc_I, [2673](#)
- nppsThreshold_GTVa1_64f, [2673](#)
- nppsThreshold_GTVa1_64f_I, [2673](#)
- nppsThreshold_GTVa1_64fc, [2674](#)
- nppsThreshold_GTVa1_64fc_I, [2674](#)
- nppsThreshold_LT_16s, [2674](#)
- nppsThreshold_LT_16s_I, [2675](#)
- nppsThreshold_LT_16sc, [2675](#)
- nppsThreshold_LT_16sc_I, [2675](#)
- nppsThreshold_LT_32f, [2676](#)
- nppsThreshold_LT_32f_I, [2676](#)
- nppsThreshold_LT_32fc, [2676](#)
- nppsThreshold_LT_32fc_I, [2677](#)
- nppsThreshold_LT_64f, [2677](#)
- nppsThreshold_LT_64f_I, [2677](#)
- nppsThreshold_LT_64fc, [2678](#)
- nppsThreshold_LT_64fc_I, [2678](#)
- nppsThreshold_LTVa1_16s, [2678](#)
- nppsThreshold_LTVa1_16s_I, [2679](#)
- nppsThreshold_LTVa1_16sc, [2679](#)
- nppsThreshold_LTVa1_16sc_I, [2679](#)
- nppsThreshold_LTVa1_32f, [2680](#)
- nppsThreshold_LTVa1_32f_I, [2680](#)
- nppsThreshold_LTVa1_32fc, [2680](#)
- nppsThreshold_LTVa1_32fc_I, [2681](#)
- nppsThreshold_LTVa1_64f, [2681](#)
- nppsThreshold_LTVa1_64f_I, [2681](#)
- nppsThreshold_LTVa1_64fc, [2682](#)
- nppsThreshold_LTVa1_64fc_I, [2682](#)
- signal_xor
 - nppsXor_16u, [2640](#)
 - nppsXor_16u_I, [2640](#)
 - nppsXor_32u, [2641](#)
 - nppsXor_32u_I, [2641](#)
 - nppsXor_8u, [2641](#)
 - nppsXor_8u_I, [2642](#)
- signal_xorc
 - nppsXorC_16u, [2637](#)
 - nppsXorC_16u_I, [2637](#)
 - nppsXorC_32u, [2638](#)
 - nppsXorC_32u_I, [2638](#)
 - nppsXorC_8u, [2638](#)
 - nppsXorC_8u_I, [2639](#)
- signal_zero
 - nppsZero_16s, [2691](#)
 - nppsZero_16sc, [2692](#)
 - nppsZero_32f, [2692](#)
 - nppsZero_32fc, [2692](#)
 - nppsZero_32s, [2692](#)
 - nppsZero_32sc, [2692](#)
 - nppsZero_64f, [2693](#)
 - nppsZero_64fc, [2693](#)
 - nppsZero_64s, [2693](#)
 - nppsZero_64sc, [2693](#)
 - nppsZero_8u, [2694](#)
- Sqr, [331](#), [2589](#)
- SqrDistanceFull_Norm, [2128](#)
- sqrdistancefullnorm
 - nppiSqrDistanceFull_Norm_16u32f_AC4R, [2130](#)
 - nppiSqrDistanceFull_Norm_16u32f_C1R, [2130](#)
 - nppiSqrDistanceFull_Norm_16u32f_C3R, [2130](#)
 - nppiSqrDistanceFull_Norm_16u32f_C4R, [2131](#)
 - nppiSqrDistanceFull_Norm_32f_AC4R, [2131](#)
 - nppiSqrDistanceFull_Norm_32f_C1R, [2132](#)
 - nppiSqrDistanceFull_Norm_32f_C3R, [2132](#)
