

NVIDIA Performance Primitives (NPP)  
Version 8.0

January 28, 2016



# Contents

<b>1 NVIDIA Performance Primitives</b>	<b>1</b>
1.1 What is NPP? . . . . .	2
1.2 Documentation . . . . .	2
1.3 Technical Specifications . . . . .	3
1.4 Files . . . . .	3
1.4.1 Header Files . . . . .	3
1.4.2 Library Files . . . . .	3
1.5 Supported NVIDIA Hardware . . . . .	4
<b>2 General API Conventions</b>	<b>5</b>
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
<b>3 Signal-Processing Specific API Conventions</b>	<b>9</b>
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
<b>4 Imaging-Processing Specific API Conventions</b>	<b>13</b>

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 . . . . .	25
6.1	Data Structures . . . . .	25
7	Module Documentation . . . . .	27
7.1	NPP Core . . . . .	27
7.1.1	Detailed Description . . . . .	28
7.1.2	Function Documentation . . . . .	28
7.1.2.1	nppGetGpuComputeCapability . . . . .	28
7.1.2.2	nppGetGpuDeviceProperties . . . . .	28
7.1.2.3	nppGetGpuName . . . . .	28

7.1.2.4	nppGetGpuNumSMs . . . . .	28
7.1.2.5	nppGetLibVersion . . . . .	29
7.1.2.6	nppGetMaxThreadsPerBlock . . . . .	29
7.1.2.7	nppGetMaxThreadsPerSM . . . . .	29
7.1.2.8	nppGetStream . . . . .	29
7.1.2.9	nppGetStreamMaxThreadsPerSM . . . . .	29
7.1.2.10	nppGetStreamNumSMs . . . . .	29
7.1.2.11	nppSetStream . . . . .	30
7.2	NPP Type Definitions and Constants . . . . .	31
7.2.1	Define Documentation . . . . .	37
7.2.1.1	NPP_MAX_16S . . . . .	37
7.2.1.2	NPP_MAX_16U . . . . .	37
7.2.1.3	NPP_MAX_32S . . . . .	37
7.2.1.4	NPP_MAX_32U . . . . .	37
7.2.1.5	NPP_MAX_64S . . . . .	37
7.2.1.6	NPP_MAX_64U . . . . .	37
7.2.1.7	NPP_MAX_8S . . . . .	37
7.2.1.8	NPP_MAX_8U . . . . .	37
7.2.1.9	NPP_MAXABS_32F . . . . .	37
7.2.1.10	NPP_MAXABS_64F . . . . .	37
7.2.1.11	NPP_MIN_16S . . . . .	38
7.2.1.12	NPP_MIN_16U . . . . .	38
7.2.1.13	NPP_MIN_32S . . . . .	38
7.2.1.14	NPP_MIN_32U . . . . .	38
7.2.1.15	NPP_MIN_64S . . . . .	38
7.2.1.16	NPP_MIN_64U . . . . .	38
7.2.1.17	NPP_MIN_8S . . . . .	38
7.2.1.18	NPP_MIN_8U . . . . .	38
7.2.1.19	NPP_MINABS_32F . . . . .	38
7.2.1.20	NPP_MINABS_64F . . . . .	38
7.2.2	Enumeration Type Documentation . . . . .	38
7.2.2.1	NppCmpOp . . . . .	38
7.2.2.2	NppGpuComputeCapability . . . . .	39
7.2.2.3	NppHintAlgorithm . . . . .	39
7.2.2.4	NppiAlphaOp . . . . .	39
7.2.2.5	NppiAxis . . . . .	40

7.2.2.6	NppiBayerGridPosition . . . . .	40
7.2.2.7	NppiBorderType . . . . .	40
7.2.2.8	NppiDifferentialKernel . . . . .	40
7.2.2.9	NppiHuffmanTableType . . . . .	41
7.2.2.10	NppiInterpolationMode . . . . .	41
7.2.2.11	NppiMaskSize . . . . .	41
7.2.2.12	NppiNorm . . . . .	42
7.2.2.13	NppRoundMode . . . . .	42
7.2.2.14	NppStatus . . . . .	43
7.2.2.15	NppsZCType . . . . .	45
7.3	Basic NPP Data Types . . . . .	46
7.3.1	Typedef Documentation . . . . .	47
7.3.1.1	Npp16s . . . . .	47
7.3.1.2	Npp16u . . . . .	47
7.3.1.3	Npp32f . . . . .	47
7.3.1.4	Npp32fc . . . . .	47
7.3.1.5	Npp32s . . . . .	47
7.3.1.6	Npp32sc . . . . .	48
7.3.1.7	Npp32u . . . . .	48
7.3.1.8	Npp32uc . . . . .	48
7.3.1.9	Npp64f . . . . .	48
7.3.1.10	Npp64fc . . . . .	48
7.3.1.11	Npp64s . . . . .	48
7.3.1.12	Npp64sc . . . . .	48
7.3.1.13	Npp64u . . . . .	48
7.3.1.14	Npp8s . . . . .	48
7.3.1.15	Npp8u . . . . .	48
7.3.2	Function Documentation . . . . .	48
7.3.2.1	__align__ . . . . .	48
7.3.2.2	__align__ . . . . .	49
7.3.3	Variable Documentation . . . . .	49
7.3.3.1	Npp16sc . . . . .	49
7.3.3.2	Npp16uc . . . . .	49
7.3.3.3	Npp8uc . . . . .	49
7.4	Filtering Functions . . . . .	50
7.4.1	Detailed Description . . . . .	50

7.5 1D Linear Filter . . . . .	51
7.5.1 Function Documentation . . . . .	63
7.5.1.1 nppiFilterColumn32f_16s_AC4R . . . . .	63
7.5.1.2 nppiFilterColumn32f_16s_C1R . . . . .	63
7.5.1.3 nppiFilterColumn32f_16s_C3R . . . . .	64
7.5.1.4 nppiFilterColumn32f_16s_C4R . . . . .	64
7.5.1.5 nppiFilterColumn32f_16u_AC4R . . . . .	65
7.5.1.6 nppiFilterColumn32f_16u_C1R . . . . .	65
7.5.1.7 nppiFilterColumn32f_16u_C3R . . . . .	66
7.5.1.8 nppiFilterColumn32f_16u_C4R . . . . .	66
7.5.1.9 nppiFilterColumn32f_8u_AC4R . . . . .	67
7.5.1.10 nppiFilterColumn32f_8u_C1R . . . . .	67
7.5.1.11 nppiFilterColumn32f_8u_C3R . . . . .	68
7.5.1.12 nppiFilterColumn32f_8u_C4R . . . . .	68
7.5.1.13 nppiFilterColumn_16s_AC4R . . . . .	69
7.5.1.14 nppiFilterColumn_16s_C1R . . . . .	69
7.5.1.15 nppiFilterColumn_16s_C3R . . . . .	70
7.5.1.16 nppiFilterColumn_16s_C4R . . . . .	70
7.5.1.17 nppiFilterColumn_16u_AC4R . . . . .	71
7.5.1.18 nppiFilterColumn_16u_C1R . . . . .	71
7.5.1.19 nppiFilterColumn_16u_C3R . . . . .	72
7.5.1.20 nppiFilterColumn_16u_C4R . . . . .	72
7.5.1.21 nppiFilterColumn_32f_AC4R . . . . .	73
7.5.1.22 nppiFilterColumn_32f_C1R . . . . .	73
7.5.1.23 nppiFilterColumn_32f_C3R . . . . .	74
7.5.1.24 nppiFilterColumn_32f_C4R . . . . .	74
7.5.1.25 nppiFilterColumn_64f_C1R . . . . .	75
7.5.1.26 nppiFilterColumn_8u_AC4R . . . . .	75
7.5.1.27 nppiFilterColumn_8u_C1R . . . . .	76
7.5.1.28 nppiFilterColumn_8u_C3R . . . . .	76
7.5.1.29 nppiFilterColumn_8u_C4R . . . . .	77
7.5.1.30 nppiFilterColumnBorder32f_16s_AC4R . . . . .	77
7.5.1.31 nppiFilterColumnBorder32f_16s_C1R . . . . .	78
7.5.1.32 nppiFilterColumnBorder32f_16s_C3R . . . . .	78
7.5.1.33 nppiFilterColumnBorder32f_16s_C4R . . . . .	79
7.5.1.34 nppiFilterColumnBorder32f_16u_AC4R . . . . .	79

7.5.1.35 nppiFilterColumnBorder32f_16u_C1R . . . . .	80
7.5.1.36 nppiFilterColumnBorder32f_16u_C3R . . . . .	80
7.5.1.37 nppiFilterColumnBorder32f_16u_C4R . . . . .	81
7.5.1.38 nppiFilterColumnBorder32f_8u_AC4R . . . . .	81
7.5.1.39 nppiFilterColumnBorder32f_8u_C1R . . . . .	82
7.5.1.40 nppiFilterColumnBorder32f_8u_C3R . . . . .	82
7.5.1.41 nppiFilterColumnBorder32f_8u_C4R . . . . .	83
7.5.1.42 nppiFilterColumnBorder_16s_AC4R . . . . .	83
7.5.1.43 nppiFilterColumnBorder_16s_C1R . . . . .	84
7.5.1.44 nppiFilterColumnBorder_16s_C3R . . . . .	84
7.5.1.45 nppiFilterColumnBorder_16s_C4R . . . . .	85
7.5.1.46 nppiFilterColumnBorder_16u_AC4R . . . . .	86
7.5.1.47 nppiFilterColumnBorder_16u_C1R . . . . .	86
7.5.1.48 nppiFilterColumnBorder_16u_C3R . . . . .	87
7.5.1.49 nppiFilterColumnBorder_16u_C4R . . . . .	87
7.5.1.50 nppiFilterColumnBorder_32f_AC4R . . . . .	88
7.5.1.51 nppiFilterColumnBorder_32f_C1R . . . . .	89
7.5.1.52 nppiFilterColumnBorder_32f_C3R . . . . .	89
7.5.1.53 nppiFilterColumnBorder_32f_C4R . . . . .	90
7.5.1.54 nppiFilterColumnBorder_8u_AC4R . . . . .	90
7.5.1.55 nppiFilterColumnBorder_8u_C1R . . . . .	91
7.5.1.56 nppiFilterColumnBorder_8u_C3R . . . . .	91
7.5.1.57 nppiFilterColumnBorder_8u_C4R . . . . .	92
7.5.1.58 nppiFilterRow32f_16s_AC4R . . . . .	93
7.5.1.59 nppiFilterRow32f_16s_C1R . . . . .	93
7.5.1.60 nppiFilterRow32f_16s_C3R . . . . .	94
7.5.1.61 nppiFilterRow32f_16s_C4R . . . . .	94
7.5.1.62 nppiFilterRow32f_16u_AC4R . . . . .	95
7.5.1.63 nppiFilterRow32f_16u_C1R . . . . .	95
7.5.1.64 nppiFilterRow32f_16u_C3R . . . . .	96
7.5.1.65 nppiFilterRow32f_16u_C4R . . . . .	96
7.5.1.66 nppiFilterRow32f_8u_AC4R . . . . .	97
7.5.1.67 nppiFilterRow32f_8u_C1R . . . . .	97
7.5.1.68 nppiFilterRow32f_8u_C3R . . . . .	98
7.5.1.69 nppiFilterRow32f_8u_C4R . . . . .	98
7.5.1.70 nppiFilterRow_16s_AC4R . . . . .	99

7.5.1.71 nppiFilterRow_16s_C1R . . . . .	99
7.5.1.72 nppiFilterRow_16s_C3R . . . . .	100
7.5.1.73 nppiFilterRow_16s_C4R . . . . .	100
7.5.1.74 nppiFilterRow_16u_AC4R . . . . .	101
7.5.1.75 nppiFilterRow_16u_C1R . . . . .	101
7.5.1.76 nppiFilterRow_16u_C3R . . . . .	102
7.5.1.77 nppiFilterRow_16u_C4R . . . . .	102
7.5.1.78 nppiFilterRow_32f_AC4R . . . . .	103
7.5.1.79 nppiFilterRow_32f_C1R . . . . .	103
7.5.1.80 nppiFilterRow_32f_C3R . . . . .	104
7.5.1.81 nppiFilterRow_32f_C4R . . . . .	104
7.5.1.82 nppiFilterRow_64f_C1R . . . . .	105
7.5.1.83 nppiFilterRow_8u_AC4R . . . . .	105
7.5.1.84 nppiFilterRow_8u_C1R . . . . .	106
7.5.1.85 nppiFilterRow_8u_C3R . . . . .	106
7.5.1.86 nppiFilterRow_8u_C4R . . . . .	107
7.5.1.87 nppiFilterRowBorder32f_16s_AC4R . . . . .	107
7.5.1.88 nppiFilterRowBorder32f_16s_C1R . . . . .	108
7.5.1.89 nppiFilterRowBorder32f_16s_C3R . . . . .	108
7.5.1.90 nppiFilterRowBorder32f_16s_C4R . . . . .	109
7.5.1.91 nppiFilterRowBorder32f_16u_AC4R . . . . .	109
7.5.1.92 nppiFilterRowBorder32f_16u_C1R . . . . .	110
7.5.1.93 nppiFilterRowBorder32f_16u_C3R . . . . .	110
7.5.1.94 nppiFilterRowBorder32f_16u_C4R . . . . .	111
7.5.1.95 nppiFilterRowBorder32f_8u_AC4R . . . . .	111
7.5.1.96 nppiFilterRowBorder32f_8u_C1R . . . . .	112
7.5.1.97 nppiFilterRowBorder32f_8u_C3R . . . . .	112
7.5.1.98 nppiFilterRowBorder32f_8u_C4R . . . . .	113
7.5.1.99 nppiFilterRowBorder_16s_AC4R . . . . .	113
7.5.1.100 nppiFilterRowBorder_16s_C1R . . . . .	114
7.5.1.101 nppiFilterRowBorder_16s_C3R . . . . .	114
7.5.1.102 nppiFilterRowBorder_16s_C4R . . . . .	115
7.5.1.103 nppiFilterRowBorder_16u_AC4R . . . . .	116
7.5.1.104 nppiFilterRowBorder_16u_C1R . . . . .	116
7.5.1.105 nppiFilterRowBorder_16u_C3R . . . . .	117
7.5.1.106 nppiFilterRowBorder_16u_C4R . . . . .	117

7.5.1.107 nppiFilterRowBorder_32f_AC4R . . . . .	118
7.5.1.108 nppiFilterRowBorder_32f_C1R . . . . .	119
7.5.1.109 nppiFilterRowBorder_32f_C3R . . . . .	119
7.5.1.110 nppiFilterRowBorder_32f_C4R . . . . .	120
7.5.1.111 nppiFilterRowBorder_8u_AC4R . . . . .	120
7.5.1.112 nppiFilterRowBorder_8u_C1R . . . . .	121
7.5.1.113 nppiFilterRowBorder_8u_C3R . . . . .	121
7.5.1.114 nppiFilterRowBorder_8u_C4R . . . . .	122
7.6 1D Window Sum . . . . .	123
7.6.1 Function Documentation . . . . .	124
7.6.1.1 nppiSumWindowColumn_16s32f_C1R . . . . .	124
7.6.1.2 nppiSumWindowColumn_16s32f_C3R . . . . .	125
7.6.1.3 nppiSumWindowColumn_16s32f_C4R . . . . .	125
7.6.1.4 nppiSumWindowColumn_16u32f_C1R . . . . .	126
7.6.1.5 nppiSumWindowColumn_16u32f_C3R . . . . .	126
7.6.1.6 nppiSumWindowColumn_16u32f_C4R . . . . .	127
7.6.1.7 nppiSumWindowColumn_8u32f_C1R . . . . .	127
7.6.1.8 nppiSumWindowColumn_8u32f_C3R . . . . .	128
7.6.1.9 nppiSumWindowColumn_8u32f_C4R . . . . .	128
7.6.1.10 nppiSumWindowRow_16s32f_C1R . . . . .	129
7.6.1.11 nppiSumWindowRow_16s32f_C3R . . . . .	129
7.6.1.12 nppiSumWindowRow_16s32f_C4R . . . . .	130
7.6.1.13 nppiSumWindowRow_16u32f_C1R . . . . .	130
7.6.1.14 nppiSumWindowRow_16u32f_C3R . . . . .	131
7.6.1.15 nppiSumWindowRow_16u32f_C4R . . . . .	131
7.6.1.16 nppiSumWindowRow_8u32f_C1R . . . . .	132
7.6.1.17 nppiSumWindowRow_8u32f_C3R . . . . .	132
7.6.1.18 nppiSumWindowRow_8u32f_C4R . . . . .	133
7.7 1D Window Sum with Border Control . . . . .	134
7.7.1 Function Documentation . . . . .	136
7.7.1.1 nppiSumWindowColumnBorder_16s32f_C1R . . . . .	136
7.7.1.2 nppiSumWindowColumnBorder_16s32f_C3R . . . . .	136
7.7.1.3 nppiSumWindowColumnBorder_16s32f_C4R . . . . .	137
7.7.1.4 nppiSumWindowColumnBorder_16u32f_C1R . . . . .	137
7.7.1.5 nppiSumWindowColumnBorder_16u32f_C3R . . . . .	138
7.7.1.6 nppiSumWindowColumnBorder_16u32f_C4R . . . . .	139

7.7.1.7	nppiSumWindowColumnBorder_8u32f_C1R . . . . .	139
7.7.1.8	nppiSumWindowColumnBorder_8u32f_C3R . . . . .	140
7.7.1.9	nppiSumWindowColumnBorder_8u32f_C4R . . . . .	140
7.7.1.10	nppiSumWindowRowBorder_16s32f_C1R . . . . .	141
7.7.1.11	nppiSumWindowRowBorder_16s32f_C3R . . . . .	141
7.7.1.12	nppiSumWindowRowBorder_16s32f_C4R . . . . .	142
7.7.1.13	nppiSumWindowRowBorder_16u32f_C1R . . . . .	143
7.7.1.14	nppiSumWindowRowBorder_16u32f_C3R . . . . .	143
7.7.1.15	nppiSumWindowRowBorder_16u32f_C4R . . . . .	144
7.7.1.16	nppiSumWindowRowBorder_8u32f_C1R . . . . .	144
7.7.1.17	nppiSumWindowRowBorder_8u32f_C3R . . . . .	145
7.7.1.18	nppiSumWindowRowBorder_8u32f_C4R . . . . .	145
7.8	Convolution . . . . .	147
7.8.1	Function Documentation . . . . .	156
7.8.1.1	nppiFilter32f_16s_AC4R . . . . .	156
7.8.1.2	nppiFilter32f_16s_C1R . . . . .	157
7.8.1.3	nppiFilter32f_16s_C3R . . . . .	157
7.8.1.4	nppiFilter32f_16s_C4R . . . . .	158
7.8.1.5	nppiFilter32f_16u_AC4R . . . . .	158
7.8.1.6	nppiFilter32f_16u_C1R . . . . .	159
7.8.1.7	nppiFilter32f_16u_C3R . . . . .	159
7.8.1.8	nppiFilter32f_16u_C4R . . . . .	160
7.8.1.9	nppiFilter32f_32s_AC4R . . . . .	160
7.8.1.10	nppiFilter32f_32s_C1R . . . . .	161
7.8.1.11	nppiFilter32f_32s_C3R . . . . .	161
7.8.1.12	nppiFilter32f_32s_C4R . . . . .	162
7.8.1.13	nppiFilter32f_8s16s_AC4R . . . . .	162
7.8.1.14	nppiFilter32f_8s16s_C1R . . . . .	163
7.8.1.15	nppiFilter32f_8s16s_C3R . . . . .	163
7.8.1.16	nppiFilter32f_8s16s_C4R . . . . .	164
7.8.1.17	nppiFilter32f_8s_AC4R . . . . .	164
7.8.1.18	nppiFilter32f_8s_C1R . . . . .	165
7.8.1.19	nppiFilter32f_8s_C2R . . . . .	165
7.8.1.20	nppiFilter32f_8s_C3R . . . . .	166
7.8.1.21	nppiFilter32f_8s_C4R . . . . .	166
7.8.1.22	nppiFilter32f_8u16s_AC4R . . . . .	167

7.8.1.23	nppiFilter32f_8u16s_C1R . . . . .	167
7.8.1.24	nppiFilter32f_8u16s_C3R . . . . .	168
7.8.1.25	nppiFilter32f_8u16s_C4R . . . . .	168
7.8.1.26	nppiFilter32f_8u_AC4R . . . . .	169
7.8.1.27	nppiFilter32f_8u_C1R . . . . .	169
7.8.1.28	nppiFilter32f_8u_C2R . . . . .	170
7.8.1.29	nppiFilter32f_8u_C3R . . . . .	170
7.8.1.30	nppiFilter32f_8u_C4R . . . . .	171
7.8.1.31	nppiFilter_16s_AC4R . . . . .	171
7.8.1.32	nppiFilter_16s_C1R . . . . .	172
7.8.1.33	nppiFilter_16s_C3R . . . . .	172
7.8.1.34	nppiFilter_16s_C4R . . . . .	173
7.8.1.35	nppiFilter_16u_AC4R . . . . .	173
7.8.1.36	nppiFilter_16u_C1R . . . . .	174
7.8.1.37	nppiFilter_16u_C3R . . . . .	174
7.8.1.38	nppiFilter_16u_C4R . . . . .	175
7.8.1.39	nppiFilter_32f_AC4R . . . . .	175
7.8.1.40	nppiFilter_32f_C1R . . . . .	176
7.8.1.41	nppiFilter_32f_C2R . . . . .	176
7.8.1.42	nppiFilter_32f_C3R . . . . .	177
7.8.1.43	nppiFilter_32f_C4R . . . . .	177
7.8.1.44	nppiFilter_64f_C1R . . . . .	178
7.8.1.45	nppiFilter_8u_AC4R . . . . .	178
7.8.1.46	nppiFilter_8u_C1R . . . . .	179
7.8.1.47	nppiFilter_8u_C3R . . . . .	179
7.8.1.48	nppiFilter_8u_C4R . . . . .	180
7.8.1.49	nppiFilterBorder32f_16s_AC4R . . . . .	180
7.8.1.50	nppiFilterBorder32f_16s_C1R . . . . .	181
7.8.1.51	nppiFilterBorder32f_16s_C3R . . . . .	181
7.8.1.52	nppiFilterBorder32f_16s_C4R . . . . .	182
7.8.1.53	nppiFilterBorder32f_16u_AC4R . . . . .	182
7.8.1.54	nppiFilterBorder32f_16u_C1R . . . . .	183
7.8.1.55	nppiFilterBorder32f_16u_C3R . . . . .	183
7.8.1.56	nppiFilterBorder32f_16u_C4R . . . . .	184
7.8.1.57	nppiFilterBorder32f_32s_AC4R . . . . .	184
7.8.1.58	nppiFilterBorder32f_32s_C1R . . . . .	185

7.8.1.59 nppiFilterBorder32f_32s_C3R . . . . .	185
7.8.1.60 nppiFilterBorder32f_32s_C4R . . . . .	186
7.8.1.61 nppiFilterBorder32f_8s16s_AC4R . . . . .	186
7.8.1.62 nppiFilterBorder32f_8s16s_C1R . . . . .	187
7.8.1.63 nppiFilterBorder32f_8s16s_C3R . . . . .	187
7.8.1.64 nppiFilterBorder32f_8s16s_C4R . . . . .	188
7.8.1.65 nppiFilterBorder32f_8s_AC4R . . . . .	188
7.8.1.66 nppiFilterBorder32f_8s_C1R . . . . .	189
7.8.1.67 nppiFilterBorder32f_8s_C2R . . . . .	189
7.8.1.68 nppiFilterBorder32f_8s_C3R . . . . .	190
7.8.1.69 nppiFilterBorder32f_8s_C4R . . . . .	190
7.8.1.70 nppiFilterBorder32f_8u16s_AC4R . . . . .	191
7.8.1.71 nppiFilterBorder32f_8u16s_C1R . . . . .	191
7.8.1.72 nppiFilterBorder32f_8u16s_C3R . . . . .	192
7.8.1.73 nppiFilterBorder32f_8u16s_C4R . . . . .	192
7.8.1.74 nppiFilterBorder32f_8u_AC4R . . . . .	193
7.8.1.75 nppiFilterBorder32f_8u_C1R . . . . .	193
7.8.1.76 nppiFilterBorder32f_8u_C2R . . . . .	194
7.8.1.77 nppiFilterBorder32f_8u_C3R . . . . .	194
7.8.1.78 nppiFilterBorder32f_8u_C4R . . . . .	195
7.8.1.79 nppiFilterBorder_16s_AC4R . . . . .	195
7.8.1.80 nppiFilterBorder_16s_C1R . . . . .	196
7.8.1.81 nppiFilterBorder_16s_C3R . . . . .	196
7.8.1.82 nppiFilterBorder_16s_C4R . . . . .	197
7.8.1.83 nppiFilterBorder_16u_AC4R . . . . .	198
7.8.1.84 nppiFilterBorder_16u_C1R . . . . .	198
7.8.1.85 nppiFilterBorder_16u_C3R . . . . .	199
7.8.1.86 nppiFilterBorder_16u_C4R . . . . .	199
7.8.1.87 nppiFilterBorder_32f_AC4R . . . . .	200
7.8.1.88 nppiFilterBorder_32f_C1R . . . . .	201
7.8.1.89 nppiFilterBorder_32f_C2R . . . . .	201
7.8.1.90 nppiFilterBorder_32f_C3R . . . . .	202
7.8.1.91 nppiFilterBorder_32f_C4R . . . . .	202
7.8.1.92 nppiFilterBorder_8u_AC4R . . . . .	203
7.8.1.93 nppiFilterBorder_8u_C1R . . . . .	203
7.8.1.94 nppiFilterBorder_8u_C3R . . . . .	204

7.8.1.95 nppiFilterBorder_8u_C4R . . . . .	204
7.9 2D Fixed Linear Filters . . . . .	206
7.9.1 Function Documentation . . . . .	209
7.9.1.1 nppiFilterBox_16s_AC4R . . . . .	209
7.9.1.2 nppiFilterBox_16s_C1R . . . . .	209
7.9.1.3 nppiFilterBox_16s_C3R . . . . .	210
7.9.1.4 nppiFilterBox_16s_C4R . . . . .	210
7.9.1.5 nppiFilterBox_16u_AC4R . . . . .	210
7.9.1.6 nppiFilterBox_16u_C1R . . . . .	211
7.9.1.7 nppiFilterBox_16u_C3R . . . . .	211
7.9.1.8 nppiFilterBox_16u_C4R . . . . .	212
7.9.1.9 nppiFilterBox_32f_AC4R . . . . .	212
7.9.1.10 nppiFilterBox_32f_C1R . . . . .	212
7.9.1.11 nppiFilterBox_32f_C3R . . . . .	213
7.9.1.12 nppiFilterBox_32f_C4R . . . . .	213
7.9.1.13 nppiFilterBox_64f_C1R . . . . .	214
7.9.1.14 nppiFilterBox_8u_AC4R . . . . .	214
7.9.1.15 nppiFilterBox_8u_C1R . . . . .	214
7.9.1.16 nppiFilterBox_8u_C3R . . . . .	215
7.9.1.17 nppiFilterBox_8u_C4R . . . . .	215
7.9.1.18 nppiFilterBoxBorder_16s_AC4R . . . . .	216
7.9.1.19 nppiFilterBoxBorder_16s_C1R . . . . .	216
7.9.1.20 nppiFilterBoxBorder_16s_C3R . . . . .	217
7.9.1.21 nppiFilterBoxBorder_16s_C4R . . . . .	217
7.9.1.22 nppiFilterBoxBorder_16u_AC4R . . . . .	218
7.9.1.23 nppiFilterBoxBorder_16u_C1R . . . . .	218
7.9.1.24 nppiFilterBoxBorder_16u_C3R . . . . .	219
7.9.1.25 nppiFilterBoxBorder_16u_C4R . . . . .	219
7.9.1.26 nppiFilterBoxBorder_32f_AC4R . . . . .	220
7.9.1.27 nppiFilterBoxBorder_32f_C1R . . . . .	220
7.9.1.28 nppiFilterBoxBorder_32f_C3R . . . . .	221
7.9.1.29 nppiFilterBoxBorder_32f_C4R . . . . .	221
7.9.1.30 nppiFilterBoxBorder_8u_AC4R . . . . .	222
7.9.1.31 nppiFilterBoxBorder_8u_C1R . . . . .	222
7.9.1.32 nppiFilterBoxBorder_8u_C3R . . . . .	223
7.9.1.33 nppiFilterBoxBorder_8u_C4R . . . . .	223

7.10 Rank Filters . . . . .	224
7.10.1 Function Documentation . . . . .	233
7.10.1.1 nppiFilterMax_16s_AC4R . . . . .	233
7.10.1.2 nppiFilterMax_16s_C1R . . . . .	233
7.10.1.3 nppiFilterMax_16s_C3R . . . . .	234
7.10.1.4 nppiFilterMax_16s_C4R . . . . .	234
7.10.1.5 nppiFilterMax_16u_AC4R . . . . .	234
7.10.1.6 nppiFilterMax_16u_C1R . . . . .	235
7.10.1.7 nppiFilterMax_16u_C3R . . . . .	235
7.10.1.8 nppiFilterMax_16u_C4R . . . . .	236
7.10.1.9 nppiFilterMax_32f_AC4R . . . . .	236
7.10.1.10 nppiFilterMax_32f_C1R . . . . .	236
7.10.1.11 nppiFilterMax_32f_C3R . . . . .	237
7.10.1.12 nppiFilterMax_32f_C4R . . . . .	237
7.10.1.13 nppiFilterMax_8u_AC4R . . . . .	238
7.10.1.14 nppiFilterMax_8u_C1R . . . . .	238
7.10.1.15 nppiFilterMax_8u_C3R . . . . .	238
7.10.1.16 nppiFilterMax_8u_C4R . . . . .	239
7.10.1.17 nppiFilterMaxBorder_16s_AC4R . . . . .	239
7.10.1.18 nppiFilterMaxBorder_16s_C1R . . . . .	240
7.10.1.19 nppiFilterMaxBorder_16s_C3R . . . . .	240
7.10.1.20 nppiFilterMaxBorder_16s_C4R . . . . .	241
7.10.1.21 nppiFilterMaxBorder_16u_AC4R . . . . .	241
7.10.1.22 nppiFilterMaxBorder_16u_C1R . . . . .	242
7.10.1.23 nppiFilterMaxBorder_16u_C3R . . . . .	242
7.10.1.24 nppiFilterMaxBorder_16u_C4R . . . . .	243
7.10.1.25 nppiFilterMaxBorder_32f_AC4R . . . . .	243
7.10.1.26 nppiFilterMaxBorder_32f_C1R . . . . .	244
7.10.1.27 nppiFilterMaxBorder_32f_C3R . . . . .	244
7.10.1.28 nppiFilterMaxBorder_32f_C4R . . . . .	245
7.10.1.29 nppiFilterMaxBorder_8u_AC4R . . . . .	245
7.10.1.30 nppiFilterMaxBorder_8u_C1R . . . . .	246
7.10.1.31 nppiFilterMaxBorder_8u_C3R . . . . .	246
7.10.1.32 nppiFilterMaxBorder_8u_C4R . . . . .	247
7.10.1.33 nppiFilterMedian_16s_AC4R . . . . .	247
7.10.1.34 nppiFilterMedian_16s_C1R . . . . .	248

7.10.1.35 nppiFilterMedian_16s_C3R . . . . .	248
7.10.1.36 nppiFilterMedian_16s_C4R . . . . .	248
7.10.1.37 nppiFilterMedian_16u_AC4R . . . . .	249
7.10.1.38 nppiFilterMedian_16u_C1R . . . . .	249
7.10.1.39 nppiFilterMedian_16u_C3R . . . . .	250
7.10.1.40 nppiFilterMedian_16u_C4R . . . . .	250
7.10.1.41 nppiFilterMedian_32f_AC4R . . . . .	251
7.10.1.42 nppiFilterMedian_32f_C1R . . . . .	251
7.10.1.43 nppiFilterMedian_32f_C3R . . . . .	251
7.10.1.44 nppiFilterMedian_32f_C4R . . . . .	252
7.10.1.45 nppiFilterMedian_8u_AC4R . . . . .	252
7.10.1.46 nppiFilterMedian_8u_C1R . . . . .	253
7.10.1.47 nppiFilterMedian_8u_C3R . . . . .	253
7.10.1.48 nppiFilterMedian_8u_C4R . . . . .	254
7.10.1.49 nppiFilterMedianGetBufferSize_16s_AC4R . . . . .	254
7.10.1.50 nppiFilterMedianGetBufferSize_16s_C1R . . . . .	254
7.10.1.51 nppiFilterMedianGetBufferSize_16s_C3R . . . . .	255
7.10.1.52 nppiFilterMedianGetBufferSize_16s_C4R . . . . .	255
7.10.1.53 nppiFilterMedianGetBufferSize_16u_AC4R . . . . .	255
7.10.1.54 nppiFilterMedianGetBufferSize_16u_C1R . . . . .	255
7.10.1.55 nppiFilterMedianGetBufferSize_16u_C3R . . . . .	256
7.10.1.56 nppiFilterMedianGetBufferSize_16u_C4R . . . . .	256
7.10.1.57 nppiFilterMedianGetBufferSize_32f_AC4R . . . . .	256
7.10.1.58 nppiFilterMedianGetBufferSize_32f_C1R . . . . .	257
7.10.1.59 nppiFilterMedianGetBufferSize_32f_C3R . . . . .	257
7.10.1.60 nppiFilterMedianGetBufferSize_32f_C4R . . . . .	257
7.10.1.61 nppiFilterMedianGetBufferSize_8u_AC4R . . . . .	257
7.10.1.62 nppiFilterMedianGetBufferSize_8u_C1R . . . . .	258
7.10.1.63 nppiFilterMedianGetBufferSize_8u_C3R . . . . .	258
7.10.1.64 nppiFilterMedianGetBufferSize_8u_C4R . . . . .	258
7.10.1.65 nppiFilterMin_16s_AC4R . . . . .	259
7.10.1.66 nppiFilterMin_16s_C1R . . . . .	259
7.10.1.67 nppiFilterMin_16s_C3R . . . . .	259
7.10.1.68 nppiFilterMin_16s_C4R . . . . .	260
7.10.1.69 nppiFilterMin_16u_AC4R . . . . .	260
7.10.1.70 nppiFilterMin_16u_C1R . . . . .	261

7.10.1.71 nppiFilterMin_16u_C3R . . . . .	261
7.10.1.72 nppiFilterMin_16u_C4R . . . . .	261
7.10.1.73 nppiFilterMin_32f_AC4R . . . . .	262
7.10.1.74 nppiFilterMin_32f_C1R . . . . .	262
7.10.1.75 nppiFilterMin_32f_C3R . . . . .	263
7.10.1.76 nppiFilterMin_32f_C4R . . . . .	263
7.10.1.77 nppiFilterMin_8u_AC4R . . . . .	263
7.10.1.78 nppiFilterMin_8u_C1R . . . . .	264
7.10.1.79 nppiFilterMin_8u_C3R . . . . .	264
7.10.1.80 nppiFilterMin_8u_C4R . . . . .	265
7.10.1.81 nppiFilterMinBorder_16s_AC4R . . . . .	265
7.10.1.82 nppiFilterMinBorder_16s_C1R . . . . .	265
7.10.1.83 nppiFilterMinBorder_16s_C3R . . . . .	266
7.10.1.84 nppiFilterMinBorder_16s_C4R . . . . .	266
7.10.1.85 nppiFilterMinBorder_16u_AC4R . . . . .	267
7.10.1.86 nppiFilterMinBorder_16u_C1R . . . . .	267
7.10.1.87 nppiFilterMinBorder_16u_C3R . . . . .	268
7.10.1.88 nppiFilterMinBorder_16u_C4R . . . . .	268
7.10.1.89 nppiFilterMinBorder_32f_AC4R . . . . .	269
7.10.1.90 nppiFilterMinBorder_32f_C1R . . . . .	269
7.10.1.91 nppiFilterMinBorder_32f_C3R . . . . .	270
7.10.1.92 nppiFilterMinBorder_32f_C4R . . . . .	270
7.10.1.93 nppiFilterMinBorder_8u_AC4R . . . . .	271
7.10.1.94 nppiFilterMinBorder_8u_C1R . . . . .	271
7.10.1.95 nppiFilterMinBorder_8u_C3R . . . . .	272
7.10.1.96 nppiFilterMinBorder_8u_C4R . . . . .	272
7.11 Fixed Filters . . . . .	274
7.11.1 Detailed Description . . . . .	329
7.11.2 Function Documentation . . . . .	329
7.11.2.1 nppiFilterBilateralGaussBorder_16u_C1R . . . . .	329
7.11.2.2 nppiFilterBilateralGaussBorder_16u_C3R . . . . .	330
7.11.2.3 nppiFilterBilateralGaussBorder_32f_C1R . . . . .	330
7.11.2.4 nppiFilterBilateralGaussBorder_32f_C3R . . . . .	331
7.11.2.5 nppiFilterBilateralGaussBorder_8u_C1R . . . . .	332
7.11.2.6 nppiFilterBilateralGaussBorder_8u_C3R . . . . .	332
7.11.2.7 nppiFilterCannyBorder_8u_C1R . . . . .	333

7.11.2.8 nppiFilterCannyBorderGetBufferSize . . . . .	334
7.11.2.9 nppiFilterGauss_16s_AC4R . . . . .	334
7.11.2.10 nppiFilterGauss_16s_C1R . . . . .	334
7.11.2.11 nppiFilterGauss_16s_C3R . . . . .	335
7.11.2.12 nppiFilterGauss_16s_C4R . . . . .	335
7.11.2.13 nppiFilterGauss_16u_AC4R . . . . .	335
7.11.2.14 nppiFilterGauss_16u_C1R . . . . .	336
7.11.2.15 nppiFilterGauss_16u_C3R . . . . .	336
7.11.2.16 nppiFilterGauss_16u_C4R . . . . .	336
7.11.2.17 nppiFilterGauss_32f_AC4R . . . . .	337
7.11.2.18 nppiFilterGauss_32f_C1R . . . . .	337
7.11.2.19 nppiFilterGauss_32f_C3R . . . . .	337
7.11.2.20 nppiFilterGauss_32f_C4R . . . . .	338
7.11.2.21 nppiFilterGauss_8u_AC4R . . . . .	338
7.11.2.22 nppiFilterGauss_8u_C1R . . . . .	338
7.11.2.23 nppiFilterGauss_8u_C3R . . . . .	339
7.11.2.24 nppiFilterGauss_8u_C4R . . . . .	339
7.11.2.25 nppiFilterGaussAdvanced_16s_AC4R . . . . .	339
7.11.2.26 nppiFilterGaussAdvanced_16s_C1R . . . . .	340
7.11.2.27 nppiFilterGaussAdvanced_16s_C3R . . . . .	340
7.11.2.28 nppiFilterGaussAdvanced_16s_C4R . . . . .	341
7.11.2.29 nppiFilterGaussAdvanced_16u_AC4R . . . . .	341
7.11.2.30 nppiFilterGaussAdvanced_16u_C1R . . . . .	341
7.11.2.31 nppiFilterGaussAdvanced_16u_C3R . . . . .	342
7.11.2.32 nppiFilterGaussAdvanced_16u_C4R . . . . .	342
7.11.2.33 nppiFilterGaussAdvanced_32f_AC4R . . . . .	343
7.11.2.34 nppiFilterGaussAdvanced_32f_C1R . . . . .	343
7.11.2.35 nppiFilterGaussAdvanced_32f_C3R . . . . .	343
7.11.2.36 nppiFilterGaussAdvanced_32f_C4R . . . . .	344
7.11.2.37 nppiFilterGaussAdvanced_8u_AC4R . . . . .	344
7.11.2.38 nppiFilterGaussAdvanced_8u_C1R . . . . .	345
7.11.2.39 nppiFilterGaussAdvanced_8u_C3R . . . . .	345
7.11.2.40 nppiFilterGaussAdvanced_8u_C4R . . . . .	345
7.11.2.41 nppiFilterGaussAdvancedBorder_16s_AC4R . . . . .	346
7.11.2.42 nppiFilterGaussAdvancedBorder_16s_C1R . . . . .	346
7.11.2.43 nppiFilterGaussAdvancedBorder_16s_C3R . . . . .	347

7.11.2.44 nppiFilterGaussAdvancedBorder_16s_C4R . . . . .	347
7.11.2.45 nppiFilterGaussAdvancedBorder_16u_AC4R . . . . .	348
7.11.2.46 nppiFilterGaussAdvancedBorder_16u_C1R . . . . .	348
7.11.2.47 nppiFilterGaussAdvancedBorder_16u_C3R . . . . .	349
7.11.2.48 nppiFilterGaussAdvancedBorder_16u_C4R . . . . .	349
7.11.2.49 nppiFilterGaussAdvancedBorder_32f_AC4R . . . . .	350
7.11.2.50 nppiFilterGaussAdvancedBorder_32f_C1R . . . . .	350
7.11.2.51 nppiFilterGaussAdvancedBorder_32f_C3R . . . . .	351
7.11.2.52 nppiFilterGaussAdvancedBorder_32f_C4R . . . . .	351
7.11.2.53 nppiFilterGaussAdvancedBorder_8u_AC4R . . . . .	352
7.11.2.54 nppiFilterGaussAdvancedBorder_8u_C1R . . . . .	352
7.11.2.55 nppiFilterGaussAdvancedBorder_8u_C3R . . . . .	353
7.11.2.56 nppiFilterGaussAdvancedBorder_8u_C4R . . . . .	353
7.11.2.57 nppiFilterGaussBorder_16s_AC4R . . . . .	354
7.11.2.58 nppiFilterGaussBorder_16s_C1R . . . . .	354
7.11.2.59 nppiFilterGaussBorder_16s_C3R . . . . .	355
7.11.2.60 nppiFilterGaussBorder_16s_C4R . . . . .	355
7.11.2.61 nppiFilterGaussBorder_16u_AC4R . . . . .	356
7.11.2.62 nppiFilterGaussBorder_16u_C1R . . . . .	356
7.11.2.63 nppiFilterGaussBorder_16u_C3R . . . . .	357
7.11.2.64 nppiFilterGaussBorder_16u_C4R . . . . .	357
7.11.2.65 nppiFilterGaussBorder_32f_AC4R . . . . .	358
7.11.2.66 nppiFilterGaussBorder_32f_C1R . . . . .	358
7.11.2.67 nppiFilterGaussBorder_32f_C3R . . . . .	359
7.11.2.68 nppiFilterGaussBorder_32f_C4R . . . . .	359
7.11.2.69 nppiFilterGaussBorder_8u_AC4R . . . . .	360
7.11.2.70 nppiFilterGaussBorder_8u_C1R . . . . .	360
7.11.2.71 nppiFilterGaussBorder_8u_C3R . . . . .	361
7.11.2.72 nppiFilterGaussBorder_8u_C4R . . . . .	361
7.11.2.73 nppiFilterGaussPyramidLayerDownBorder_16u_C1R . . . . .	362
7.11.2.74 nppiFilterGaussPyramidLayerDownBorder_16u_C3R . . . . .	362
7.11.2.75 nppiFilterGaussPyramidLayerDownBorder_32f_C1R . . . . .	363
7.11.2.76 nppiFilterGaussPyramidLayerDownBorder_32f_C3R . . . . .	363
7.11.2.77 nppiFilterGaussPyramidLayerDownBorder_8u_C1R . . . . .	364
7.11.2.78 nppiFilterGaussPyramidLayerDownBorder_8u_C3R . . . . .	364
7.11.2.79 nppiFilterGaussPyramidLayerUpBorder_16u_C1R . . . . .	365

7.11.2.80 nppiFilterGaussPyramidLayerUpBorder_16u_C3R . . . . .	366
7.11.2.81 nppiFilterGaussPyramidLayerUpBorder_32f_C1R . . . . .	366
7.11.2.82 nppiFilterGaussPyramidLayerUpBorder_32f_C3R . . . . .	367
7.11.2.83 nppiFilterGaussPyramidLayerUpBorder_8u_C1R . . . . .	367
7.11.2.84 nppiFilterGaussPyramidLayerUpBorder_8u_C3R . . . . .	368
7.11.2.85 nppiFilterHarrisCornersBorder_8u32f_C1R . . . . .	368
7.11.2.86 nppiFilterHarrisCornersBorderGetBufferSize . . . . .	369
7.11.2.87 nppiFilterHighPass_16s_AC4R . . . . .	369
7.11.2.88 nppiFilterHighPass_16s_C1R . . . . .	370
7.11.2.89 nppiFilterHighPass_16s_C3R . . . . .	370
7.11.2.90 nppiFilterHighPass_16s_C4R . . . . .	371
7.11.2.91 nppiFilterHighPass_16u_AC4R . . . . .	371
7.11.2.92 nppiFilterHighPass_16u_C1R . . . . .	371
7.11.2.93 nppiFilterHighPass_16u_C3R . . . . .	372
7.11.2.94 nppiFilterHighPass_16u_C4R . . . . .	372
7.11.2.95 nppiFilterHighPass_32f_AC4R . . . . .	372
7.11.2.96 nppiFilterHighPass_32f_C1R . . . . .	373
7.11.2.97 nppiFilterHighPass_32f_C3R . . . . .	373
7.11.2.98 nppiFilterHighPass_32f_C4R . . . . .	373
7.11.2.99 nppiFilterHighPass_8u_AC4R . . . . .	374
7.11.2.100nppiFilterHighPass_8u_C1R . . . . .	374
7.11.2.101nppiFilterHighPass_8u_C3R . . . . .	374
7.11.2.102nppiFilterHighPass_8u_C4R . . . . .	375
7.11.2.103nppiFilterHighPassBorder_16s_AC4R . . . . .	375
7.11.2.104nppiFilterHighPassBorder_16s_C1R . . . . .	375
7.11.2.105nppiFilterHighPassBorder_16s_C3R . . . . .	376
7.11.2.106nppiFilterHighPassBorder_16s_C4R . . . . .	376
7.11.2.107nppiFilterHighPassBorder_16u_AC4R . . . . .	377
7.11.2.108nppiFilterHighPassBorder_16u_C1R . . . . .	377
7.11.2.109nppiFilterHighPassBorder_16u_C3R . . . . .	378
7.11.2.110nppiFilterHighPassBorder_16u_C4R . . . . .	378
7.11.2.111nppiFilterHighPassBorder_32f_AC4R . . . . .	379
7.11.2.112nppiFilterHighPassBorder_32f_C1R . . . . .	379
7.11.2.113nppiFilterHighPassBorder_32f_C3R . . . . .	380
7.11.2.114nppiFilterHighPassBorder_32f_C4R . . . . .	380
7.11.2.115nppiFilterHighPassBorder_8u_AC4R . . . . .	381

7.11.2.116nppiFilterHighPassBorder_8u_C1R . . . . .	381
7.11.2.117nppiFilterHighPassBorder_8u_C3R . . . . .	382
7.11.2.118nppiFilterHighPassBorder_8u_C4R . . . . .	382
7.11.2.119nppiFilterLaplace_16s_AC4R . . . . .	383
7.11.2.120nppiFilterLaplace_16s_C1R . . . . .	383
7.11.2.121nppiFilterLaplace_16s_C3R . . . . .	383
7.11.2.122nppiFilterLaplace_16s_C4R . . . . .	384
7.11.2.123nppiFilterLaplace_32f_AC4R . . . . .	384
7.11.2.124nppiFilterLaplace_32f_C1R . . . . .	384
7.11.2.125nppiFilterLaplace_32f_C3R . . . . .	385
7.11.2.126nppiFilterLaplace_32f_C4R . . . . .	385
7.11.2.127nppiFilterLaplace_8s16s_C1R . . . . .	385
7.11.2.128nppiFilterLaplace_8u16s_C1R . . . . .	386
7.11.2.129nppiFilterLaplace_8u_AC4R . . . . .	386
7.11.2.130nppiFilterLaplace_8u_C1R . . . . .	386
7.11.2.131nppiFilterLaplace_8u_C3R . . . . .	387
7.11.2.132nppiFilterLaplace_8u_C4R . . . . .	387
7.11.2.133nppiFilterLaplaceBorder_16s_AC4R . . . . .	387
7.11.2.134nppiFilterLaplaceBorder_16s_C1R . . . . .	388
7.11.2.135nppiFilterLaplaceBorder_16s_C3R . . . . .	388
7.11.2.136nppiFilterLaplaceBorder_16s_C4R . . . . .	389
7.11.2.137nppiFilterLaplaceBorder_32f_AC4R . . . . .	389
7.11.2.138nppiFilterLaplaceBorder_32f_C1R . . . . .	390
7.11.2.139nppiFilterLaplaceBorder_32f_C3R . . . . .	390
7.11.2.140nppiFilterLaplaceBorder_32f_C4R . . . . .	391
7.11.2.141nppiFilterLaplaceBorder_8s16s_C1R . . . . .	391
7.11.2.142nppiFilterLaplaceBorder_8u16s_C1R . . . . .	392
7.11.2.143nppiFilterLaplaceBorder_8u_AC4R . . . . .	392
7.11.2.144nppiFilterLaplaceBorder_8u_C1R . . . . .	393
7.11.2.145nppiFilterLaplaceBorder_8u_C3R . . . . .	393
7.11.2.146nppiFilterLaplaceBorder_8u_C4R . . . . .	394
7.11.2.147nppiFilterLowPass_16s_AC4R . . . . .	394
7.11.2.148nppiFilterLowPass_16s_C1R . . . . .	394
7.11.2.149nppiFilterLowPass_16s_C3R . . . . .	395
7.11.2.150nppiFilterLowPass_16s_C4R . . . . .	395
7.11.2.151nppiFilterLowPass_16u_AC4R . . . . .	396

7.11.2.152nppiFilterLowPass_16u_C1R . . . . .	396
7.11.2.153nppiFilterLowPass_16u_C3R . . . . .	396
7.11.2.154nppiFilterLowPass_16u_C4R . . . . .	397
7.11.2.155nppiFilterLowPass_32f_AC4R . . . . .	397
7.11.2.156nppiFilterLowPass_32f_C1R . . . . .	397
7.11.2.157nppiFilterLowPass_32f_C3R . . . . .	398
7.11.2.158nppiFilterLowPass_32f_C4R . . . . .	398
7.11.2.159nppiFilterLowPass_8u_AC4R . . . . .	398
7.11.2.160nppiFilterLowPass_8u_C1R . . . . .	399
7.11.2.161nppiFilterLowPass_8u_C3R . . . . .	399
7.11.2.162nppiFilterLowPass_8u_C4R . . . . .	399
7.11.2.163nppiFilterLowPassBorder_16s_AC4R . . . . .	400
7.11.2.164nppiFilterLowPassBorder_16s_C1R . . . . .	400
7.11.2.165nppiFilterLowPassBorder_16s_C3R . . . . .	401
7.11.2.166nppiFilterLowPassBorder_16s_C4R . . . . .	401
7.11.2.167nppiFilterLowPassBorder_16u_AC4R . . . . .	402
7.11.2.168nppiFilterLowPassBorder_16u_C1R . . . . .	402
7.11.2.169nppiFilterLowPassBorder_16u_C3R . . . . .	403
7.11.2.170nppiFilterLowPassBorder_16u_C4R . . . . .	403
7.11.2.171nppiFilterLowPassBorder_32f_AC4R . . . . .	404
7.11.2.172nppiFilterLowPassBorder_32f_C1R . . . . .	404
7.11.2.173nppiFilterLowPassBorder_32f_C3R . . . . .	405
7.11.2.174nppiFilterLowPassBorder_32f_C4R . . . . .	405
7.11.2.175nppiFilterLowPassBorder_8u_AC4R . . . . .	406
7.11.2.176nppiFilterLowPassBorder_8u_C1R . . . . .	406
7.11.2.177nppiFilterLowPassBorder_8u_C3R . . . . .	407
7.11.2.178nppiFilterLowPassBorder_8u_C4R . . . . .	407
7.11.2.179nppiFilterPrewittHoriz_16s_AC4R . . . . .	408
7.11.2.180nppiFilterPrewittHoriz_16s_C1R . . . . .	408
7.11.2.181nppiFilterPrewittHoriz_16s_C3R . . . . .	408
7.11.2.182nppiFilterPrewittHoriz_16s_C4R . . . . .	409
7.11.2.183nppiFilterPrewittHoriz_32f_AC4R . . . . .	409
7.11.2.184nppiFilterPrewittHoriz_32f_C1R . . . . .	409
7.11.2.185nppiFilterPrewittHoriz_32f_C3R . . . . .	410
7.11.2.186nppiFilterPrewittHoriz_32f_C4R . . . . .	410
7.11.2.187nppiFilterPrewittHoriz_8u_AC4R . . . . .	410

---

7.11.2.188nppiFilterPrewittHoriz_8u_C1R . . . . .	411
7.11.2.189nppiFilterPrewittHoriz_8u_C3R . . . . .	411
7.11.2.190nppiFilterPrewittHoriz_8u_C4R . . . . .	411
7.11.2.191nppiFilterPrewittHorizBorder_16s_AC4R . . . . .	412
7.11.2.192nppiFilterPrewittHorizBorder_16s_C1R . . . . .	412
7.11.2.193nppiFilterPrewittHorizBorder_16s_C3R . . . . .	412
7.11.2.194nppiFilterPrewittHorizBorder_16s_C4R . . . . .	413
7.11.2.195nppiFilterPrewittHorizBorder_32f_AC4R . . . . .	413
7.11.2.196nppiFilterPrewittHorizBorder_32f_C1R . . . . .	414
7.11.2.197nppiFilterPrewittHorizBorder_32f_C3R . . . . .	414
7.11.2.198nppiFilterPrewittHorizBorder_32f_C4R . . . . .	415
7.11.2.199nppiFilterPrewittHorizBorder_8u_AC4R . . . . .	415
7.11.2.200nppiFilterPrewittHorizBorder_8u_C1R . . . . .	415
7.11.2.201nppiFilterPrewittHorizBorder_8u_C3R . . . . .	416
7.11.2.202nppiFilterPrewittHorizBorder_8u_C4R . . . . .	416
7.11.2.203nppiFilterPrewittVert_16s_AC4R . . . . .	417
7.11.2.204nppiFilterPrewittVert_16s_C1R . . . . .	417
7.11.2.205nppiFilterPrewittVert_16s_C3R . . . . .	418
7.11.2.206nppiFilterPrewittVert_16s_C4R . . . . .	418
7.11.2.207nppiFilterPrewittVert_32f_AC4R . . . . .	418
7.11.2.208nppiFilterPrewittVert_32f_C1R . . . . .	419
7.11.2.209nppiFilterPrewittVert_32f_C3R . . . . .	419
7.11.2.210nppiFilterPrewittVert_32f_C4R . . . . .	419
7.11.2.211nppiFilterPrewittVert_8u_AC4R . . . . .	420
7.11.2.212nppiFilterPrewittVert_8u_C1R . . . . .	420
7.11.2.213nppiFilterPrewittVert_8u_C3R . . . . .	420
7.11.2.214nppiFilterPrewittVert_8u_C4R . . . . .	421
7.11.2.215nppiFilterPrewittVertBorder_16s_AC4R . . . . .	421
7.11.2.216nppiFilterPrewittVertBorder_16s_C1R . . . . .	421
7.11.2.217nppiFilterPrewittVertBorder_16s_C3R . . . . .	422
7.11.2.218nppiFilterPrewittVertBorder_16s_C4R . . . . .	422
7.11.2.219nppiFilterPrewittVertBorder_32f_AC4R . . . . .	423
7.11.2.220nppiFilterPrewittVertBorder_32f_C1R . . . . .	423
7.11.2.221nppiFilterPrewittVertBorder_32f_C3R . . . . .	423
7.11.2.222nppiFilterPrewittVertBorder_32f_C4R . . . . .	424
7.11.2.223nppiFilterPrewittVertBorder_8u_AC4R . . . . .	424

7.11.2.224nppiFilterPrewittVertBorder_8u_C1R . . . . .	425
7.11.2.225nppiFilterPrewittVertBorder_8u_C3R . . . . .	425
7.11.2.226nppiFilterPrewittVertBorder_8u_C4R . . . . .	426
7.11.2.227nppiFilterRobertsDown_16s_AC4R . . . . .	426
7.11.2.228nppiFilterRobertsDown_16s_C1R . . . . .	426
7.11.2.229nppiFilterRobertsDown_16s_C3R . . . . .	427
7.11.2.230nppiFilterRobertsDown_16s_C4R . . . . .	427
7.11.2.231nppiFilterRobertsDown_32f_AC4R . . . . .	427
7.11.2.232nppiFilterRobertsDown_32f_C1R . . . . .	428
7.11.2.233nppiFilterRobertsDown_32f_C3R . . . . .	428
7.11.2.234nppiFilterRobertsDown_32f_C4R . . . . .	429
7.11.2.235nppiFilterRobertsDown_8u_AC4R . . . . .	429
7.11.2.236nppiFilterRobertsDown_8u_C1R . . . . .	429
7.11.2.237nppiFilterRobertsDown_8u_C3R . . . . .	430
7.11.2.238nppiFilterRobertsDown_8u_C4R . . . . .	430
7.11.2.239nppiFilterRobertsDownBorder_16s_AC4R . . . . .	430
7.11.2.240nppiFilterRobertsDownBorder_16s_C1R . . . . .	431
7.11.2.241nppiFilterRobertsDownBorder_16s_C3R . . . . .	431
7.11.2.242nppiFilterRobertsDownBorder_16s_C4R . . . . .	432
7.11.2.243nppiFilterRobertsDownBorder_32f_AC4R . . . . .	432
7.11.2.244nppiFilterRobertsDownBorder_32f_C1R . . . . .	432
7.11.2.245nppiFilterRobertsDownBorder_32f_C3R . . . . .	433
7.11.2.246nppiFilterRobertsDownBorder_32f_C4R . . . . .	433
7.11.2.247nppiFilterRobertsDownBorder_8u_AC4R . . . . .	434
7.11.2.248nppiFilterRobertsDownBorder_8u_C1R . . . . .	434
7.11.2.249nppiFilterRobertsDownBorder_8u_C3R . . . . .	435
7.11.2.250nppiFilterRobertsDownBorder_8u_C4R . . . . .	435
7.11.2.251nppiFilterRobertsUp_16s_AC4R . . . . .	435
7.11.2.252nppiFilterRobertsUp_16s_C1R . . . . .	436
7.11.2.253nppiFilterRobertsUp_16s_C3R . . . . .	436
7.11.2.254nppiFilterRobertsUp_16s_C4R . . . . .	437
7.11.2.255nppiFilterRobertsUp_32f_AC4R . . . . .	437
7.11.2.256nppiFilterRobertsUp_32f_C1R . . . . .	437
7.11.2.257nppiFilterRobertsUp_32f_C3R . . . . .	438
7.11.2.258nppiFilterRobertsUp_32f_C4R . . . . .	438
7.11.2.259nppiFilterRobertsUp_8u_AC4R . . . . .	438

7.11.2.260nppiFilterRobertsUp_8u_C1R . . . . .	439
7.11.2.261nppiFilterRobertsUp_8u_C3R . . . . .	439
7.11.2.262nppiFilterRobertsUp_8u_C4R . . . . .	439
7.11.2.263nppiFilterRobertsUpBorder_16s_AC4R . . . . .	440
7.11.2.264nppiFilterRobertsUpBorder_16s_C1R . . . . .	440
7.11.2.265nppiFilterRobertsUpBorder_16s_C3R . . . . .	440
7.11.2.266nppiFilterRobertsUpBorder_16s_C4R . . . . .	441
7.11.2.267nppiFilterRobertsUpBorder_32f_AC4R . . . . .	441
7.11.2.268nppiFilterRobertsUpBorder_32f_C1R . . . . .	442
7.11.2.269nppiFilterRobertsUpBorder_32f_C3R . . . . .	442
7.11.2.270nppiFilterRobertsUpBorder_32f_C4R . . . . .	443
7.11.2.271nppiFilterRobertsUpBorder_8u_AC4R . . . . .	443
7.11.2.272nppiFilterRobertsUpBorder_8u_C1R . . . . .	443
7.11.2.273nppiFilterRobertsUpBorder_8u_C3R . . . . .	444
7.11.2.274nppiFilterRobertsUpBorder_8u_C4R . . . . .	444
7.11.2.275nppiFilterScharHoriz_32f_C1R . . . . .	445
7.11.2.276nppiFilterScharHoriz_8s16s_C1R . . . . .	445
7.11.2.277nppiFilterScharHoriz_8u16s_C1R . . . . .	446
7.11.2.278nppiFilterScharHorizBorder_32f_C1R . . . . .	446
7.11.2.279nppiFilterScharHorizBorder_8s16s_C1R . . . . .	446
7.11.2.280nppiFilterScharHorizBorder_8u16s_C1R . . . . .	447
7.11.2.281nppiFilterScharVert_32f_C1R . . . . .	447
7.11.2.282nppiFilterScharVert_8s16s_C1R . . . . .	448
7.11.2.283nppiFilterScharVert_8u16s_C1R . . . . .	448
7.11.2.284nppiFilterScharVertBorder_32f_C1R . . . . .	448
7.11.2.285nppiFilterScharVertBorder_8s16s_C1R . . . . .	449
7.11.2.286nppiFilterScharVertBorder_8u16s_C1R . . . . .	449
7.11.2.287nppiFilterSharpen_16s_AC4R . . . . .	450
7.11.2.288nppiFilterSharpen_16s_C1R . . . . .	450
7.11.2.289nppiFilterSharpen_16s_C3R . . . . .	450
7.11.2.290nppiFilterSharpen_16s_C4R . . . . .	451
7.11.2.291nppiFilterSharpen_16u_AC4R . . . . .	451
7.11.2.292nppiFilterSharpen_16u_C1R . . . . .	451
7.11.2.293nppiFilterSharpen_16u_C3R . . . . .	452
7.11.2.294nppiFilterSharpen_16u_C4R . . . . .	452
7.11.2.295nppiFilterSharpen_32f_AC4R . . . . .	452

7.11.2.296nppiFilterSharpen_32f_C1R . . . . .	453
7.11.2.297nppiFilterSharpen_32f_C3R . . . . .	453
7.11.2.298nppiFilterSharpen_32f_C4R . . . . .	453
7.11.2.299nppiFilterSharpen_8u_AC4R . . . . .	454
7.11.2.300nppiFilterSharpen_8u_C1R . . . . .	454
7.11.2.301nppiFilterSharpen_8u_C3R . . . . .	454
7.11.2.302nppiFilterSharpen_8u_C4R . . . . .	455
7.11.2.303nppiFilterSharpenBorder_16s_AC4R . . . . .	455
7.11.2.304nppiFilterSharpenBorder_16s_C1R . . . . .	455
7.11.2.305nppiFilterSharpenBorder_16s_C3R . . . . .	456
7.11.2.306nppiFilterSharpenBorder_16s_C4R . . . . .	456
7.11.2.307nppiFilterSharpenBorder_16u_AC4R . . . . .	457
7.11.2.308nppiFilterSharpenBorder_16u_C1R . . . . .	457
7.11.2.309nppiFilterSharpenBorder_16u_C3R . . . . .	457
7.11.2.310nppiFilterSharpenBorder_16u_C4R . . . . .	458
7.11.2.311nppiFilterSharpenBorder_32f_AC4R . . . . .	458
7.11.2.312nppiFilterSharpenBorder_32f_C1R . . . . .	459
7.11.2.313nppiFilterSharpenBorder_32f_C3R . . . . .	459
7.11.2.314nppiFilterSharpenBorder_32f_C4R . . . . .	460
7.11.2.315nppiFilterSharpenBorder_8u_AC4R . . . . .	460
7.11.2.316nppiFilterSharpenBorder_8u_C1R . . . . .	460
7.11.2.317nppiFilterSharpenBorder_8u_C3R . . . . .	461
7.11.2.318nppiFilterSharpenBorder_8u_C4R . . . . .	461
7.11.2.319nppiFilterSobelCross_32f_C1R . . . . .	462
7.11.2.320nppiFilterSobelCross_8s16s_C1R . . . . .	462
7.11.2.321nppiFilterSobelCross_8u16s_C1R . . . . .	463
7.11.2.322nppiFilterSobelCrossBorder_32f_C1R . . . . .	463
7.11.2.323nppiFilterSobelCrossBorder_8s16s_C1R . . . . .	463
7.11.2.324nppiFilterSobelCrossBorder_8u16s_C1R . . . . .	464
7.11.2.325nppiFilterSobelHoriz_16s_AC4R . . . . .	464
7.11.2.326nppiFilterSobelHoriz_16s_C1R . . . . .	465
7.11.2.327nppiFilterSobelHoriz_16s_C3R . . . . .	465
7.11.2.328nppiFilterSobelHoriz_16s_C4R . . . . .	465
7.11.2.329nppiFilterSobelHoriz_32f_AC4R . . . . .	466
7.11.2.330nppiFilterSobelHoriz_32f_C1R . . . . .	466
7.11.2.331nppiFilterSobelHoriz_32f_C3R . . . . .	466

7.11.2.332nppiFilterSobelHoriz_32f_C4R . . . . .	467
7.11.2.333nppiFilterSobelHoriz_8s16s_C1R . . . . .	467
7.11.2.334nppiFilterSobelHoriz_8u16s_C1R . . . . .	467
7.11.2.335nppiFilterSobelHoriz_8u_AC4R . . . . .	468
7.11.2.336nppiFilterSobelHoriz_8u_C1R . . . . .	468
7.11.2.337nppiFilterSobelHoriz_8u_C3R . . . . .	468
7.11.2.338nppiFilterSobelHoriz_8u_C4R . . . . .	469
7.11.2.339nppiFilterSobelHorizBorder_16s_AC4R . . . . .	469
7.11.2.340nppiFilterSobelHorizBorder_16s_C1R . . . . .	469
7.11.2.341nppiFilterSobelHorizBorder_16s_C3R . . . . .	470
7.11.2.342nppiFilterSobelHorizBorder_16s_C4R . . . . .	470
7.11.2.343nppiFilterSobelHorizBorder_32f_AC4R . . . . .	471
7.11.2.344nppiFilterSobelHorizBorder_32f_C1R . . . . .	471
7.11.2.345nppiFilterSobelHorizBorder_32f_C3R . . . . .	471
7.11.2.346nppiFilterSobelHorizBorder_32f_C4R . . . . .	472
7.11.2.347nppiFilterSobelHorizBorder_8s16s_C1R . . . . .	472
7.11.2.348nppiFilterSobelHorizBorder_8u16s_C1R . . . . .	473
7.11.2.349nppiFilterSobelHorizBorder_8u_AC4R . . . . .	473
7.11.2.350nppiFilterSobelHorizBorder_8u_C1R . . . . .	474
7.11.2.351nppiFilterSobelHorizBorder_8u_C3R . . . . .	474
7.11.2.352nppiFilterSobelHorizBorder_8u_C4R . . . . .	475
7.11.2.353nppiFilterSobelHorizMask_32f_C1R . . . . .	475
7.11.2.354nppiFilterSobelHorizMaskBorder_32f_C1R . . . . .	475
7.11.2.355nppiFilterSobelHorizSecond_32f_C1R . . . . .	476
7.11.2.356nppiFilterSobelHorizSecond_8s16s_C1R . . . . .	476
7.11.2.357nppiFilterSobelHorizSecond_8u16s_C1R . . . . .	477
7.11.2.358nppiFilterSobelHorizSecondBorder_32f_C1R . . . . .	477
7.11.2.359nppiFilterSobelHorizSecondBorder_8s16s_C1R . . . . .	477
7.11.2.360nppiFilterSobelHorizSecondBorder_8u16s_C1R . . . . .	478
7.11.2.361nppiFilterSobelVert_16s_AC4R . . . . .	478
7.11.2.362nppiFilterSobelVert_16s_C1R . . . . .	479
7.11.2.363nppiFilterSobelVert_16s_C3R . . . . .	479
7.11.2.364nppiFilterSobelVert_16s_C4R . . . . .	479
7.11.2.365nppiFilterSobelVert_32f_AC4R . . . . .	480
7.11.2.366nppiFilterSobelVert_32f_C1R . . . . .	480
7.11.2.367nppiFilterSobelVert_32f_C3R . . . . .	481

7.11.2.368nppiFilterSobelVert_32f_C4R . . . . .	481
7.11.2.369nppiFilterSobelVert_8s16s_C1R . . . . .	481
7.11.2.370nppiFilterSobelVert_8u16s_C1R . . . . .	482
7.11.2.371nppiFilterSobelVert_8u_AC4R . . . . .	482
7.11.2.372nppiFilterSobelVert_8u_C1R . . . . .	482
7.11.2.373nppiFilterSobelVert_8u_C3R . . . . .	483
7.11.2.374nppiFilterSobelVert_8u_C4R . . . . .	483
7.11.2.375nppiFilterSobelVertBorder_16s_AC4R . . . . .	483
7.11.2.376nppiFilterSobelVertBorder_16s_C1R . . . . .	484
7.11.2.377nppiFilterSobelVertBorder_16s_C3R . . . . .	484
7.11.2.378nppiFilterSobelVertBorder_16s_C4R . . . . .	485
7.11.2.379nppiFilterSobelVertBorder_32f_AC4R . . . . .	485
7.11.2.380nppiFilterSobelVertBorder_32f_C1R . . . . .	485
7.11.2.381nppiFilterSobelVertBorder_32f_C3R . . . . .	486
7.11.2.382nppiFilterSobelVertBorder_32f_C4R . . . . .	486
7.11.2.383nppiFilterSobelVertBorder_8s16s_C1R . . . . .	487
7.11.2.384nppiFilterSobelVertBorder_8u16s_C1R . . . . .	487
7.11.2.385nppiFilterSobelVertBorder_8u_AC4R . . . . .	488
7.11.2.386nppiFilterSobelVertBorder_8u_C1R . . . . .	488
7.11.2.387nppiFilterSobelVertBorder_8u_C3R . . . . .	489
7.11.2.388nppiFilterSobelVertBorder_8u_C4R . . . . .	489
7.11.2.389nppiFilterSobelVertMask_32f_C1R . . . . .	489
7.11.2.390nppiFilterSobelVertMaskBorder_32f_C1R . . . . .	490
7.11.2.391nppiFilterSobelVertSecond_32f_C1R . . . . .	490
7.11.2.392nppiFilterSobelVertSecond_8s16s_C1R . . . . .	491
7.11.2.393nppiFilterSobelVertSecond_8u16s_C1R . . . . .	491
7.11.2.394nppiFilterSobelVertSecondBorder_32f_C1R . . . . .	492
7.11.2.395nppiFilterSobelVertSecondBorder_8s16s_C1R . . . . .	492
7.11.2.396nppiFilterSobelVertSecondBorder_8u16s_C1R . . . . .	493
7.11.2.397nppiFilterUnsharpBorder_16s_AC4R . . . . .	493
7.11.2.398nppiFilterUnsharpBorder_16s_C1R . . . . .	494
7.11.2.399nppiFilterUnsharpBorder_16s_C3R . . . . .	494
7.11.2.400nppiFilterUnsharpBorder_16s_C4R . . . . .	495
7.11.2.401nppiFilterUnsharpBorder_16u_AC4R . . . . .	495
7.11.2.402nppiFilterUnsharpBorder_16u_C1R . . . . .	496
7.11.2.403nppiFilterUnsharpBorder_16u_C3R . . . . .	496

7.11.2.404nppiFilterUnsharpBorder_16u_C4R . . . . .	497
7.11.2.405nppiFilterUnsharpBorder_32f_AC4R . . . . .	498
7.11.2.406nppiFilterUnsharpBorder_32f_C1R . . . . .	498
7.11.2.407nppiFilterUnsharpBorder_32f_C3R . . . . .	499
7.11.2.408nppiFilterUnsharpBorder_32f_C4R . . . . .	499
7.11.2.409nppiFilterUnsharpBorder_8u_AC4R . . . . .	500
7.11.2.410nppiFilterUnsharpBorder_8u_C1R . . . . .	500
7.11.2.411nppiFilterUnsharpBorder_8u_C3R . . . . .	501
7.11.2.412nppiFilterUnsharpBorder_8u_C4R . . . . .	502
7.11.2.413nppiFilterUnsharpGetBufferSize_16s_AC4R . . . . .	502
7.11.2.414nppiFilterUnsharpGetBufferSize_16s_C1R . . . . .	502
7.11.2.415nppiFilterUnsharpGetBufferSize_16s_C3R . . . . .	503
7.11.2.416nppiFilterUnsharpGetBufferSize_16s_C4R . . . . .	503
7.11.2.417nppiFilterUnsharpGetBufferSize_16u_AC4R . . . . .	503
7.11.2.418nppiFilterUnsharpGetBufferSize_16u_C1R . . . . .	504
7.11.2.419nppiFilterUnsharpGetBufferSize_16u_C3R . . . . .	504
7.11.2.420nppiFilterUnsharpGetBufferSize_16u_C4R . . . . .	504
7.11.2.421nppiFilterUnsharpGetBufferSize_32f_AC4R . . . . .	504
7.11.2.422nppiFilterUnsharpGetBufferSize_32f_C1R . . . . .	505
7.11.2.423nppiFilterUnsharpGetBufferSize_32f_C3R . . . . .	505
7.11.2.424nppiFilterUnsharpGetBufferSize_32f_C4R . . . . .	505
7.11.2.425nppiFilterUnsharpGetBufferSize_8u_AC4R . . . . .	506
7.11.2.426nppiFilterUnsharpGetBufferSize_8u_C1R . . . . .	506
7.11.2.427nppiFilterUnsharpGetBufferSize_8u_C3R . . . . .	506
7.11.2.428nppiFilterUnsharpGetBufferSize_8u_C4R . . . . .	506
7.11.2.429nppiGetFilterGaussPyramidLayerDownBorderDstROI . . . . .	507
7.11.2.430nppiGetFilterGaussPyramidLayerUpBorderDstROI . . . . .	507
7.11.2.431nppiGradientVectorPrewittBorder_16s32f_C1R . . . . .	508
7.11.2.432nppiGradientVectorPrewittBorder_16s32f_C3C1R . . . . .	508
7.11.2.433nppiGradientVectorPrewittBorder_16u32f_C1R . . . . .	509
7.11.2.434nppiGradientVectorPrewittBorder_16u32f_C3C1R . . . . .	510
7.11.2.435nppiGradientVectorPrewittBorder_32f_C1R . . . . .	511
7.11.2.436nppiGradientVectorPrewittBorder_32f_C3C1R . . . . .	511
7.11.2.437nppiGradientVectorPrewittBorder_8u16s_C1R . . . . .	512
7.11.2.438nppiGradientVectorPrewittBorder_8u16s_C3C1R . . . . .	513
7.11.2.439nppiGradientVectorScharrBorder_16s32f_C1R . . . . .	513

---

7.11.2.440nppiGradientVectorScharrBorder_16s32f_C3C1R . . . . .	514
7.11.2.441nppiGradientVectorScharrBorder_16u32f_C1R . . . . .	515
7.11.2.442nppiGradientVectorScharrBorder_16u32f_C3C1R . . . . .	515
7.11.2.443nppiGradientVectorScharrBorder_32f_C1R . . . . .	516
7.11.2.444nppiGradientVectorScharrBorder_32f_C3C1R . . . . .	517
7.11.2.445nppiGradientVectorScharrBorder_8u16s_C1R . . . . .	517
7.11.2.446nppiGradientVectorScharrBorder_8u16s_C3C1R . . . . .	518
7.11.2.447nppiGradientVectorSobelBorder_16s32f_C1R . . . . .	519
7.11.2.448nppiGradientVectorSobelBorder_16s32f_C3C1R . . . . .	519
7.11.2.449nppiGradientVectorSobelBorder_16u32f_C1R . . . . .	520
7.11.2.450nppiGradientVectorSobelBorder_16u32f_C3C1R . . . . .	521
7.11.2.451nppiGradientVectorSobelBorder_32f_C1R . . . . .	521
7.11.2.452nppiGradientVectorSobelBorder_32f_C3C1R . . . . .	522
7.11.2.453nppiGradientVectorSobelBorder_8u16s_C1R . . . . .	523
7.11.2.454nppiGradientVectorSobelBorder_8u16s_C3C1R . . . . .	523
<b>8 Data Structure Documentation . . . . .</b>	<b>525</b>
8.1 NPP_ALIGN_16 Struct Reference . . . . .	525
8.1.1 Detailed Description . . . . .	525
8.1.2 Field Documentation . . . . .	525
8.1.2.1 im . . . . .	525
8.1.2.2 im . . . . .	526
8.1.2.3 re . . . . .	526
8.1.2.4 re . . . . .	526
8.2 NPP_ALIGN_8 Struct Reference . . . . .	527
8.2.1 Detailed Description . . . . .	527
8.2.2 Field Documentation . . . . .	527
8.2.2.1 im . . . . .	527
8.2.2.2 im . . . . .	527
8.2.2.3 im . . . . .	527
8.2.2.4 re . . . . .	528
8.2.2.5 re . . . . .	528
8.2.2.6 re . . . . .	528
8.3 NppiHaarBuffer Struct Reference . . . . .	529
8.3.1 Field Documentation . . . . .	529
8.3.1.1 haarBuffer . . . . .	529
8.3.1.2 haarBufferSize . . . . .	529

8.4	NppiHaarClassifier_32f Struct Reference . . . . .	530
8.4.1	Field Documentation . . . . .	530
8.4.1.1	classifiers . . . . .	530
8.4.1.2	classifierSize . . . . .	530
8.4.1.3	classifierStep . . . . .	530
8.4.1.4	counterDevice . . . . .	530
8.4.1.5	numClassifiers . . . . .	530
8.5	NppiPoint Struct Reference . . . . .	531
8.5.1	Detailed Description . . . . .	531
8.5.2	Field Documentation . . . . .	531
8.5.2.1	x . . . . .	531
8.5.2.2	y . . . . .	531
8.6	NppiRect Struct Reference . . . . .	532
8.6.1	Detailed Description . . . . .	532
8.6.2	Field Documentation . . . . .	532
8.6.2.1	height . . . . .	532
8.6.2.2	width . . . . .	532
8.6.2.3	x . . . . .	532
8.6.2.4	y . . . . .	532
8.7	NppiSize Struct Reference . . . . .	533
8.7.1	Detailed Description . . . . .	533
8.7.2	Field Documentation . . . . .	533
8.7.2.1	height . . . . .	533
8.7.2.2	width . . . . .	533
8.8	NppLibraryVersion Struct Reference . . . . .	534
8.8.1	Field Documentation . . . . .	534
8.8.1.1	build . . . . .	534
8.8.1.2	major . . . . .	534
8.8.1.3	minor . . . . .	534



# Chapter 1

## NVIDIA Performance Primitives

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, Android, 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. The libnppi library is becoming quite large so to minimize library loading and CUDA runtime startup times it is recommended to use the static library(s) whenever possible. To improve loading and runtime performance when using dynamic libraries NPP 8.0 now includes the full set of nppi sub-libraries in addition to the full sized nppi library itself. Linking to only the sub-libraries that contain functions that your application uses can significantly improve load time and runtime startup performance. Some nppi functions make calls to other nppi and/or npps functions internally so you may need to link to a few extra libraries depending on what function calls your application makes. The nppi sub-libraries are split into sections corresponding to the way that nppi header files are split. There are also static versions of each of the new sub-libraries. The full sized nppi library will be deprecated in the next CUDA release. This list of sub-libraries is as follows:

注意：根据头文件的结构，分成不同子库

```
nppial arithmetic and logical operation functions in nppi_arithmetic_and_logical_operations.h  
nppicc color conversion and sampling functions in nppi_color_conversion.h  
nppicom JPEG compression and decompression functions in nppi_compression_functions.h  
nppidei data exchange and initialization functions in nppi_data_exchange_and_initialization.h  
nppif filtering and computer vision functions in nppi_filter_functions.h  
nppig geometry transformation functions found in nppi_geometry_transforms.h  
nppim morphological operation functions found in nppi_morphological_operations.h  
nppist statistics and linear transform in nppi_statistics_functions.h and nppi_linear_transforms.h  
nppisu memory support functions in nppi_support_functions.h  
nppitc threshold and compare operation functions in nppi_threshold_and_compare_operations.h
```