 - nppiSqrDistanceFull_Norm_32f_C4R, [2133](#)
 - nppiSqrDistanceFull_Norm_8s32f_AC4R, [2133](#)
 - nppiSqrDistanceFull_Norm_8s32f_C1R, [2133](#)
 - nppiSqrDistanceFull_Norm_8s32f_C3R, [2134](#)
 - nppiSqrDistanceFull_Norm_8s32f_C4R, [2134](#)
 - nppiSqrDistanceFull_Norm_8u32f_AC4R, [2135](#)
 - nppiSqrDistanceFull_Norm_8u32f_C1R, [2135](#)
 - nppiSqrDistanceFull_Norm_8u32f_C3R, [2136](#)
 - nppiSqrDistanceFull_Norm_8u32f_C4R, [2136](#)
 - nppiSqrDistanceFull_Norm_8u_AC4RSfs, [2136](#)
 - nppiSqrDistanceFull_Norm_8u_C1RSfs, [2137](#)
 - nppiSqrDistanceFull_Norm_8u_C3RSfs, [2137](#)
 - nppiSqrDistanceFull_Norm_8u_C4RSfs, [2138](#)
- SqrDistanceSame_Norm, [2139](#)
- sqrdistancesamenorm
 - nppiSqrDistanceSame_Norm_16u32f_AC4R, [2141](#)
 - nppiSqrDistanceSame_Norm_16u32f_C1R, [2141](#)
 - nppiSqrDistanceSame_Norm_16u32f_C3R, [2142](#)

- [nppiSqrDistanceSame_Norm_16u32f_C4R, 2142](#)
- [nppiSqrDistanceSame_Norm_32f_AC4R, 2142](#)
- [nppiSqrDistanceSame_Norm_32f_C1R, 2143](#)
- [nppiSqrDistanceSame_Norm_32f_C3R, 2143](#)
- [nppiSqrDistanceSame_Norm_32f_C4R, 2144](#)
- [nppiSqrDistanceSame_Norm_8s32f_AC4R, 2144](#)
- [nppiSqrDistanceSame_Norm_8s32f_C1R, 2145](#)
- [nppiSqrDistanceSame_Norm_8s32f_C3R, 2145](#)
- [nppiSqrDistanceSame_Norm_8s32f_C4R, 2145](#)
- [nppiSqrDistanceSame_Norm_8u32f_AC4R, 2146](#)
- [nppiSqrDistanceSame_Norm_8u32f_C1R, 2146](#)
- [nppiSqrDistanceSame_Norm_8u32f_C3R, 2147](#)
- [nppiSqrDistanceSame_Norm_8u32f_C4R, 2147](#)
- [nppiSqrDistanceSame_Norm_8u_AC4RSfs, 2148](#)
- [nppiSqrDistanceSame_Norm_8u_C1RSfs, 2148](#)
- [nppiSqrDistanceSame_Norm_8u_C3RSfs, 2149](#)
- [nppiSqrDistanceSame_Norm_8u_C4RSfs, 2149](#)
- [SqrDistanceValid_Norm, 2150](#)
- [sqrdistancevalidnorm](#)
 - [nppiSqrDistanceValid_Norm_16u32f_AC4R, 2152](#)
 - [nppiSqrDistanceValid_Norm_16u32f_C1R, 2152](#)
 - [nppiSqrDistanceValid_Norm_16u32f_C3R, 2153](#)
 - [nppiSqrDistanceValid_Norm_16u32f_C4R, 2153](#)
 - [nppiSqrDistanceValid_Norm_32f_AC4R, 2153](#)
 - [nppiSqrDistanceValid_Norm_32f_C1R, 2154](#)
 - [nppiSqrDistanceValid_Norm_32f_C3R, 2154](#)
 - [nppiSqrDistanceValid_Norm_32f_C4R, 2155](#)