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 current stream ID, i.e. the stream ID that was set in the most recent nppSetStream call. The default stream ID is 0. If an application intends to use NPP with multiple streams then it is the responsibility of the application to call nppSetStream whenever it wishes to change stream IDs. Several NPP functions may call other NPP functions internally to complete their functionality. For this reason it is recommended that cudaDeviceSynchronize be called before making an nppSetStream call to change to a new stream ID. This will insure that any internal function calls that have not yet occurred will be completed using the current stream ID before it changes to a new ID. Calling cudaDeviceSynchronize frequently call kill performance so minimizing the frequency of these calls is critical for good performance. It is not necessary to call cudaDeviceSynchronize for stream management while the same stream ID is used for multiple NPP calls. All NPP functions should be thread safe except for the following functions:

```
nppiGraphcut_32s8u - this function has been deprecated in NPP 8.0
nppiGraphcut_32f8u - this function has been deprecated in NPP 8.0
nppiGraphcut8_32s8u - this function has been deprecated in NPP 8.0
nppiGraphcut8_32f8u - this function has been deprecated in NPP 8.0
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 nppiMeanStdDev8uC1RGetBufferSize has been renamed nppiMeanStdDevGetBufferSize\_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, 8, and 10 (64-bit and 32-bit)
- Microsoft Windows Vista (64-bit and 32-bit)
- Linux (Centos, Ubuntu, and several others) (64-bit and 32-bit)
- Mac OS X (64-bit)
- Android on Arm (32-bit and 64-bit)

## 1.4 Files

NPP is comprises 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](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.

注意：所有形参中的指针都是代表设备指针

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 magnitude (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  $2^{-8} = \frac{1}{256} = \frac{1}{256}$ . 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, pScr2, ...`

#### 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, pScr2, ...`

##### 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.

#### 4.2.2.1 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 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.

#### 4.2.2.3 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, ...`

#### 4.2.2.4 Destination-Image Line Step

The destination image line step parameter is

`nDstStep`

or in the case of multiple destination images

`nDstStep1, nDstStep2, ...`

#### 4.2.2.5 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[]`

#### 4.2.2.6 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-left corner (lowest memory address), 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 (`x`, `y`), one would pass

`pSrcOffset = pSrc + y * nSrcStep + x * 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 value 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 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 `image_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 . . . . .	27
NPP Type Definitions and Constants . . . . .	31
Basic NPP Data Types . . . . .	46
Filtering Functions . . . . .	50
1D Linear Filter . . . . .	51
1D Window Sum . . . . .	123
1D Window Sum with Border Control . . . . .	134
Convolution . . . . .	147
2D Fixed Linear Filters . . . . .	206
Rank Filters . . . . .	224
Fixed Filters . . . . .	274



# Chapter 6

## Data Structure Index

### 6.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">NPP_ALIGN_16</a> (Complex Number This struct represents a long long complex number ) . . . . .	525
<a href="#">NPP_ALIGN_8</a> (Complex Number This struct represents an unsigned int complex number ) . . . . .	527
<a href="#">NppiHaarBuffer</a> . . . . .	529
<a href="#">NppiHaarClassifier_32f</a> . . . . .	530
<a href="#">NppiPoint</a> (2D Point ) . . . . .	531
<a href="#">NppiRect</a> (2D Rectangle This struct contains position and size information of a rectangle in two space ) . . . . .	532
<a href="#">NppiSize</a> (2D Size This struct typically represents the size of a a rectangular region in two space )	533
<a href="#">NppLibraryVersion</a> . . . . .	534



# 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.*
- `int nppGetGpuDeviceProperties (int *pMaxThreadsPerSM, int *pMaxThreadsPerBlock, int *pNumberOfSMs)`  
*Get the maximum number of threads per SM, maximum threads per block, and number of SMs 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.*
- `unsigned int nppGetStreamNumSMs (void)`  
*Get the number of SMs on the device associated with the current NPP CUDA stream.*

- `unsigned int nppGetStreamMaxThreadsPerSM (void)`

*Get the maximum number of threads per SM on the device associated with the current 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 `int nppGetGpuDeviceProperties (int * pMaxThreadsPerSM, int * pMaxThreadsPerBlock, int * pNumberOfSMs)`

Get the maximum number of threads per SM, maximum threads per block, and number of SMs for the active GPU.

**Returns:**

`cudaSuccess` for success, -1 for failure

#### 7.1.2.3 `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.

#### 7.1.2.4 `int nppGetGpuNumSMs (void)`

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

**Returns:**

Number of SMs of the default CUDA device.

**7.1.2.5 const NppLibraryVersion\* nppGetLibVersion (void)**

Get the NPP library version.

**Returns:**

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

**7.1.2.6 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.

**7.1.2.7 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

**7.1.2.8 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.

**7.1.2.9 unsigned int nppGetStreamMaxThreadsPerSM (void)**

Get the maximum number of threads per SM on the device associated with the current 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. This call avoids a `cudaGetDeviceProperties()` call.

**7.1.2.10 unsigned int nppGetStreamNumSMs (void)**

Get the number of SMs on the device associated with the current 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. This call avoids a `cudaGetDeviceProperties()` call.

### 7.1.2.11 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 `NppiDifferentialKernel` {  
    `NPP_FILTER_SOBEL`,  
    `NPP_FILTER_SCHARR` }

*Differential Filter types.*

- 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_OF_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 [Nppi HuffmanTableType](#) {  
    [nppiDCTable](#),  
    [nppiACTable](#) }

- enum `NppiNorm` {  
    `nppiNormInf` = 0,  
    `nppiNormL1` = 1,  
    `nppiNormL2` = 2 }

## 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 NppiDifferentialKernel

Differential Filter types.

Enumerator:

*NPP\_FILTER\_SOBEL*  
*NPP\_FILTER\_SCHARR*

### 7.2.2.9 enum NppiHuffmanTableType

Enumerator:

*nppiDCTable* DC Table.

*nppiACTable* AC Table.

### 7.2.2.10 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.

### 7.2.2.11 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*

### 7.2.2.12 enum NppiNorm

**Enumerator:**

- nppiNormInf* maximum
- nppiNormL1* sum
- nppiNormL2* square root of sum of squares

### 7.2.2.13 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. <integer>.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](#).

### 7.2.2.14 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\_OF\_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.

### 7.2.2.15 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 Filtering Functions

Linear and non-linear image filtering functions.

### Modules

- [1D Linear Filter](#)
- [1D Window Sum](#)
- [1D Window Sum with Border Control](#)
- [Convolution](#)
- [2D Fixed Linear Filters](#)
- [Rank Filters](#)
- [Fixed Filters](#)

*Fixed filters perform linear filtering operations (such as convolutions) with predefined kernels of fixed sizes.*

### 7.4.1 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](#).

These functions can be found in either the nppi or nppif libraries. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

## 7.5 1D Linear Filter

### 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 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.*

## 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`, `NppSize oSrcSize`, `NppPoint oSrcOffset`, `Npp8u *pDst`, int `nDstStep`, `NppSize 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`, `NppSize oSrcSize`, `NppPoint oSrcOffset`, `Npp8u *pDst`, int `nDstStep`, `NppSize 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`, `NppSize oSrcSize`, `NppPoint oSrcOffset`, `Npp8u *pDst`, int `nDstStep`, `NppSize 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`, `NppSize oSrcSize`, `NppPoint oSrcOffset`, `Npp8u *pDst`, int `nDstStep`, `NppSize 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`, `NppSize oSrcSize`, `NppPoint oSrcOffset`, `Npp16u *pDst`, int `nDstStep`, `NppSize 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`, `NppSize oSrcSize`, `NppPoint oSrcOffset`, `Npp16u *pDst`, int `nDstStep`, `NppSize 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`, `NppSize oSrcSize`, `NppPoint oSrcOffset`, `Npp16u *pDst`, int `nDstStep`, `NppSize 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`, `NppSize oSrcSize`, `NppPoint oSrcOffset`, `Npp16u *pDst`, int `nDstStep`, `NppSize 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`, `NppSize oSrcSize`, `NppPoint oSrcOffset`, `Npp16s *pDst`, int `nDstStep`, `NppSize 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`, `NppSize oSrcSize`, `NppPoint oSrcOffset`, `Npp16s *pDst`, int `nDstStep`, `NppSize 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 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 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 ID 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 ID 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 ID 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 ID 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 ID 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 ID 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 ID 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 ID 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*, **NppSize** *oSrcSize*, **NppPoint** *oSrcOffset*, **Npp16s** \**pDst*, int *nDstStep*, **NppSize** *oSizeROI*, const **Npp32f** \**pKernel*, **Npp32s** *nMaskSize*, **Npp32s** *nAnchor*, **NppBorderType** *eBorderType*)

*Three channel 16-bit 1D row convolution filter with border control.*

- **NppStatus nppiFilterRowBorder32f\_16s\_C4R** (const **Npp16s** \**pSrc*, int *nSrcStep*, **NppSize** *oSrcSize*, **NppPoint** *oSrcOffset*, **Npp16s** \**pDst*, int *nDstStep*, **NppSize** *oSizeROI*, const **Npp32f** \**pKernel*, **Npp32s** *nMaskSize*, **Npp32s** *nAnchor*, **NppBorderType** *eBorderType*)

*Four channel 16-bit 1D row convolution filter with border control.*

- **NppStatus nppiFilterRowBorder32f\_16s\_AC4R** (const **Npp16s** \**pSrc*, int *nSrcStep*, **NppSize** *oSrcSize*, **NppPoint** *oSrcOffset*, **Npp16s** \**pDst*, int *nDstStep*, **NppSize** *oSizeROI*, const **Npp32f** \**pKernel*, **Npp32s** *nMaskSize*, **Npp32s** *nAnchor*, **NppBorderType** *eBorderType*)

*Four channel 16-bit 1D row convolution filter with border control, ignoring alpha channel.*

## 7.5.1 Function Documentation

### 7.5.1.1 NppStatus nppiFilterColumn32f\_16s\_AC4R (const **Npp16s** \**pSrc*, **Npp32s** *nSrcStep*, **Npp16s** \**pDst*, **Npp32s** *nDstStep*, **NppSize** *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](#)

### 7.5.1.2 NppStatus nppiFilterColumn32f\_16s\_C1R (const **Npp16s** \**pSrc*, **Npp32s** *nSrcStep*, **Npp16s** \**pDst*, **Npp32s** *nDstStep*, **NppSize** *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](#)

**7.5.1.3 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](#)

**7.5.1.4 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](#)

**7.5.1.5 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](#)

**7.5.1.6 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](#)

**7.5.1.7 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](#)

**7.5.1.8 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](#)

**7.5.1.9 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](#)

**7.5.1.10 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](#)

---

**7.5.1.11 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](#)

---

**7.5.1.12 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](#)

**7.5.1.13 NppStatus nppiFilterColumn\_16s\_AC4R (const Npp16s \**pSrc*, Npp32s *nSrcStep*,  
Npp16s \**pDst*, Npp32s *nDstStep*, NppSize *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](#)

**7.5.1.14 NppStatus nppiFilterColumn\_16s\_C1R (const Npp16s \**pSrc*, Npp32s *nSrcStep*, Npp16s  
\**pDst*, Npp32s *nDstStep*, NppSize *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](#)

**7.5.1.15 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](#)

**7.5.1.16 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](#)

### 7.5.1.17 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](#)

### 7.5.1.18 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](#)

**7.5.1.19 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](#)

**7.5.1.20 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](#)

**7.5.1.21 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](#)

**7.5.1.22 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](#)

**7.5.1.23 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](#)

**7.5.1.24 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](#)

**7.5.1.25 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](#)

**7.5.1.26 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](#)

**7.5.1.27 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](#)

**7.5.1.28 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](#)

**7.5.1.29 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](#)

**7.5.1.30 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](#)

---

**7.5.1.31 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. Coeffcients 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](#)

**7.5.1.32 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. Coeffcients 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](#)

**7.5.1.33 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. Coeffcients 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](#)

**7.5.1.34 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. Coeffcients 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](#)

---

**7.5.1.35 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. Coeffcients 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](#)

**7.5.1.36 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. Coeffcients 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](#)

**7.5.1.37 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. Coeffcients 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](#)

**7.5.1.38 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, 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. Coeffcients 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](#)

---

**7.5.1.39 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. Coeffcients 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](#)

**7.5.1.40 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. Coeffcients 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](#)

**7.5.1.41 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. Coeffcients 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](#)

**7.5.1.42 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. Coeffcients 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](#)

**7.5.1.43 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](#)

**7.5.1.44 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. Coeffcients 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](#)

**7.5.1.45 NppStatus nppiFilterColumnBorder\_16s\_C4R (const Npp16s \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppPoint 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. Coeffcients 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](#)

---

**7.5.1.46 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. Coeffcients 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](#)

---

**7.5.1.47 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. Coeffcients 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](#)

**7.5.1.48 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. Coeffcients 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](#)

**7.5.1.49 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.

*nSrcStep* The pixel offset that pSrc 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. Coeffcients 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](#)

**7.5.1.50 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. Coeffcients 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](#)

**7.5.1.51 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. Coeffcients 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](#)

**7.5.1.52 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. Coeffcients 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](#)

---

**7.5.1.53 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](#)

---

**7.5.1.54 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](#)

**7.5.1.55 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](#)

**7.5.1.56 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. Coeffcients 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](#)

**7.5.1.57 NppStatus nppiFilterColumnBorder\_8u\_C4R (const Npp8u \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppPoint 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. Coeffcients 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](#)

**7.5.1.58 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](#)

**7.5.1.59 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](#)

**7.5.1.60 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](#)

**7.5.1.61 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](#)

**7.5.1.62 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](#)

**7.5.1.63 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](#)

**7.5.1.64 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](#)

**7.5.1.65 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](#)

**7.5.1.66 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](#)

**7.5.1.67 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](#)

### 7.5.1.68 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](#)

### 7.5.1.69 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](#)

### 7.5.1.70 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](#)

### 7.5.1.71 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](#)

### 7.5.1.72 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](#)

### 7.5.1.73 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](#)

**7.5.1.74 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](#)

**7.5.1.75 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](#)

---

**7.5.1.76 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](#)

---

**7.5.1.77 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](#)

**7.5.1.78 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](#)

**7.5.1.79 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](#)

**7.5.1.80 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](#)

**7.5.1.81 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](#)

**7.5.1.82 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](#)

**7.5.1.83 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](#)

**7.5.1.84 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](#)

**7.5.1.85 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](#)

**7.5.1.86 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](#)

**7.5.1.87 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](#)

---

**7.5.1.88 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. Coeffcients 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](#)

---

**7.5.1.89 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. Coeffcients 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](#)

**7.5.1.90 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. Coeffcients 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](#)

**7.5.1.91 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. Coeffcients 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](#)

---

**7.5.1.92 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. Coeffcients 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](#)

---

**7.5.1.93 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. Coeffcients 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](#)

**7.5.1.94 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. Coeffcients 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](#)

**7.5.1.95 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. Coeffcients 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](#)

---

**7.5.1.96 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. Coeffcients 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](#)

---

**7.5.1.97 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. Coeffcients 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](#)

**7.5.1.98 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. Coeffcients 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](#)

**7.5.1.99 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. Coeffcients 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](#)

**7.5.1.100 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](#)

**7.5.1.101 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. Coeffcients 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](#)

**7.5.1.102 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. Coeffcients 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](#)

---

**7.5.1.103 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. Coeffcients 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](#)

---

**7.5.1.104 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. Coeffcients 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](#)

**7.5.1.105 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. Coeffcients 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](#)

**7.5.1.106 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. Coeffcients 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](#)

**7.5.1.107 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. Coeffcients 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](#)

**7.5.1.108 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. Coeffcients 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](#)

**7.5.1.109 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. Coeffcients 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](#)

---

**7.5.1.110 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](#)

---

**7.5.1.111 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](#)

**7.5.1.112 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](#)

**7.5.1.113 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. Coeffcients 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](#)

**7.5.1.114 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. Coeffcients 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](#)

## 7.6 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 ID (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 ID (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 ID (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 ID (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 ID (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 ID (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 ID (row) sum to 32f.*

## 7.6.1 Function Documentation

### 7.6.1.1 `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](#)

**7.6.1.2 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](#)

**7.6.1.3 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](#)

#### 7.6.1.4 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](#)

#### 7.6.1.5 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](#)

### 7.6.1.6 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](#)

### 7.6.1.7 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](#)

### 7.6.1.8 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](#)

### 7.6.1.9 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](#)

**7.6.1.10 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](#)

**7.6.1.11 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](#)

### 7.6.1.12 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](#)

### 7.6.1.13 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](#)

**7.6.1.14 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](#)

**7.6.1.15 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](#)

### 7.6.1.16 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](#)

### 7.6.1.17 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](#)

**7.6.1.18 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](#)

## 7.7 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, NppSize oSrcSize, NppPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppSize 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, NppSize oSrcSize, NppPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppSize 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, NppSize oSrcSize, NppPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppSize 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, NppSize oSrcSize, NppPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppSize 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, NppSize oSrcSize, NppPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppSize 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, NppSize oSrcSize, NppPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppSize 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, NppSize oSrcSize, NppPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppSize 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, NppSize oSrcSize, NppPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppSize 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.*

### 7.7.1 Function Documentation

#### 7.7.1.1 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](#)

#### 7.7.1.2 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](#)

**7.7.1.3 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](#)

**7.7.1.4 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](#)

### 7.7.1.5 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.

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](#)

### 7.7.1.6 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](#)

### 7.7.1.7 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](#)

**7.7.1.8 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](#)

**7.7.1.9 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.

*nSrcStep* 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](#)

**7.7.1.10 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](#)

**7.7.1.11 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](#)

**7.7.1.12 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](#)

**7.7.1.13 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](#)

**7.7.1.14 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](#)

**7.7.1.15 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](#)

**7.7.1.16 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.

*nSrcStep* 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](#)

#### 7.7.1.17 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](#)

#### 7.7.1.18 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.

*nSrcSize* Source image width and height in pixels relative to pSrc.

*nSrcOffset* The pixel offset that pSrc points to relative to the origin of the source image.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*nROI* 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](#)

## 7.8 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 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 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 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.

注意：濾波操作

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, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** oKernelSize, **NppiPoint** oAnchor)

*Two channel 8-bit unsigned convolution filter.*

- **NppStatus nppiFilter32f\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** oKernelSize, **NppiPoint** oAnchor)

*Three channel 8-bit unsigned convolution filter.*

- **NppStatus nppiFilter32f\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** oKernelSize, **NppiPoint** oAnchor)

*Four channel 8-bit unsigned convolution filter.*

- **NppStatus nppiFilter32f\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** oKernelSize, **NppiPoint** oAnchor)

*Four channel 8-bit unsigned convolution filter, ignoring alpha channel.*

- **NppStatus nppiFilter32f\_8s\_C1R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** oKernelSize, **NppiPoint** oAnchor)

*Single channel 8-bit signed convolution filter.*

- **NppStatus nppiFilter32f\_8s\_C2R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** oKernelSize, **NppiPoint** oAnchor)

*Two channel 8-bit signed convolution filter.*

- **NppStatus nppiFilter32f\_8s\_C3R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** oKernelSize, **NppiPoint** oAnchor)

*Three channel 8-bit signed convolution filter.*

- **NppStatus nppiFilter32f\_8s\_C4R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** oKernelSize, **NppiPoint** oAnchor)

*Four channel 8-bit signed convolution filter.*

- **NppStatus nppiFilter32f\_8s\_AC4R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** 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, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** oKernelSize, **NppiPoint** oAnchor)

*Single channel 16-bit unsigned convolution filter.*

- **NppStatus nppiFilter32f\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** oKernelSize, **NppiPoint** oAnchor)

*Three channel 16-bit unsigned convolution filter.*

- **NppStatus nppiFilter32f\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppSize** oSizeROI, const **Npp32f** \*pKernel, **NppSize** 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 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 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 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, ignoring 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.*

## 7.8.1 Function Documentation

### 7.8.1.1 `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. Coeffcients 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](#)

**7.8.1.2 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. Coeffcients 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](#)

**7.8.1.3 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. Coeffcients 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](#)

#### 7.8.1.4 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. Coeffcients 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](#)

#### 7.8.1.5 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. Coeffcients 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](#)

---

**7.8.1.6 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. Coeffcients 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](#)

---

**7.8.1.7 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. Coeffcients 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](#)

### 7.8.1.8 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. Coeffcients 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](#)

### 7.8.1.9 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. Coeffcients 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](#)

**7.8.1.10 NppStatus nppiFilter32f\_32s\_C1R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppSize oSizeROI, const Npp32f \* pKernel, NppSize 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. Coeffcients 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](#)

**7.8.1.11 NppStatus nppiFilter32f\_32s\_C3R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppSize oSizeROI, const Npp32f \* pKernel, NppSize 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. Coeffcients 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](#)

### 7.8.1.12 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. Coeffcients 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](#)

### 7.8.1.13 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. Coeffcients 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](#)

**7.8.1.14 NppStatus nppiFilter32f\_8s16s\_C1R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppSize *oSizeROI*, const Npp32f \* *pKernel*, NppSize *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](#)

**7.8.1.15 NppStatus nppiFilter32f\_8s16s\_C3R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppSize *oSizeROI*, const Npp32f \* *pKernel*, NppSize *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](#)

### 7.8.1.16 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. Coeffcients 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](#)

### 7.8.1.17 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. Coeffcients 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](#)

**7.8.1.18 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. Coeffcients 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](#)

**7.8.1.19 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. Coeffcients 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](#)

### 7.8.1.20 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. Coeffcients 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](#)

### 7.8.1.21 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. Coeffcients 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](#)

**7.8.1.22 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. Coeffcients 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](#)

**7.8.1.23 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. Coeffcients 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](#)

### 7.8.1.24 NppStatus nppiFilter32f\_8u16s\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppSize *oSizeROI*, const Npp32f \* *pKernel*, NppSize *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. Coeffcients 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](#)

### 7.8.1.25 NppStatus nppiFilter32f\_8u16s\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppSize *oSizeROI*, const Npp32f \* *pKernel*, NppSize *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. Coeffcients 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](#)

**7.8.1.26 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, 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](#)

**7.8.1.27 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](#)

**7.8.1.28 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. Coeffcients 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](#)

**7.8.1.29 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. Coeffcients 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](#)

### 7.8.1.30 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. Coeffcients 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](#)

### 7.8.1.31 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. Coeffcients 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](#)

### 7.8.1.32 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. Coeffcients 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](#)

### 7.8.1.33 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. Coeffcients 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](#)

### 7.8.1.34 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. Coeffcients 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](#)

### 7.8.1.35 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. Coeffcients 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](#)

**7.8.1.36 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. Coeffcients 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](#)

**7.8.1.37 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. Coeffcients 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](#)

**7.8.1.38 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. Coeffcients 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](#)

**7.8.1.39 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. Coeffcients 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](#)

### 7.8.1.40 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. Coeffcients 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](#)

### 7.8.1.41 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. Coeffcients 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](#)

**7.8.1.42 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. Coeffcients 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](#)

**7.8.1.43 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. Coeffcients 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](#)

#### 7.8.1.44 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. Coeffcients 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](#)

#### 7.8.1.45 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. Coeffcients 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](#)

**7.8.1.46 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. Coeffcients 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](#)

**7.8.1.47 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. Coeffcients 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](#)

**7.8.1.48 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. Coeffcients 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](#)

**7.8.1.49 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. Coeffcients 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](#)

**7.8.1.50 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. Coeffcients 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](#)

**7.8.1.51 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. Coeffcients 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](#)

### 7.8.1.52 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. Coeffcients 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](#)

### 7.8.1.53 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. Coeffcients 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](#)

**7.8.1.54 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. Coeffcients 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](#)

**7.8.1.55 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. Coeffcients 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](#)

### 7.8.1.56 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. Coeffcients 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](#)

### 7.8.1.57 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. Coeffcients 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](#)

**7.8.1.58 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. Coeffcients 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](#)

**7.8.1.59 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. Coeffcients 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](#)

**7.8.1.60 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. Coeffcients 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](#)

**7.8.1.61 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. Coeffcients 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](#)

**7.8.1.62 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. Coeffcients 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](#)

**7.8.1.63 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. Coeffcients 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](#)

---

**7.8.1.64 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. Coeffcients 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](#)

**7.8.1.65 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. Coeffcients 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](#)

**7.8.1.66 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. Coeffcients 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](#)

**7.8.1.67 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. Coeffcients 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](#)

---

**7.8.1.68 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. Coeffcients 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](#)

---

**7.8.1.69 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. Coeffcients 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](#)

**7.8.1.70 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. Coeffcients 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](#)

**7.8.1.71 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. Coeffcients 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](#)

---

**7.8.1.72 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. Coeffcients 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](#)

**7.8.1.73 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. Coeffcients 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](#)

**7.8.1.74 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, 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](#)

**7.8.1.75 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](#)

---

**7.8.1.76 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. Coeffcients 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](#)

---

**7.8.1.77 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. Coeffcients 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](#)

**7.8.1.78 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. Coeffcients 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](#)

**7.8.1.79 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. Coeffcients 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](#)

**7.8.1.80 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. Coeffcients 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](#)

**7.8.1.81 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. Coeffcients 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](#)

**7.8.1.82 NppStatus nppiFilterBorder\_16s\_C4R (const Npp16s \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppPoint oSrcOffset, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s \* pKernel, NppiSize oKernelSize, NppPoint 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. Coeffcients 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](#)

---

**7.8.1.83 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. Coeffcients 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](#)

---

**7.8.1.84 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. Coeffcients 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](#)

**7.8.1.85 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. Coeffcients 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](#)

**7.8.1.86 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. Coeffcients 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](#)

**7.8.1.87 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. Coeffcients 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](#)

**7.8.1.88 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. Coeffcients 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](#)

**7.8.1.89 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. Coeffcients 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](#)

**7.8.1.90 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. Coeffcients 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](#)

**7.8.1.91 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. Coeffcients 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](#)

**7.8.1.92 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. Coeffcients 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](#)

**7.8.1.93 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. Coeffcients 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](#)

**7.8.1.94 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. Coeffcients 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](#)

**7.8.1.95 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. Coeffcients 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](#)

## 7.9 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_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 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.*

## 7.9.1 Function Documentation

### 7.9.1.1 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, ignorting 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](#)

### 7.9.1.2 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](#)

### 7.9.1.3 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](#)

### 7.9.1.4 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](#)

### 7.9.1.5 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](#)

#### 7.9.1.6 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](#)

#### 7.9.1.7 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](#)

### 7.9.1.8 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](#)

### 7.9.1.9 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, ignorting 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](#)

### 7.9.1.10 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](#)

**7.9.1.11 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](#)

**7.9.1.12 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](#)

### 7.9.1.13 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](#)

### 7.9.1.14 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](#)

### 7.9.1.15 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](#)

**7.9.1.16 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](#)

**7.9.1.17 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](#)

**7.9.1.18 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](#)

**7.9.1.19 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](#)

**7.9.1.20 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](#)

**7.9.1.21 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](#)

**7.9.1.22 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](#)

**7.9.1.23 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](#)

**7.9.1.24 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](#)

**7.9.1.25 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](#)

**7.9.1.26 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](#)

**7.9.1.27 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](#)

**7.9.1.28 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](#)

**7.9.1.29 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](#)

**7.9.1.30 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](#)

**7.9.1.31 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](#)

### 7.9.1.32 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](#)

### 7.9.1.33 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](#)

## 7.10 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.*

## 7.10.1 Function Documentation

### 7.10.1.1 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](#)

### 7.10.1.2 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](#)

### 7.10.1.3 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](#)

### 7.10.1.4 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](#)

### 7.10.1.5 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](#)

#### 7.10.1.6 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](#)

#### 7.10.1.7 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](#)

### 7.10.1.8 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](#)

### 7.10.1.9 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](#)

### 7.10.1.10 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](#)

#### 7.10.1.11 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](#)

#### 7.10.1.12 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](#)

**7.10.1.13 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](#)

**7.10.1.14 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](#)

**7.10.1.15 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](#)

#### 7.10.1.16 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](#)

#### 7.10.1.17 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](#)

**7.10.1.18 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](#)

**7.10.1.19 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](#)

**7.10.1.20 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](#)

**7.10.1.21 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](#)

**7.10.1.22 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](#)

**7.10.1.23 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](#)

**7.10.1.24 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](#)

**7.10.1.25 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](#)

**7.10.1.26 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](#)

**7.10.1.27 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](#)

**7.10.1.28 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](#)

**7.10.1.29 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](#)

**7.10.1.30 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](#)

**7.10.1.31 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](#)

**7.10.1.32 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](#)

**7.10.1.33 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](#)

---

**7.10.1.34 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](#)

---

**7.10.1.35 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](#)

---

**7.10.1.36 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](#)

**7.10.1.37 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](#)

**7.10.1.38 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](#)

**7.10.1.39 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](#)

**7.10.1.40 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](#)

**7.10.1.41 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](#)

**7.10.1.42 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](#)

**7.10.1.43 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](#)

#### 7.10.1.44 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](#)

#### 7.10.1.45 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](#)

**7.10.1.46 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](#)

**7.10.1.47 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](#)

### 7.10.1.48 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](#)

### 7.10.1.49 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](#)

### 7.10.1.50 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](#)

**7.10.1.51 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](#)

**7.10.1.52 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](#)

**7.10.1.53 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](#)

**7.10.1.54 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](#)

**7.10.1.55 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](#)

**7.10.1.56 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](#)

**7.10.1.57 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](#)

**7.10.1.58 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](#)

**7.10.1.59 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](#)

**7.10.1.60 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](#)

**7.10.1.61 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](#)

**7.10.1.62 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](#)

**7.10.1.63 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](#)

**7.10.1.64 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](#)

**7.10.1.65 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](#)

**7.10.1.66 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](#)

**7.10.1.67 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](#)

**7.10.1.68 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](#)

**7.10.1.69 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](#)

**7.10.1.70 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](#)

**7.10.1.71 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](#)

**7.10.1.72 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](#)

#### 7.10.1.73 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](#)

#### 7.10.1.74 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](#)

**7.10.1.75 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](#)

**7.10.1.76 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](#)

**7.10.1.77 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](#)

#### 7.10.1.78 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](#)

#### 7.10.1.79 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](#)

**7.10.1.80 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](#)

**7.10.1.81 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](#)

**7.10.1.82 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](#)

**7.10.1.83 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](#)

**7.10.1.84 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](#)**7.10.1.85 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](#)**7.10.1.86 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](#)

**7.10.1.87 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](#)

**7.10.1.88 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](#)

**7.10.1.89 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](#)

**7.10.1.90 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](#)

**7.10.1.91 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](#)

**7.10.1.92 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](#)

**7.10.1.93 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](#)

**7.10.1.94 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](#)

**7.10.1.95 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](#)

**7.10.1.96 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](#)

## 7.11 Fixed Filters

Fixed filters perform linear filtering operations (such as convolutions) with predefined kernels of fixed sizes.

### 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 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.*

## 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.*

### 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 and NPP\_BORDER\_MIRROR border type operations are 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.*

## FilterGaussPyramidLayerDownBorder

Filters the image using a separable Gaussian filter kernel with user supplied floating point coefficients with downsampling and border control.

If the downsampling rate is equivalent to an integer value then unnecessary source pixels are just skipped. 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\_MIRROR and NPP\_BORDER\_REPLICATE border type operations are supported.

- `NppStatus nppiGetFilterGaussPyramidLayerDownBorderDstROI (int nSrcROIWidth, int nSrcROIHeight, NppiSize *pDstSizeROI, Npp32f nRate)`

*Calculate destination image SizeROI width and height from source image ROI width and height and downsampling rate.*

- `NppStatus nppiFilterGaussPyramidLayerDownBorder_8u_C1R (const Npp8u *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u *pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f *pKernel, NppiBorderType eBorderType)`

*Single channel 8-bit unsigned Gauss filter with downsampling and border control.*

- `NppStatus nppiFilterGaussPyramidLayerDownBorder_8u_C3R (const Npp8u *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u *pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f *pKernel, NppiBorderType eBorderType)`

*Three channel 8-bit unsigned Gauss filter with downsampling and border control.*

- `NppStatus nppiFilterGaussPyramidLayerDownBorder_16u_C1R (const Npp16u *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f *pKernel, NppiBorderType eBorderType)`

*Single channel 16-bit unsigned Gauss filter with downsampling and border control.*

- `NppStatus nppiFilterGaussPyramidLayerDownBorder_16u_C3R (const Npp16u *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f *pKernel, NppiBorderType eBorderType)`

*Three channel 16-bit unsigned Gauss filter with downsampling and border control.*

- `NppStatus nppiFilterGaussPyramidLayerDownBorder_32f_C1R (const Npp32f *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f *pKernel, NppiBorderType eBorderType)`

*Single channel 32-bit floating-point Gauss filter downsampling and with border control.*

- `NppStatus nppiFilterGaussPyramidLayerDownBorder_32f_C3R (const Npp32f *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f *pKernel, NppiBorderType eBorderType)`

*Three channel 32-bit floating-point Gauss filter with downsampling and border control.*

## FilterGaussPyramidLayerUpBorder

Filters the image using a separable Gaussian filter kernel with user supplied floating point coefficients with upsampling and border control.

If the upsampling rate is equivalent to an integer value then unnecessary source pixels are just skipped. 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\_MIRROR and NPP\_BORDER\_REPLICATE border type operations are supported.

- **NppStatus nppiGetFilterGaussPyramidLayerUpBorderDstROI** (int nSrcROIWidth, int nSrcROIHeight, NppiSize \*pDstSizeROIMin, NppiSize \*pDstSizeROIMax, Npp32f nRate)

*Calculate destination image minimum and maximum SizeROI width and height from source image ROI width and height and upsampling rate.*

- **NppStatus nppiFilterGaussPyramidLayerUpBorder\_8u\_C1R** (const Npp8u \*pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u \*pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \*pKernel, NppiBorderType eBorderType)

*Single channel 8-bit unsigned Gauss filter with upsampling and border control.*

- **NppStatus nppiFilterGaussPyramidLayerUpBorder\_8u\_C3R** (const Npp8u \*pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u \*pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \*pKernel, NppiBorderType eBorderType)

*Three channel 8-bit unsigned Gauss filter with upsampling and border control.*

- **NppStatus nppiFilterGaussPyramidLayerUpBorder\_16u\_C1R** (const Npp16u \*pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u \*pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \*pKernel, NppiBorderType eBorderType)

*Single channel 16-bit unsigned Gauss filter with upsampling and border control.*

- **NppStatus nppiFilterGaussPyramidLayerUpBorder\_16u\_C3R** (const Npp16u \*pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u \*pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \*pKernel, NppiBorderType eBorderType)

*Three channel 16-bit unsigned Gauss filter with upsampling and border control.*

- **NppStatus nppiFilterGaussPyramidLayerUpBorder\_32f\_C1R** (const Npp32f \*pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \*pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \*pKernel, NppiBorderType eBorderType)

*Single channel 32-bit floating-point Gauss filter upsampling and with border control.*

- **NppStatus nppiFilterGaussPyramidLayerUpBorder\_32f\_C3R** (const Npp32f \*pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \*pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \*pKernel, NppiBorderType eBorderType)

*Three channel 32-bit floating-point Gauss filter with upsampling and border control.*

## FilterBilateralGaussBorder

Filters the image using a bilateral Gaussian 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.

For this filter the anchor point is always the central element of the kernel. Coefficients of the bilateral filter kernel depend on their position in the kernel and on the value of some source image pixels overlayed by the filter kernel. Only source image pixels with both coordinates divisible by nDistanceBetweenSrcPixels are used in calculations.

The value of an output pixel  $d$  is

$$d = \frac{\sum_{h=-nRadius}^{nRadius} \sum_{w=-nRadius}^{nRadius} W1(h, w) \cdot W2(h, w) \cdot S(h, w)}{\sum_{h=-nRadius}^{nRadius} \sum_{w=-nRadius}^{nRadius} W1(h, w) \cdot W2(h, w)}$$

where  $h$  and  $w$  are the corresponding kernel width and height indexes,  $S(h,w)$  is the value of the source image pixel overlayed by filter kernel position  $(h,w)$ ,  $W1(h,w)$  is `func(nValSquareSigma, (S(h,w) - S(0,0))) where  $S(0,0)$  is the value of the source image pixel at the center of the kernel,  $W2(h,w)$  is func(nPosSquareSigma, sqrt(h*h+w*w)), and func is the following formula`

$$func(S, I) = \exp\left(-\frac{I^2}{2.0F \cdot S^2}\right)$$

Currently only the `NPP_BORDER_REPLICATE` border type operations are supported.

- `NppStatus nppiFilterBilateralGaussBorder_8u_C1R (const Npp8u *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u *pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const Npp32f nValSquareSigma, const Npp32f nPosSquareSigma, NppiBorderType eBorderType)`

*Single channel 8-bit unsigned bilateral Gauss filter with border control.*

- `NppStatus nppiFilterBilateralGaussBorder_8u_C3R (const Npp8u *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u *pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const Npp32f nValSquareSigma, const Npp32f nPosSquareSigma, NppiBorderType eBorderType)`

*Three channel 8-bit unsigned bilateral Gauss filter with border control.*

- `NppStatus nppiFilterBilateralGaussBorder_16u_C1R (const Npp16u *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const Npp32f nValSquareSigma, const Npp32f nPosSquareSigma, NppiBorderType eBorderType)`

*Single channel 16-bit unsigned bilateral Gauss filter with border control.*

- `NppStatus nppiFilterBilateralGaussBorder_16u_C3R (const Npp16u *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const Npp32f nValSquareSigma, const Npp32f nPosSquareSigma, NppiBorderType eBorderType)`

*Three channel 16-bit unsigned bilateral Gauss filter with border control.*

- `NppStatus nppiFilterBilateralGaussBorder_32f_C1R (const Npp32f *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const Npp32f nValSquareSigma, const Npp32f nPosSquareSigma, NppiBorderType eBorderType)`

*One channel 32-bit floating-point bilateral Gauss filter with border control.*

- `NppStatus nppiFilterBilateralGaussBorder_32f_C3R (const Npp32f *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const Npp32f nValSquareSigma, const Npp32f nPosSquareSigma, NppiBorderType eBorderType)`

*Three channel 32-bit floating-point bilateral Gauss filter with border control.*

## FilterHighPass

Filters the image using a high-pass filter kernel:

$$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

- `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 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, 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: HighPass = Image - Gaussian(Image) Result = Image + nWeight \* HighPass \* ( |HighPass| >= 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).*

## GradientVectorPrewittBorder

RGB Color to Prewitt Gradient Vector conversion using user selected fixed mask size and gradient distance method.

Functions support up to 4 optional single channel output gradient vectors, X (vertical), Y (horizontal), magnitude, and angle with user selectable distance methods. Output for a particular vector is disabled by supplying a NULL pointer for that vector. X and Y gradient vectors are in cartesian form in the destination data type. Magnitude vectors are polar gradient form in the destination data type, angle is always in floating point polar gradient format. Only fixed mask sizes of 3x3 are supported. Only nppiNormL1 (sum) and nppiNormL2 (sqrt of sum of squares) distance methods are currently supported.

Currently only the NPP\_BORDER\_REPLICATE border type operation is supported. Borderless output can be accomplished by using a larger source image than the destination and adjusting oSrcSize and oSrcOffset parameters accordingly.

The following fixed kernel mask is used for producing the pDstX (vertical) output image.

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

The following fixed kernel mask is used for producing the pDstY (horizontal) output image.

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

For the C1R versions of the function the pDstMag output image value for L1 normalization consists of the absolute value of the pDstX value plus the absolute value of the pDstY value at that particular image pixel location. For the C1R versions of the function the pDstMag output image value for L2 normalization consists of the square root of the pDstX value squared plus the pDstY value squared at that particular image pixel location. For the C1R versions of the function the pDstAngle output image value consists of the arctangent (atan2) of the pDstY value and the pDstX value at that particular image pixel location.

For the C3C1R versions of the function, regardless of the selected normalization method, the L2 normalization value is first determined for each of the pDstX and pDstY values for each source channel then the

largest L2 normalization value (largest gradient) is used to select which of the 3 pDstX channel values are output to the pDstX image or pDstY channel values are output to the pDstY image. For the C3C1R versions of the function the pDstMag output image value for L1 normalizaton consists of the same technique used for the C1R version for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 pDstMag channel values to output to the pDstMag image. For the C3C1R versions of the function the pDstMag output image value for L2 normalizaton consists of just outputting the largest per source channel L2 normalization value to the pDstMag image. For the C3C1R versions of the function the pDstAngle output image value consists of the same technique used for the C1R version calculated for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 angle values to output to the pDstAngle image.

- **NppStatus nppiGradientVectorPrewittBorder\_8u16s\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** \*pDstX, int nDstXStep, **Npp16s** \*pDstY, int nDstYStep, **Npp16s** \*pDstMag, int nDstMagStep, **Npp32f** \*pDstAngle, int nDstAngleStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiNorm** eNorm, **NppiBorderType** eBorderType)

*1 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- **NppStatus nppiGradientVectorPrewittBorder\_8u16s\_C3C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** \*pDstX, int nDstXStep, **Npp16s** \*pDstY, int nDstYStep, **Npp16s** \*pDstMag, int nDstMagStep, **Npp32f** \*pDstAngle, int nDstAngleStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiNorm** eNorm, **NppiBorderType** eBorderType)

*3 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- **NppStatus nppiGradientVectorPrewittBorder\_16s32f\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** \*pDstX, int nDstXStep, **Npp32f** \*pDstY, int nDstYStep, **Npp32f** \*pDstMag, int nDstMagStep, **Npp32f** \*pDstAngle, int nDstAngleStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiNorm** eNorm, **NppiBorderType** eBorderType)

*1 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- **NppStatus nppiGradientVectorPrewittBorder\_16s32f\_C3C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** \*pDstX, int nDstXStep, **Npp32f** \*pDstY, int nDstYStep, **Npp32f** \*pDstMag, int nDstMagStep, **Npp32f** \*pDstAngle, int nDstAngleStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiNorm** eNorm, **NppiBorderType** eBorderType)

*3 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- **NppStatus nppiGradientVectorPrewittBorder\_16u32f\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** \*pDstX, int nDstXStep, **Npp32f** \*pDstY, int nDstYStep, **Npp32f** \*pDstMag, int nDstMagStep, **Npp32f** \*pDstAngle, int nDstAngleStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiNorm** eNorm, **NppiBorderType** eBorderType)

*1 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- **NppStatus nppiGradientVectorPrewittBorder\_16u32f\_C3C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** \*pDstX, int nDstXStep, **Npp32f** \*pDstY, int nDstYStep, **Npp32f** \*pDstMag, int nDstMagStep, **Npp32f** \*pDstAngle, int nDstAngleStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiNorm** eNorm, **NppiBorderType** eBorderType)

`stYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*3 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorPrewittBorder_32f_C1R (const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*1 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorPrewittBorder_32f_C3C1R (const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*3 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

## GradientVectorScharrBorder

RGB Color to Scharr Gradient Vector conversion using user selected fixed mask size and gradient distance method.

Functions support up to 4 optional single channel output gradient vectors, X (vertical), Y (horizontal), magnitude, and angle with user selectable distance methods. Output for a particular vector is disabled by supplying a NULL pointer for that vector. X and Y gradient vectors are in cartesian form in the destination data type. Magnitude vectors are polar gradient form in the destination data type, angle is always in floating point polar gradient format. Only fixed mask sizes of 3x3 are supported. Only nppiNormL1 (sum) and nppiNormL2 (sqrt of sum of squares) distance methods are currently supported.

Currently only the NPP\_BORDER\_REPLICATE border type operation is supported. Borderless output can be accomplished by using a larger source image than the destination and adjusting oSrcSize and oSrcOffset parameters accordingly.

The following fixed kernel mask is used for producing the pDstX (vertical) output image.

$$\begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}$$

The following fixed kernel mask is used for producing the pDstY (horizontal) output image.

$$\begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$$

For the C1R versions of the function the pDstMag output image value for L1 normalization consists of the absolute value of the pDstX value plus the absolute value of the pDstY value at that particular image

pixel location. For the C1R versions of the function the pDstMag output image value for L2 normalization consists of the square root of the pDstX value squared plus the pDstY value squared at that particular image pixel location. For the C1R versions of the function the pDstAngle output image value consists of the arctangent (atan2) of the pDstY value and the pDstX value at that particular image pixel location.

For the C3C1R versions of the function, regardless of the selected normalization method, the L2 normalization value is first determined for each or the pDstX and pDstY values for each source channel then the largest L2 normalization value (largest gradient) is used to select which of the 3 pDstX channel values are output to the pDstX image or pDstY channel values are output to the pDstY image. For the C3C1R versions of the function the pDstMag output image value for L1 normalizaton consists of the same technique used for the C1R version for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 pDstMag channel values to output to the pDstMag image. For the C3C1R versions of the function the pDstMag output image value for L2 normalizaton consists of just outputting the largest per source channel L2 normalization value to the pDstMag image. For the C3C1R versions of the function the pDstAngle output image value consists of the same technique used for the C1R version calculated for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 angle values to output to the pDstAngle image.

- `NppStatus nppiGradientVectorScharrBorder_8u16s_C1R (const Npp8u *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s *pDstX, int nDstXStep, Npp16s *pDstY, int nDstYStep, Npp16s *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*1 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorScharrBorder_8u16s_C3C1R (const Npp8u *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s *pDstX, int nDstXStep, Npp16s *pDstY, int nDstYStep, Npp16s *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*3 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorScharrBorder_16s32f_C1R (const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*1 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorScharrBorder_16s32f_C3C1R (const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*3 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorScharrBorder_16u32f_C1R (const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*1 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorScharrBorder_16u32f_C3C1R (const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*3 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorScharrBorder_32f_C1R (const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*1 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorScharrBorder_32f_C3C1R (const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*3 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

## GradientVectorSobelBorder

RGB Color to Sobel Gradient Vector conversion using user selected fixed mask size and gradient distance method.

Functions support up to 4 optional single channel output gradient vectors, X (vertical), Y (horizontal), magnitude, and angle with user selectable distance methods. Output for a particular vector is disabled by supplying a NULL pointer for that vector. X and Y gradient vectors are in cartesian form in the destination data type. Magnitude vectors are polar gradient form in the destination data type, angle is always in floating point polar gradient format. Only fixed mask sizes of 3x3 and 5x5 are supported. Only nppiNormL1 (sum) and nppiNormL2 (sqrt of sum of squares) distance methods are currently supported.

Currently only the NPP\_BORDER\_REPLICATE border type operation is supported. Borderless output can be accomplished by using a larger source image than the destination and adjusting oSrcSize and oSrcOffset parameters accordingly.

One of the following fixed kernel masks are used for producing the 3x3 or 5x5 pDstX (vertical) output image depending on selected mask size.

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{pmatrix}$$

One of the following fixed kernel masks are used for producing the 3x3 or 5x5 pDstY (horizontal) output image depending on selected mask size.

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{pmatrix}$$

For the C1R versions of the function the pDstMag output image value for L1 normalization consists of the absolute value of the pDstX value plus the absolute value of the pDstY value at that particular image pixel location. For the C1R versions of the function the pDstMag output image value for L2 normalization consists of the square root of the pDstX value squared plus the pDstY value squared at that particular image pixel location. For the C1R versions of the function the pDstAngle output image value consists of the arctangent (atan2) of the pDstY value and the pDstX value at that particular image pixel location.

For the C3C1R versions of the function, regardless of the selected normalization method, the L2 normalization value is first determined for each of the pDstX and pDstY values for each source channel then the largest L2 normalization value (largest gradient) is used to select which of the 3 pDstX channel values are output to the pDstX image or pDstY channel values are output to the pDstY image. For the C3C1R versions of the function the pDstMag output image value for L1 normalization consists of the same technique used for the C1R version for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 pDstMag channel values to output to the pDstMag image. For the C3C1R versions of the function the pDstMag output image value for L2 normalization consists of just outputting the largest per source channel L2 normalization value to the pDstMag image. For the C3C1R versions of the function the pDstAngle output image value consists of the same technique used for the C1R version calculated for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 angle values to output to the pDstAngle image.

- `NppStatus nppiGradientVectorSobelBorder_8u16s_C1R (const Npp8u *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s *pDstX, int nDstXStep, Npp16s *pDstY, int nDstYStep, Npp16s *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*1 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorSobelBorder_8u16s_C3C1R (const Npp8u *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s *pDstX, int nDstXStep, Npp16s *pDstY, int nDstYStep, Npp16s *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*3 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorSobelBorder_16s32f_C1R (const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*1 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorSobelBorder_16s32f_C3C1R (const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*3 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorSobelBorder_16u32f_C1R (const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*1 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorSobelBorder_16u32f_C3C1R (const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*3 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorSobelBorder_32f_C1R (const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*1 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

- `NppStatus nppiGradientVectorSobelBorder_32f_C3C1R (const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDstX, int nDstXStep, Npp32f *pDstY, int nDstYStep, Npp32f *pDstMag, int nDstMagStep, Npp32f *pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)`

*3 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.*

## FilterCannyBorder

Performs Canny edge detection on a single channel 8-bit grayscale image and outputs a single channel 8-bit image consisting of 0x00 and 0xFF values with 0xFF representing edge pixels.

The algorithm consists of three phases. The first phase generates two output images consisting of a single channel 16-bit signed image containing magnitude values and a single channel 32-bit floating point image containing the angular direction of those magnitude values. This phase is accomplished by calling the appropriate GradientVectorBorder filter function based on the filter type, filter mask size, and norm type requested. The next phase uses those magnitude and direction images to suppress non-maximum magnitude values which are lower than the values of either of its two nearest neighbors in the same direction as the test magnitude pixel in the 3x3 surrounding magnitude pixel neighborhood. This phase outputs a new magnitude image with non-maximum pixel values suppressed. Finally, in the third phase, the new magnitude image is passed through a hysteresis threshold filter that filters out any magnitude values that are not connected to another edge magnitude value. In this phase, any magnitude value above the high threshold value is automatically accepted, any magnitude value below the low threshold value is automatically rejected. For magnitude values that lie between the low and high threshold, values are only accepted if one of their two neighbors in the same direction in the 3x3 neighborhood around them lies above the low threshold value. In other words, if they are connected to an active edge. J. Canny recommends that the ratio of high to low threshold limit be in the range two or three to one, based on predicted signal-to-noise ratios. The final output of the third phase consists of a single channel 8-bit unsigned image of 0x00 and 0xFF values based on whether they are accepted or rejected during threshold testing.

Currently only the NPP\_BORDER\_REPLICATE border type operation is supported. Borderless output can be accomplished by using a larger source image than the destination and adjusting oSrcSize and oSrcOffset parameters accordingly.

- [NppStatus nppiFilterCannyBorderGetBufferSize \(NppiSize oSizeROI, int \\*hpBufferSize\)](#)  
*Calculate scratch buffer size needed for the FilterCannyBorder function based on destination image SizeROI width and height.*