 - [nppiSqrDistanceValid_Norm_8s32f_AC4R, 2155](#)
 - [nppiSqrDistanceValid_Norm_8s32f_C1R, 2156](#)
 - [nppiSqrDistanceValid_Norm_8s32f_C3R, 2156](#)
 - [nppiSqrDistanceValid_Norm_8s32f_C4R, 2156](#)
 - [nppiSqrDistanceValid_Norm_8u32f_AC4R, 2157](#)
 - [nppiSqrDistanceValid_Norm_8u32f_C1R, 2157](#)
 - [nppiSqrDistanceValid_Norm_8u32f_C3R, 2158](#)
 - [nppiSqrDistanceValid_Norm_8u32f_C4R, 2158](#)
 - [nppiSqrDistanceValid_Norm_8u_AC4RSfs, 2159](#)
 - [nppiSqrDistanceValid_Norm_8u_C1RSfs, 2159](#)
 - [nppiSqrDistanceValid_Norm_8u_C3RSfs, 2160](#)
 - [nppiSqrDistanceValid_Norm_8u_C4RSfs, 2160](#)
- [SqrIntegral, 2090](#)
- [Sqrt, 345, 2595](#)
- [Standard Deviation, 2737](#)
- [Statistical Functions, 2699](#)
- [Statistical Operations, 1635](#)
- [Sub, 247, 2565](#)
- [SubC, 115, 2509](#)
- [SubCRev, 2518](#)
- [Sum, 1702, 2704](#)
- [SumLn, 2613](#)
- [Swap Channels, 942](#)
- [Threshold, 2658](#)
- [Threshold and Compare Operations, 2372](#)
- [Threshold Operations, 2373](#)
- [Transpose, 935](#)
- [typedefs_npp](#)
 - [NPP_AFFINE_QUAD_INCORRECT_WARNING, 47](#)
 - [NPP_ALG_HINT_ACCURATE, 42](#)
 - [NPP_ALG_HINT_FAST, 42](#)
 - [NPP_ALG_HINT_NONE, 42](#)
 - [NPP_ALIGNMENT_ERROR, 45](#)
 - [NPP_ANCHOR_ERROR, 46](#)
 - [NPP_BAD_ARGUMENT_ERROR, 46](#)
 - [NPP_BORDER_CONSTANT, 43](#)
 - [NPP_BORDER_MIRROR, 43](#)
 - [NPP_BORDER_NONE, 43](#)
 - [NPP_BORDER_REPLICATE, 43](#)
 - [NPP_BORDER_UNDEFINED, 43](#)
 - [NPP_BORDER_WRAP, 43](#)
 - [NPP_BOTH_AXIS, 43](#)
 - [NPP_CHANNEL_ERROR, 46](#)
 - [NPP_CHANNEL_ORDER_ERROR, 46](#)
 - [NPP_CMP_EQ, 41](#)
 - [NPP_CMP_GREATER, 41](#)
 - [NPP_CMP_GREATER_EQ, 41](#)
 - [NPP_CMP_LESS, 41](#)

- NPP_CMP_LESS_EQ, 41
- NPP_COEFFICIENT_ERROR, 46
- NPP_COI_ERROR, 46
- NPP_CONTEXT_MATCH_ERROR, 46
- NPP_CORRUPTED_DATA_ERROR, 46
- NPP_CUDA_1_0, 41
- NPP_CUDA_1_1, 42
- NPP_CUDA_1_2, 42
- NPP_CUDA_1_3, 42
- NPP_CUDA_2_0, 42
- NPP_CUDA_2_1, 42
- NPP_CUDA_3_0, 42
- NPP_CUDA_3_2, 42
- NPP_CUDA_3_5, 42
- NPP_CUDA_3_7, 42
- NPP_CUDA_5_0, 42
- NPP_CUDA_5_2, 42
- NPP_CUDA_5_3, 42
- NPP_CUDA_6_0, 42
- NPP_CUDA_KERNEL_EXECUTION_ERROR, 45
- NPP_CUDA_NOT_CAPABLE, 41
- NPP_CUDA_UNKNOWN_VERSION, 41