- [NppStatus nppiFilterCannyBorder\\_8u\\_C1R \(const Npp8u \\*pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u \\*pDst, int nDstStep, NppiSize oSizeROI, NppiDifferentialKernel eFilterType, NppiMaskSize eMaskSize, Npp16s nLowThreshold, Npp16s nHighThreshold, NppiNorm eNorm, NppiBorderType eBorderType, Npp8u \\*pDeviceBuffer\)](#)  
*1 channel 8-bit unsigned grayscale to 1 channel 8-bit unsigned black (0x00) and white (0xFF) image with border control.*

## FilterHarrisCornersBorder

Performs Harris Corner detection on a single channel 8-bit grayscale image and outputs a single channel 32-bit floating point image consisting the corner response at each pixel of the image.

The algorithm consists of two phases. The first phase generates the floating point product of XX, YY, and XY gradients at each pixel in the image. The type of gradient used is controlled by the eFilterType and eMaskSize parameters. The second phase averages those products over a window of either 3x3 or 5x5 pixels around the center pixel then generates the Harris corner response at that pixel which is output in the destination image. The Harris response value is determined as  $H = ((XX * YY - XY * XY) - (nK * ((XX + YY) * (XX + YY)))) * nScale$ .

Currently only the NPP\_BORDER\_REPLICATE border type operation is supported. Borderless output can be accomplished by using a larger source image than the destination and adjusting oSrcSize and oSrcOffset parameters accordingly.

- **NppStatus nppiFilterHarrisCornersBorderGetBufferSize (NppiSize oSizeROI, int \*hpBufferSize)**  
*Calculate scratch buffer size needed for the FilterHarrisCornersBorder function based on destination image SizeROI width and height.*
- **NppStatus nppiFilterHarrisCornersBorder\_8u32f\_C1R (const Npp8u \*pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \*pDst, int nDstStep, NppiSize oSizeROI, NppiDifferentialKernel eFilterType, NppiMaskSize eMaskSize, NppiMaskSize eAvgWindowSize, Npp32f nK, Npp32f nScale, NppiBorderType eBorderType, Npp8u \*pDeviceBuffer)**  
*1 channel 8-bit unsigned grayscale to 1 channel 32-bit floating point Harris corners response image with border control.*

### 7.11.1 Detailed Description

Fixed filters perform linear filtering operations (such as convolutions) with predefined kernels of fixed sizes.

Note that this section also contains a few dynamic kernel filters, namely GaussAdvanced and Bilateral.

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.

### 7.11.2 Function Documentation

- 7.11.2.1 NppStatus nppiFilterBilateralGaussBorder\_16u\_C1R (const Npp16u \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const Npp32f nValSquareSigma, const Npp32f nPosSquareSigma, NppiBorderType eBorderType)**

Single channel 16-bit unsigned bilateral 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).  
**nRadius** The radius of the round filter kernel to be used. A radius of 1 indicates a filter kernel size of 3 by 3, 2 indicates 5 by 5, etc. Radius values from 1 to 32 are supported.  
**nStepBetweenSrcPixels** The step size between adjacent source image pixels processed by the filter kernel, most commonly 1.  
**nValSquareSigma** The square of the sigma for the relative intensity distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.  
**nPosSquareSigma** The square of the sigma for the relative geometric distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

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

**7.11.2.2 NppStatus nppiFilterBilateralGaussBorder\_16u\_C3R (const Npp16u \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const Npp32f nValSquareSigma, const Npp32f nPosSquareSigma, NppiBorderType eBorderType)**

Three channel 16-bit unsigned bilateral 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).

*nRadius* The radius of the round filter kernel to be used. A radius of 1 indicates a filter kernel size of 3 by 3, 2 indicates 5 by 5, etc. Radius values from 1 to 32 are supported.

*nStepBetweenSrcPixels* The step size between adjacent source image pixels processed by the filter kernel, most commonly 1.

*nValSquareSigma* The square of the sigma for the relative intensity distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.

*nPosSquareSigma* The square of the sigma for the relative geometric distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

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

**7.11.2.3 NppStatus nppiFilterBilateralGaussBorder\_32f\_C1R (const Npp32f \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDst, Npp32s nDstStep, NppiSize oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const Npp32f nValSquareSigma, const Npp32f nPosSquareSigma, NppiBorderType eBorderType)**

One channel 32-bit floating-point bilateral 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).

*nRadius* The radius of the round filter kernel to be used. A radius of 1 indicates a filter kernel size of 3 by 3, 2 indicates 5 by 5, etc. Radius values from 1 to 32 are supported.

*nStepBetweenSrcPixels* The step size between adjacent source image pixels processed by the filter kernel, most commonly 1.

*nValSquareSigma* The square of the sigma for the relative intensity distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.

*nPosSquareSigma* The square of the sigma for the relative geometric distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.

*eBorderType* The border type operation to be applied at source image border boundaries.

#### Returns:

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

**7.11.2.4 NppStatus nppiFilterBilateralGaussBorder\_32f\_C3R (const Npp32f \* pSrc, Npp32s  
nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDst, Npp32s nDstStep,  
NppiSize oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const Npp32f  
nValSquareSigma, const Npp32f nPosSquareSigma, NppiBorderType eBorderType)**

Three channel 32-bit floating-point bilateral 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).

*nRadius* The radius of the round filter kernel to be used. A radius of 1 indicates a filter kernel size of 3 by 3, 2 indicates 5 by 5, etc. Radius values from 1 to 32 are supported.

*nStepBetweenSrcPixels* The step size between adjacent source image pixels processed by the filter kernel, most commonly 1.

*nValSquareSigma* The square of the sigma for the relative intensity distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.

*nPosSquareSigma* The square of the sigma for the relative geometric distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.

*eBorderType* The border type operation to be applied at source image border boundaries.

#### Returns:

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

---

**7.11.2.5 NppStatus nppiFilterBilateralGaussBorder\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const int *nRadius*, const int *nStepBetweenSrcPixels*, const Npp32f *nValSquareSigma*, const Npp32f *nPosSquareSigma*, NppiBorderType *eBorderType*)**

Single channel 8-bit unsigned bilateral 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).  
*nRadius* The radius of the round filter kernel to be used. A radius of 1 indicates a filter kernel size of 3 by 3, 2 indicates 5 by 5, etc. Radius values from 1 to 32 are supported.  
*nStepBetweenSrcPixels* The step size between adjacent source image pixels processed by the filter kernel, most commonly 1.  
*nValSquareSigma* The square of the sigma for the relative intensity distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.  
*nPosSquareSigma* The square of the sigma for the relative geometric distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

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

---

**7.11.2.6 NppStatus nppiFilterBilateralGaussBorder\_8u\_C3R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const int *nRadius*, const int *nStepBetweenSrcPixels*, const Npp32f *nValSquareSigma*, const Npp32f *nPosSquareSigma*, NppiBorderType *eBorderType*)**

Three channel 8-bit unsigned bilateral 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).  
*nRadius* The radius of the round filter kernel to be used. A radius of 1 indicates a filter kernel size of 3 by 3, 2 indicates 5 by 5, etc. Radius values from 1 to 32 are supported.

***nStepBetweenSrcPixels*** The step size between adjacent source image pixels processed by the filter kernel, most commonly 1.

***nValSquareSigma*** The square of the sigma for the relative intensity distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.

***nPosSquareSigma*** The square of the sigma for the relative geometric distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.

***eBorderType*** The border type operation to be applied at source image border boundaries.

**Returns:**

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

**7.11.2.7 NppStatus nppiFilterCannyBorder\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppiDifferentialKernel eFilterType, NppiMaskSize eMaskSize, Npp16s nLowThreshold, Npp16s nHighThreshold, NppiNorm eNorm, NppiBorderType eBorderType, Npp8u \* pDeviceBuffer)**

1 channel 8-bit unsigned grayscale to 1 channel 8-bit unsigned black (0x00) and white (0xFF) image with 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*** output edge destination\_image\_pointer.

***nDstStep*** output edge destination\_image\_line\_step.

***oSizeROI*** Region-of-Interest (ROI).

***eFilterType*** selects between Sobel or Scharr filter type.

***eMaskSize*** fixed filter mask size to use.

***nLowThreshold*** low hysteresis threshold value.

***nHighThreshold*** high hysteresis threshold value.

***eNorm*** gradient distance method to use.

***eBorderType*** source image border type to use use.

***pDeviceBuffer*** pointer to scratch DEVICE memory buffer of size hpBufferSize (see [nppiFilterCannyBorderGetBufferSize\(\)](#) above)

**Returns:**

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

### 7.11.2.8 NppStatus nppiFilterCannyBorderGetBufferSize (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Calculate scratch buffer size needed for the FilterCannyBorder function based on destination image Size-ROI width and height.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. Scratch Buffer and Host Pointer.

**Returns:**

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

### 7.11.2.9 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](#)

### 7.11.2.10 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](#)

**7.11.2.11 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](#)

**7.11.2.12 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](#)

**7.11.2.13 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](#)

**7.11.2.14 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](#)

**7.11.2.15 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](#)

**7.11.2.16 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](#)

**7.11.2.17 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](#)

**7.11.2.18 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](#)

**7.11.2.19 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](#)

**7.11.2.20 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](#)

**7.11.2.21 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](#)

**7.11.2.22 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](#)

**7.11.2.23 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](#)

**7.11.2.24 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](#)

**7.11.2.25 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](#)

**7.11.2.26 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](#)

**7.11.2.27 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](#)

**7.11.2.28 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](#)

**7.11.2.29 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](#)

**7.11.2.30 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](#)

**7.11.2.31 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](#)

**7.11.2.32 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](#)

**7.11.2.33 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](#)

**7.11.2.34 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](#)

**7.11.2.35 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](#)

**7.11.2.36 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](#)

**7.11.2.37 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](#)

---

**7.11.2.38 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](#)

---

**7.11.2.39 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](#)

---

**7.11.2.40 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](#)

**7.11.2.41 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](#)

**7.11.2.42 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](#)

**7.11.2.43 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](#)

**7.11.2.44 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](#)

**7.11.2.45 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](#)

**7.11.2.46 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](#)

**7.11.2.47 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](#)

**7.11.2.48 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](#)

**7.11.2.49 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](#)

**7.11.2.50 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](#)

**7.11.2.51 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](#)

**7.11.2.52 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](#)

**7.11.2.53 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](#)

**7.11.2.54 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](#)

**7.11.2.55 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](#)

**7.11.2.56 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](#)

**7.11.2.57 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](#)

**7.11.2.58 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](#)

**7.11.2.59 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](#)

**7.11.2.60 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](#)

**7.11.2.61 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](#)

**7.11.2.62 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](#)

**7.11.2.63 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](#)

**7.11.2.64 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](#)

**7.11.2.65 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](#)

**7.11.2.66 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](#)

**7.11.2.67 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](#)

**7.11.2.68 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](#)

**7.11.2.69 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](#)

**7.11.2.70 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](#)

**7.11.2.71 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](#)

**7.11.2.72 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](#)

---

**7.11.2.73 NppStatus nppiFilterGaussPyramidLayerDownBorder\_16u\_C1R (const Npp16u \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Single channel 16-bit unsigned Gauss filter with downsampling and 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).

**nRate** The downsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.

**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](#)

---

**7.11.2.74 NppStatus nppiFilterGaussPyramidLayerDownBorder\_16u\_C3R (const Npp16u \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Three channel 16-bit unsigned Gauss filter with downsampling and 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).

**nRate** The downsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.

**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](#)

**7.11.2.75 NppStatus nppiFilterGaussPyramidLayerDownBorder\_32f\_C1R (const Npp32f \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Single channel 32-bit floating-point Gauss filter downsampling and with 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).

*nRate* The downsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.

*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](#)

**7.11.2.76 NppStatus nppiFilterGaussPyramidLayerDownBorder\_32f\_C3R (const Npp32f \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Three channel 32-bit floating-point Gauss filter with downsampling and 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).

**nRate** The downsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.

**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](#)

**7.11.2.77 NppStatus nppiFilterGaussPyramidLayerDownBorder\_8u\_C1R (const Npp8u \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Single channel 8-bit unsigned Gauss filter with downsampling and 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).

**nRate** The downsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.

**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](#)

**7.11.2.78 NppStatus nppiFilterGaussPyramidLayerDownBorder\_8u\_C3R (const Npp8u \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Three channel 8-bit unsigned Gauss filter with downsampling and 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).  
*nRate* The downsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.  
*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](#)

**7.11.2.79 NppStatus nppiFilterGaussPyramidLayerUpBorder\_16u\_C1R (const Npp16u \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Single channel 16-bit unsigned Gauss filter with upsampling and 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).  
*nRate* The upsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.  
*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](#)

---

**7.11.2.80 NppStatus nppiFilterGaussPyramidLayerUpBorder\_16u\_C3R (const Npp16u \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Three channel 16-bit unsigned Gauss filter with upsampling and 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).  
**nRate** The upsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.  
**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](#)

---

**7.11.2.81 NppStatus nppiFilterGaussPyramidLayerUpBorder\_32f\_C1R (const Npp32f \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Single channel 32-bit floating-point Gauss filter upsampling and with 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).  
**nRate** The upsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.  
**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](#)

**7.11.2.82 NppStatus nppiFilterGaussPyramidLayerUpBorder\_32f\_C3R (const Npp32f \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Three channel 32-bit floating-point Gauss filter with upsampling and 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).

*nRate* The upsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.

*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](#)

**7.11.2.83 NppStatus nppiFilterGaussPyramidLayerUpBorder\_8u\_C1R (const Npp8u \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Single channel 8-bit unsigned Gauss filter with upsampling and 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).

*nRate* The upsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be  $> 1.0F$  and  $\leq 10.0F$ .

*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](#)

**7.11.2.84 NppStatus nppiFilterGaussPyramidLayerUpBorder\_8u\_C3R (const Npp8u \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, Npp32f nRate, const int nFilterTaps, const Npp32f \* pKernel, NppiBorderType eBorderType)**

Three channel 8-bit unsigned Gauss filter with upsampling and 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).

*nRate* The upsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be  $> 1.0F$  and  $\leq 10.0F$ .

*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](#)

**7.11.2.85 NppStatus nppiFilterHarrisCornersBorder\_8u32f\_C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, NppiDifferentialKernel eFilterType, NppiMaskSize eMaskSize, NppiMaskSize eAvgWindowSize, Npp32f nK, Npp32f nScale, NppiBorderType eBorderType, Npp8u \* pDeviceBuffer)**

1 channel 8-bit unsigned grayscale to 1 channel 32-bit floating point Harris corners response image with 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* output edge destination\_image\_pointer.  
*nDstStep* output edge destination\_image\_line\_step.  
*oSizeROI* Region-of-Interest (ROI).  
*eFilterType* selects between Sobel or Scharr filter type.  
*eMaskSize* fixed filter mask size to use (3x3 or 5x5 for Sobel).  
*eAvgWindowSize* fixed window mask size to use (3x3 or 5x5).  
*nK* Harris Corners constant (commonly used value is 0.04F).  
*nScale* output is scaled by this scale factor.  
*eBorderType* source image border type to use use.  
*pDeviceBuffer* pointer to scratch DEVICE memory buffer of size hpBufferSize (see [nppiFilterHarrisCornersBorderGetBufferSize\(\)](#) above)

**Returns:**

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

**7.11.2.86 NppStatus nppiFilterHarrisCornersBorderGetBufferSize (NppiSize oSizeROI, int \* hpBufferSize)**

Calculate scratch buffer size needed for the FilterHarrisCornersBorder function based on destination image SizeROI width and height.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).  
*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

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

**7.11.2.87 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

**7.11.2.88 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

**7.11.2.89 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

**7.11.2.90 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](#)

**7.11.2.91 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](#)

**7.11.2.92 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](#)

**7.11.2.93 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](#)

**7.11.2.94 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](#)

**7.11.2.95 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](#)

**7.11.2.96 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](#)

**7.11.2.97 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](#)

**7.11.2.98 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](#)

**7.11.2.99 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](#)

**7.11.2.100 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](#)

**7.11.2.101 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](#)

**7.11.2.102 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](#)

**7.11.2.103 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](#)

**7.11.2.104 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](#)

**7.11.2.105 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](#)

**7.11.2.106 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](#)

**7.11.2.107 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](#)

**7.11.2.108 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](#)

**7.11.2.109 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](#)

**7.11.2.110 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](#)

**7.11.2.111 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](#)

**7.11.2.112 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](#)

---

**7.11.2.113 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](#)

---

**7.11.2.114 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](#)

**7.11.2.115 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](#)

**7.11.2.116 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](#)

---

**7.11.2.117 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](#)

---

**7.11.2.118 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](#)

**7.11.2.119 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](#)

**7.11.2.120 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](#)

**7.11.2.121 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](#)

**7.11.2.122 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](#)

**7.11.2.123 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](#)

**7.11.2.124 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](#)

**7.11.2.125 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](#)

**7.11.2.126 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](#)

**7.11.2.127 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](#)

**7.11.2.128 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](#)

**7.11.2.129 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](#)

**7.11.2.130 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](#)

**7.11.2.131 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](#)

**7.11.2.132 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](#)

**7.11.2.133 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](#)

**7.11.2.134 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](#)

**7.11.2.135 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](#)

**7.11.2.136 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](#)

**7.11.2.137 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](#)

---

**7.11.2.138 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](#)

---

**7.11.2.139 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](#)

**7.11.2.140 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](#)

**7.11.2.141 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](#)

---

**7.11.2.142 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](#)

---

**7.11.2.143 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](#)

**7.11.2.144 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](#)

**7.11.2.145 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](#)

**7.11.2.146 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](#)

**7.11.2.147 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](#)

**7.11.2.148 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](#)

**7.11.2.149 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](#)

**7.11.2.150 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](#)

**7.11.2.151 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](#)

**7.11.2.152 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](#)

**7.11.2.153 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](#)

**7.11.2.154 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](#)

**7.11.2.155 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](#)

**7.11.2.156 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](#)

**7.11.2.157 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](#)

**7.11.2.158 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](#)

**7.11.2.159 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](#)

**7.11.2.160 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](#)

**7.11.2.161 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](#)

**7.11.2.162 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](#)

### 7.11.2.163 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](#)

### 7.11.2.164 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](#)

**7.11.2.165 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](#)

**7.11.2.166 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](#)

---

**7.11.2.167 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](#)

---

**7.11.2.168 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](#)

**7.11.2.169 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](#)

**7.11.2.170 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](#)

### 7.11.2.171 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](#)

### 7.11.2.172 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](#)

**7.11.2.173 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](#)

**7.11.2.174 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](#)

**7.11.2.175 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](#)

**7.11.2.176 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](#)

**7.11.2.177 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](#)

**7.11.2.178 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](#)

**7.11.2.179 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](#)

**7.11.2.180 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](#)

**7.11.2.181 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](#)

**7.11.2.182 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](#)

**7.11.2.183 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](#)

**7.11.2.184 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](#)

**7.11.2.185 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](#)

**7.11.2.186 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](#)

**7.11.2.187 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](#)

**7.11.2.188 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](#)

**7.11.2.189 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](#)

**7.11.2.190 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](#)

**7.11.2.191 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](#)

**7.11.2.192 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](#)

**7.11.2.193 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](#)

**7.11.2.194 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](#)

**7.11.2.195 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](#)

**7.11.2.196 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](#)

**7.11.2.197 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](#)

**7.11.2.198 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](#)

**7.11.2.199 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](#)

**7.11.2.200 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](#)

**7.11.2.201 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](#)

**7.11.2.202 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](#)

**7.11.2.203 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](#)

**7.11.2.204 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](#)

**7.11.2.205 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](#)

**7.11.2.206 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](#)

**7.11.2.207 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](#)

**7.11.2.208 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](#)

**7.11.2.209 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](#)

**7.11.2.210 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](#)

**7.11.2.211 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](#)

**7.11.2.212 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](#)

**7.11.2.213 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](#)

**7.11.2.214 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](#)

**7.11.2.215 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](#)

**7.11.2.216 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](#)

**7.11.2.217 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](#)

**7.11.2.218 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](#)

**7.11.2.219 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](#)

**7.11.2.220 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](#)

**7.11.2.221 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](#)

**7.11.2.222 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](#)

**7.11.2.223 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](#)

**7.11.2.224 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](#)

**7.11.2.225 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](#)

**7.11.2.226 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](#)

**7.11.2.227 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](#)

**7.11.2.228 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](#)

**7.11.2.229 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](#)

**7.11.2.230 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](#)

**7.11.2.231 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](#)

**7.11.2.232 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](#)

**7.11.2.233 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](#)

**7.11.2.234 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](#)

**7.11.2.235 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](#)

**7.11.2.236 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](#)

### 7.11.2.237 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](#)

### 7.11.2.238 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](#)

### 7.11.2.239 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](#)

**7.11.2.240 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](#)

**7.11.2.241 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](#)

**7.11.2.242 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](#)

**7.11.2.243 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](#)

**7.11.2.244 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](#)

**7.11.2.245 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](#)

**7.11.2.246 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](#)

**7.11.2.247 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](#)

**7.11.2.248 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](#)

**7.11.2.249 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](#)

**7.11.2.250 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](#)

**7.11.2.251 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

**7.11.2.252 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

**7.11.2.253 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

**7.11.2.254 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](#)

**7.11.2.255 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](#)

**7.11.2.256 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](#)

**7.11.2.257 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](#)

**7.11.2.258 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](#)

**7.11.2.259 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](#)

**7.11.2.260 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](#)

**7.11.2.261 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](#)

**7.11.2.262 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](#)

**7.11.2.263 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](#)

**7.11.2.264 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](#)

**7.11.2.265 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](#)

**7.11.2.266 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](#)

**7.11.2.267 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](#)

**7.11.2.268 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](#)

**7.11.2.269 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](#)

**7.11.2.270 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](#)

**7.11.2.271 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](#)

**7.11.2.272 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](#)

**7.11.2.273 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](#)

**7.11.2.274 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](#)

**7.11.2.275 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](#)

**7.11.2.276 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](#)

**7.11.2.277 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](#)

**7.11.2.278 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](#)

**7.11.2.279 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](#)

**7.11.2.280 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](#)

**7.11.2.281 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](#)

### 7.11.2.282 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](#)

### 7.11.2.283 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](#)

### 7.11.2.284 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](#)

**7.11.2.285 NppStatus nppiFilterScharVertBorder\_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](#)

**7.11.2.286 NppStatus nppiFilterScharVertBorder\_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](#)

**7.11.2.287 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](#)

**7.11.2.288 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](#)

**7.11.2.289 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](#)

**7.11.2.290 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](#)

**7.11.2.291 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](#)

**7.11.2.292 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](#)

**7.11.2.293 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](#)

**7.11.2.294 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](#)

**7.11.2.295 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](#)

**7.11.2.296 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](#)

**7.11.2.297 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](#)

**7.11.2.298 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](#)

**7.11.2.299 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](#)

**7.11.2.300 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](#)

**7.11.2.301 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](#)

**7.11.2.302 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](#)

**7.11.2.303 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](#)

**7.11.2.304 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](#)

**7.11.2.305 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](#)

**7.11.2.306 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](#)

**7.11.2.307 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](#)

**7.11.2.308 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](#)

**7.11.2.309 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](#)

**7.11.2.310 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](#)

**7.11.2.311 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](#)

**7.11.2.312 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](#)

**7.11.2.313 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](#)

**7.11.2.314 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](#)

**7.11.2.315 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](#)

**7.11.2.316 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](#)

**7.11.2.317 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](#)

**7.11.2.318 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](#)

**7.11.2.319 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](#)

**7.11.2.320 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](#)

**7.11.2.321 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](#)

**7.11.2.322 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](#)

**7.11.2.323 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](#)

**7.11.2.324 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](#)

**7.11.2.325 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](#)

**7.11.2.326 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](#)

**7.11.2.327 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](#)

**7.11.2.328 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](#)

**7.11.2.329 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](#)

**7.11.2.330 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](#)

**7.11.2.331 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](#)

**7.11.2.332 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](#)

**7.11.2.333 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](#)

**7.11.2.334 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](#)

**7.11.2.335 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](#)

**7.11.2.336 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](#)

**7.11.2.337 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](#)

**7.11.2.338 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](#)

**7.11.2.339 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](#)

**7.11.2.340 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](#)

**7.11.2.341 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](#)

**7.11.2.342 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](#)

**7.11.2.343 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](#)

**7.11.2.344 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](#)

**7.11.2.345 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](#)

**7.11.2.346 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](#)

**7.11.2.347 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](#)

**7.11.2.348 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](#)

**7.11.2.349 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](#)

**7.11.2.350 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](#)

**7.11.2.351 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](#)

**7.11.2.352 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](#)

**7.11.2.353 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](#)

**7.11.2.354 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](#)

**7.11.2.355 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](#)

**7.11.2.356 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](#)

**7.11.2.357 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](#)

**7.11.2.358 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](#)

**7.11.2.359 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](#)

**7.11.2.360 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](#)

**7.11.2.361 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

**7.11.2.362 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

**7.11.2.363 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

**7.11.2.364 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](#)

**7.11.2.365 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](#)

**7.11.2.366 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](#)

**7.11.2.367 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

**7.11.2.368 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

**7.11.2.369 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

**7.11.2.370 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](#)

**7.11.2.371 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](#)

**7.11.2.372 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](#)

**7.11.2.373 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](#)

**7.11.2.374 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](#)

**7.11.2.375 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](#)

**7.11.2.376 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](#)

**7.11.2.377 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](#)

**7.11.2.378 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](#)

**7.11.2.379 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](#)

**7.11.2.380 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](#)

**7.11.2.381 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](#)

**7.11.2.382 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](#)

**7.11.2.383 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](#)

**7.11.2.384 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](#)

**7.11.2.385 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](#)

**7.11.2.386 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](#)

**7.11.2.387 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](#)

**7.11.2.388 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](#)

**7.11.2.389 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](#)

**7.11.2.390 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](#)

**7.11.2.391 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](#)

**7.11.2.392 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](#)

**7.11.2.393 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](#)

**7.11.2.394 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](#)

**7.11.2.395 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](#)

**7.11.2.396 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](#)

**7.11.2.397 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](#)

---