- NPP_DATA_TYPE_ERROR, 46
- NPP_DIVIDE_BY_ZERO_ERROR, 46
- NPP_DIVIDE_BY_ZERO_WARNING, 47
- NPP_DIVISOR_ERROR, 46
- NPP_DOUBLE_SIZE_WARNING, 47
- NPP_ERROR, 46
- NPP_ERROR_RESERVED, 46
- NPP_FFT_FLAG_ERROR, 46
- NPP_FFT_ORDER_ERROR, 46
- NPP_HAAR_CLASSIFIER_PIXEL_MATCH_ERROR, 45
- NPP_HISTOGRAM_NUMBER_OF_LEVELS_ERROR, 46
- NPP_HORIZONTAL_AXIS, 43
- NPP_INTERPOLATION_ERROR, 46
- NPP_INVALID_DEVICE_POINTER_ERROR, 45
- NPP_INVALID_HOST_POINTER_ERROR, 45
- NPP_LUT_NUMBER_OF_LEVELS_ERROR, 46
- NPP_LUT_PALETTE_BITSIZE_ERROR, 45
- NPP_MASK_SIZE_11_X_11, 44
- NPP_MASK_SIZE_13_X_13, 44
- NPP_MASK_SIZE_15_X_15, 44
- NPP_MASK_SIZE_1_X_3, 44
- NPP_MASK_SIZE_1_X_5, 44
- NPP_MASK_SIZE_3_X_1, 44
- NPP_MASK_SIZE_3_X_3, 44
- NPP_MASK_SIZE_5_X_1, 44
- NPP_MASK_SIZE_5_X_5, 44
- NPP_MASK_SIZE_7_X_7, 44
- NPP_MASK_SIZE_9_X_9, 44
- NPP_MASK_SIZE_ERROR, 46
- NPP_MEMCPY_ERROR, 45
- NPP_MEMFREE_ERROR, 45
- NPP_MEMORY_ALLOCATION_ERR, 46
- NPP_MEMSET_ERROR, 45
- NPP_MIRROR_FLIP_ERROR, 46
- NPP_MISALIGNED_DST_ROI_WARNING, 47
- NPP_MOMENT_00_ZERO_ERROR, 46
- NPP_NO_ERROR, 46
- NPP_NO_MEMORY_ERROR, 46
- NPP_NO_OPERATION_WARNING, 47
- NPP_NOT_EVEN_STEP_ERROR, 45
- NPP_NOT_IMPLEMENTED_ERROR, 46
- NPP_NOT_SUFFICIENT_COMPUTE_CAPABILITY, 45
- NPP_NOT_SUPPORTED_MODE_ERROR, 45
- NPP_NULL_POINTER_ERROR, 46
- NPP_NUMBER_OF_CHANNELS_ERROR, 46
- NPP_OUT_OFF_RANGE_ERROR, 46
- NPP_OVERFLOW_ERROR, 45
- NPP_QUADRANGLE_ERROR, 46
- NPP_QUALITY_INDEX_ERROR, 45
- NPP_RANGE_ERROR, 46
- NPP_RECTANGLE_ERROR, 46
- NPP_RESIZE_FACTOR_ERROR, 46
- NPP_RESIZE_NO_OPERATION_ERROR, 45
- NPP_RND_FINANCIAL, 45
- NPP_RND_NEAR, 44
- NPP_RND_ZERO, 45
- NPP_ROUND_MODE_NOT_SUPPORTED_ERROR, 45
- NPP_ROUND_NEAREST_TIES_AWAY_FROM_ZERO, 45
- NPP_ROUND_NEAREST_TIES_TO_EVEN, 45
- NPP_ROUND_TOWARD_ZERO, 45
- NPP_SCALE_RANGE_ERROR, 46
- NPP_SIZE_ERROR, 46
- NPP_STEP_ERROR, 46
- NPP_STRIDE_ERROR, 46
- NPP_SUCCESS, 46
- NPP_TEXTURE_BIND_ERROR, 45
- NPP_THRESHOLD_ERROR, 46
- NPP_THRESHOLD_NEGATIVE_LEVEL_ERROR, 46
- NPP_VERTICAL_AXIS, 43
- NPP_WRONG_INTERSECTION_QUAD_WARNING, 47

- NPP_WRONG_INTERSECTION_ROI_ERROR, [45](#)
- NPP_WRONG_INTERSECTION_ROI_WARNING, [47](#)