**7.11.2.398 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 neede 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](#)

---

**7.11.2.399 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 neede 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](#)

**7.11.2.400 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](#)

**7.11.2.401 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 neede 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](#)

**7.11.2.402 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 neede 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](#)

**7.11.2.403 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 neede 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](#)

**7.11.2.404 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 neede 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](#)

---

**7.11.2.405 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 neede 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](#)

---

**7.11.2.406 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 neede 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](#)

**7.11.2.407 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.

**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](#)

**7.11.2.408 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 neede 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](#)

**7.11.2.409 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 neede 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](#)

**7.11.2.410 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 neede 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](#)

**7.11.2.411 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 neede 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](#)

---

**7.11.2.412 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](#)

**7.11.2.413 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](#)

**7.11.2.414 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](#)

**7.11.2.415 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](#)

**7.11.2.416 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](#)

**7.11.2.417 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](#)

**7.11.2.418 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](#)

**7.11.2.419 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](#)

**7.11.2.420 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](#)

**7.11.2.421 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](#)

**7.11.2.422 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](#)

**7.11.2.423 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](#)

**7.11.2.424 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](#)

**7.11.2.425 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](#)

**7.11.2.426 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](#)

**7.11.2.427 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](#)

**7.11.2.428 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](#)

**7.11.2.429 NppStatus nppiGetFilterGaussPyramidLayerDownBorderDstROI (int nSrcROIWidth, int nSrcROIHeight, NppiSize \* pDstSizeROI, Npp32f nRate)**

Calculate destination image SizeROI width and height from source image ROI width and height and down-sampling rate.

It is highly recommended that this function be used to determine the destination image ROI for consistent results.

**Parameters:**

**nSrcROIWidth** The desired source image ROI width, must be  $\leq$  oSrcSize.width.

**nSrcROIHeight** The desired source image ROI height, must be  $\leq$  oSrcSize.height.

**pDstSizeROI** Host memory pointer to the destination image roi\_specification.

**nRate** The downsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be  $> 1.0F$  and  $\leq 10.0F$ .

**Returns:**

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

**7.11.2.430 NppStatus nppiGetFilterGaussPyramidLayerUpBorderDstROI (int nSrcROIWidth, int nSrcROIHeight, NppiSize \* pDstSizeROIMin, NppiSize \* pDstSizeROIMax, Npp32f nRate)**

Calculate destination image minimum and maximum SizeROI width and height from source image ROI width and height and upsampling rate.

It is highly recommended that this function be used to determine the best destination image ROI for consistent results.

**Parameters:**

**nSrcROIWidth** The desired source image ROI width, must be  $\leq$  oSrcSize.width.

**nSrcROIHeight** The desired source image ROI height, must be  $\leq$  oSrcSize.height.

**pDstSizeROIMin** Host memory pointer to the minimum recommended destination image roi\_specification.

**pDstSizeROIMax** Host memory pointer to the maximum recommended destination image roi\_specification.

**nRate** The upsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be  $> 1.0F$  and  $\leq 10.0F$ .

**Returns:**

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

**7.11.2.431 NppStatus nppiGradientVectorPrewittBorder\_16s32f\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

1 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

**Returns:**

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

**7.11.2.432 NppStatus nppiGradientVectorPrewittBorder\_16s32f\_C3C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

3 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

**Returns:**

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

**7.11.2.433 NppStatus nppiGradientVectorPrewittBorder\_16u32f\_C1R (const Npp16u \**pSrc*, int *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f \**pDstX*, int *nDstXStep*, Npp32f \**pDstY*, int *nDstYStep*, Npp32f \**pDstMag*, int *nDstMagStep*, Npp32f \**pDstAngle*, int *nDstAngleStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*, NppiNorm *eNorm*, NppiBorderType *eBorderType*)**

1 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

**Returns:**

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

**7.11.2.434 NppStatus nppiGradientVectorPrewittBorder\_16u32f\_C3C1R (const Npp16u \*  
*pSrc*, int *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f \* *pDstX*, int  
*nDstXStep*, Npp32f \* *pDstY*, int *nDstYStep*, Npp32f \* *pDstMag*, int *nDstMagStep*,  
*Npp32f \* pDstAngle*, int *nDstAngleStep*, NppiSize *oSizeROI*, NppiMaskSize  
*eMaskSize*, NppiNorm *eNorm*, NppiBorderType *eBorderType*)**

3 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

**Returns:**

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

**7.11.2.435 NppStatus nppiGradientVectorPrewittBorder\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

1 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

**Returns:**

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

**7.11.2.436 NppStatus nppiGradientVectorPrewittBorder\_32f\_C3C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

3 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.437 NppStatus nppiGradientVectorPrewittBorder\_8u16s\_C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s \* pDstX, int nDstXStep, Npp16s \* pDstY, int nDstYStep, Npp16s \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

1 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.438 NppStatus nppiGradientVectorPrewittBorder\_8u16s\_C3C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s \* pDstX, int nDstXStep, Npp16s \* pDstY, int nDstYStep, Npp16s \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

3 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

**Returns:**

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

**7.11.2.439 NppStatus nppiGradientVectorScharrBorder\_16s32f\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

1 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.440 NppStatus nppiGradientVectorScharrBorder\_16s32f\_C3C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

3 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.441 NppStatus nppiGradientVectorScharrBorder\_16u32f\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

1 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

**pDstX** X vector destination\_image\_pointer.

**nDstXStep** X vector destination\_image\_line\_step.

**pDstY** Y vector destination\_image\_pointer.

**nDstYStep** Y vector destination\_image\_line\_step.

**pDstMag** magnitude destination\_image\_pointer.

**nDstMagStep** magnitude destination\_image\_line\_step.

**pDstAngle** angle destination\_image\_pointer.

**nDstAngleStep** angle destination\_image\_line\_step.

**oSizeROI** Region-of-Interest (ROI).

**eMaskSize** fixed filter mask size to use.

**eNorm** gradient distance method to use.

**eBorderType** source image border type to use use.

**Returns:**

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

**7.11.2.442 NppStatus nppiGradientVectorScharrBorder\_16u32f\_C3C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

3 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.443 NppStatus nppiGradientVectorScharrBorder\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

1 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.444 NppStatus nppiGradientVectorScharrBorder\_32f\_C3C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

3 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

**Returns:**

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

**7.11.2.445 NppStatus nppiGradientVectorScharrBorder\_8u16s\_C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s \* pDstX, int nDstXStep, Npp16s \* pDstY, int nDstYStep, Npp16s \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

1 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.446 NppStatus nppiGradientVectorScharrBorder\_8u16s\_C3C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s \* pDstX, int nDstXStep, Npp16s \* pDstY, int nDstYStep, Npp16s \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

3 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.447 NppStatus nppiGradientVectorSobelBorder\_16s32f\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

1 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

**Returns:**

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

**7.11.2.448 NppStatus nppiGradientVectorSobelBorder\_16s32f\_C3C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

3 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.449 NppStatus nppiGradientVectorSobelBorder\_16u32f\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

1 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.450 NppStatus nppiGradientVectorSobelBorder\_16u32f\_C3C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

3 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

**Returns:**

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

**7.11.2.451 NppStatus nppiGradientVectorSobelBorder\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

1 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.452 NppStatus nppiGradientVectorSobelBorder\_32f\_C3C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDstX, int nDstXStep, Npp32f \* pDstY, int nDstYStep, Npp32f \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

3 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

#### Returns:

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

**7.11.2.453 NppStatus nppiGradientVectorSobelBorder\_8u16s\_C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s \* pDstX, int nDstXStep, Npp16s \* pDstY, int nDstYStep, Npp16s \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

1 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

**Returns:**

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

**7.11.2.454 NppStatus nppiGradientVectorSobelBorder\_8u16s\_C3C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s \* pDstX, int nDstXStep, Npp16s \* pDstY, int nDstYStep, Npp16s \* pDstMag, int nDstMagStep, Npp32f \* pDstAngle, int nDstAngleStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiNorm eNorm, NppiBorderType eBorderType)**

3 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with 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.

*pDstX* X vector destination\_image\_pointer.

*nDstXStep* X vector destination\_image\_line\_step.

*pDstY* Y vector destination\_image\_pointer.

*nDstYStep* Y vector destination\_image\_line\_step.

*pDstMag* magnitude destination\_image\_pointer.

*nDstMagStep* magnitude destination\_image\_line\_step.

*pDstAngle* angle destination\_image\_pointer.

*nDstAngleStep* angle destination\_image\_line\_step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* fixed filter mask size to use.

*eNorm* gradient distance method to use.

*eBorderType* source image border type to use use.

**Returns:**

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

## 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/r8.0/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/r8.0/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/r8.0/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/r8.0/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/r8.0/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 (lowest memory address).*
- int **y**  
*y-coordinate of upper left corner (lowest memory address).*
- 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 (lowest memory address).

#### 8.6.2.4 int NppiRect::y

y-coordinate of upper left corner (lowest memory address).

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r8.0/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/r8.0/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/r8.0/NPP/npp/include/nppdefs.h

# Index

- \_\_align\_\_
  - npp\_basic\_types, 48, 49
- 1D Linear Filter, 51
- 1D Window Sum, 123
- 1D Window Sum with Border Control, 134
- 2D Fixed Linear Filters, 206
- Basic NPP Data Types, 46
- build
  - NppLibraryVersion, 534
- classifiers
  - NppiHaarClassifier\_32f, 530
- classifierSize
  - NppiHaarClassifier\_32f, 530
- classifierStep
  - NppiHaarClassifier\_32f, 530
- Convolution, 147
- core\_npp
  - nppGetGpuComputeCapability, 28
  - nppGetGpuDeviceProperties, 28
  - nppGetGpuName, 28
  - nppGetGpuNumSMs, 28
  - nppGetLibVersion, 28
  - nppGetMaxThreadsPerBlock, 29
  - nppGetMaxThreadsPerSM, 29
  - nppGetStream, 29
  - nppGetStreamMaxThreadsPerSM, 29
  - nppGetStreamNumSMs, 29
  - nppSetStream, 29
- counterDevice
  - NppiHaarClassifier\_32f, 530
- Filtering Functions, 50
- Fixed Filters, 274
- fixed\_filters
  - nppiFilterBilateralGaussBorder\_16u\_C1R, 329
  - nppiFilterBilateralGaussBorder\_16u\_C3R, 330
  - nppiFilterBilateralGaussBorder\_32f\_C1R, 330
  - nppiFilterBilateralGaussBorder\_32f\_C3R, 331
  - nppiFilterBilateralGaussBorder\_8u\_C1R, 331
  - nppiFilterBilateralGaussBorder\_8u\_C3R, 332
  - nppiFilterCannyBorder\_8u\_C1R, 333
- nppiFilterCannyBorderGetBufferSize, 333
- nppiFilterGauss\_16s\_AC4R, 334
- nppiFilterGauss\_16s\_C1R, 334
- nppiFilterGauss\_16s\_C3R, 334
- nppiFilterGauss\_16s\_C4R, 335
- nppiFilterGauss\_16u\_AC4R, 335
- nppiFilterGauss\_16u\_C1R, 335
- nppiFilterGauss\_16u\_C3R, 336
- nppiFilterGauss\_16u\_C4R, 336
- nppiFilterGauss\_32f\_AC4R, 336
- nppiFilterGauss\_32f\_C1R, 337
- nppiFilterGauss\_32f\_C3R, 337
- nppiFilterGauss\_32f\_C4R, 337
- nppiFilterGauss\_8u\_AC4R, 338
- nppiFilterGauss\_8u\_C1R, 338
- nppiFilterGauss\_8u\_C3R, 338
- nppiFilterGauss\_8u\_C4R, 339
- nppiFilterGaussAdvanced\_16s\_AC4R, 339
- nppiFilterGaussAdvanced\_16s\_C1R, 340
- nppiFilterGaussAdvanced\_16s\_C3R, 340
- nppiFilterGaussAdvanced\_16s\_C4R, 340
- nppiFilterGaussAdvanced\_16u\_AC4R, 341
- nppiFilterGaussAdvanced\_16u\_C1R, 341
- nppiFilterGaussAdvanced\_16u\_C3R, 342
- nppiFilterGaussAdvanced\_16u\_C4R, 342
- nppiFilterGaussAdvanced\_32f\_AC4R, 342
- nppiFilterGaussAdvanced\_32f\_C1R, 343
- nppiFilterGaussAdvanced\_32f\_C3R, 343
- nppiFilterGaussAdvanced\_32f\_C4R, 344
- nppiFilterGaussAdvanced\_8u\_AC4R, 344
- nppiFilterGaussAdvanced\_8u\_C1R, 344
- nppiFilterGaussAdvanced\_8u\_C3R, 345
- nppiFilterGaussAdvanced\_8u\_C4R, 345
- nppiFilterGaussAdvancedBorder\_16s\_AC4R, 346
- nppiFilterGaussAdvancedBorder\_16s\_C1R, 346
- nppiFilterGaussAdvancedBorder\_16s\_C3R, 347
- nppiFilterGaussAdvancedBorder\_16s\_C4R, 347
- nppiFilterGaussAdvancedBorder\_16u\_AC4R, 348
- nppiFilterGaussAdvancedBorder\_16u\_C1R, 348

- nppiFilterGaussAdvancedBorder\_16u\_C3R,  
349  
nppiFilterGaussAdvancedBorder\_16u\_C4R,  
349  
nppiFilterGaussAdvancedBorder\_32f\_AC4R,  
350  
nppiFilterGaussAdvancedBorder\_32f\_C1R,  
350  
nppiFilterGaussAdvancedBorder\_32f\_C3R,  
351  
nppiFilterGaussAdvancedBorder\_32f\_C4R,  
351  
nppiFilterGaussAdvancedBorder\_8u\_AC4R,  
352  
nppiFilterGaussAdvancedBorder\_8u\_C1R,  
352  
nppiFilterGaussAdvancedBorder\_8u\_C3R,  
353  
nppiFilterGaussAdvancedBorder\_8u\_C4R,  
353  
nppiFilterGaussBorder\_16s\_AC4R, 354  
nppiFilterGaussBorder\_16s\_C1R, 354  
nppiFilterGaussBorder\_16s\_C3R, 355  
nppiFilterGaussBorder\_16s\_C4R, 355  
nppiFilterGaussBorder\_16u\_AC4R, 356  
nppiFilterGaussBorder\_16u\_C1R, 356  
nppiFilterGaussBorder\_16u\_C3R, 356  
nppiFilterGaussBorder\_16u\_C4R, 357  
nppiFilterGaussBorder\_32f\_AC4R, 357  
nppiFilterGaussBorder\_32f\_C1R, 358  
nppiFilterGaussBorder\_32f\_C3R, 358  
nppiFilterGaussBorder\_32f\_C4R, 359  
nppiFilterGaussBorder\_8u\_AC4R, 359  
nppiFilterGaussBorder\_8u\_C1R, 360  
nppiFilterGaussBorder\_8u\_C3R, 360  
nppiFilterGaussBorder\_8u\_C4R, 361  
nppiFilterGaussPyramidLayerDownBorder\_-  
16u\_C1R, 361  
nppiFilterGaussPyramidLayerDownBorder\_-  
16u\_C3R, 362  
nppiFilterGaussPyramidLayerDownBorder\_-  
32f\_C1R, 363  
nppiFilterGaussPyramidLayerDownBorder\_-  
32f\_C3R, 363  
nppiFilterGaussPyramidLayerDownBorder\_-  
8u\_C1R, 364  
nppiFilterGaussPyramidLayerDownBorder\_-  
8u\_C3R, 364  
nppiFilterGaussPyramidLayerUpBorder\_-  
16u\_C1R, 365  
nppiFilterGaussPyramidLayerUpBorder\_-  
16u\_C3R, 365  
nppiFilterGaussPyramidLayerUpBorder\_-  
32f\_C1R, 366
- nppiFilterGaussPyramidLayerUpBorder\_-  
32f\_C3R, 367  
nppiFilterGaussPyramidLayerUpBorder\_8u\_-  
C1R, 367  
nppiFilterGaussPyramidLayerUpBorder\_8u\_-  
C3R, 368  
nppiFilterHarrisCornersBorder\_8u32f\_C1R,  
368  
nppiFilterHarrisCornersBorderGetBufferSize,  
369  
nppiFilterHighPass\_16s\_AC4R, 369  
nppiFilterHighPass\_16s\_C1R, 370  
nppiFilterHighPass\_16s\_C3R, 370  
nppiFilterHighPass\_16s\_C4R, 370  
nppiFilterHighPass\_16u\_AC4R, 371  
nppiFilterHighPass\_16u\_C1R, 371  
nppiFilterHighPass\_16u\_C3R, 371  
nppiFilterHighPass\_16u\_C4R, 372  
nppiFilterHighPass\_32f\_AC4R, 372  
nppiFilterHighPass\_32f\_C1R, 372  
nppiFilterHighPass\_32f\_C3R, 373  
nppiFilterHighPass\_32f\_C4R, 373  
nppiFilterHighPass\_8u\_AC4R, 373  
nppiFilterHighPass\_8u\_C1R, 374  
nppiFilterHighPass\_8u\_C3R, 374  
nppiFilterHighPass\_8u\_C4R, 374  
nppiFilterHighPassBorder\_16s\_AC4R, 375  
nppiFilterHighPassBorder\_16s\_C1R, 375  
nppiFilterHighPassBorder\_16s\_C3R, 376  
nppiFilterHighPassBorder\_16s\_C4R, 376  
nppiFilterHighPassBorder\_16u\_AC4R, 377  
nppiFilterHighPassBorder\_16u\_C1R, 377  
nppiFilterHighPassBorder\_16u\_C3R, 378  
nppiFilterHighPassBorder\_16u\_C4R, 378  
nppiFilterHighPassBorder\_32f\_AC4R, 378  
nppiFilterHighPassBorder\_32f\_C1R, 379  
nppiFilterHighPassBorder\_32f\_C3R, 379  
nppiFilterHighPassBorder\_32f\_C4R, 380  
nppiFilterHighPassBorder\_8u\_AC4R, 380  
nppiFilterHighPassBorder\_8u\_C1R, 381  
nppiFilterHighPassBorder\_8u\_C3R, 381  
nppiFilterHighPassBorder\_8u\_C4R, 382  
nppiFilterLaplace\_16s\_AC4R, 382  
nppiFilterLaplace\_16s\_C1R, 383  
nppiFilterLaplace\_16s\_C3R, 383  
nppiFilterLaplace\_16s\_C4R, 383  
nppiFilterLaplace\_32f\_AC4R, 384  
nppiFilterLaplace\_32f\_C1R, 384  
nppiFilterLaplace\_32f\_C3R, 384  
nppiFilterLaplace\_32f\_C4R, 385  
nppiFilterLaplace\_8s16s\_C1R, 385  
nppiFilterLaplace\_8u16s\_C1R, 385  
nppiFilterLaplace\_8u\_AC4R, 386  
nppiFilterLaplace\_8u\_C1R, 386

- nppiFilterLaplace\_8u\_C3R, 386  
nppiFilterLaplace\_8u\_C4R, 387  
nppiFilterLaplaceBorder\_16s\_AC4R, 387  
nppiFilterLaplaceBorder\_16s\_C1R, 388  
nppiFilterLaplaceBorder\_16s\_C3R, 388  
nppiFilterLaplaceBorder\_16s\_C4R, 389  
nppiFilterLaplaceBorder\_32f\_AC4R, 389  
nppiFilterLaplaceBorder\_32f\_C1R, 389  
nppiFilterLaplaceBorder\_32f\_C3R, 390  
nppiFilterLaplaceBorder\_32f\_C4R, 390  
nppiFilterLaplaceBorder\_8s16s\_C1R, 391  
nppiFilterLaplaceBorder\_8u16s\_C1R, 391  
nppiFilterLaplaceBorder\_8u\_AC4R, 392  
nppiFilterLaplaceBorder\_8u\_C1R, 392  
nppiFilterLaplaceBorder\_8u\_C3R, 393  
nppiFilterLaplaceBorder\_8u\_C4R, 393  
nppiFilterLowPass\_16s\_AC4R, 394  
nppiFilterLowPass\_16s\_C1R, 394  
nppiFilterLowPass\_16s\_C3R, 395  
nppiFilterLowPass\_16s\_C4R, 395  
nppiFilterLowPass\_16u\_AC4R, 395  
nppiFilterLowPass\_16u\_C1R, 396  
nppiFilterLowPass\_16u\_C3R, 396  
nppiFilterLowPass\_16u\_C4R, 396  
nppiFilterLowPass\_32f\_AC4R, 397  
nppiFilterLowPass\_32f\_C1R, 397  
nppiFilterLowPass\_32f\_C3R, 397  
nppiFilterLowPass\_32f\_C4R, 398  
nppiFilterLowPass\_8u\_AC4R, 398  
nppiFilterLowPass\_8u\_C1R, 398  
nppiFilterLowPass\_8u\_C3R, 399  
nppiFilterLowPass\_8u\_C4R, 399  
nppiFilterLowPassBorder\_16s\_AC4R, 399  
nppiFilterLowPassBorder\_16s\_C1R, 400  
nppiFilterLowPassBorder\_16s\_C3R, 400  
nppiFilterLowPassBorder\_16s\_C4R, 401  
nppiFilterLowPassBorder\_16u\_AC4R, 401  
nppiFilterLowPassBorder\_16u\_C1R, 402  
nppiFilterLowPassBorder\_16u\_C3R, 402  
nppiFilterLowPassBorder\_16u\_C4R, 403  
nppiFilterLowPassBorder\_32f\_AC4R, 403  
nppiFilterLowPassBorder\_32f\_C1R, 404  
nppiFilterLowPassBorder\_32f\_C3R, 404  
nppiFilterLowPassBorder\_32f\_C4R, 405  
nppiFilterLowPassBorder\_8u\_AC4R, 405  
nppiFilterLowPassBorder\_8u\_C1R, 406  
nppiFilterLowPassBorder\_8u\_C3R, 406  
nppiFilterLowPassBorder\_8u\_C4R, 407  
nppiFilterPrewittHoriz\_16s\_AC4R, 407  
nppiFilterPrewittHoriz\_16s\_C1R, 408  
nppiFilterPrewittHoriz\_16s\_C3R, 408  
nppiFilterPrewittHoriz\_16s\_C4R, 408  
nppiFilterPrewittHoriz\_32f\_AC4R, 409  
nppiFilterPrewittHoriz\_32f\_C1R, 409  
nppiFilterPrewittHoriz\_32f\_C3R, 409  
nppiFilterPrewittHoriz\_32f\_C4R, 410  
nppiFilterPrewittHoriz\_8u\_AC4R, 410  
nppiFilterPrewittHoriz\_8u\_C1R, 410  
nppiFilterPrewittHoriz\_8u\_C3R, 411  
nppiFilterPrewittHoriz\_8u\_C4R, 411  
nppiFilterPrewittHorizBorder\_16s\_AC4R, 411  
nppiFilterPrewittHorizBorder\_16s\_C1R, 412  
nppiFilterPrewittHorizBorder\_16s\_C3R, 412  
nppiFilterPrewittHorizBorder\_16s\_C4R, 413  
nppiFilterPrewittHorizBorder\_32f\_AC4R, 413  
nppiFilterPrewittHorizBorder\_32f\_C1R, 414  
nppiFilterPrewittHorizBorder\_32f\_C3R, 414  
nppiFilterPrewittHorizBorder\_32f\_C4R, 414  
nppiFilterPrewittHorizBorder\_8u\_AC4R, 415  
nppiFilterPrewittHorizBorder\_8u\_C1R, 415  
nppiFilterPrewittHorizBorder\_8u\_C3R, 416  
nppiFilterPrewittHorizBorder\_8u\_C4R, 416  
nppiFilterPrewittVert\_16s\_AC4R, 417  
nppiFilterPrewittVert\_16s\_C1R, 417  
nppiFilterPrewittVert\_16s\_C3R, 417  
nppiFilterPrewittVert\_16s\_C4R, 418  
nppiFilterPrewittVert\_32f\_AC4R, 418  
nppiFilterPrewittVert\_32f\_C1R, 418  
nppiFilterPrewittVert\_32f\_C3R, 419  
nppiFilterPrewittVert\_32f\_C4R, 419  
nppiFilterPrewittVert\_8u\_AC4R, 419  
nppiFilterPrewittVert\_8u\_C1R, 420  
nppiFilterPrewittVert\_8u\_C3R, 420  
nppiFilterPrewittVert\_8u\_C4R, 420  
nppiFilterPrewittVertBorder\_16s\_AC4R, 421  
nppiFilterPrewittVertBorder\_16s\_C1R, 421  
nppiFilterPrewittVertBorder\_16s\_C3R, 422  
nppiFilterPrewittVertBorder\_16s\_C4R, 422  
nppiFilterPrewittVertBorder\_32f\_AC4R, 422  
nppiFilterPrewittVertBorder\_32f\_C1R, 423  
nppiFilterPrewittVertBorder\_32f\_C3R, 423  
nppiFilterPrewittVertBorder\_32f\_C4R, 424  
nppiFilterPrewittVertBorder\_8u\_AC4R, 424  
nppiFilterPrewittVertBorder\_8u\_C1R, 425  
nppiFilterPrewittVertBorder\_8u\_C3R, 425  
nppiFilterPrewittVertBorder\_8u\_C4R, 425  
nppiFilterRobertsDown\_16s\_AC4R, 426  
nppiFilterRobertsDown\_16s\_C1R, 426  
nppiFilterRobertsDown\_16s\_C3R, 427  
nppiFilterRobertsDown\_16s\_C4R, 427  
nppiFilterRobertsDown\_32f\_AC4R, 427  
nppiFilterRobertsDown\_32f\_C1R, 428  
nppiFilterRobertsDown\_32f\_C3R, 428  
nppiFilterRobertsDown\_32f\_C4R, 428  
nppiFilterRobertsDown\_8u\_AC4R, 429  
nppiFilterRobertsDown\_8u\_C1R, 429  
nppiFilterRobertsDown\_8u\_C3R, 429  
nppiFilterRobertsDown\_8u\_C4R, 430

- nppiFilterRobertsDownBorder\_16s\_AC4R,  
  430  
 nppiFilterRobertsDownBorder\_16s\_C1R, 431  
 nppiFilterRobertsDownBorder\_16s\_C3R, 431  
 nppiFilterRobertsDownBorder\_16s\_C4R, 431  
 nppiFilterRobertsDownBorder\_32f\_AC4R,  
  432  
 nppiFilterRobertsDownBorder\_32f\_C1R, 432  
 nppiFilterRobertsDownBorder\_32f\_C3R, 433  
 nppiFilterRobertsDownBorder\_32f\_C4R, 433  
 nppiFilterRobertsDownBorder\_8u\_AC4R, 434  
 nppiFilterRobertsDownBorder\_8u\_C1R, 434  
 nppiFilterRobertsDownBorder\_8u\_C3R, 434  
 nppiFilterRobertsDownBorder\_8u\_C4R, 435  
 nppiFilterRobertsUp\_16s\_AC4R, 435  
 nppiFilterRobertsUp\_16s\_C1R, 436  
 nppiFilterRobertsUp\_16s\_C3R, 436  
 nppiFilterRobertsUp\_16s\_C4R, 436  
 nppiFilterRobertsUp\_32f\_AC4R, 437  
 nppiFilterRobertsUp\_32f\_C1R, 437  
 nppiFilterRobertsUp\_32f\_C3R, 437  
 nppiFilterRobertsUp\_32f\_C4R, 438  
 nppiFilterRobertsUp\_8u\_AC4R, 438  
 nppiFilterRobertsUp\_8u\_C1R, 438  
 nppiFilterRobertsUp\_8u\_C3R, 439  
 nppiFilterRobertsUp\_8u\_C4R, 439  
 nppiFilterRobertsUpBorder\_16s\_AC4R, 439  
 nppiFilterRobertsUpBorder\_16s\_C1R, 440  
 nppiFilterRobertsUpBorder\_16s\_C3R, 440  
 nppiFilterRobertsUpBorder\_16s\_C4R, 441  
 nppiFilterRobertsUpBorder\_32f\_AC4R, 441  
 nppiFilterRobertsUpBorder\_32f\_C1R, 442  
 nppiFilterRobertsUpBorder\_32f\_C3R, 442  
 nppiFilterRobertsUpBorder\_32f\_C4R, 442  
 nppiFilterRobertsUpBorder\_8u\_AC4R, 443  
 nppiFilterRobertsUpBorder\_8u\_C1R, 443  
 nppiFilterRobertsUpBorder\_8u\_C3R, 444  
 nppiFilterRobertsUpBorder\_8u\_C4R, 444  
 nppiFilterScharrHoriz\_32f\_C1R, 445  
 nppiFilterScharrHoriz\_8s16s\_C1R, 445  
 nppiFilterScharrHoriz\_8u16s\_C1R, 445  
 nppiFilterScharrHorizBorder\_32f\_C1R, 446  
 nppiFilterScharrHorizBorder\_8s16s\_C1R, 446  
 nppiFilterScharrHorizBorder\_8u16s\_C1R,  
  447  
 nppiFilterScharrVert\_32f\_C1R, 447  
 nppiFilterScharrVert\_8s16s\_C1R, 447  
 nppiFilterScharrVert\_8u16s\_C1R, 448  
 nppiFilterScharrVertBorder\_32f\_C1R, 448  
 nppiFilterScharrVertBorder\_8s16s\_C1R, 449  
 nppiFilterScharrVertBorder\_8u16s\_C1R, 449  
 nppiFilterSharpen\_16s\_AC4R, 449  
 nppiFilterSharpen\_16s\_C1R, 450  
 nppiFilterSharpen\_16s\_C3R, 450  
 nppiFilterSharpen\_16s\_C4R, 450  
 nppiFilterSharpen\_32f\_AC4R, 450  
 nppiFilterSharpen\_16u\_AC4R, 451  
 nppiFilterSharpen\_16u\_C1R, 451  
 nppiFilterSharpen\_16u\_C3R, 451  
 nppiFilterSharpen\_16u\_C4R, 452  
 nppiFilterSharpen\_32f\_AC4R, 452  
 nppiFilterSharpen\_32f\_C1R, 452  
 nppiFilterSharpen\_32f\_C3R, 453  
 nppiFilterSharpen\_32f\_C4R, 453  
 nppiFilterSharpen\_8u\_AC4R, 453  
 nppiFilterSharpen\_8u\_C1R, 454  
 nppiFilterSharpen\_8u\_C3R, 454  
 nppiFilterSharpen\_8u\_C4R, 454  
 nppiFilterSharpenBorder\_16s\_AC4R, 455  
 nppiFilterSharpenBorder\_16s\_C1R, 455  
 nppiFilterSharpenBorder\_16s\_C3R, 456  
 nppiFilterSharpenBorder\_16s\_C4R, 456  
 nppiFilterSharpenBorder\_16u\_AC4R, 456  
 nppiFilterSharpenBorder\_16u\_C1R, 457  
 nppiFilterSharpenBorder\_16u\_C3R, 457  
 nppiFilterSharpenBorder\_16u\_C4R, 458  
 nppiFilterSharpenBorder\_32f\_AC4R, 458  
 nppiFilterSharpenBorder\_32f\_C1R, 459  
 nppiFilterSharpenBorder\_32f\_C3R, 459  
 nppiFilterSharpenBorder\_32f\_C4R, 459  
 nppiFilterSharpenBorder\_8u\_AC4R, 460  
 nppiFilterSharpenBorder\_8u\_C1R, 460  
 nppiFilterSharpenBorder\_8u\_C3R, 461  
 nppiFilterSharpenBorder\_8u\_C4R, 461  
 nppiFilterSobelCross\_32f\_C1R, 462  
 nppiFilterSobelCross\_8s16s\_C1R, 462  
 nppiFilterSobelCross\_8u16s\_C1R, 462  
 nppiFilterSobelCrossBorder\_32f\_C1R, 463  
 nppiFilterSobelCrossBorder\_8s16s\_C1R, 463  
 nppiFilterSobelCrossBorder\_8u16s\_C1R, 464  
 nppiFilterSobelHoriz\_16s\_AC4R, 464  
 nppiFilterSobelHoriz\_16s\_C1R, 464  
 nppiFilterSobelHoriz\_16s\_C3R, 465  
 nppiFilterSobelHoriz\_16s\_C4R, 465  
 nppiFilterSobelHoriz\_32f\_AC4R, 465  
 nppiFilterSobelHoriz\_32f\_C1R, 466  
 nppiFilterSobelHoriz\_32f\_C3R, 466  
 nppiFilterSobelHoriz\_32f\_C4R, 466  
 nppiFilterSobelHoriz\_8s16s\_C1R, 467  
 nppiFilterSobelHoriz\_8u16s\_C1R, 467  
 nppiFilterSobelHoriz\_8u\_AC4R, 467  
 nppiFilterSobelHoriz\_8u\_C1R, 468  
 nppiFilterSobelHoriz\_8u\_C3R, 468  
 nppiFilterSobelHoriz\_8u\_C4R, 468  
 nppiFilterSobelHorizBorder\_16s\_AC4R, 469  
 nppiFilterSobelHorizBorder\_16s\_C1R, 469  
 nppiFilterSobelHorizBorder\_16s\_C3R, 470  
 nppiFilterSobelHorizBorder\_16s\_C4R, 470  
 nppiFilterSobelHorizBorder\_32f\_AC4R, 470

nppiFilterSobelHorizBorder\_32f\_C1R, 471  
nppiFilterSobelHorizBorder\_32f\_C3R, 471  
nppiFilterSobelHorizBorder\_32f\_C4R, 472  
nppiFilterSobelHorizBorder\_8s16s\_C1R, 472  
nppiFilterSobelHorizBorder\_8u16s\_C1R, 473  
nppiFilterSobelHorizBorder\_8u\_AC4R, 473  
nppiFilterSobelHorizBorder\_8u\_C1R, 474  
nppiFilterSobelHorizBorder\_8u\_C3R, 474  
nppiFilterSobelHorizBorder\_8u\_C4R, 474  
nppiFilterSobelHorizMask\_32f\_C1R, 475  
nppiFilterSobelHorizMaskBorder\_32f\_C1R,  
    475  
nppiFilterSobelHorizSecond\_32f\_C1R, 476  
nppiFilterSobelHorizSecond\_8s16s\_C1R, 476  
nppiFilterSobelHorizSecond\_8u16s\_C1R, 476  
nppiFilterSobelHorizSecondBorder\_32f\_C1R,  
    477  
nppiFilterSobelHorizSecondBorder\_8s16s\_-  
    C1R, 477  
nppiFilterSobelHorizSecondBorder\_8u16s\_-  
    C1R, 478  
nppiFilterSobelVert\_16s\_AC4R, 478  
nppiFilterSobelVert\_16s\_C1R, 479  
nppiFilterSobelVert\_16s\_C3R, 479  
nppiFilterSobelVert\_16s\_C4R, 479  
nppiFilterSobelVert\_32f\_AC4R, 480  
nppiFilterSobelVert\_32f\_C1R, 480  
nppiFilterSobelVert\_32f\_C3R, 480  
nppiFilterSobelVert\_32f\_C4R, 481  
nppiFilterSobelVert\_8s16s\_C1R, 481  
nppiFilterSobelVert\_8u16s\_C1R, 481  
nppiFilterSobelVert\_8u\_AC4R, 482  
nppiFilterSobelVert\_8u\_C1R, 482  
nppiFilterSobelVert\_8u\_C3R, 482  
nppiFilterSobelVert\_8u\_C4R, 483  
nppiFilterSobelVertBorder\_16s\_AC4R, 483  
nppiFilterSobelVertBorder\_16s\_C1R, 484  
nppiFilterSobelVertBorder\_16s\_C3R, 484  
nppiFilterSobelVertBorder\_16s\_C4R, 484  
nppiFilterSobelVertBorder\_32f\_AC4R, 485  
nppiFilterSobelVertBorder\_32f\_C1R, 485  
nppiFilterSobelVertBorder\_32f\_C3R, 486  
nppiFilterSobelVertBorder\_32f\_C4R, 486  
nppiFilterSobelVertBorder\_8s16s\_C1R, 487  
nppiFilterSobelVertBorder\_8u16s\_C1R, 487  
nppiFilterSobelVertBorder\_8u\_AC4R, 488  
nppiFilterSobelVertBorder\_8u\_C1R, 488  
nppiFilterSobelVertBorder\_8u\_C3R, 488  
nppiFilterSobelVertBorder\_8u\_C4R, 489  
nppiFilterSobelVertMask\_32f\_C1R, 489  
nppiFilterSobelVertMaskBorder\_32f\_C1R,  
    490  
nppiFilterSobelVertSecond\_32f\_C1R, 490  
nppiFilterSobelVertSecond\_8s16s\_C1R, 491  
nppiFilterSobelVertSecond\_8u16s\_C1R, 491  
nppiFilterSobelVertSecondBorder\_32f\_C1R,  
    491  
nppiFilterSobelVertSecondBorder\_8s16s\_-  
    C1R, 492  
nppiFilterSobelVertSecondBorder\_8u16s\_-  
    C1R, 492  
nppiFilterUnsharpBorder\_16s\_AC4R, 493  
nppiFilterUnsharpBorder\_16s\_C1R, 493  
nppiFilterUnsharpBorder\_16s\_C3R, 494  
nppiFilterUnsharpBorder\_16s\_C4R, 495  
nppiFilterUnsharpBorder\_16u\_AC4R, 495  
nppiFilterUnsharpBorder\_16u\_C1R, 496  
nppiFilterUnsharpBorder\_16u\_C3R, 496  
nppiFilterUnsharpBorder\_16u\_C4R, 497  
nppiFilterUnsharpBorder\_32f\_AC4R, 497  
nppiFilterUnsharpBorder\_32f\_C1R, 498  
nppiFilterUnsharpBorder\_32f\_C3R, 499  
nppiFilterUnsharpBorder\_32f\_C4R, 499  
nppiFilterUnsharpBorder\_8u\_AC4R, 500  
nppiFilterUnsharpBorder\_8u\_C1R, 500  
nppiFilterUnsharpBorder\_8u\_C3R, 501  
nppiFilterUnsharpBorder\_8u\_C4R, 501  
nppiFilterUnsharpGetBufferSize\_16s\_AC4R,  
    502  
nppiFilterUnsharpGetBufferSize\_16s\_C1R,  
    502  
nppiFilterUnsharpGetBufferSize\_16s\_C3R,  
    503  
nppiFilterUnsharpGetBufferSize\_16s\_C4R,  
    503  
nppiFilterUnsharpGetBufferSize\_16u\_AC4R,  
    503  
nppiFilterUnsharpGetBufferSize\_16u\_C1R,  
    503  
nppiFilterUnsharpGetBufferSize\_16u\_C3R,  
    504  
nppiFilterUnsharpGetBufferSize\_16u\_C4R,  
    504  
nppiFilterUnsharpGetBufferSize\_32f\_AC4R,  
    504  
nppiFilterUnsharpGetBufferSize\_32f\_C1R,  
    505  
nppiFilterUnsharpGetBufferSize\_32f\_C3R,  
    505  
nppiFilterUnsharpGetBufferSize\_32f\_C4R,  
    505  
nppiFilterUnsharpGetBufferSize\_8u\_AC4R,  
    505  
nppiFilterUnsharpGetBufferSize\_8u\_C1R,  
    506  
nppiFilterUnsharpGetBufferSize\_8u\_C3R,  
    506

- nppiFilterUnsharpGetBufferSize\_8u\_C4R,  
    506
- nppiGetFilterGaussPyramidLayerDownBorderDstROI, 507
- nppiGetFilterGaussPyramidLayerUpBorderDstROI, 507
- nppiGradientVectorPrewittBorder\_16s32f\_C1R, 508
- nppiGradientVectorPrewittBorder\_16s32f\_C3C1R, 508
- nppiGradientVectorPrewittBorder\_16u32f\_C1R, 509
- nppiGradientVectorPrewittBorder\_16u32f\_C3C1R, 510
- nppiGradientVectorPrewittBorder\_32f\_C1R, 510
- nppiGradientVectorPrewittBorder\_32f\_C3C1R, 511
- nppiGradientVectorPrewittBorder\_8u16s\_C1R, 512
- nppiGradientVectorPrewittBorder\_8u16s\_C3C1R, 512
- nppiGradientVectorScharrBorder\_16s32f\_C1R, 513
- nppiGradientVectorScharrBorder\_16s32f\_C3C1R, 514
- nppiGradientVectorScharrBorder\_16u32f\_C1R, 514
- nppiGradientVectorScharrBorder\_16u32f\_C3C1R, 515
- nppiGradientVectorScharrBorder\_32f\_C1R, 516
- nppiGradientVectorScharrBorder\_32f\_C3C1R, 516
- nppiGradientVectorScharrBorder\_8u16s\_C1R, 517
- nppiGradientVectorScharrBorder\_8u16s\_C3C1R, 518
- nppiGradientVectorSobelBorder\_16s32f\_C1R, 518
- nppiGradientVectorSobelBorder\_16s32f\_C3C1R, 519
- nppiGradientVectorSobelBorder\_16u32f\_C1R, 520
- nppiGradientVectorSobelBorder\_16u32f\_C3C1R, 520
- nppiGradientVectorSobelBorder\_32f\_C1R, 521
- nppiGradientVectorSobelBorder\_32f\_C3C1R, 522
- nppiGradientVectorSobelBorder\_8u16s\_C1R, 522
- nppiGradientVectorSobelBorder\_8u16s\_C3C1R, 523
- haarBuffer  
    NppiHaarBuffer, 529
- haarBufferSize  
    NppiHaarBuffer, 529
- height  
    NppiRect, 532
- NppiSize, 533
- im  
    NPP\_ALIGN\_16, 525
- NPP\_ALIGN\_8, 527
- image\_1D\_linear\_filter  
    nppiFilterColumn32f\_16s\_AC4R, 63
- nppiFilterColumn32f\_16s\_C1R, 63
- nppiFilterColumn32f\_16s\_C3R, 64
- nppiFilterColumn32f\_16s\_C4R, 64
- nppiFilterColumn32f\_16u\_AC4R, 65
- nppiFilterColumn32f\_16u\_C1R, 65
- nppiFilterColumn32f\_16u\_C3R, 66
- nppiFilterColumn32f\_16u\_C4R, 66
- nppiFilterColumn32f\_8u\_AC4R, 66
- nppiFilterColumn32f\_8u\_C1R, 67
- nppiFilterColumn32f\_8u\_C3R, 67
- nppiFilterColumn32f\_8u\_C4R, 68
- nppiFilterColumn\_16s\_AC4R, 68
- nppiFilterColumn\_16s\_C1R, 69
- nppiFilterColumn\_16s\_C3R, 69
- nppiFilterColumn\_16s\_C4R, 70
- nppiFilterColumn\_16u\_AC4R, 70
- nppiFilterColumn\_16u\_C1R, 71
- nppiFilterColumn\_16u\_C3R, 71
- nppiFilterColumn\_16u\_C4R, 72
- nppiFilterColumn\_32f\_AC4R, 72
- nppiFilterColumn\_32f\_C1R, 73
- nppiFilterColumn\_32f\_C3R, 73
- nppiFilterColumn\_32f\_C4R, 74
- nppiFilterColumn\_64f\_C1R, 74
- nppiFilterColumn\_8u\_AC4R, 75
- nppiFilterColumn\_8u\_C1R, 75
- nppiFilterColumn\_8u\_C3R, 76
- nppiFilterColumn\_8u\_C4R, 76
- nppiFilterColumnBorder32f\_16s\_AC4R, 77
- nppiFilterColumnBorder32f\_16s\_C1R, 77
- nppiFilterColumnBorder32f\_16s\_C3R, 78
- nppiFilterColumnBorder32f\_16s\_C4R, 78
- nppiFilterColumnBorder32f\_16u\_AC4R, 79
- nppiFilterColumnBorder32f\_16u\_C1R, 79
- nppiFilterColumnBorder32f\_16u\_C3R, 80
- nppiFilterColumnBorder32f\_16u\_C4R, 80
- nppiFilterColumnBorder32f\_8u\_AC4R, 81
- nppiFilterColumnBorder32f\_8u\_C1R, 81
- nppiFilterColumnBorder32f\_8u\_C3R, 82
- nppiFilterColumnBorder32f\_8u\_C4R, 82
- nppiFilterColumnBorder\_16s\_AC4R, 83

- nppiFilterColumnBorder\_16s\_C1R, 84  
nppiFilterColumnBorder\_16s\_C3R, 84  
nppiFilterColumnBorder\_16s\_C4R, 85  
nppiFilterColumnBorder\_16u\_AC4R, 85  
nppiFilterColumnBorder\_16u\_C1R, 86  
nppiFilterColumnBorder\_16u\_C3R, 87  
nppiFilterColumnBorder\_16u\_C4R, 87  
nppiFilterColumnBorder\_32f\_AC4R, 88  
nppiFilterColumnBorder\_32f\_C1R, 88  
nppiFilterColumnBorder\_32f\_C3R, 89  
nppiFilterColumnBorder\_32f\_C4R, 89  
nppiFilterColumnBorder\_8u\_AC4R, 90  
nppiFilterColumnBorder\_8u\_C1R, 91  
nppiFilterColumnBorder\_8u\_C3R, 91  
nppiFilterColumnBorder\_8u\_C4R, 92  
nppiFilterRow32f\_16s\_AC4R, 92  
nppiFilterRow32f\_16s\_C1R, 93  
nppiFilterRow32f\_16s\_C3R, 93  
nppiFilterRow32f\_16s\_C4R, 94  
nppiFilterRow32f\_16u\_AC4R, 94  
nppiFilterRow32f\_16u\_C1R, 95  
nppiFilterRow32f\_16u\_C3R, 95  
nppiFilterRow32f\_16u\_C4R, 96  
nppiFilterRow32f\_8u\_AC4R, 96  
nppiFilterRow32f\_8u\_C1R, 97  
nppiFilterRow32f\_8u\_C3R, 97  
nppiFilterRow32f\_8u\_C4R, 98  
nppiFilterRow\_16s\_AC4R, 98  
nppiFilterRow\_16s\_C1R, 99  
nppiFilterRow\_16s\_C3R, 99  
nppiFilterRow\_16s\_C4R, 100  
nppiFilterRow\_16u\_AC4R, 100  
nppiFilterRow\_16u\_C1R, 101  
nppiFilterRow\_16u\_C3R, 101  
nppiFilterRow\_16u\_C4R, 102  
nppiFilterRow\_32f\_AC4R, 102  
nppiFilterRow\_32f\_C1R, 103  
nppiFilterRow\_32f\_C3R, 103  
nppiFilterRow\_32f\_C4R, 104  
nppiFilterRow\_64f\_C1R, 104  
nppiFilterRow\_8u\_AC4R, 105  
nppiFilterRow\_8u\_C1R, 105  
nppiFilterRow\_8u\_C3R, 106  
nppiFilterRow\_8u\_C4R, 106  
nppiFilterRowBorder32f\_16s\_AC4R, 107  
nppiFilterRowBorder32f\_16s\_C1R, 107  
nppiFilterRowBorder32f\_16s\_C3R, 108  
nppiFilterRowBorder32f\_16s\_C4R, 108  
nppiFilterRowBorder32f\_16u\_AC4R, 109  
nppiFilterRowBorder32f\_16u\_C1R, 109  