- NPP_ZC_MODE_NOT_SUPPORTED_ERROR, [45](#)
- NPP_ZERO_MASK_VALUE_ERROR, [46](#)
- NPPI_BAYER_BGGR, [43](#)
- NPPI_BAYER_GBRG, [43](#)
- NPPI_BAYER_GRBG, [43](#)
- NPPI_BAYER_RGBG, [43](#)
- NPPI_INTER_CUBIC, [43](#)
- NPPI_INTER_CUBIC2P_B05C03, [44](#)
- NPPI_INTER_CUBIC2P_BSPLINE, [44](#)
- NPPI_INTER_CUBIC2P_CATMULLROM, [44](#)
- NPPI_INTER_LANCZOS, [44](#)
- NPPI_INTER_LANCZOS3_ADVANCED, [44](#)
- NPPI_INTER_LINEAR, [43](#)
- NPPI_INTER_NN, [43](#)
- NPPI_INTER_SUPER, [44](#)
- NPPI_INTER_UNDEFINED, [43](#)
- NPPI_OP_ALPHA_ATOP, [42](#)
- NPPI_OP_ALPHA_ATOP_PREMUL, [42](#)
- NPPI_OP_ALPHA_IN, [42](#)
- NPPI_OP_ALPHA_IN_PREMUL, [42](#)
- NPPI_OP_ALPHA_OUT, [42](#)
- NPPI_OP_ALPHA_OUT_PREMUL, [42](#)
- NPPI_OP_ALPHA_OVER, [42](#)
- NPPI_OP_ALPHA_OVER_PREMUL, [42](#)
- NPPI_OP_ALPHA_PLUS, [42](#)
- NPPI_OP_ALPHA_PLUS_PREMUL, [42](#)
- NPPI_OP_ALPHA_PREMUL, [42](#)
- NPPI_OP_ALPHA_XOR, [42](#)
- NPPI_OP_ALPHA_XOR_PREMUL, [42](#)
- NPPI_SMOOTH_EDGE, [44](#)
- nppiACTable, [43](#)
- nppiDCTable, [43](#)
- nppZCC, [47](#)
- nppZCR, [47](#)
- nppZCXor, [47](#)
- typedefs_npp
 - NPP_MAX_16S, [39](#)
 - NPP_MAX_16U, [39](#)
 - NPP_MAX_32S, [39](#)
 - NPP_MAX_32U, [40](#)
 - NPP_MAX_64S, [40](#)
 - NPP_MAX_64U, [40](#)
 - NPP_MAX_8S, [40](#)
 - NPP_MAX_8U, [40](#)
 - NPP_MAXABS_32F, [40](#)
 - NPP_MAXABS_64F, [40](#)
 - NPP_MIN_16S, [40](#)
 - NPP_MIN_16U, [40](#)
 - NPP_MIN_32S, [40](#)
 - NPP_MIN_32U, [40](#)
 - NPP_MIN_64S, [41](#)
 - NPP_MIN_64U, [41](#)
 - NPP_MIN_8S, [41](#)
 - NPP_MIN_8U, [41](#)
 - NPP_MINABS_32F, [41](#)
 - NPP_MINABS_64F, [41](#)
 - NppCmpOp, [41](#)
 - NppGpuComputeCapability, [41](#)
 - NppHintAlgorithm, [42](#)
 - NppiAlphaOp, [42](#)
 - NppiAxis, [42](#)
 - NppiBayerGridPosition, [43](#)
 - NppiBorderType, [43](#)
 - NppiHuffmanTableType, [43](#)
 - NppiInterpolationMode, [43](#)
 - NppiMaskSize, [44](#)
 - NppRoundMode, [44](#)
 - NppStatus, [45](#)
 - NppsZCType, [47](#)
- width
 - NppiRect, [2874](#)
 - NppiSize, [2875](#)
- x
 - NppiPoint, [2873](#)
 - NppiRect, [2874](#)
- Xor, [457](#), [2640](#)
- XorC, [394](#), [2637](#)
- y
 - NppiPoint, [2873](#)
 - NppiRect, [2874](#)
- Zero, [2691](#)