nppiFilterRowBorder32f\_16u\_C3R, 110  
nppiFilterRowBorder32f\_16u\_C4R, 110  
nppiFilterRowBorder32f\_8u\_AC4R, 111  
nppiFilterRowBorder32f\_8u\_C1R, 111  
nppiFilterRowBorder32f\_8u\_C3R, 112  
nppiFilterRowBorder32f\_8u\_C4R, 112  
nppiFilterRowBorder\_16s\_AC4R, 113  
nppiFilterRowBorder\_16s\_C1R, 114  
nppiFilterRowBorder\_16s\_C3R, 114  
nppiFilterRowBorder\_16s\_C4R, 115  
nppiFilterRowBorder\_16u\_AC4R, 115  
nppiFilterRowBorder\_16u\_C1R, 116  
nppiFilterRowBorder\_16u\_C3R, 117  
nppiFilterRowBorder\_16u\_C4R, 117  
nppiFilterRowBorder\_32f\_AC4R, 118  
nppiFilterRowBorder\_32f\_C1R, 118  
nppiFilterRowBorder\_32f\_C3R, 119  
nppiFilterRowBorder\_32f\_C4R, 119  
nppiFilterRowBorder\_8u\_AC4R, 120  
nppiFilterRowBorder\_8u\_C1R, 121  
nppiFilterRowBorder\_8u\_C3R, 121  
nppiFilterRowBorder\_8u\_C4R, 122  
image\_1D\_window\_sum  
    nppiSumWindowColumn\_16s32f\_C1R, 124  
    nppiSumWindowColumn\_16s32f\_C3R, 125  
    nppiSumWindowColumn\_16s32f\_C4R, 125  
    nppiSumWindowColumn\_16u32f\_C1R, 125  
    nppiSumWindowColumn\_16u32f\_C3R, 126  
    nppiSumWindowColumn\_16u32f\_C4R, 126  
    nppiSumWindowColumn\_8u32f\_C1R, 127  
    nppiSumWindowColumn\_8u32f\_C3R, 127  
    nppiSumWindowColumn\_8u32f\_C4R, 128  
    nppiSumWindowRow\_16s32f\_C1R, 128  
    nppiSumWindowRow\_16s32f\_C3R, 129  
    nppiSumWindowRow\_16s32f\_C4R, 129  
    nppiSumWindowRow\_16u32f\_C1R, 130  
    nppiSumWindowRow\_16u32f\_C3R, 130  
    nppiSumWindowRow\_16u32f\_C4R, 131  
    nppiSumWindowRow\_8u32f\_C1R, 131  
    nppiSumWindowRow\_8u32f\_C3R, 132  
    nppiSumWindowRow\_8u32f\_C4R, 132  
image\_1D\_window\_sum\_border  
    nppiSumWindowColumnBorder\_16s32f\_-  
        C1R, 136  
    nppiSumWindowColumnBorder\_16s32f\_-  
        C3R, 136  
    nppiSumWindowColumnBorder\_16s32f\_-  
        C4R, 137  
    nppiSumWindowColumnBorder\_16u32f\_-  
        C1R, 137  
    nppiSumWindowColumnBorder\_16u32f\_-  
        C3R, 138  
    nppiSumWindowColumnBorder\_16u32f\_-  
        C4R, 138  
    nppiSumWindowColumnBorder\_8u32f\_C1R,  
        139  
    nppiSumWindowColumnBorder\_8u32f\_C3R,  
        140

- nppiSumWindowColumnBorder\_8u32f\_C4R,  
   140  
 nppiSumWindowRowBorder\_16s32f\_C1R,  
   141  
 nppiSumWindowRowBorder\_16s32f\_C3R,  
   141  
 nppiSumWindowRowBorder\_16s32f\_C4R,  
   142  
 nppiSumWindowRowBorder\_16u32f\_C1R,  
   142  
 nppiSumWindowRowBorder\_16u32f\_C3R,  
   143  
 nppiSumWindowRowBorder\_16u32f\_C4R,  
   144  
 nppiSumWindowRowBorder\_8u32f\_C1R,  
   144  
 nppiSumWindowRowBorder\_8u32f\_C3R,  
   145  
 nppiSumWindowRowBorder\_8u32f\_C4R,  
   145
- image\_2D\_fixed\_linear\_filters
- nppiFilterBox\_16s\_AC4R, 209
  - nppiFilterBox\_16s\_C1R, 209
  - nppiFilterBox\_16s\_C3R, 209
  - nppiFilterBox\_16s\_C4R, 210
  - nppiFilterBox\_16u\_AC4R, 210
  - nppiFilterBox\_16u\_C1R, 211
  - nppiFilterBox\_16u\_C3R, 211
  - nppiFilterBox\_16u\_C4R, 211
  - nppiFilterBox\_32f\_AC4R, 212
  - nppiFilterBox\_32f\_C1R, 212
  - nppiFilterBox\_32f\_C3R, 213
  - nppiFilterBox\_32f\_C4R, 213
  - nppiFilterBox\_64f\_C1R, 213
  - nppiFilterBox\_8u\_AC4R, 214
  - nppiFilterBox\_8u\_C1R, 214
  - nppiFilterBox\_8u\_C3R, 215
  - nppiFilterBox\_8u\_C4R, 215
  - nppiFilterBoxBorder\_16s\_AC4R, 215
  - nppiFilterBoxBorder\_16s\_C1R, 216
  - nppiFilterBoxBorder\_16s\_C3R, 216
  - nppiFilterBoxBorder\_16s\_C4R, 217
  - nppiFilterBoxBorder\_16u\_AC4R, 217
  - nppiFilterBoxBorder\_16u\_C1R, 218
  - nppiFilterBoxBorder\_16u\_C3R, 218
  - nppiFilterBoxBorder\_16u\_C4R, 219
  - nppiFilterBoxBorder\_32f\_AC4R, 219
  - nppiFilterBoxBorder\_32f\_C1R, 220
  - nppiFilterBoxBorder\_32f\_C3R, 220
  - nppiFilterBoxBorder\_32f\_C4R, 221
  - nppiFilterBoxBorder\_8u\_AC4R, 221
  - nppiFilterBoxBorder\_8u\_C1R, 222
  - nppiFilterBoxBorder\_8u\_C3R, 222
  - nppiFilterBoxBorder\_8u\_C4R, 223
- image\_convolution
- nppiFilter32f\_16s\_AC4R, 156
  - nppiFilter32f\_16s\_C1R, 157
  - nppiFilter32f\_16s\_C3R, 157
  - nppiFilter32f\_16s\_C4R, 157
  - nppiFilter32f\_16u\_AC4R, 158
  - nppiFilter32f\_16u\_C1R, 158
  - nppiFilter32f\_16u\_C3R, 159
  - nppiFilter32f\_16u\_C4R, 159
  - nppiFilter32f\_32s\_AC4R, 160
  - nppiFilter32f\_32s\_C1R, 160
  - nppiFilter32f\_32s\_C3R, 161
  - nppiFilter32f\_32s\_C4R, 161
  - nppiFilter32f\_8s16s\_AC4R, 162
  - nppiFilter32f\_8s16s\_C1R, 162
  - nppiFilter32f\_8s16s\_C3R, 163
  - nppiFilter32f\_8s16s\_C4R, 163
  - nppiFilter32f\_8s\_AC4R, 164
  - nppiFilter32f\_8s\_C1R, 164
  - nppiFilter32f\_8s\_C2R, 165
  - nppiFilter32f\_8s\_C3R, 165
  - nppiFilter32f\_8s\_C4R, 166
  - nppiFilter32f\_8u16s\_AC4R, 166
  - nppiFilter32f\_8u16s\_C1R, 167
  - nppiFilter32f\_8u16s\_C3R, 167
  - nppiFilter32f\_8u16s\_C4R, 168
  - nppiFilter32f\_8u\_AC4R, 168
  - nppiFilter32f\_8u\_C1R, 169
  - nppiFilter32f\_8u\_C2R, 169
  - nppiFilter32f\_8u\_C3R, 170
  - nppiFilter32f\_8u\_C4R, 170
  - nppiFilter\_16s\_AC4R, 171
  - nppiFilter\_16s\_C1R, 171
  - nppiFilter\_16s\_C3R, 172
  - nppiFilter\_16s\_C4R, 172
  - nppiFilter\_16u\_AC4R, 173
  - nppiFilter\_16u\_C1R, 173
  - nppiFilter\_16u\_C3R, 174
  - nppiFilter\_16u\_C4R, 174
  - nppiFilter\_32f\_AC4R, 175
  - nppiFilter\_32f\_C1R, 175
  - nppiFilter\_32f\_C2R, 176
  - nppiFilter\_32f\_C3R, 176
  - nppiFilter\_32f\_C4R, 177
  - nppiFilter\_64f\_C1R, 177
  - nppiFilter\_8u\_AC4R, 178
  - nppiFilter\_8u\_C1R, 178
  - nppiFilter\_8u\_C3R, 179
  - nppiFilter\_8u\_C4R, 179
  - nppiFilterBorder32f\_16s\_AC4R, 180
  - nppiFilterBorder32f\_16s\_C1R, 180
  - nppiFilterBorder32f\_16s\_C3R, 181
  - nppiFilterBorder32f\_16s\_C4R, 181
  - nppiFilterBorder32f\_16u\_AC4R, 182

- nppiFilterBorder32f\_16u\_C1R, 182  
nppiFilterBorder32f\_16u\_C3R, 183  
nppiFilterBorder32f\_16u\_C4R, 183  
nppiFilterBorder32f\_32s\_AC4R, 184  
nppiFilterBorder32f\_32s\_C1R, 184  
nppiFilterBorder32f\_32s\_C3R, 185  
nppiFilterBorder32f\_32s\_C4R, 185  
nppiFilterBorder32f\_8s16s\_AC4R, 186  
nppiFilterBorder32f\_8s16s\_C1R, 186  
nppiFilterBorder32f\_8s16s\_C3R, 187  
nppiFilterBorder32f\_8s16s\_C4R, 187  
nppiFilterBorder32f\_8s\_AC4R, 188  
nppiFilterBorder32f\_8s\_C1R, 188  
nppiFilterBorder32f\_8s\_C2R, 189  
nppiFilterBorder32f\_8s\_C3R, 189  
nppiFilterBorder32f\_8s\_C4R, 190  
nppiFilterBorder32f\_8u16s\_AC4R, 190  
nppiFilterBorder32f\_8u16s\_C1R, 191  
nppiFilterBorder32f\_8u16s\_C3R, 191  
nppiFilterBorder32f\_8u16s\_C4R, 192  
nppiFilterBorder32f\_8u\_AC4R, 192  
nppiFilterBorder32f\_8u\_C1R, 193  
nppiFilterBorder32f\_8u\_C2R, 193  
nppiFilterBorder32f\_8u\_C3R, 194  
nppiFilterBorder32f\_8u\_C4R, 194  
nppiFilterBorder\_16s\_AC4R, 195  
nppiFilterBorder\_16s\_C1R, 196  
nppiFilterBorder\_16s\_C3R, 196  
nppiFilterBorder\_16s\_C4R, 197  
nppiFilterBorder\_16u\_AC4R, 197  
nppiFilterBorder\_16u\_C1R, 198  
nppiFilterBorder\_16u\_C3R, 199  
nppiFilterBorder\_16u\_C4R, 199  
nppiFilterBorder\_32f\_AC4R, 200  
nppiFilterBorder\_32f\_C1R, 200  
nppiFilterBorder\_32f\_C2R, 201  
nppiFilterBorder\_32f\_C3R, 201  
nppiFilterBorder\_32f\_C4R, 202  
nppiFilterBorder\_8u\_AC4R, 202  
nppiFilterBorder\_8u\_C1R, 203  
nppiFilterBorder\_8u\_C3R, 204  
nppiFilterBorder\_8u\_C4R, 204
- image\_rank\_filters
- nppiFilterMax\_16s\_AC4R, 233
  - nppiFilterMax\_16s\_C1R, 233
  - nppiFilterMax\_16s\_C3R, 233
  - nppiFilterMax\_16s\_C4R, 234
  - nppiFilterMax\_16u\_AC4R, 234
  - nppiFilterMax\_16u\_C1R, 235
  - nppiFilterMax\_16u\_C3R, 235
  - nppiFilterMax\_16u\_C4R, 235
  - nppiFilterMax\_32f\_AC4R, 236
  - nppiFilterMax\_32f\_C1R, 236
  - nppiFilterMax\_32f\_C3R, 237
- nppiFilterMax\_32f\_C4R, 237  
nppiFilterMax\_8u\_AC4R, 237  
nppiFilterMax\_8u\_C1R, 238  
nppiFilterMax\_8u\_C3R, 238  
nppiFilterMax\_8u\_C4R, 239  
nppiFilterMaxBorder\_16s\_AC4R, 239  
nppiFilterMaxBorder\_16s\_C1R, 239  
nppiFilterMaxBorder\_16s\_C3R, 240  
nppiFilterMaxBorder\_16s\_C4R, 240  
nppiFilterMaxBorder\_16u\_AC4R, 241  
nppiFilterMaxBorder\_16u\_C1R, 241  
nppiFilterMaxBorder\_16u\_C3R, 242  
nppiFilterMaxBorder\_16u\_C4R, 242  
nppiFilterMaxBorder\_32f\_AC4R, 243  
nppiFilterMaxBorder\_32f\_C1R, 243  
nppiFilterMaxBorder\_32f\_C3R, 244  
nppiFilterMaxBorder\_32f\_C4R, 244  
nppiFilterMaxBorder\_8u\_AC4R, 245  
nppiFilterMaxBorder\_8u\_C1R, 245  
nppiFilterMaxBorder\_8u\_C3R, 246  
nppiFilterMaxBorder\_8u\_C4R, 246  
nppiFilterMedian\_16s\_AC4R, 247  
nppiFilterMedian\_16s\_C1R, 247  
nppiFilterMedian\_16s\_C3R, 248  
nppiFilterMedian\_16s\_C4R, 248  
nppiFilterMedian\_16u\_AC4R, 249  
nppiFilterMedian\_16u\_C1R, 249  
nppiFilterMedian\_16u\_C3R, 250  
nppiFilterMedian\_16u\_C4R, 250  
nppiFilterMedian\_32f\_AC4R, 250  
nppiFilterMedian\_32f\_C1R, 251  
nppiFilterMedian\_32f\_C3R, 251  
nppiFilterMedian\_32f\_C4R, 252  
nppiFilterMedian\_8u\_AC4R, 252  
nppiFilterMedian\_8u\_C1R, 253  
nppiFilterMedian\_8u\_C3R, 253  
nppiFilterMedian\_8u\_C4R, 253  
nppiFilterMedianGetSize\_16s\_AC4R, 254  
nppiFilterMedianGetSize\_16s\_C1R, 254  
nppiFilterMedianGetSize\_16s\_C3R, 254  
nppiFilterMedianGetSize\_16s\_C4R, 255  
nppiFilterMedianGetSize\_16u\_AC4R, 255  
nppiFilterMedianGetSize\_16u\_C1R, 255  
nppiFilterMedianGetSize\_16u\_C3R, 256  
nppiFilterMedianGetSize\_16u\_C4R, 256

nppiFilterMedianGetSize\_32f\_AC4R,  
    256  
 nppiFilterMedianGetSize\_32f\_C1R,  
    256  
 nppiFilterMedianGetSize\_32f\_C3R,  
    257  
 nppiFilterMedianGetSize\_32f\_C4R,  
    257  
 nppiFilterMedianGetSize\_8u\_AC4R,  
    257  
 nppiFilterMedianGetSize\_8u\_C1R, 258  
 nppiFilterMedianGetSize\_8u\_C3R, 258  
 nppiFilterMedianGetSize\_8u\_C4R, 258  
 nppiFilterMin\_16s\_AC4R, 258  
 nppiFilterMin\_16s\_C1R, 259  
 nppiFilterMin\_16s\_C3R, 259  
 nppiFilterMin\_16s\_C4R, 260  
 nppiFilterMin\_16u\_AC4R, 260  
 nppiFilterMin\_16u\_C1R, 260  
 nppiFilterMin\_16u\_C3R, 261  
 nppiFilterMin\_16u\_C4R, 261  
 nppiFilterMin\_32f\_AC4R, 262  
 nppiFilterMin\_32f\_C1R, 262  
 nppiFilterMin\_32f\_C3R, 262  
 nppiFilterMin\_32f\_C4R, 263  
 nppiFilterMin\_8u\_AC4R, 263  
 nppiFilterMin\_8u\_C1R, 264  
 nppiFilterMin\_8u\_C3R, 264  
 nppiFilterMin\_8u\_C4R, 264  
 nppiFilterMinBorder\_16s\_AC4R, 265  
 nppiFilterMinBorder\_16s\_C1R, 265  
 nppiFilterMinBorder\_16s\_C3R, 266  
 nppiFilterMinBorder\_16s\_C4R, 266  
 nppiFilterMinBorder\_16u\_AC4R, 267  
 nppiFilterMinBorder\_16u\_C1R, 267  
 nppiFilterMinBorder\_16u\_C3R, 268  
 nppiFilterMinBorder\_16u\_C4R, 268  
 nppiFilterMinBorder\_32f\_AC4R, 269  
 nppiFilterMinBorder\_32f\_C1R, 269  
 nppiFilterMinBorder\_32f\_C3R, 270  
 nppiFilterMinBorder\_32f\_C4R, 270  
 nppiFilterMinBorder\_8u\_AC4R, 271  
 nppiFilterMinBorder\_8u\_C1R, 271  
 nppiFilterMinBorder\_8u\_C3R, 272  
 nppiFilterMinBorder\_8u\_C4R, 272

major  
 NppLibraryVersion, 534

minor  
 NppLibraryVersion, 534

NPP Core, 27  
 NPP Type Definitions and Constants, 31  
 Npp16s

npp\_basic\_types, 47  
 Npp16sc  
 npp\_basic\_types, 49  
 Npp16u  
 npp\_basic\_types, 47  
 Npp16uc  
 npp\_basic\_types, 49  
 Npp32f  
 npp\_basic\_types, 47  
 Npp32fc  
 npp\_basic\_types, 47  
 Npp32s  
 npp\_basic\_types, 47  
 Npp32sc  
 npp\_basic\_types, 47  
 Npp32u  
 npp\_basic\_types, 48  
 Npp32uc  
 npp\_basic\_types, 48  
 Npp64f  
 npp\_basic\_types, 48  
 Npp64fc  
 npp\_basic\_types, 48  
 Npp64s  
 npp\_basic\_types, 48  
 Npp64sc  
 npp\_basic\_types, 48  
 Npp64u  
 npp\_basic\_types, 48  
 Npp8s  
 npp\_basic\_types, 48  
 Npp8u  
 npp\_basic\_types, 48  
 Npp8uc  
 npp\_basic\_types, 49  
 NPP\_AFFINE\_QUAD\_INCORRECT\_WARNING  
 typedefs\_npp, 44  
 NPP\_ALG\_HINT\_ACCURATE  
 typedefs\_npp, 39  
 NPP\_ALG\_HINT\_FAST  
 typedefs\_npp, 39  
 NPP\_ALG\_HINT\_NONE  
 typedefs\_npp, 39  
 NPP\_ALIGNMENT\_ERROR  
 typedefs\_npp, 43  
 NPP\_ANCHOR\_ERROR  
 typedefs\_npp, 43  
 NPP\_BAD\_ARGUMENT\_ERROR  
 typedefs\_npp, 44  
 NPP\_BORDER\_CONSTANT  
 typedefs\_npp, 40  
 NPP\_BORDER\_MIRROR  
 typedefs\_npp, 40  
 NPP\_BORDER\_NONE

typedefs\_npp, 40  
NPP\_BORDER\_REPLICATE  
    typedefs\_npp, 40  
NPP\_BORDER\_UNDEFINED  
    typedefs\_npp, 40  
NPP\_BORDER\_WRAP  
    typedefs\_npp, 40  
NPP\_BOTH\_AXIS  
    typedefs\_npp, 40  
NPP\_CHANNEL\_ERROR  
    typedefs\_npp, 43  
NPP\_CHANNEL\_ORDER\_ERROR  
    typedefs\_npp, 43  
NPP\_CMP\_EQ  
    typedefs\_npp, 39  
NPP\_CMP\_GREATER  
    typedefs\_npp, 39  
NPP\_CMP\_GREATER\_EQ  
    typedefs\_npp, 39  
NPP\_CMP\_LESS  
    typedefs\_npp, 38  
NPP\_CMP\_LESS\_EQ  
    typedefs\_npp, 38  
NPP\_COEFFICIENT\_ERROR  
    typedefs\_npp, 43  
NPP\_COI\_ERROR  
    typedefs\_npp, 43  
NPP\_CONTEXT\_MATCH\_ERROR  
    typedefs\_npp, 44  
NPP\_CORRUPTED\_DATA\_ERROR  
    typedefs\_npp, 43  
NPP\_CUDA\_1\_0  
    typedefs\_npp, 39  
NPP\_CUDA\_1\_1  
    typedefs\_npp, 39  
NPP\_CUDA\_1\_2  
    typedefs\_npp, 39  
NPP\_CUDA\_1\_3  
    typedefs\_npp, 39  
NPP\_CUDA\_2\_0  
    typedefs\_npp, 39  
NPP\_CUDA\_2\_1  
    typedefs\_npp, 39  
NPP\_CUDA\_3\_0  
    typedefs\_npp, 39  
NPP\_CUDA\_3\_2  
    typedefs\_npp, 39  
NPP\_CUDA\_3\_5  
    typedefs\_npp, 39  
NPP\_CUDA\_3\_7  
    typedefs\_npp, 39  
NPP\_CUDA\_5\_0  
    typedefs\_npp, 39  
NPP\_CUDA\_5\_2  
    typedefs\_npp, 39  
NPP\_CUDA\_5\_3  
    typedefs\_npp, 39  
NPP\_CUDA\_6\_0  
    typedefs\_npp, 39  
NPP\_CUDA\_KERNEL\_EXECUTION\_ERROR  
    typedefs\_npp, 43  
NPP\_CUDA\_NOT\_CAPABLE  
    typedefs\_npp, 39  
NPP\_CUDA\_UNKNOWN\_VERSION  
    typedefs\_npp, 39  
NPP\_DATA\_TYPE\_ERROR  
    typedefs\_npp, 44  
NPP\_DIVIDE\_BY\_ZERO\_ERROR  
    typedefs\_npp, 44  
NPP\_DIVIDE\_BY\_ZERO\_WARNING  
    typedefs\_npp, 44  
NPP\_DIVISOR\_ERROR  
    typedefs\_npp, 43  
NPP\_DOUBLE\_SIZE\_WARNING  
    typedefs\_npp, 44  
NPP\_ERROR  
    typedefs\_npp, 44  
NPP\_ERROR\_RESERVED  
    typedefs\_npp, 44  
NPP\_FFT\_FLAG\_ERROR  
    typedefs\_npp, 44  
NPP\_FFT\_ORDER\_ERROR  
    typedefs\_npp, 44  
NPP\_FILTER\_SCHARR  
    typedefs\_npp, 40  
NPP\_FILTER\_SOBEL  
    typedefs\_npp, 40  
NPP\_HAAR\_CLASSIFIER\_PIXEL\_MATCH\_-  
    ERROR  
    typedefs\_npp, 43  
NPP\_HISTOGRAM\_NUMBER\_OF\_LEVELS\_-  
    ERROR  
    typedefs\_npp, 43  
NPP\_HORIZONTAL\_AXIS  
    typedefs\_npp, 40  
NPP\_INTERPOLATION\_ERROR  
    typedefs\_npp, 44  
NPP\_INVALID\_DEVICE\_POINTER\_ERROR  
    typedefs\_npp, 43  
NPP\_INVALID\_HOST\_POINTER\_ERROR  
    typedefs\_npp, 43  
NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR  
    typedefs\_npp, 43  
NPP\_LUT\_PALETTE\_BITSIZE\_ERROR  
    typedefs\_npp, 43  
NPP\_MASK\_SIZE\_11\_X\_11  
    typedefs\_npp, 41  
NPP\_MASK\_SIZE\_13\_X\_13

typedefs\_npp, 41  
**NPP\_MASK\_SIZE\_15\_X\_15**  
     typedefs\_npp, 41  
**NPP\_MASK\_SIZE\_1\_X\_3**  
     typedefs\_npp, 41  
**NPP\_MASK\_SIZE\_1\_X\_5**  
     typedefs\_npp, 41  
**NPP\_MASK\_SIZE\_3\_X\_1**  
     typedefs\_npp, 41  
**NPP\_MASK\_SIZE\_3\_X\_3**  
     typedefs\_npp, 41  
**NPP\_MASK\_SIZE\_5\_X\_1**  
     typedefs\_npp, 41  
**NPP\_MASK\_SIZE\_5\_X\_5**  
     typedefs\_npp, 41  
**NPP\_MASK\_SIZE\_7\_X\_7**  
     typedefs\_npp, 41  
**NPP\_MASK\_SIZE\_9\_X\_9**  
     typedefs\_npp, 41  
**NPP\_MASK\_SIZE\_ERROR**  
     typedefs\_npp, 43  
**NPP\_MEMCPY\_ERROR**  
     typedefs\_npp, 43  
**NPP\_MEMFREE\_ERROR**  
     typedefs\_npp, 43  
**NPP\_MEMORY\_ALLOCATION\_ERR**  
     typedefs\_npp, 44  
**NPP\_MEMSET\_ERROR**  
     typedefs\_npp, 43  
**NPP\_MIRROR\_FLIP\_ERROR**  
     typedefs\_npp, 44  
**NPP\_MISALIGNED\_DST\_ROI\_WARNING**  
     typedefs\_npp, 44  
**NPP\_MOMENT\_00\_ZERO\_ERROR**  
     typedefs\_npp, 44  
**NPP\_NO\_ERROR**  
     typedefs\_npp, 44  
**NPP\_NO\_MEMORY\_ERROR**  
     typedefs\_npp, 44  
**NPP\_NO\_OPERATION\_WARNING**  
     typedefs\_npp, 44  
**NPP\_NOT EVEN STEP\_ERROR**  
     typedefs\_npp, 43  
**NPP\_NOT\_IMPLEMENTED\_ERROR**  
     typedefs\_npp, 44  
**NPP\_NOT\_SUFFICIENT\_COMPUTE\_-  
    CAPABILITY**  
     typedefs\_npp, 43  
**NPP\_NOT\_SUPPORTED\_MODE\_ERROR**  
     typedefs\_npp, 43  
**NPP\_NULL\_POINTER\_ERROR**  
     typedefs\_npp, 44  
**NPP\_NUMBER\_OF\_CHANNELS\_ERROR**  
     typedefs\_npp, 43

NPP\_OUT\_OF\_RANGE\_ERROR  
     typedefs\_npp, 44  
**NPP\_OVERFLOW\_ERROR**  
     typedefs\_npp, 43  
**NPP\_QUADRANGLE\_ERROR**  
     typedefs\_npp, 43  
**NPP\_QUALITY\_INDEX\_ERROR**  
     typedefs\_npp, 43  
**NPP\_RANGE\_ERROR**  
     typedefs\_npp, 44  
**NPP\_RECTANGLE\_ERROR**  
     typedefs\_npp, 43  
**NPP\_RESIZE\_FACTOR\_ERROR**  
     typedefs\_npp, 44  
**NPP\_RESIZE\_NO\_OPERATION\_ERROR**  
     typedefs\_npp, 43  
**NPP\_RND\_FINANCIAL**  
     typedefs\_npp, 42  
**NPP\_RND\_NEAR**  
     typedefs\_npp, 42  
**NPP\_RND\_ZERO**  
     typedefs\_npp, 42  
**NPP\_ROUND\_MODE\_NOT\_SUPPORTED\_-  
    ERROR**  
     typedefs\_npp, 43  
**NPP\_ROUND\_NEAREST\_TIES\_AWAY\_-  
    FROM\_ZERO**  
     typedefs\_npp, 42  
**NPP\_ROUND\_NEAREST\_TIES\_TO\_EVEN**  
     typedefs\_npp, 42  
**NPP\_ROUND\_TOWARD\_ZERO**  
     typedefs\_npp, 42  
**NPP\_SCALE\_RANGE\_ERROR**  
     typedefs\_npp, 44  
**NPP\_SIZE\_ERROR**  
     typedefs\_npp, 44  
**NPP\_STEP\_ERROR**  
     typedefs\_npp, 44  
**NPP\_STRIDE\_ERROR**  
     typedefs\_npp, 43  
**NPP\_SUCCESS**  
     typedefs\_npp, 44  
**NPP\_TEXTURE\_BIND\_ERROR**  
     typedefs\_npp, 43  
**NPP\_THRESHOLD\_ERROR**  
     typedefs\_npp, 44  
**NPP\_THRESHOLD\_NEGATIVE\_LEVEL\_-  
    ERROR**  
     typedefs\_npp, 44  
**NPP\_VERTICAL\_AXIS**  
     typedefs\_npp, 40  
**NPP\_WRONG\_INTERSECTION\_QUAD\_-  
    WARNING**  
     typedefs\_npp, 44

NPP\_WRONG\_INTERSECTION\_ROI\_ERROR  
    typedefs\_npp, 43  
NPP\_WRONG\_INTERSECTION\_ROI\_-  
    WARNING  
    typedefs\_npp, 44  
NPP\_ZC\_MODE\_NOT\_SUPPORTED\_ERROR  
    typedefs\_npp, 43  
NPP\_ZERO\_MASK\_VALUE\_ERROR  
    typedefs\_npp, 43  
NPP\_ALIGN\_16, 525  
    im, 525  
    re, 526  
NPP\_ALIGN\_8, 527  
    im, 527  
    re, 527, 528  
npp\_basic\_types  
    \_\_align\_\_, 48, 49  
    Npp16s, 47  
    Npp16sc, 49  
    Npp16u, 47  
    Npp16uc, 49  
    Npp32f, 47  
    Npp32fc, 47  
    Npp32s, 47  
    Npp32sc, 47  
    Npp32u, 48  
    Npp32uc, 48  
    Npp64f, 48  
    Npp64fc, 48  
    Npp64s, 48  
    Npp64sc, 48  
    Npp64u, 48  
    Npp8s, 48  
    Npp8u, 48  
    Npp8uc, 49  
NPP\_MAX\_16S  
    typedefs\_npp, 37  
NPP\_MAX\_16U  
    typedefs\_npp, 37  
NPP\_MAX\_32S  
    typedefs\_npp, 37  
NPP\_MAX\_32U  
    typedefs\_npp, 37  
NPP\_MAX\_64S  
    typedefs\_npp, 37  
NPP\_MAX\_64U  
    typedefs\_npp, 37  
NPP\_MAX\_8S  
    typedefs\_npp, 37  
NPP\_MAX\_8U  
    typedefs\_npp, 37  
NPP\_MAXABS\_32F  
    typedefs\_npp, 37  
NPP\_MAXABS\_64F  
    typedefs\_npp, 37  
NPP\_MIN\_16S  
    typedefs\_npp, 37  
NPP\_MIN\_16U  
    typedefs\_npp, 38  
NPP\_MIN\_32S  
    typedefs\_npp, 38  
NPP\_MIN\_32U  
    typedefs\_npp, 38  
NPP\_MIN\_64S  
    typedefs\_npp, 38  
NPP\_MIN\_64U  
    typedefs\_npp, 38  
NPP\_MIN\_8S  
    typedefs\_npp, 38  
NPP\_MIN\_8U  
    typedefs\_npp, 38  
NPP\_MINABS\_32F  
    typedefs\_npp, 38  
NPP\_MINABS\_64F  
    typedefs\_npp, 38  
NppCmpOp  
    typedefs\_npp, 38  
nppGetGpuComputeCapability  
    core\_npp, 28  
nppGetGpuDeviceProperties  
    core\_npp, 28  
nppGetGpuName  
    core\_npp, 28  
nppGetGpuNumSMs  
    core\_npp, 28  
nppGetLibVersion  
    core\_npp, 28  
nppGetMaxThreadsPerBlock  
    core\_npp, 29  
nppGetMaxThreadsPerSM  
    core\_npp, 29  
nppGetStream  
    core\_npp, 29  
nppGetStreamMaxThreadsPerSM  
    core\_npp, 29  
nppGetStreamNumSMs  
    core\_npp, 29  
NppGpuComputeCapability  
    typedefs\_npp, 39  
NppHintAlgorithm  
    typedefs\_npp, 39  
NPPI\_BAYER\_BGGR  
    typedefs\_npp, 40  
NPPI\_BAYER\_GBRG  
    typedefs\_npp, 40  
NPPI\_BAYER\_GRBG  
    typedefs\_npp, 40  
NPPI\_BAYER\_RGGB

typedefs\_npp, 40  
**NPPI\_INTER\_CUBIC**  
 typedefs\_npp, 41  
**NPPI\_INTER\_CUBIC2P\_B05C03**  
 typedefs\_npp, 41  
**NPPI\_INTER\_CUBIC2P\_BSPLINE**  
 typedefs\_npp, 41  
**NPPI\_INTER\_CUBIC2P\_CATMULLROM**  
 typedefs\_npp, 41  
**NPPI\_INTER\_LANCZOS**  
 typedefs\_npp, 41  
**NPPI\_INTER\_LANCZOS3\_ADVANCED**  
 typedefs\_npp, 41  
**NPPI\_INTER\_LINEAR**  
 typedefs\_npp, 41  
**NPPI\_INTER\_NN**  
 typedefs\_npp, 41  
**NPPI\_INTER\_SUPER**  
 typedefs\_npp, 41  
**NPPI\_INTER\_UNDEFINED**  
 typedefs\_npp, 41  
**NPPI\_OP\_ALPHA\_ATOP**  
 typedefs\_npp, 39  
**NPPI\_OP\_ALPHA\_ATOP\_PREMUL**  
 typedefs\_npp, 40  
**NPPI\_OP\_ALPHA\_IN**  
 typedefs\_npp, 39  
**NPPI\_OP\_ALPHA\_IN\_PREMUL**  
 typedefs\_npp, 40  
**NPPI\_OP\_ALPHA\_OUT**  
 typedefs\_npp, 39  
**NPPI\_OP\_ALPHA\_OUT\_PREMUL**  
 typedefs\_npp, 40  
**NPPI\_OP\_ALPHA\_OVER**  
 typedefs\_npp, 39  
**NPPI\_OP\_ALPHA\_OVER\_PREMUL**  
 typedefs\_npp, 40  
**NPPI\_OP\_ALPHA\_PLUS**  
 typedefs\_npp, 39  
**NPPI\_OP\_ALPHA\_PLUS\_PREMUL**  
 typedefs\_npp, 40  
**NPPI\_OP\_ALPHA\_PREMUL**  
 typedefs\_npp, 40  
**NPPI\_OP\_ALPHA\_XOR**  
 typedefs\_npp, 39  
**NPPI\_OP\_ALPHA\_XOR\_PREMUL**  
 typedefs\_npp, 40  
**NPPI\_SMOOTH\_EDGE**  
 typedefs\_npp, 41  
**nppiACTable**  
 typedefs\_npp, 41  
**NppiAlphaOp**  
 typedefs\_npp, 39  
**NppiAxis**  
 typedefs\_npp, 40  
**NppiBayerGridPosition**  
 typedefs\_npp, 40  
**NppiBorderType**  
 typedefs\_npp, 40  
**nppiDCTable**  
 typedefs\_npp, 41  
**NppiDifferentialKernel**  
 typedefs\_npp, 40  
**nppiFilter32f\_16s\_AC4R**  
 image\_convolution, 156  
**nppiFilter32f\_16s\_C1R**  
 image\_convolution, 157  
**nppiFilter32f\_16s\_C3R**  
 image\_convolution, 157  
**nppiFilter32f\_16s\_C4R**  
 image\_convolution, 157  
**nppiFilter32f\_16u\_AC4R**  
 image\_convolution, 158  
**nppiFilter32f\_16u\_C1R**  
 image\_convolution, 158  
**nppiFilter32f\_16u\_C3R**  
 image\_convolution, 159  
**nppiFilter32f\_16u\_C4R**  
 image\_convolution, 159  
**nppiFilter32f\_32s\_AC4R**  
 image\_convolution, 160  
**nppiFilter32f\_32s\_C1R**  
 image\_convolution, 160  
**nppiFilter32f\_32s\_C3R**  
 image\_convolution, 161  
**nppiFilter32f\_32s\_C4R**  
 image\_convolution, 161  
**nppiFilter32f\_8s16s\_AC4R**  
 image\_convolution, 162  
**nppiFilter32f\_8s16s\_C1R**  
 image\_convolution, 162  
**nppiFilter32f\_8s16s\_C3R**  
 image\_convolution, 163  
**nppiFilter32f\_8s16s\_C4R**  
 image\_convolution, 163  
**nppiFilter32f\_8s\_AC4R**  
 image\_convolution, 164  
**nppiFilter32f\_8s\_C1R**  
 image\_convolution, 164  
**nppiFilter32f\_8s\_C2R**  
 image\_convolution, 165  
**nppiFilter32f\_8s\_C3R**  
 image\_convolution, 165  
**nppiFilter32f\_8s\_C4R**  
 image\_convolution, 166  
**nppiFilter32f\_8u16s\_AC4R**  
 image\_convolution, 166  
**nppiFilter32f\_8u16s\_C1R**

- nppiFilter32f\_8u16s\_C3R
  - image\_convolution, 167
- nppiFilter32f\_8u16s\_C4R
  - image\_convolution, 167
- nppiFilter32f\_8u16s\_C4R
  - image\_convolution, 168
- nppiFilter32f\_8u\_AC4R
  - image\_convolution, 168
- nppiFilter32f\_8u\_C1R
  - image\_convolution, 169
- nppiFilter32f\_8u\_C2R
  - image\_convolution, 169
- nppiFilter32f\_8u\_C3R
  - image\_convolution, 170
- nppiFilter32f\_8u\_C4R
  - image\_convolution, 170
- nppiFilter\_16s\_AC4R
  - image\_convolution, 171
- nppiFilter\_16s\_C1R
  - image\_convolution, 171
- nppiFilter\_16s\_C3R
  - image\_convolution, 172
- nppiFilter\_16s\_C4R
  - image\_convolution, 172
- nppiFilter\_16u\_AC4R
  - image\_convolution, 173
- nppiFilter\_16u\_C1R
  - image\_convolution, 173
- nppiFilter\_16u\_C3R
  - image\_convolution, 174
- nppiFilter\_16u\_C4R
  - image\_convolution, 174
- nppiFilter\_32f\_AC4R
  - image\_convolution, 175
- nppiFilter\_32f\_C1R
  - image\_convolution, 175
- nppiFilter\_32f\_C2R
  - image\_convolution, 176
- nppiFilter\_32f\_C3R
  - image\_convolution, 176
- nppiFilter\_32f\_C4R
  - image\_convolution, 177
- nppiFilter\_64f\_C1R
  - image\_convolution, 177
- nppiFilter\_8u\_AC4R
  - image\_convolution, 178
- nppiFilter\_8u\_C1R
  - image\_convolution, 178
- nppiFilter\_8u\_C3R
  - image\_convolution, 179
- nppiFilter\_8u\_C4R
  - image\_convolution, 179
- nppiFilterBilateralGaussBorder\_16u\_C1R
  - fixed\_filters, 329
- nppiFilterBilateralGaussBorder\_16u\_C3R
  - fixed\_filters, 330
- nppiFilterBilateralGaussBorder\_32f\_C1R
  - fixed\_filters, 330
- nppiFilterBilateralGaussBorder\_32f\_C3R
  - fixed\_filters, 331
- nppiFilterBilateralGaussBorder\_8u\_C1R
  - fixed\_filters, 331
- nppiFilterBilateralGaussBorder\_8u\_C3R
  - fixed\_filters, 332
- nppiFilterBorder32f\_16s\_AC4R
  - image\_convolution, 180
- nppiFilterBorder32f\_16s\_C1R
  - image\_convolution, 180
- nppiFilterBorder32f\_16s\_C3R
  - image\_convolution, 181
- nppiFilterBorder32f\_16s\_C4R
  - image\_convolution, 181
- nppiFilterBorder32f\_16u\_AC4R
  - image\_convolution, 182
- nppiFilterBorder32f\_16u\_C1R
  - image\_convolution, 182
- nppiFilterBorder32f\_16u\_C3R
  - image\_convolution, 183
- nppiFilterBorder32f\_16u\_C4R
  - image\_convolution, 183
- nppiFilterBorder32f\_32s\_AC4R
  - image\_convolution, 184
- nppiFilterBorder32f\_32s\_C1R
  - image\_convolution, 184
- nppiFilterBorder32f\_32s\_C3R
  - image\_convolution, 185
- nppiFilterBorder32f\_32s\_C4R
  - image\_convolution, 185
- nppiFilterBorder32f\_8s16s\_AC4R
  - image\_convolution, 186
- nppiFilterBorder32f\_8s16s\_C1R
  - image\_convolution, 186
- nppiFilterBorder32f\_8s16s\_C3R
  - image\_convolution, 187
- nppiFilterBorder32f\_8s16s\_C4R
  - image\_convolution, 187
- nppiFilterBorder32f\_8s\_AC4R
  - image\_convolution, 188
- nppiFilterBorder32f\_8s\_C1R
  - image\_convolution, 188
- nppiFilterBorder32f\_8s\_C2R
  - image\_convolution, 189
- nppiFilterBorder32f\_8s\_C3R
  - image\_convolution, 189
- nppiFilterBorder32f\_8s\_C4R
  - image\_convolution, 190
- nppiFilterBorder32f\_8u16s\_AC4R
  - image\_convolution, 190
- nppiFilterBorder32f\_8u16s\_C1R
  - image\_convolution, 190

- nppiFilterBorder32f\_8u16s\_C3R
  - image\_convolution, 191
- nppiFilterBorder32f\_8u16s\_C4R
  - image\_convolution, 192
- nppiFilterBorder32f\_8u\_AC4R
  - image\_convolution, 192
- nppiFilterBorder32f\_8u\_C1R
  - image\_convolution, 193
- nppiFilterBorder32f\_8u\_C2R
  - image\_convolution, 193
- nppiFilterBorder32f\_8u\_C3R
  - image\_convolution, 194
- nppiFilterBorder32f\_8u\_C4R
  - image\_convolution, 194
- nppiFilterBorder\_16s\_AC4R
  - image\_convolution, 195
- nppiFilterBorder\_16s\_C1R
  - image\_convolution, 196
- nppiFilterBorder\_16s\_C3R
  - image\_convolution, 196
- nppiFilterBorder\_16s\_C4R
  - image\_convolution, 197
- nppiFilterBorder\_16u\_AC4R
  - image\_convolution, 197
- nppiFilterBorder\_16u\_C1R
  - image\_convolution, 198
- nppiFilterBorder\_16u\_C3R
  - image\_convolution, 199
- nppiFilterBorder\_16u\_C4R
  - image\_convolution, 199
- nppiFilterBorder\_32f\_AC4R
  - image\_convolution, 200
- nppiFilterBorder\_32f\_C1R
  - image\_convolution, 200
- nppiFilterBorder\_32f\_C2R
  - image\_convolution, 201
- nppiFilterBorder\_32f\_C3R
  - image\_convolution, 201
- nppiFilterBorder\_32f\_C4R
  - image\_convolution, 202
- nppiFilterBorder\_8u\_AC4R
  - image\_convolution, 202
- nppiFilterBorder\_8u\_C1R
  - image\_convolution, 203
- nppiFilterBorder\_8u\_C3R
  - image\_convolution, 204
- nppiFilterBorder\_8u\_C4R
  - image\_convolution, 204
- nppiFilterBox\_16s\_AC4R
  - image\_2D\_fixed\_linear\_filters, 209
- nppiFilterBox\_16s\_C1R
  - image\_2D\_fixed\_linear\_filters, 209
- nppiFilterBox\_16s\_C3R
  - image\_2D\_fixed\_linear\_filters, 209
- nppiFilterBox\_16s\_C4R
  - image\_2D\_fixed\_linear\_filters, 209
- nppiFilterBox\_16u\_AC4R
  - image\_2D\_fixed\_linear\_filters, 210
- nppiFilterBox\_16u\_C1R
  - image\_2D\_fixed\_linear\_filters, 210
- nppiFilterBox\_16u\_C3R
  - image\_2D\_fixed\_linear\_filters, 211
- nppiFilterBox\_16u\_C4R
  - image\_2D\_fixed\_linear\_filters, 211
- nppiFilterBox\_32f\_AC4R
  - image\_2D\_fixed\_linear\_filters, 212
- nppiFilterBox\_32f\_C1R
  - image\_2D\_fixed\_linear\_filters, 212
- nppiFilterBox\_32f\_C3R
  - image\_2D\_fixed\_linear\_filters, 213
- nppiFilterBox\_32f\_C4R
  - image\_2D\_fixed\_linear\_filters, 213
- nppiFilterBox\_64f\_C1R
  - image\_2D\_fixed\_linear\_filters, 213
- nppiFilterBox\_8u\_AC4R
  - image\_2D\_fixed\_linear\_filters, 214
- nppiFilterBox\_8u\_C1R
  - image\_2D\_fixed\_linear\_filters, 214
- nppiFilterBox\_8u\_C3R
  - image\_2D\_fixed\_linear\_filters, 215
- nppiFilterBox\_8u\_C4R
  - image\_2D\_fixed\_linear\_filters, 215
- nppiFilterBoxBorder\_16s\_AC4R
  - image\_2D\_fixed\_linear\_filters, 215
- nppiFilterBoxBorder\_16s\_C1R
  - image\_2D\_fixed\_linear\_filters, 216
- nppiFilterBoxBorder\_16s\_C3R
  - image\_2D\_fixed\_linear\_filters, 216
- nppiFilterBoxBorder\_16s\_C4R
  - image\_2D\_fixed\_linear\_filters, 217
- nppiFilterBoxBorder\_16u\_AC4R
  - image\_2D\_fixed\_linear\_filters, 217
- nppiFilterBoxBorder\_16u\_C1R
  - image\_2D\_fixed\_linear\_filters, 218
- nppiFilterBoxBorder\_16u\_C3R
  - image\_2D\_fixed\_linear\_filters, 218
- nppiFilterBoxBorder\_16u\_C4R
  - image\_2D\_fixed\_linear\_filters, 219
- nppiFilterBoxBorder\_32f\_AC4R
  - image\_2D\_fixed\_linear\_filters, 219
- nppiFilterBoxBorder\_32f\_C1R
  - image\_2D\_fixed\_linear\_filters, 220
- nppiFilterBoxBorder\_32f\_C3R
  - image\_2D\_fixed\_linear\_filters, 220
- nppiFilterBoxBorder\_32f\_C4R
  - image\_2D\_fixed\_linear\_filters, 221
- nppiFilterBoxBorder\_8u\_AC4R

- image\_2D\_fixed\_linear\_filters, 221
- nppiFilterBoxBorder\_8u\_C1R
  - image\_2D\_fixed\_linear\_filters, 222
- nppiFilterBoxBorder\_8u\_C3R
  - image\_2D\_fixed\_linear\_filters, 222
- nppiFilterBoxBorder\_8u\_C4R
  - image\_2D\_fixed\_linear\_filters, 223
- nppiFilterCannyBorder\_8u\_C1R
  - fixed\_filters, 333
- nppiFilterCannyBorderGetBufferSize
  - fixed\_filters, 333
- nppiFilterColumn32f\_16s\_AC4R
  - image\_1D\_linear\_filter, 63
- nppiFilterColumn32f\_16s\_C1R
  - image\_1D\_linear\_filter, 63
- nppiFilterColumn32f\_16s\_C3R
  - image\_1D\_linear\_filter, 64
- nppiFilterColumn32f\_16s\_C4R
  - image\_1D\_linear\_filter, 64
- nppiFilterColumn32f\_16u\_AC4R
  - image\_1D\_linear\_filter, 65
- nppiFilterColumn32f\_16u\_C1R
  - image\_1D\_linear\_filter, 65
- nppiFilterColumn32f\_16u\_C3R
  - image\_1D\_linear\_filter, 66
- nppiFilterColumn32f\_16u\_C4R
  - image\_1D\_linear\_filter, 66
- nppiFilterColumn32f\_8u\_AC4R
  - image\_1D\_linear\_filter, 66
- nppiFilterColumn32f\_8u\_C1R
  - image\_1D\_linear\_filter, 67
- nppiFilterColumn32f\_8u\_C3R
  - image\_1D\_linear\_filter, 67
- nppiFilterColumn32f\_8u\_C4R
  - image\_1D\_linear\_filter, 68
- nppiFilterColumn16s\_AC4R
  - image\_1D\_linear\_filter, 68
- nppiFilterColumn16s\_C1R
  - image\_1D\_linear\_filter, 69
- nppiFilterColumn16s\_C3R
  - image\_1D\_linear\_filter, 69
- nppiFilterColumn16s\_C4R
  - image\_1D\_linear\_filter, 70
- nppiFilterColumn16u\_AC4R
  - image\_1D\_linear\_filter, 70
- nppiFilterColumn16u\_C1R
  - image\_1D\_linear\_filter, 71
- nppiFilterColumn16u\_C3R
  - image\_1D\_linear\_filter, 71
- nppiFilterColumn16u\_C4R
  - image\_1D\_linear\_filter, 72
- nppiFilterColumn\_32f\_AC4R
  - image\_1D\_linear\_filter, 72
- nppiFilterColumn\_32f\_C1R
  - image\_1D\_linear\_filter, 73
- nppiFilterColumn\_32f\_C3R
  - image\_1D\_linear\_filter, 73
- nppiFilterColumn\_32f\_C4R
  - image\_1D\_linear\_filter, 74
- nppiFilterColumn\_64f\_C1R
  - image\_1D\_linear\_filter, 74
- nppiFilterColumn\_8u\_AC4R
  - image\_1D\_linear\_filter, 75
- nppiFilterColumn\_8u\_C1R
  - image\_1D\_linear\_filter, 75
- nppiFilterColumn\_8u\_C3R
  - image\_1D\_linear\_filter, 76
- nppiFilterColumn\_8u\_C4R
  - image\_1D\_linear\_filter, 76
- nppiFilterColumnBorder32f\_16s\_AC4R
  - image\_1D\_linear\_filter, 77
- nppiFilterColumnBorder32f\_16s\_C1R
  - image\_1D\_linear\_filter, 77
- nppiFilterColumnBorder32f\_16s\_C3R
  - image\_1D\_linear\_filter, 78
- nppiFilterColumnBorder32f\_16s\_C4R
  - image\_1D\_linear\_filter, 78
- nppiFilterColumnBorder32f\_16u\_AC4R
  - image\_1D\_linear\_filter, 79
- nppiFilterColumnBorder32f\_16u\_C1R
  - image\_1D\_linear\_filter, 79
- nppiFilterColumnBorder32f\_16u\_C3R
  - image\_1D\_linear\_filter, 80
- nppiFilterColumnBorder32f\_16u\_C4R
  - image\_1D\_linear\_filter, 80
- nppiFilterColumnBorder32f\_8u\_AC4R
  - image\_1D\_linear\_filter, 81
- nppiFilterColumnBorder32f\_8u\_C1R
  - image\_1D\_linear\_filter, 81
- nppiFilterColumnBorder32f\_8u\_C3R
  - image\_1D\_linear\_filter, 82
- nppiFilterColumnBorder32f\_8u\_C4R
  - image\_1D\_linear\_filter, 82
- nppiFilterColumnBorder\_16s\_AC4R
  - image\_1D\_linear\_filter, 83
- nppiFilterColumnBorder\_16s\_C1R
  - image\_1D\_linear\_filter, 84
- nppiFilterColumnBorder\_16s\_C3R
  - image\_1D\_linear\_filter, 84
- nppiFilterColumnBorder\_16s\_C4R
  - image\_1D\_linear\_filter, 85
- nppiFilterColumnBorder\_16u\_AC4R
  - image\_1D\_linear\_filter, 85
- nppiFilterColumnBorder\_16u\_C1R
  - image\_1D\_linear\_filter, 86
- nppiFilterColumnBorder\_16u\_C3R
  - image\_1D\_linear\_filter, 87
- nppiFilterColumnBorder\_16u\_C4R
  - image\_1D\_linear\_filter, 87

image\_1D\_linear\_filter, 87  
 nppiFilterColumnBorder\_32f\_AC4R  
     image\_1D\_linear\_filter, 88  
 nppiFilterColumnBorder\_32f\_C1R  
     image\_1D\_linear\_filter, 88  
 nppiFilterColumnBorder\_32f\_C3R  
     image\_1D\_linear\_filter, 89  
 nppiFilterColumnBorder\_32f\_C4R  
     image\_1D\_linear\_filter, 89  
 nppiFilterColumnBorder\_8u\_AC4R  
     image\_1D\_linear\_filter, 90  
 nppiFilterColumnBorder\_8u\_C1R  
     image\_1D\_linear\_filter, 91  
 nppiFilterColumnBorder\_8u\_C3R  
     image\_1D\_linear\_filter, 91  
 nppiFilterColumnBorder\_8u\_C4R  
     image\_1D\_linear\_filter, 92  
 nppiFilterGauss\_16s\_AC4R  
     fixed\_filters, 334  
 nppiFilterGauss\_16s\_C1R  
     fixed\_filters, 334  
 nppiFilterGauss\_16s\_C3R  
     fixed\_filters, 334  
 nppiFilterGauss\_16s\_C4R  
     fixed\_filters, 335  
 nppiFilterGauss\_16u\_AC4R  
     fixed\_filters, 335  
 nppiFilterGauss\_16u\_C1R  
     fixed\_filters, 335  
 nppiFilterGauss\_16u\_C3R  
     fixed\_filters, 336  
 nppiFilterGauss\_16u\_C4R  
     fixed\_filters, 336  
 nppiFilterGauss\_32f\_AC4R  
     fixed\_filters, 336  
 nppiFilterGauss\_32f\_C1R  
     fixed\_filters, 337  
 nppiFilterGauss\_32f\_C3R  
     fixed\_filters, 337  
 nppiFilterGauss\_32f\_C4R  
     fixed\_filters, 337  
 nppiFilterGauss\_8u\_AC4R  
     fixed\_filters, 338  
 nppiFilterGauss\_8u\_C1R  
     fixed\_filters, 338  
 nppiFilterGauss\_8u\_C3R  
     fixed\_filters, 338  
 nppiFilterGauss\_8u\_C4R  
     fixed\_filters, 339  
 nppiFilterGaussAdvanced\_16s\_AC4R  
     fixed\_filters, 339  
 nppiFilterGaussAdvanced\_16s\_C1R  
     fixed\_filters, 340  
 nppiFilterGaussAdvanced\_16s\_C3R

    fixed\_filters, 340  
 nppiFilterGaussAdvanced\_16s\_C4R  
     fixed\_filters, 340  
 nppiFilterGaussAdvanced\_16u\_AC4R  
     fixed\_filters, 341  
 nppiFilterGaussAdvanced\_16u\_C1R  
     fixed\_filters, 341  
 nppiFilterGaussAdvanced\_16u\_C3R  
     fixed\_filters, 342  
 nppiFilterGaussAdvanced\_16u\_C4R  
     fixed\_filters, 342  
 nppiFilterGaussAdvanced\_32f\_AC4R  
     fixed\_filters, 342  
 nppiFilterGaussAdvanced\_32f\_C1R  
     fixed\_filters, 343  
 nppiFilterGaussAdvanced\_32f\_C3R  
     fixed\_filters, 343  
 nppiFilterGaussAdvanced\_32f\_C4R  
     fixed\_filters, 344  
 nppiFilterGaussAdvanced\_8u\_AC4R  
     fixed\_filters, 344  
 nppiFilterGaussAdvanced\_8u\_C1R  
     fixed\_filters, 344  
 nppiFilterGaussAdvanced\_8u\_C3R  
     fixed\_filters, 345  
 nppiFilterGaussAdvanced\_8u\_C4R  
     fixed\_filters, 345  
 nppiFilterGaussAdvancedBorder\_16s\_AC4R  
     fixed\_filters, 346  
 nppiFilterGaussAdvancedBorder\_16s\_C1R  
     fixed\_filters, 346  
 nppiFilterGaussAdvancedBorder\_16s\_C3R  
     fixed\_filters, 347  
 nppiFilterGaussAdvancedBorder\_16s\_C4R  
     fixed\_filters, 347  
 nppiFilterGaussAdvancedBorder\_16u\_AC4R  
     fixed\_filters, 348  
 nppiFilterGaussAdvancedBorder\_16u\_C1R  
     fixed\_filters, 348  
 nppiFilterGaussAdvancedBorder\_16u\_C3R  
     fixed\_filters, 349  
 nppiFilterGaussAdvancedBorder\_16u\_C4R  
     fixed\_filters, 349  
 nppiFilterGaussAdvancedBorder\_32f\_AC4R  
     fixed\_filters, 350  
 nppiFilterGaussAdvancedBorder\_32f\_C1R  
     fixed\_filters, 350  
 nppiFilterGaussAdvancedBorder\_32f\_C3R  
     fixed\_filters, 351  
 nppiFilterGaussAdvancedBorder\_32f\_C4R  
     fixed\_filters, 351  
 nppiFilterGaussAdvancedBorder\_8u\_AC4R  
     fixed\_filters, 352  
 nppiFilterGaussAdvancedBorder\_8u\_C1R

fixed\_filters, 352  
nppiFilterGaussAdvancedBorder\_8u\_C3R  
    fixed\_filters, 353  
nppiFilterGaussAdvancedBorder\_8u\_C4R  
    fixed\_filters, 353  
nppiFilterGaussBorder\_16s\_AC4R  
    fixed\_filters, 354  
nppiFilterGaussBorder\_16s\_C1R  
    fixed\_filters, 354  
nppiFilterGaussBorder\_16s\_C3R  
    fixed\_filters, 355  
nppiFilterGaussBorder\_16s\_C4R  
    fixed\_filters, 355  
nppiFilterGaussBorder\_16u\_AC4R  
    fixed\_filters, 356  
nppiFilterGaussBorder\_16u\_C1R  
    fixed\_filters, 356  
nppiFilterGaussBorder\_16u\_C3R  
    fixed\_filters, 356  
nppiFilterGaussBorder\_16u\_C4R  
    fixed\_filters, 357  
nppiFilterGaussBorder\_32f\_AC4R  
    fixed\_filters, 357  
nppiFilterGaussBorder\_32f\_C1R  
    fixed\_filters, 358  
nppiFilterGaussBorder\_32f\_C3R  
    fixed\_filters, 358  
nppiFilterGaussBorder\_32f\_C4R  
    fixed\_filters, 359  
nppiFilterGaussBorder\_8u\_AC4R  
    fixed\_filters, 359  
nppiFilterGaussBorder\_8u\_C1R  
    fixed\_filters, 360  
nppiFilterGaussBorder\_8u\_C3R  
    fixed\_filters, 360  
nppiFilterGaussBorder\_8u\_C4R  
    fixed\_filters, 361  
nppiFilterGaussPyramidLayerDownBorder\_16u\_-  
    C1R  
    fixed\_filters, 361  
nppiFilterGaussPyramidLayerDownBorder\_16u\_-  
    C3R  
    fixed\_filters, 362  
nppiFilterGaussPyramidLayerDownBorder\_32f\_-  
    C1R  
    fixed\_filters, 363  
nppiFilterGaussPyramidLayerDownBorder\_32f\_-  
    C3R  
    fixed\_filters, 363  
nppiFilterGaussPyramidLayerDownBorder\_8u\_-  
    C1R  
    fixed\_filters, 364  
nppiFilterGaussPyramidLayerDownBorder\_8u\_-  
    C3R  
    fixed\_filters, 364  
nppiFilterGaussPyramidLayerUpBorder\_16u\_C1R  
    fixed\_filters, 365  
nppiFilterGaussPyramidLayerUpBorder\_16u\_C3R  
    fixed\_filters, 365  
nppiFilterGaussPyramidLayerUpBorder\_32f\_C1R  
    fixed\_filters, 366  
nppiFilterGaussPyramidLayerUpBorder\_32f\_C3R  
    fixed\_filters, 367  
nppiFilterGaussPyramidLayerUpBorder\_8u\_C1R  
    fixed\_filters, 367  
nppiFilterGaussPyramidLayerUpBorder\_8u\_C3R  
    fixed\_filters, 368  
nppiFilterHarrisCornersBorder\_8u32f\_C1R  
    fixed\_filters, 368  
nppiFilterHarrisCornersBorderGetBufferSize  
    fixed\_filters, 369  
nppiFilterHighPass\_16s\_AC4R  
    fixed\_filters, 369  
nppiFilterHighPass\_16s\_C1R  
    fixed\_filters, 370  
nppiFilterHighPass\_16s\_C3R  
    fixed\_filters, 370  
nppiFilterHighPass\_16s\_C4R  
    fixed\_filters, 370  
nppiFilterHighPass\_16u\_AC4R  
    fixed\_filters, 371  
nppiFilterHighPass\_16u\_C1R  
    fixed\_filters, 371  
nppiFilterHighPass\_16u\_C3R  
    fixed\_filters, 371  
nppiFilterHighPass\_16u\_C4R  
    fixed\_filters, 372  
nppiFilterHighPass\_32f\_AC4R  
    fixed\_filters, 372  
nppiFilterHighPass\_32f\_C1R  
    fixed\_filters, 372  
nppiFilterHighPass\_32f\_C3R  
    fixed\_filters, 373  
nppiFilterHighPass\_32f\_C4R  
    fixed\_filters, 373  
nppiFilterHighPass\_8u\_AC4R  
    fixed\_filters, 373  
nppiFilterHighPass\_8u\_C1R  
    fixed\_filters, 374  
nppiFilterHighPass\_8u\_C3R  
    fixed\_filters, 374  
nppiFilterHighPass\_8u\_C4R  
    fixed\_filters, 374  
nppiFilterHighPassBorder\_16s\_AC4R  
    fixed\_filters, 375  
nppiFilterHighPassBorder\_16s\_C1R  
    fixed\_filters, 375  
nppiFilterHighPassBorder\_16s\_C3R

fixed\_filters, 376  
 nppiFilterHighPassBorder\_16s\_C4R  
     fixed\_filters, 376  
 nppiFilterHighPassBorder\_16u\_AC4R  
     fixed\_filters, 377  
 nppiFilterHighPassBorder\_16u\_C1R  
     fixed\_filters, 377  
 nppiFilterHighPassBorder\_16u\_C3R  
     fixed\_filters, 378  
 nppiFilterHighPassBorder\_16u\_C4R  
     fixed\_filters, 378  
 nppiFilterHighPassBorder\_32f\_AC4R  
     fixed\_filters, 378  
 nppiFilterHighPassBorder\_32f\_C1R  
     fixed\_filters, 379  
 nppiFilterHighPassBorder\_32f\_C3R  
     fixed\_filters, 379  
 nppiFilterHighPassBorder\_32f\_C4R  
     fixed\_filters, 380  
 nppiFilterHighPassBorder\_8u\_AC4R  
     fixed\_filters, 380  
 nppiFilterHighPassBorder\_8u\_C1R  
     fixed\_filters, 381  
 nppiFilterHighPassBorder\_8u\_C3R  
     fixed\_filters, 381  
 nppiFilterHighPassBorder\_8u\_C4R  
     fixed\_filters, 382  
 nppiFilterLaplace\_16s\_AC4R  
     fixed\_filters, 382  
 nppiFilterLaplace\_16s\_C1R  
     fixed\_filters, 383  
 nppiFilterLaplace\_16s\_C3R  
     fixed\_filters, 383  
 nppiFilterLaplace\_16s\_C4R  
     fixed\_filters, 383  
 nppiFilterLaplace\_32f\_AC4R  
     fixed\_filters, 384  
 nppiFilterLaplace\_32f\_C1R  
     fixed\_filters, 384  
 nppiFilterLaplace\_32f\_C3R  
     fixed\_filters, 384  
 nppiFilterLaplace\_32f\_C4R  
     fixed\_filters, 385  
 nppiFilterLaplace\_8s16s\_C1R  
     fixed\_filters, 385  
 nppiFilterLaplace\_8u16s\_C1R  
     fixed\_filters, 385  
 nppiFilterLaplace\_8u\_AC4R  
     fixed\_filters, 386  
 nppiFilterLaplace\_8u\_C1R  
     fixed\_filters, 386  
 nppiFilterLaplace\_8u\_C3R  
     fixed\_filters, 386  
 nppiFilterLaplace\_8u\_C4R

    fixed\_filters, 387  
 nppiFilterLaplaceBorder\_16s\_AC4R  
     fixed\_filters, 387  
 nppiFilterLaplaceBorder\_16s\_C1R  
     fixed\_filters, 388  
 nppiFilterLaplaceBorder\_16s\_C3R  
     fixed\_filters, 388  
 nppiFilterLaplaceBorder\_16s\_C4R  
     fixed\_filters, 389  
 nppiFilterLaplaceBorder\_32f\_AC4R  
     fixed\_filters, 389  
 nppiFilterLaplaceBorder\_32f\_C1R  
     fixed\_filters, 389  
 nppiFilterLaplaceBorder\_32f\_C3R  
     fixed\_filters, 390  
 nppiFilterLaplaceBorder\_32f\_C4R  
     fixed\_filters, 390  
 nppiFilterLaplaceBorder\_8s16s\_C1R  
     fixed\_filters, 391  
 nppiFilterLaplaceBorder\_8u16s\_C1R  
     fixed\_filters, 391  
 nppiFilterLaplaceBorder\_8u\_AC4R  
     fixed\_filters, 392  
 nppiFilterLaplaceBorder\_8u\_C1R  
     fixed\_filters, 392  
 nppiFilterLaplaceBorder\_8u\_C3R  
     fixed\_filters, 393  
 nppiFilterLaplaceBorder\_8u\_C4R  
     fixed\_filters, 393  
 nppiFilterLowPass\_16s\_AC4R  
     fixed\_filters, 394  
 nppiFilterLowPass\_16s\_C1R  
     fixed\_filters, 394  
 nppiFilterLowPass\_16s\_C3R  
     fixed\_filters, 395  
 nppiFilterLowPass\_16s\_C4R  
     fixed\_filters, 395  
 nppiFilterLowPass\_16u\_AC4R  
     fixed\_filters, 395  
 nppiFilterLowPass\_16u\_C1R  
     fixed\_filters, 396  
 nppiFilterLowPass\_16u\_C3R  
     fixed\_filters, 396  
 nppiFilterLowPass\_16u\_C4R  
     fixed\_filters, 396  
 nppiFilterLowPass\_32f\_AC4R  
     fixed\_filters, 397  
 nppiFilterLowPass\_32f\_C1R  
     fixed\_filters, 397  
 nppiFilterLowPass\_32f\_C3R  
     fixed\_filters, 397  
 nppiFilterLowPass\_32f\_C4R  
     fixed\_filters, 398  
 nppiFilterLowPass\_8u\_AC4R

fixed\_filters, 398  
nppiFilterLowPass\_8u\_C1R  
    fixed\_filters, 398  
nppiFilterLowPass\_8u\_C3R  
    fixed\_filters, 399  
nppiFilterLowPass\_8u\_C4R  
    fixed\_filters, 399  
nppiFilterLowPassBorder\_16s\_AC4R  
    fixed\_filters, 399  
nppiFilterLowPassBorder\_16s\_C1R  
    fixed\_filters, 400  
nppiFilterLowPassBorder\_16s\_C3R  
    fixed\_filters, 400  
nppiFilterLowPassBorder\_16s\_C4R  
    fixed\_filters, 401  
nppiFilterLowPassBorder\_16u\_AC4R  
    fixed\_filters, 401  
nppiFilterLowPassBorder\_16u\_C1R  
    fixed\_filters, 402  
nppiFilterLowPassBorder\_16u\_C3R  
    fixed\_filters, 402  
nppiFilterLowPassBorder\_16u\_C4R  
    fixed\_filters, 403  
nppiFilterLowPassBorder\_32f\_AC4R  
    fixed\_filters, 403  
nppiFilterLowPassBorder\_32f\_C1R  
    fixed\_filters, 404  
nppiFilterLowPassBorder\_32f\_C3R  
    fixed\_filters, 404  
nppiFilterLowPassBorder\_32f\_C4R  
    fixed\_filters, 405  
nppiFilterLowPassBorder\_8u\_AC4R  
    fixed\_filters, 405  
nppiFilterLowPassBorder\_8u\_C1R  
    fixed\_filters, 406  
nppiFilterLowPassBorder\_8u\_C3R  
    fixed\_filters, 406  
nppiFilterLowPassBorder\_8u\_C4R  
    fixed\_filters, 407  
nppiFilterMax\_16s\_AC4R  
    image\_rank\_filters, 233  
nppiFilterMax\_16s\_C1R  
    image\_rank\_filters, 233  
nppiFilterMax\_16s\_C3R  
    image\_rank\_filters, 233  
nppiFilterMax\_16s\_C4R  
    image\_rank\_filters, 234  
nppiFilterMax\_16u\_AC4R  
    image\_rank\_filters, 234  
nppiFilterMax\_16u\_C1R  
    image\_rank\_filters, 235  
nppiFilterMax\_16u\_C3R  
    image\_rank\_filters, 235  
nppiFilterMax\_16u\_C4R

    image\_rank\_filters, 235  
nppiFilterMax\_32f\_AC4R  
    image\_rank\_filters, 236  
nppiFilterMax\_32f\_C1R  
    image\_rank\_filters, 236  
nppiFilterMax\_32f\_C3R  
    image\_rank\_filters, 237  
nppiFilterMax\_32f\_C4R  
    image\_rank\_filters, 237  
nppiFilterMax\_8u\_AC4R  
    image\_rank\_filters, 237  
nppiFilterMax\_8u\_C1R  
    image\_rank\_filters, 238  
nppiFilterMax\_8u\_C3R  
    image\_rank\_filters, 238  
nppiFilterMax\_8u\_C4R  
    image\_rank\_filters, 239  
nppiFilterMaxBorder\_16s\_AC4R  
    image\_rank\_filters, 239  
nppiFilterMaxBorder\_16s\_C1R  
    image\_rank\_filters, 239  
nppiFilterMaxBorder\_16s\_C3R  
    image\_rank\_filters, 240  
nppiFilterMaxBorder\_16s\_C4R  
    image\_rank\_filters, 240  
nppiFilterMaxBorder\_16u\_AC4R  
    image\_rank\_filters, 241  
nppiFilterMaxBorder\_16u\_C1R  
    image\_rank\_filters, 241  
nppiFilterMaxBorder\_16u\_C3R  
    image\_rank\_filters, 242  
nppiFilterMaxBorder\_16u\_C4R  
    image\_rank\_filters, 242  
nppiFilterMaxBorder\_32f\_AC4R  
    image\_rank\_filters, 243  
nppiFilterMaxBorder\_32f\_C1R  
    image\_rank\_filters, 243  
nppiFilterMaxBorder\_32f\_C3R  
    image\_rank\_filters, 244  
nppiFilterMaxBorder\_32f\_C4R  
    image\_rank\_filters, 244  
nppiFilterMaxBorder\_8u\_AC4R  
    image\_rank\_filters, 245  
nppiFilterMaxBorder\_8u\_C1R  
    image\_rank\_filters, 245  
nppiFilterMaxBorder\_8u\_C3R  
    image\_rank\_filters, 246  
nppiFilterMaxBorder\_8u\_C4R  
    image\_rank\_filters, 246  
nppiFilterMedian\_16s\_AC4R  
    image\_rank\_filters, 247  
nppiFilterMedian\_16s\_C1R  
    image\_rank\_filters, 247  
nppiFilterMedian\_16s\_C3R

- nppiFilterMedian\_16s\_C4R
  - image\_rank\_filters, 248
- nppiFilterMedian\_16u\_AC4R
  - image\_rank\_filters, 249
- nppiFilterMedian\_16u\_C1R
  - image\_rank\_filters, 249
- nppiFilterMedian\_16u\_C3R
  - image\_rank\_filters, 250
- nppiFilterMedian\_16u\_C4R
  - image\_rank\_filters, 250
- nppiFilterMedian\_32f\_AC4R
  - image\_rank\_filters, 250
- nppiFilterMedian\_32f\_C1R
  - image\_rank\_filters, 251
- nppiFilterMedian\_32f\_C3R
  - image\_rank\_filters, 251
- nppiFilterMedian\_32f\_C4R
  - image\_rank\_filters, 252
- nppiFilterMedian\_8u\_AC4R
  - image\_rank\_filters, 252
- nppiFilterMedian\_8u\_C1R
  - image\_rank\_filters, 253
- nppiFilterMedian\_8u\_C3R
  - image\_rank\_filters, 253
- nppiFilterMedian\_8u\_C4R
  - image\_rank\_filters, 253
- nppiFilterMedianGetBufferSize\_16s\_AC4R
  - image\_rank\_filters, 254
- nppiFilterMedianGetBufferSize\_16s\_C1R
  - image\_rank\_filters, 254
- nppiFilterMedianGetBufferSize\_16s\_C3R
  - image\_rank\_filters, 254
- nppiFilterMedianGetBufferSize\_16s\_C4R
  - image\_rank\_filters, 255
- nppiFilterMedianGetBufferSize\_16u\_AC4R
  - image\_rank\_filters, 255
- nppiFilterMedianGetBufferSize\_16u\_C1R
  - image\_rank\_filters, 255
- nppiFilterMedianGetBufferSize\_16u\_C3R
  - image\_rank\_filters, 256
- nppiFilterMedianGetBufferSize\_16u\_C4R
  - image\_rank\_filters, 256
- nppiFilterMedianGetBufferSize\_32f\_AC4R
  - image\_rank\_filters, 256
- nppiFilterMedianGetBufferSize\_32f\_C1R
  - image\_rank\_filters, 256
- nppiFilterMedianGetBufferSize\_32f\_C3R
  - image\_rank\_filters, 257
- nppiFilterMedianGetBufferSize\_32f\_C4R
  - image\_rank\_filters, 257
- nppiFilterMedianGetBufferSize\_8u\_AC4R
  - image\_rank\_filters, 257
- nppiFilterMedianGetBufferSize\_8u\_C1R
  - image\_rank\_filters, 258
- nppiFilterMedianGetBufferSize\_8u\_C3R
  - image\_rank\_filters, 258
- nppiFilterMedianGetBufferSize\_8u\_C4R
  - image\_rank\_filters, 258
- nppiFilterMin\_16s\_AC4R
  - image\_rank\_filters, 258
- nppiFilterMin\_16s\_C1R
  - image\_rank\_filters, 259
- nppiFilterMin\_16s\_C3R
  - image\_rank\_filters, 259
- nppiFilterMin\_16s\_C4R
  - image\_rank\_filters, 260
- nppiFilterMin\_16u\_AC4R
  - image\_rank\_filters, 260
- nppiFilterMin\_16u\_C1R
  - image\_rank\_filters, 260
- nppiFilterMin\_16u\_C3R
  - image\_rank\_filters, 261
- nppiFilterMin\_16u\_C4R
  - image\_rank\_filters, 261
- nppiFilterMin\_32f\_AC4R
  - image\_rank\_filters, 262
- nppiFilterMin\_32f\_C1R
  - image\_rank\_filters, 262
- nppiFilterMin\_32f\_C3R
  - image\_rank\_filters, 262
- nppiFilterMin\_32f\_C4R
  - image\_rank\_filters, 263
- nppiFilterMin\_8u\_AC4R
  - image\_rank\_filters, 263
- nppiFilterMin\_8u\_C1R
  - image\_rank\_filters, 264
- nppiFilterMin\_8u\_C3R
  - image\_rank\_filters, 264
- nppiFilterMin\_8u\_C4R
  - image\_rank\_filters, 264
- nppiFilterMinBorder\_16s\_AC4R
  - image\_rank\_filters, 265
- nppiFilterMinBorder\_16s\_C1R
  - image\_rank\_filters, 265
- nppiFilterMinBorder\_16s\_C3R
  - image\_rank\_filters, 266
- nppiFilterMinBorder\_16s\_C4R
  - image\_rank\_filters, 266
- nppiFilterMinBorder\_16u\_AC4R
  - image\_rank\_filters, 267
- nppiFilterMinBorder\_16u\_C1R
  - image\_rank\_filters, 267
- nppiFilterMinBorder\_16u\_C3R
  - image\_rank\_filters, 268
- nppiFilterMinBorder\_16u\_C4R
  - image\_rank\_filters, 268
- nppiFilterMinBorder\_32f\_AC4R
  - image\_rank\_filters, 268

- image\_rank\_filters, 269
- nppiFilterMinBorder\_32f\_C1R
  - image\_rank\_filters, 269
- nppiFilterMinBorder\_32f\_C3R
  - image\_rank\_filters, 270
- nppiFilterMinBorder\_32f\_C4R
  - image\_rank\_filters, 270
- nppiFilterMinBorder\_8u\_AC4R
  - image\_rank\_filters, 271
- nppiFilterMinBorder\_8u\_C1R
  - image\_rank\_filters, 271
- nppiFilterMinBorder\_8u\_C3R
  - image\_rank\_filters, 272
- nppiFilterMinBorder\_8u\_C4R
  - image\_rank\_filters, 272
- nppiFilterPrewittHoriz\_16s\_AC4R
  - fixed\_filters, 407
- nppiFilterPrewittHoriz\_16s\_C1R
  - fixed\_filters, 408
- nppiFilterPrewittHoriz\_16s\_C3R
  - fixed\_filters, 408
- nppiFilterPrewittHoriz\_16s\_C4R
  - fixed\_filters, 408
- nppiFilterPrewittHoriz\_32f\_AC4R
  - fixed\_filters, 409
- nppiFilterPrewittHoriz\_32f\_C1R
  - fixed\_filters, 409
- nppiFilterPrewittHoriz\_32f\_C3R
  - fixed\_filters, 409
- nppiFilterPrewittHoriz\_32f\_C4R
  - fixed\_filters, 410
- nppiFilterPrewittHoriz\_8u\_AC4R
  - fixed\_filters, 410
- nppiFilterPrewittHoriz\_8u\_C1R
  - fixed\_filters, 410
- nppiFilterPrewittHoriz\_8u\_C3R
  - fixed\_filters, 411
- nppiFilterPrewittHoriz\_8u\_C4R
  - fixed\_filters, 411
- nppiFilterPrewittHorizBorder\_16s\_AC4R
  - fixed\_filters, 411
- nppiFilterPrewittHorizBorder\_16s\_C1R
  - fixed\_filters, 412
- nppiFilterPrewittHorizBorder\_16s\_C3R
  - fixed\_filters, 412
- nppiFilterPrewittHorizBorder\_16s\_C4R
  - fixed\_filters, 413
- nppiFilterPrewittHorizBorder\_32f\_AC4R
  - fixed\_filters, 413
- nppiFilterPrewittHorizBorder\_32f\_C1R
  - fixed\_filters, 414
- nppiFilterPrewittHorizBorder\_32f\_C3R
  - fixed\_filters, 414
- nppiFilterPrewittHorizBorder\_32f\_C4R
  - fixed\_filters, 414
- nppiFilterPrewittHorizBorder\_8u\_AC4R
  - fixed\_filters, 415
- nppiFilterPrewittHorizBorder\_8u\_C1R
  - fixed\_filters, 415
- nppiFilterPrewittHorizBorder\_8u\_C3R
  - fixed\_filters, 416
- nppiFilterPrewittHorizBorder\_8u\_C4R
  - fixed\_filters, 416
- nppiFilterPrewittVert\_16s\_AC4R
  - fixed\_filters, 417
- nppiFilterPrewittVert\_16s\_C1R
  - fixed\_filters, 417
- nppiFilterPrewittVert\_16s\_C3R
  - fixed\_filters, 417
- nppiFilterPrewittVert\_16s\_C4R
  - fixed\_filters, 418
- nppiFilterPrewittVert\_32f\_AC4R
  - fixed\_filters, 418
- nppiFilterPrewittVert\_32f\_C1R
  - fixed\_filters, 418
- nppiFilterPrewittVert\_32f\_C3R
  - fixed\_filters, 419
- nppiFilterPrewittVert\_32f\_C4R
  - fixed\_filters, 419
- nppiFilterPrewittVert\_8u\_AC4R
  - fixed\_filters, 419
- nppiFilterPrewittVert\_8u\_C1R
  - fixed\_filters, 420
- nppiFilterPrewittVert\_8u\_C3R
  - fixed\_filters, 420
- nppiFilterPrewittVert\_8u\_C4R
  - fixed\_filters, 420
- nppiFilterPrewittVertBorder\_16s\_AC4R
  - fixed\_filters, 421
- nppiFilterPrewittVertBorder\_16s\_C1R
  - fixed\_filters, 421
- nppiFilterPrewittVertBorder\_16s\_C3R
  - fixed\_filters, 422
- nppiFilterPrewittVertBorder\_16s\_C4R
  - fixed\_filters, 422
- nppiFilterPrewittVertBorder\_32f\_AC4R
  - fixed\_filters, 422
- nppiFilterPrewittVertBorder\_32f\_C1R
  - fixed\_filters, 423
- nppiFilterPrewittVertBorder\_32f\_C3R
  - fixed\_filters, 423
- nppiFilterPrewittVertBorder\_32f\_C4R
  - fixed\_filters, 424
- nppiFilterPrewittVertBorder\_8u\_AC4R
  - fixed\_filters, 424
- nppiFilterPrewittVertBorder\_8u\_C1R
  - fixed\_filters, 425
- nppiFilterPrewittVertBorder\_8u\_C3R
  - fixed\_filters, 425

fixed\_filters, 425  
 nppiFilterPrewittVertBorder\_8u\_C4R  
     fixed\_filters, 425  
 nppiFilterRobertsDown\_16s\_AC4R  
     fixed\_filters, 426  
 nppiFilterRobertsDown\_16s\_C1R  
     fixed\_filters, 426  
 nppiFilterRobertsDown\_16s\_C3R  
     fixed\_filters, 427  
 nppiFilterRobertsDown\_16s\_C4R  
     fixed\_filters, 427  
 nppiFilterRobertsDown\_32f\_AC4R  
     fixed\_filters, 427  
 nppiFilterRobertsDown\_32f\_C1R  
     fixed\_filters, 428  
 nppiFilterRobertsDown\_32f\_C3R  
     fixed\_filters, 428  
 nppiFilterRobertsDown\_32f\_C4R  
     fixed\_filters, 428  
 nppiFilterRobertsDown\_8u\_AC4R  
     fixed\_filters, 429  
 nppiFilterRobertsDown\_8u\_C1R  
     fixed\_filters, 429  
 nppiFilterRobertsDown\_8u\_C3R  
     fixed\_filters, 429  
 nppiFilterRobertsDown\_8u\_C4R  
     fixed\_filters, 430  
 nppiFilterRobertsDownBorder\_16s\_AC4R  
     fixed\_filters, 430  
 nppiFilterRobertsDownBorder\_16s\_C1R  
     fixed\_filters, 431  
 nppiFilterRobertsDownBorder\_16s\_C3R  
     fixed\_filters, 431  
 nppiFilterRobertsDownBorder\_16s\_C4R  
     fixed\_filters, 431  
 nppiFilterRobertsDownBorder\_32f\_AC4R  
     fixed\_filters, 432  
 nppiFilterRobertsDownBorder\_32f\_C1R  
     fixed\_filters, 432  
 nppiFilterRobertsDownBorder\_32f\_C3R  
     fixed\_filters, 433  
 nppiFilterRobertsDownBorder\_32f\_C4R  
     fixed\_filters, 433  
 nppiFilterRobertsDownBorder\_8u\_AC4R  
     fixed\_filters, 434  
 nppiFilterRobertsDownBorder\_8u\_C1R  
     fixed\_filters, 434  
 nppiFilterRobertsDownBorder\_8u\_C3R  
     fixed\_filters, 434  
 nppiFilterRobertsDownBorder\_8u\_C4R  
     fixed\_filters, 435  
 nppiFilterRobertsUp\_16s\_AC4R  
     fixed\_filters, 435  
 nppiFilterRobertsUp\_16s\_C1R  
     fixed\_filters, 436  
 nppiFilterRobertsUp\_16s\_C3R  
     fixed\_filters, 436  
 nppiFilterRobertsUp\_16s\_C4R  
     fixed\_filters, 436  
 nppiFilterRobertsUp\_32f\_AC4R  
     fixed\_filters, 437  
 nppiFilterRobertsUp\_32f\_C1R  
     fixed\_filters, 437  
 nppiFilterRobertsUp\_32f\_C3R  
     fixed\_filters, 437  
 nppiFilterRobertsUp\_32f\_C4R  
     fixed\_filters, 438  
 nppiFilterRobertsUp\_8u\_AC4R  
     fixed\_filters, 438  
 nppiFilterRobertsUp\_8u\_C1R  
     fixed\_filters, 438  
 nppiFilterRobertsUp\_8u\_C3R  
     fixed\_filters, 439  
 nppiFilterRobertsUp\_8u\_C4R  
     fixed\_filters, 439  
 nppiFilterRobotsUpBorder\_16s\_AC4R  
     fixed\_filters, 439  
 nppiFilterRobotsUpBorder\_16s\_C1R  
     fixed\_filters, 440  
 nppiFilterRobotsUpBorder\_16s\_C3R  
     fixed\_filters, 440  
 nppiFilterRobotsUpBorder\_16s\_C4R  
     fixed\_filters, 441  
 nppiFilterRobotsUpBorder\_32f\_AC4R  
     fixed\_filters, 441  
 nppiFilterRobotsUpBorder\_32f\_C1R  
     fixed\_filters, 442  
 nppiFilterRobotsUpBorder\_32f\_C3R  
     fixed\_filters, 442  
 nppiFilterRobotsUpBorder\_32f\_C4R  
     fixed\_filters, 442  
 nppiFilterRobotsUpBorder\_8u\_AC4R  
     fixed\_filters, 443  
 nppiFilterRobotsUpBorder\_8u\_C1R  
     fixed\_filters, 443  
 nppiFilterRobotsUpBorder\_8u\_C3R  
     fixed\_filters, 444  
 nppiFilterRobotsUpBorder\_8u\_C4R  
     fixed\_filters, 444  
 nppiFilterRow32f\_16s\_AC4R  
     image\_1D\_linear\_filter, 92  
 nppiFilterRow32f\_16s\_C1R  
     image\_1D\_linear\_filter, 93  
 nppiFilterRow32f\_16s\_C3R  
     image\_1D\_linear\_filter, 93  
 nppiFilterRow32f\_16s\_C4R  
     image\_1D\_linear\_filter, 94  
 nppiFilterRow32f\_16u\_AC4R

- image\_1D\_linear\_filter, 94
- nppiFilterRow32f\_16u\_C1R
  - image\_1D\_linear\_filter, 95
- nppiFilterRow32f\_16u\_C3R
  - image\_1D\_linear\_filter, 95
- nppiFilterRow32f\_16u\_C4R
  - image\_1D\_linear\_filter, 96
- nppiFilterRow32f\_8u\_AC4R
  - image\_1D\_linear\_filter, 96
- nppiFilterRow32f\_8u\_C1R
  - image\_1D\_linear\_filter, 97
- nppiFilterRow32f\_8u\_C3R
  - image\_1D\_linear\_filter, 97
- nppiFilterRow32f\_8u\_C4R
  - image\_1D\_linear\_filter, 98
- nppiFilterRow\_16s\_AC4R
  - image\_1D\_linear\_filter, 98
- nppiFilterRow\_16s\_C1R
  - image\_1D\_linear\_filter, 99
- nppiFilterRow\_16s\_C3R
  - image\_1D\_linear\_filter, 99
- nppiFilterRow\_16s\_C4R
  - image\_1D\_linear\_filter, 100
- nppiFilterRow\_16u\_AC4R
  - image\_1D\_linear\_filter, 100
- nppiFilterRow\_16u\_C1R
  - image\_1D\_linear\_filter, 101
- nppiFilterRow\_16u\_C3R
  - image\_1D\_linear\_filter, 101
- nppiFilterRow\_16u\_C4R
  - image\_1D\_linear\_filter, 102
- nppiFilterRow\_32f\_AC4R
  - image\_1D\_linear\_filter, 102
- nppiFilterRow\_32f\_C1R
  - image\_1D\_linear\_filter, 103
- nppiFilterRow\_32f\_C3R
  - image\_1D\_linear\_filter, 103
- nppiFilterRow\_32f\_C4R
  - image\_1D\_linear\_filter, 104
- nppiFilterRow\_64f\_C1R
  - image\_1D\_linear\_filter, 104
- nppiFilterRow\_8u\_AC4R
  - image\_1D\_linear\_filter, 105
- nppiFilterRow\_8u\_C1R
  - image\_1D\_linear\_filter, 105
- nppiFilterRow\_8u\_C3R
  - image\_1D\_linear\_filter, 106
- nppiFilterRow\_8u\_C4R
  - image\_1D\_linear\_filter, 106
- nppiFilterRowBorder32f\_16s\_AC4R
  - image\_1D\_linear\_filter, 107
- nppiFilterRowBorder32f\_16s\_C1R
  - image\_1D\_linear\_filter, 107
- nppiFilterRowBorder32f\_16s\_C3R
  - image\_1D\_linear\_filter, 108
- nppiFilterRowBorder32f\_16s\_C4R
  - image\_1D\_linear\_filter, 108
- nppiFilterRowBorder32f\_16u\_AC4R
  - image\_1D\_linear\_filter, 109
- nppiFilterRowBorder32f\_16u\_C1R
  - image\_1D\_linear\_filter, 109
- nppiFilterRowBorder32f\_16u\_C3R
  - image\_1D\_linear\_filter, 110
- nppiFilterRowBorder32f\_16u\_C4R
  - image\_1D\_linear\_filter, 110
- nppiFilterRowBorder32f\_8u\_AC4R
  - image\_1D\_linear\_filter, 111
- nppiFilterRowBorder32f\_8u\_C1R
  - image\_1D\_linear\_filter, 111
- nppiFilterRowBorder32f\_8u\_C3R
  - image\_1D\_linear\_filter, 112
- nppiFilterRowBorder32f\_8u\_C4R
  - image\_1D\_linear\_filter, 112
- nppiFilterRowBorder\_16s\_AC4R
  - image\_1D\_linear\_filter, 113
- nppiFilterRowBorder\_16s\_C1R
  - image\_1D\_linear\_filter, 114
- nppiFilterRowBorder\_16s\_C3R
  - image\_1D\_linear\_filter, 114
- nppiFilterRowBorder\_16s\_C4R
  - image\_1D\_linear\_filter, 115
- nppiFilterRowBorder\_16u\_AC4R
  - image\_1D\_linear\_filter, 115
- nppiFilterRowBorder\_16u\_C1R
  - image\_1D\_linear\_filter, 116
- nppiFilterRowBorder\_16u\_C3R
  - image\_1D\_linear\_filter, 117
- nppiFilterRowBorder\_16u\_C4R
  - image\_1D\_linear\_filter, 117
- nppiFilterRowBorder\_32f\_AC4R
  - image\_1D\_linear\_filter, 118
- nppiFilterRowBorder\_32f\_C1R
  - image\_1D\_linear\_filter, 118
- nppiFilterRowBorder\_32f\_C3R
  - image\_1D\_linear\_filter, 119
- nppiFilterRowBorder\_32f\_C4R
  - image\_1D\_linear\_filter, 119
- nppiFilterRowBorder\_8u\_AC4R
  - image\_1D\_linear\_filter, 120
- nppiFilterRowBorder\_8u\_C1R
  - image\_1D\_linear\_filter, 121
- nppiFilterRowBorder\_8u\_C3R
  - image\_1D\_linear\_filter, 121
- nppiFilterRowBorder\_8u\_C4R
  - image\_1D\_linear\_filter, 122
- nppiFilterScharHoriz\_32f\_C1R
  - fixed\_filters, 445
- nppiFilterScharHoriz\_8s16s\_C1R

fixed\_filters, 445  
nppiFilterScharrHoriz\_8u16s\_C1R  
    fixed\_filters, 445  
nppiFilterScharrHorizBorder\_32f\_C1R  
    fixed\_filters, 446  
nppiFilterScharrHorizBorder\_8s16s\_C1R  
    fixed\_filters, 446  
nppiFilterScharrHorizBorder\_8u16s\_C1R  
    fixed\_filters, 447  
nppiFilterScharrVert\_32f\_C1R  
    fixed\_filters, 447  
nppiFilterScharrVert\_8s16s\_C1R  
    fixed\_filters, 447  
nppiFilterScharrVert\_8u16s\_C1R  
    fixed\_filters, 448  
nppiFilterScharrVertBorder\_32f\_C1R  
    fixed\_filters, 448  
nppiFilterScharrVertBorder\_8s16s\_C1R  
    fixed\_filters, 449  
nppiFilterScharrVertBorder\_8u16s\_C1R  
    fixed\_filters, 449  
nppiFilterSharpen\_16s\_AC4R  
    fixed\_filters, 449  
nppiFilterSharpen\_16s\_C1R  
    fixed\_filters, 450  
nppiFilterSharpen\_16s\_C3R  
    fixed\_filters, 450  
nppiFilterSharpen\_16s\_C4R  
    fixed\_filters, 450  
nppiFilterSharpen\_16u\_AC4R  
    fixed\_filters, 451  
nppiFilterSharpen\_16u\_C1R  
    fixed\_filters, 451  
nppiFilterSharpen\_16u\_C3R  
    fixed\_filters, 451  
nppiFilterSharpen\_16u\_C4R  
    fixed\_filters, 452  
nppiFilterSharpen\_32f\_AC4R  
    fixed\_filters, 452  
nppiFilterSharpen\_32f\_C1R  
    fixed\_filters, 452  
nppiFilterSharpen\_32f\_C3R  
    fixed\_filters, 453  
nppiFilterSharpen\_32f\_C4R  
    fixed\_filters, 453  
nppiFilterSharpen\_8u\_AC4R  
    fixed\_filters, 453  
nppiFilterSharpen\_8u\_C1R  
    fixed\_filters, 454  
nppiFilterSharpen\_8u\_C3R  
    fixed\_filters, 454  
nppiFilterSharpen\_8u\_C4R  
    fixed\_filters, 454  
nppiFilterSharpenBorder\_16s\_AC4R  
    fixed\_filters, 455  
nppiFilterSharpenBorder\_16s\_C1R  
    fixed\_filters, 455  
nppiFilterSharpenBorder\_16s\_C3R  
    fixed\_filters, 456  
nppiFilterSharpenBorder\_16s\_C4R  
    fixed\_filters, 456  
nppiFilterSharpenBorder\_16u\_AC4R  
    fixed\_filters, 456  
nppiFilterSharpenBorder\_16u\_C1R  
    fixed\_filters, 457  
nppiFilterSharpenBorder\_16u\_C3R  
    fixed\_filters, 457  
nppiFilterSharpenBorder\_16u\_C4R  
    fixed\_filters, 458  
nppiFilterSharpenBorder\_32f\_AC4R  
    fixed\_filters, 458  
nppiFilterSharpenBorder\_32f\_C1R  
    fixed\_filters, 459  
nppiFilterSharpenBorder\_32f\_C3R  
    fixed\_filters, 459  
nppiFilterSharpenBorder\_32f\_C4R  
    fixed\_filters, 459  
nppiFilterSharpenBorder\_8u\_AC4R  
    fixed\_filters, 460  
nppiFilterSharpenBorder\_8u\_C1R  
    fixed\_filters, 460  
nppiFilterSharpenBorder\_8u\_C3R  
    fixed\_filters, 461  
nppiFilterSharpenBorder\_8u\_C4R  
    fixed\_filters, 461  
nppiFilterSobelCross\_32f\_C1R  
    fixed\_filters, 462  
nppiFilterSobelCross\_8s16s\_C1R  
    fixed\_filters, 462  
nppiFilterSobelCross\_8u16s\_C1R  
    fixed\_filters, 462  
nppiFilterSobelCrossBorder\_32f\_C1R  
    fixed\_filters, 463  
nppiFilterSobelCrossBorder\_8s16s\_C1R  
    fixed\_filters, 463  
nppiFilterSobelCrossBorder\_8u16s\_C1R  
    fixed\_filters, 464  
nppiFilterSobelHoriz\_16s\_AC4R  
    fixed\_filters, 464  
nppiFilterSobelHoriz\_16s\_C1R  
    fixed\_filters, 464  
nppiFilterSobelHoriz\_16s\_C3R  
    fixed\_filters, 465  
nppiFilterSobelHoriz\_16s\_C4R  
    fixed\_filters, 465  
nppiFilterSobelHoriz\_32f\_AC4R  
    fixed\_filters, 465  
nppiFilterSobelHoriz\_32f\_C1R

- fixed\_filters, 466
- nppiFilterSobelHoriz\_32f\_C3R
  - fixed\_filters, 466
- nppiFilterSobelHoriz\_32f\_C4R
  - fixed\_filters, 466
- nppiFilterSobelHoriz\_8s16s\_C1R
  - fixed\_filters, 467
- nppiFilterSobelHoriz\_8u16s\_C1R
  - fixed\_filters, 467
- nppiFilterSobelHoriz\_8u\_AC4R
  - fixed\_filters, 467
- nppiFilterSobelHoriz\_8u\_C1R
  - fixed\_filters, 468
- nppiFilterSobelHoriz\_8u\_C3R
  - fixed\_filters, 468
- nppiFilterSobelHoriz\_8u\_C4R
  - fixed\_filters, 468
- nppiFilterSobelHorizBorder\_16s\_AC4R
  - fixed\_filters, 469
- nppiFilterSobelHorizBorder\_16s\_C1R
  - fixed\_filters, 469
- nppiFilterSobelHorizBorder\_16s\_C3R
  - fixed\_filters, 470
- nppiFilterSobelHorizBorder\_16s\_C4R
  - fixed\_filters, 470
- nppiFilterSobelHorizBorder\_32f\_AC4R
  - fixed\_filters, 470
- nppiFilterSobelHorizBorder\_32f\_C1R
  - fixed\_filters, 471
- nppiFilterSobelHorizBorder\_32f\_C3R
  - fixed\_filters, 471
- nppiFilterSobelHorizBorder\_32f\_C4R
  - fixed\_filters, 472
- nppiFilterSobelHorizBorder\_8s16s\_C1R
  - fixed\_filters, 472
- nppiFilterSobelHorizBorder\_8u16s\_C1R
  - fixed\_filters, 473
- nppiFilterSobelHorizBorder\_8u\_AC4R
  - fixed\_filters, 473
- nppiFilterSobelHorizBorder\_8u\_C1R
  - fixed\_filters, 474
- nppiFilterSobelHorizBorder\_8u\_C3R
  - fixed\_filters, 474
- nppiFilterSobelHorizBorder\_8u\_C4R
  - fixed\_filters, 474
- nppiFilterSobelHorizMask\_32f\_C1R
  - fixed\_filters, 475
- nppiFilterSobelHorizMaskBorder\_32f\_C1R
  - fixed\_filters, 475
- nppiFilterSobelHorizSecond\_32f\_C1R
  - fixed\_filters, 476
- nppiFilterSobelHorizSecond\_8s16s\_C1R
  - fixed\_filters, 476
- nppiFilterSobelHorizSecond\_8u16s\_C1R
  - fixed\_filters, 476
- nppiFilterSobelHorizSecondBorder\_32f\_C1R
  - fixed\_filters, 477
- nppiFilterSobelHorizSecondBorder\_8s16s\_C1R
  - fixed\_filters, 477
- nppiFilterSobelHorizSecondBorder\_8u16s\_C1R
  - fixed\_filters, 478
- nppiFilterSobelVert\_16s\_AC4R
  - fixed\_filters, 478
- nppiFilterSobelVert\_16s\_C1R
  - fixed\_filters, 479
- nppiFilterSobelVert\_16s\_C3R
  - fixed\_filters, 479
- nppiFilterSobelVert\_16s\_C4R
  - fixed\_filters, 479
- nppiFilterSobelVert\_32f\_AC4R
  - fixed\_filters, 480
- nppiFilterSobelVert\_32f\_C1R
  - fixed\_filters, 480
- nppiFilterSobelVert\_32f\_C3R
  - fixed\_filters, 480
- nppiFilterSobelVert\_32f\_C4R
  - fixed\_filters, 481
- nppiFilterSobelVert\_8s16s\_C1R
  - fixed\_filters, 481
- nppiFilterSobelVert\_8u16s\_C1R
  - fixed\_filters, 481
- nppiFilterSobelVert\_8u\_AC4R
  - fixed\_filters, 482
- nppiFilterSobelVert\_8u\_C1R
  - fixed\_filters, 482
- nppiFilterSobelVert\_8u\_C3R
  - fixed\_filters, 482
- nppiFilterSobelVert\_8u\_C4R
  - fixed\_filters, 483
- nppiFilterSobelVertBorder\_16s\_AC4R
  - fixed\_filters, 483
- nppiFilterSobelVertBorder\_16s\_C1R
  - fixed\_filters, 484
- nppiFilterSobelVertBorder\_16s\_C3R
  - fixed\_filters, 484
- nppiFilterSobelVertBorder\_16s\_C4R
  - fixed\_filters, 484
- nppiFilterSobelVertBorder\_32f\_AC4R
  - fixed\_filters, 485
- nppiFilterSobelVertBorder\_32f\_C1R
  - fixed\_filters, 485
- nppiFilterSobelVertBorder\_32f\_C3R
  - fixed\_filters, 486
- nppiFilterSobelVertBorder\_32f\_C4R
  - fixed\_filters, 486
- nppiFilterSobelVertBorder\_8s16s\_C1R
  - fixed\_filters, 487
- nppiFilterSobelVertBorder\_8u16s\_C1R
  - fixed\_filters, 487

- fixed\_filters, 487
- nppiFilterSobelVertBorder\_8u\_AC4R
  - fixed\_filters, 488
- nppiFilterSobelVertBorder\_8u\_C1R
  - fixed\_filters, 488
- nppiFilterSobelVertBorder\_8u\_C3R
  - fixed\_filters, 488
- nppiFilterSobelVertBorder\_8u\_C4R
  - fixed\_filters, 489
- nppiFilterSobelVertMask\_32f\_C1R
  - fixed\_filters, 489
- nppiFilterSobelVertMaskBorder\_32f\_C1R
  - fixed\_filters, 490
- nppiFilterSobelVertSecond\_32f\_C1R
  - fixed\_filters, 490
- nppiFilterSobelVertSecond\_8s16s\_C1R
  - fixed\_filters, 491
- nppiFilterSobelVertSecond\_8u16s\_C1R
  - fixed\_filters, 491
- nppiFilterSobelVertSecondBorder\_32f\_C1R
  - fixed\_filters, 491
- nppiFilterSobelVertSecondBorder\_8s16s\_C1R
  - fixed\_filters, 492
- nppiFilterSobelVertSecondBorder\_8u16s\_C1R
  - fixed\_filters, 492
- nppiFilterUnsharpBorder\_16s\_AC4R
  - fixed\_filters, 493
- nppiFilterUnsharpBorder\_16s\_C1R
  - fixed\_filters, 493
- nppiFilterUnsharpBorder\_16s\_C3R
  - fixed\_filters, 494
- nppiFilterUnsharpBorder\_16s\_C4R
  - fixed\_filters, 495
- nppiFilterUnsharpBorder\_16u\_AC4R
  - fixed\_filters, 495
- nppiFilterUnsharpBorder\_16u\_C1R
  - fixed\_filters, 496
- nppiFilterUnsharpBorder\_16u\_C3R
  - fixed\_filters, 496
- nppiFilterUnsharpBorder\_16u\_C4R
  - fixed\_filters, 497
- nppiFilterUnsharpBorder\_32f\_AC4R
  - fixed\_filters, 497
- nppiFilterUnsharpBorder\_32f\_C1R
  - fixed\_filters, 498
- nppiFilterUnsharpBorder\_32f\_C3R
  - fixed\_filters, 499
- nppiFilterUnsharpBorder\_32f\_C4R
  - fixed\_filters, 499
- nppiFilterUnsharpBorder\_8u\_AC4R
  - fixed\_filters, 500
- nppiFilterUnsharpBorder\_8u\_C1R
  - fixed\_filters, 500
- nppiFilterUnsharpBorder\_8u\_C3R
  - fixed\_filters, 501
- nppiFilterUnsharpBorder\_8u\_C4R
  - fixed\_filters, 501
- nppiFilterUnsharpGetBufferSize\_16s\_AC4R
  - fixed\_filters, 502
- nppiFilterUnsharpGetBufferSize\_16s\_C1R
  - fixed\_filters, 502
- nppiFilterUnsharpGetBufferSize\_16s\_C3R
  - fixed\_filters, 503
- nppiFilterUnsharpGetBufferSize\_16s\_C4R
  - fixed\_filters, 503
- nppiFilterUnsharpGetBufferSize\_16u\_AC4R
  - fixed\_filters, 503
- nppiFilterUnsharpGetBufferSize\_16u\_C1R
  - fixed\_filters, 503
- nppiFilterUnsharpGetBufferSize\_16u\_C3R
  - fixed\_filters, 504
- nppiFilterUnsharpGetBufferSize\_16u\_C4R
  - fixed\_filters, 504
- nppiFilterUnsharpGetBufferSize\_32f\_AC4R
  - fixed\_filters, 504
- nppiFilterUnsharpGetBufferSize\_32f\_C1R
  - fixed\_filters, 505
- nppiFilterUnsharpGetBufferSize\_32f\_C3R
  - fixed\_filters, 505
- nppiFilterUnsharpGetBufferSize\_32f\_C4R
  - fixed\_filters, 505
- nppiFilterUnsharpGetBufferSize\_8u\_AC4R
  - fixed\_filters, 505
- nppiFilterUnsharpGetBufferSize\_8u\_C1R
  - fixed\_filters, 506
- nppiFilterUnsharpGetBufferSize\_8u\_C3R
  - fixed\_filters, 506
- nppiFilterUnsharpGetBufferSize\_8u\_C4R
  - fixed\_filters, 506
- nppiGetFilterGaussPyramidLayerDownBorderDstROI
  - fixed\_filters, 507
- nppiGetFilterGaussPyramidLayerUpBorderDstROI
  - fixed\_filters, 507
- nppiGradientVectorPrewittBorder\_16s32f\_C1R
  - fixed\_filters, 508
- nppiGradientVectorPrewittBorder\_16s32f\_C3C1R
  - fixed\_filters, 508
- nppiGradientVectorPrewittBorder\_16u32f\_C1R
  - fixed\_filters, 509
- nppiGradientVectorPrewittBorder\_16u32f\_C3C1R
  - fixed\_filters, 510
- nppiGradientVectorPrewittBorder\_32f\_C1R
  - fixed\_filters, 510
- nppiGradientVectorPrewittBorder\_32f\_C3C1R
  - fixed\_filters, 511
- nppiGradientVectorPrewittBorder\_8u16s\_C1R
  - fixed\_filters, 512
- nppiGradientVectorPrewittBorder\_8u16s\_C3C1R
  - fixed\_filters, 512

fixed\_filters, 512  
nppiGradientVectorScharrBorder\_16s32f\_C1R  
    fixed\_filters, 513  
nppiGradientVectorScharrBorder\_16s32f\_C3C1R  
    fixed\_filters, 514  
nppiGradientVectorScharrBorder\_16u32f\_C1R  
    fixed\_filters, 514  
nppiGradientVectorScharrBorder\_16u32f\_C3C1R  
    fixed\_filters, 515  
nppiGradientVectorScharrBorder\_32f\_C1R  
    fixed\_filters, 516  
nppiGradientVectorScharrBorder\_32f\_C3C1R  
    fixed\_filters, 516  
nppiGradientVectorScharrBorder\_8u16s\_C1R  
    fixed\_filters, 517  
nppiGradientVectorScharrBorder\_8u16s\_C3C1R  
    fixed\_filters, 518  
nppiGradientVectorSobelBorder\_16s32f\_C1R  
    fixed\_filters, 518  
nppiGradientVectorSobelBorder\_16s32f\_C3C1R  
    fixed\_filters, 519  
nppiGradientVectorSobelBorder\_16u32f\_C1R  
    fixed\_filters, 520  
nppiGradientVectorSobelBorder\_16u32f\_C3C1R  
    fixed\_filters, 520  
nppiGradientVectorSobelBorder\_32f\_C1R  
    fixed\_filters, 521  
nppiGradientVectorSobelBorder\_32f\_C3C1R  
    fixed\_filters, 522  
nppiGradientVectorSobelBorder\_8u16s\_C1R  
    fixed\_filters, 522  
nppiGradientVectorSobelBorder\_8u16s\_C3C1R  
    fixed\_filters, 523  
NppiHaarBuffer, 529  
    haarBuffer, 529  
    haarBufferSize, 529  
NppiHaarClassifier\_32f, 530  
    classifiers, 530  
    classifierSize, 530  
    classifierStep, 530  
    counterDevice, 530  
    numClassifiers, 530  
NppiHuffmanTableType  
    typedefs\_npp, 40  
NppiInterpolationMode  
    typedefs\_npp, 41  
NppiMaskSize  
    typedefs\_npp, 41  
NppiNorm  
    typedefs\_npp, 41  
nppiNormInf  
    typedefs\_npp, 42  
nppiNormL1  
    typedefs\_npp, 42  
nppiNormL2  
    typedefs\_npp, 42  
NppiPoint, 531  
    x, 531  
    y, 531  
NppiRect, 532  
    height, 532  
    width, 532  
    x, 532  
    y, 532  
NppiSize, 533  
    height, 533  
    width, 533  
nppiSumWindowColumn\_16s32f\_C1R  
    image\_1D\_window\_sum, 124  
nppiSumWindowColumn\_16s32f\_C3R  
    image\_1D\_window\_sum, 125  
nppiSumWindowColumn\_16s32f\_C4R  
    image\_1D\_window\_sum, 125  
nppiSumWindowColumn\_16u32f\_C1R  
    image\_1D\_window\_sum, 125  
nppiSumWindowColumn\_16u32f\_C3R  
    image\_1D\_window\_sum, 126  
nppiSumWindowColumn\_16u32f\_C4R  
    image\_1D\_window\_sum, 126  
nppiSumWindowColumn\_8u32f\_C1R  
    image\_1D\_window\_sum, 127  
nppiSumWindowColumn\_8u32f\_C3R  
    image\_1D\_window\_sum, 127  
nppiSumWindowColumn\_8u32f\_C4R  
    image\_1D\_window\_sum, 128  
nppiSumWindowColumnBorder\_16s32f\_C1R  
    image\_1D\_window\_sum\_border, 136  
nppiSumWindowColumnBorder\_16s32f\_C3R  
    image\_1D\_window\_sum\_border, 136  
nppiSumWindowColumnBorder\_16s32f\_C4R  
    image\_1D\_window\_sum\_border, 137  
nppiSumWindowColumnBorder\_16u32f\_C1R  
    image\_1D\_window\_sum\_border, 137  
nppiSumWindowColumnBorder\_16u32f\_C3R  
    image\_1D\_window\_sum\_border, 138  
nppiSumWindowColumnBorder\_16u32f\_C4R  
    image\_1D\_window\_sum\_border, 138  
nppiSumWindowColumnBorder\_8u32f\_C1R  
    image\_1D\_window\_sum\_border, 139  
nppiSumWindowColumnBorder\_8u32f\_C3R  
    image\_1D\_window\_sum\_border, 140  
nppiSumWindowColumnBorder\_8u32f\_C4R  
    image\_1D\_window\_sum\_border, 140  
nppiSumWindowRow\_16s32f\_C1R  
    image\_1D\_window\_sum, 128  
nppiSumWindowRow\_16s32f\_C3R  
    image\_1D\_window\_sum, 129  
nppiSumWindowRow\_16s32f\_C4R

image\_1D\_window\_sum, 129  
 nppiSumWindowRow\_16u32f\_C1R  
     image\_1D\_window\_sum, 130  
 nppiSumWindowRow\_16u32f\_C3R  
     image\_1D\_window\_sum, 130  
 nppiSumWindowRow\_16u32f\_C4R  
     image\_1D\_window\_sum, 131  
 nppiSumWindowRow\_8u32f\_C1R  
     image\_1D\_window\_sum, 131  
 nppiSumWindowRow\_8u32f\_C3R  
     image\_1D\_window\_sum, 132  
 nppiSumWindowRow\_8u32f\_C4R  
     image\_1D\_window\_sum, 132  
 nppiSumWindowRowBorder\_16s32f\_C1R  
     image\_1D\_window\_sum\_border, 141  
 nppiSumWindowRowBorder\_16s32f\_C3R  
     image\_1D\_window\_sum\_border, 141  
 nppiSumWindowRowBorder\_16s32f\_C4R  
     image\_1D\_window\_sum\_border, 142  
 nppiSumWindowRowBorder\_16u32f\_C1R  
     image\_1D\_window\_sum\_border, 142  
 nppiSumWindowRowBorder\_16u32f\_C3R  
     image\_1D\_window\_sum\_border, 143  
 nppiSumWindowRowBorder\_16u32f\_C4R  
     image\_1D\_window\_sum\_border, 144  
 nppiSumWindowRowBorder\_8u32f\_C1R  
     image\_1D\_window\_sum\_border, 144  
 nppiSumWindowRowBorder\_8u32f\_C3R  
     image\_1D\_window\_sum\_border, 145  
 nppiSumWindowRowBorder\_8u32f\_C4R  
     image\_1D\_window\_sum\_border, 145  
 NppLibraryVersion, 534  
     build, 534  
     major, 534  
     minor, 534  
 NppRoundMode  
     typedefs\_npp, 42  
 nppSetStream  
     core\_npp, 29  
 NppStatus  
     typedefs\_npp, 42  
 NppsZCType  
     typedefs\_npp, 44  
 nppZCC  
     typedefs\_npp, 45  
 nppZCR  
     typedefs\_npp, 45  
 nppZCXor  
     typedefs\_npp, 45  
 numClassifiers  
     NppHaarClassifier\_32f, 530  
 Rank Filters, 224  
 re

NPP\_ALIGN\_16, 526  
 NPP\_ALIGN\_8, 527, 528  
 typedefs\_npp  
     NPP\_AFFINE\_QUAD\_INCORRECT\_-  
         WARNING, 44  
     NPP\_ALG\_HINT\_ACCURATE, 39  
     NPP\_ALG\_HINT\_FAST, 39  
     NPP\_ALG\_HINT\_NONE, 39  
     NPP\_ALIGNMENT\_ERROR, 43  
     NPP\_ANCHOR\_ERROR, 43  
     NPP\_BAD\_ARGUMENT\_ERROR, 44  
     NPP\_BORDER\_CONSTANT, 40  
     NPP\_BORDER\_MIRROR, 40  
     NPP\_BORDER\_NONE, 40  
     NPP\_BORDER\_REPLICATE, 40  
     NPP\_BORDER\_UNDEFINED, 40  
     NPP\_BORDER\_WRAP, 40  
     NPP\_BOTH\_AXIS, 40  
     NPP\_CHANNEL\_ERROR, 43  
     NPP\_CHANNEL\_ORDER\_ERROR, 43  
     NPP\_CMP\_EQ, 39  
     NPP\_CMP\_GREATER, 39  
     NPP\_CMP\_GREATER\_EQ, 39  
     NPP\_CMP\_LESS, 38  
     NPP\_CMP\_LESS\_EQ, 38  
     NPP\_COEFFICIENT\_ERROR, 43  
     NPP\_COI\_ERROR, 43  
     NPP\_CONTEXT\_MATCH\_ERROR, 44  
     NPP\_CORRUPTED\_DATA\_ERROR, 43  
     NPP\_CUDA\_1\_0, 39  
     NPP\_CUDA\_1\_1, 39  
     NPP\_CUDA\_1\_2, 39  
     NPP\_CUDA\_1\_3, 39  
     NPP\_CUDA\_2\_0, 39  
     NPP\_CUDA\_2\_1, 39  
     NPP\_CUDA\_3\_0, 39  
     NPP\_CUDA\_3\_2, 39  
     NPP\_CUDA\_3\_5, 39  
     NPP\_CUDA\_3\_7, 39  
     NPP\_CUDA\_5\_0, 39  
     NPP\_CUDA\_5\_2, 39  
     NPP\_CUDA\_5\_3, 39  
     NPP\_CUDA\_6\_0, 39  
     NPP\_CUDA\_KERNEL\_EXECUTION\_-  
         ERROR, 43  
     NPP\_CUDA\_NOT\_CAPABLE, 39  
     NPP\_CUDA\_UNKNOWN\_VERSION, 39  
     NPP\_DATA\_TYPE\_ERROR, 44  
     NPP\_DIVIDE\_BY\_ZERO\_ERROR, 44  
     NPP\_DIVIDE\_BY\_ZERO\_WARNING, 44  
     NPP\_DIVISOR\_ERROR, 43  
     NPP\_DOUBLE\_SIZE\_WARNING, 44  
     NPP\_ERROR, 44

NPP\_ERROR\_RESERVED, 44  
NPP\_FFT\_FLAG\_ERROR, 44  
NPP\_FFT\_ORDER\_ERROR, 44  
NPP\_FILTER\_SCHARR, 40  
NPP\_FILTER\_SOBEL, 40  
NPP\_HAAR\_CLASSIFIER\_PIXEL\_-  
MATCH\_ERROR, 43  
NPP\_HISTOGRAM\_NUMBER\_OF\_-  
LEVELS\_ERROR, 43  
NPP\_HORIZONTAL\_AXIS, 40  
NPP\_INTERPOLATION\_ERROR, 44  
NPP\_INVALID\_DEVICE\_POINTER\_-  
ERROR, 43  
NPP\_INVALID\_HOST\_POINTER\_ERROR,  
43  
NPP\_LUT\_NUMBER\_OF\_LEVELS\_-  
ERROR, 43  
NPP\_LUT\_PALETTE\_BITSIZE\_ERROR, 43  
NPP\_MASK\_SIZE\_11\_X\_11, 41  
NPP\_MASK\_SIZE\_13\_X\_13, 41  
NPP\_MASK\_SIZE\_15\_X\_15, 41  
NPP\_MASK\_SIZE\_1\_X\_3, 41  
NPP\_MASK\_SIZE\_1\_X\_5, 41  
NPP\_MASK\_SIZE\_3\_X\_1, 41  
NPP\_MASK\_SIZE\_3\_X\_3, 41  
NPP\_MASK\_SIZE\_5\_X\_1, 41  
NPP\_MASK\_SIZE\_5\_X\_5, 41  
NPP\_MASK\_SIZE\_7\_X\_7, 41  
NPP\_MASK\_SIZE\_9\_X\_9, 41  
NPP\_MASK\_SIZE\_ERROR, 43  
NPP\_MEMCPY\_ERROR, 43  
NPP\_MEMFREE\_ERROR, 43  
NPP\_MEMORY\_ALLOCATION\_ERR, 44  
NPP\_MEMSET\_ERROR, 43  
NPP\_MIRROR\_FLIP\_ERROR, 44  
NPP\_MISALIGNED\_DST\_ROI\_WARNING,  
44  
NPP\_MOMENT\_00\_ZERO\_ERROR, 44  
NPP\_NO\_ERROR, 44  
NPP\_NO\_MEMORY\_ERROR, 44  
NPP\_NO\_OPERATION\_WARNING, 44  
NPP\_NOT\_EVEN\_STEP\_ERROR, 43  
NPP\_NOT\_IMPLEMENTED\_ERROR, 44  
NPP\_NOT\_SUFFICIENT\_COMPUTE\_-  
CAPABILITY, 43  
NPP\_NOT\_SUPPORTED\_MODE\_ERROR,  
43  
NPP\_NULL\_POINTER\_ERROR, 44  
NPP\_NUMBER\_OF\_CHANNELS\_ERROR,  
43  
NPP\_OUT\_OF\_RANGE\_ERROR, 44  
NPP\_OVERFLOW\_ERROR, 43  
NPP\_QUADRANGLE\_ERROR, 43  
NPP\_QUALITY\_INDEX\_ERROR, 43  
NPP\_RANGE\_ERROR, 44  
NPP\_RECTANGLE\_ERROR, 43  
NPP\_RESIZE\_FACTOR\_ERROR, 44  
NPP\_RESIZE\_NO\_OPERATION\_ERROR,  
43  
NPP\_RND\_FINANCIAL, 42  
NPP\_RND\_NEAR, 42  
NPP\_RND\_ZERO, 42  
NPP\_ROUND\_MODE\_NOT\_-  
SUPPORTED\_ERROR, 43  
NPP\_ROUND\_NEAREST\_TIES\_AWAY\_-  
FROM\_ZERO, 42  
NPP\_ROUND\_NEAREST\_TIES\_TO\_EVEN,  
42  
NPP\_ROUND\_TOWARD\_ZERO, 42  
NPP\_SCALE\_RANGE\_ERROR, 44  
NPP\_SIZE\_ERROR, 44  
NPP\_STEP\_ERROR, 44  
NPP\_STRIDE\_ERROR, 43  
NPP\_SUCCESS, 44  
NPP\_TEXTURE\_BIND\_ERROR, 43  
NPP\_THRESHOLD\_ERROR, 44  
NPP\_THRESHOLD\_NEGATIVE\_LEVEL\_-  
ERROR, 44  
NPP\_VERTICAL\_AXIS, 40  
NPP\_WRONG\_INTERSECTION\_QUAD\_-  
WARNING, 44  
NPP\_WRONG\_INTERSECTION\_ROI\_-  
ERROR, 43  
NPP\_WRONG\_INTERSECTION\_ROI\_-  
WARNING, 44  
NPP\_ZC\_MODE\_NOT\_SUPPORTED\_-  
ERROR, 43  
NPP\_ZERO\_MASK\_VALUE\_ERROR, 43  
NPPI\_BAYER\_BGGR, 40  
NPPI\_BAYER\_GBRG, 40  
NPPI\_BAYER\_GRBG, 40  
NPPI\_BAYER\_RGGB, 40  
NPPI\_INTER\_CUBIC, 41  
NPPI\_INTER\_CUBIC2P\_B05C03, 41  
NPPI\_INTER\_CUBIC2P\_BSPLINE, 41  
NPPI\_INTER\_CUBIC2P\_CATMULLROM,  
41  
NPPI\_INTER\_LANCZOS, 41  
NPPI\_INTER\_LANCZOS3\_ADVANCED, 41  
NPPI\_INTER\_LINEAR, 41  
NPPI\_INTER\_NN, 41  
NPPI\_INTER\_SUPER, 41  
NPPI\_INTER\_UNDEFINED, 41  
NPPI\_OP\_ALPHA\_ATOP, 39  
NPPI\_OP\_ALPHA\_ATOP\_PREMUL, 40  
NPPI\_OP\_ALPHA\_IN, 39  
NPPI\_OP\_ALPHA\_IN\_PREMUL, 40  
NPPI\_OP\_ALPHA\_OUT, 39

NPPI\_OP\_ALPHA\_OUT\_PREMUL, 40  
NPPI\_OP\_ALPHA\_OVER, 39  
NPPI\_OP\_ALPHA\_OVER\_PREMUL, 40  
NPPI\_OP\_ALPHA\_PLUS, 39  
NPPI\_OP\_ALPHA\_PLUS\_PREMUL, 40  
NPPI\_OP\_ALPHA\_PREMUL, 40  
NPPI\_OP\_ALPHA\_XOR, 39  
NPPI\_OP\_ALPHA\_XOR\_PREMUL, 40  
NPPI\_SMOOTH\_EDGE, 41  
nppiACTable, 41  
nppiDCTable, 41  
nppiNormInf, 42  
nppiNormL1, 42  
nppiNormL2, 42  
nppZCC, 45  
nppZCR, 45  
nppZCXor, 45  
typedefs\_npp  
    NPP\_MAX\_16S, 37  
    NPP\_MAX\_16U, 37  
    NPP\_MAX\_32S, 37  
    NPP\_MAX\_32U, 37  
    NPP\_MAX\_64S, 37  
    NPP\_MAX\_64U, 37  
    NPP\_MAX\_8S, 37  
    NPP\_MAX\_8U, 37  
    NPP\_MAXABS\_32F, 37  
    NPP\_MAXABS\_64F, 37  
    NPP\_MIN\_16S, 37  
    NPP\_MIN\_16U, 38  
    NPP\_MIN\_32S, 38  
    NPP\_MIN\_32U, 38  
    NPP\_MIN\_64S, 38  
    NPP\_MIN\_64U, 38  
    NPP\_MIN\_8S, 38  
    NPP\_MIN\_8U, 38  
    NPP\_MINABS\_32F, 38  
    NPP\_MINABS\_64F, 38  
    NppCmpOp, 38  
    NppGpuComputeCapability, 39  
    NppHintAlgorithm, 39  
    NppiAlphaOp, 39  
    NppiAxis, 40  
    NppiBayerGridPosition, 40  
    NppiBorderType, 40  
    NppiDifferentialKernel, 40  
    NppiHuffmanTableType, 40  
    NppiInterpolationMode, 41  
    NppiMaskSize, 41  
    NppiNorm, 41  
    NppRoundMode, 42  
    NppStatus, 42  
    NppsZCType, 44  
width  
    NppiRect, 532  
    NppiSize, 533  
x  
    NppiPoint, 531  
    NppiRect, 532  
y  
    NppiPoint, 531  
    NppiRect, 532