
vs-mask

Release 0.5.1

IEW

Dec 04, 2022

CONTENTS

1	API Reference	1
1.1	Edge Masks	1
1.2	Utility functions	122
2	License	127
2.1	Conditions for Contributors	127
	Python Module Index	129
	Index	131

API REFERENCE

Various masking tools for Vapoursynth

1.1 Edge Masks

Edge and ridge detection submodule

class `vsmask.edge.EdgeDetect`

Bases: ABC

Abstract edge detection interface.

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.MatrixEdgeDetect

Bases: [EdgeDetect](#), ABC

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SingleMatrix

Bases: [MatrixEdgeDetect](#), ABC

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | *NoReturn*

Returns

Mask clip

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

class *vsmask.edge.EuclidianDistance*

Bases: *MatrixEdgeDetect*, *ABC*

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding

- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.Max

Bases: [MatrixEdgeDetect](#), ABC

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.RidgeDetectBases: *MatrixEdgeDetect***ridgemask**(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.Matrix1DBases: *EdgeDetect*, ABC**edgemask**(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.TEdgeBases: *EuclidianDistance*, *Matrix1D*

(TEdgeMasktype=2) Avisynth plugin.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.TEdgeTedgemask

Bases: [Matrix1D](#), [EdgeDetect](#)

(tedgemask.TEdgeMask(threshold=0.0, type=2)) Vapoursynth plugin.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayBases: [EuclidianDistance](#), [Matrix1D](#)**edgemask**(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class *vsmask.edge.SavitzkyGolayDeriv1Quad5*

Bases: *SavitzkyGolay*

Savitzky-Golay first quadratic derivative operator of size 5

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quad7

Bases: [SavitzkyGolay](#)

Savitzky-Golay first quadratic derivative operator of size 7

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quad9Bases: [SavitzkyGolay](#)

Savitzky-Golay first quadratic derivative operator of size 9

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quad11Bases: [SavitzkyGolay](#)

Savitzky-Golay first quadratic derivative operator of size 11

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip

- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quad13

Bases: [SavitzkyGolay](#)

Savitzky-Golay first quadratic derivative operator of size 13

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quad15Bases: [SavitzkyGolay](#)

Savitzky-Golay first quadratic derivative operator of size 15

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0

- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quad17

Bases: [SavitzkyGolay](#)

Savitzky-Golay first quadratic derivative operator of size 17

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quad19Bases: [SavitzkyGolay](#)

Savitzky-Golay first quadratic derivative operator of size 19

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quad21Bases: [SavitzkyGolay](#)

Savitzky-Golay first quadratic derivative operator of size 21

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class *vsmask.edge.SavitzkyGolayDeriv1Quad23*

Bases: [*SavitzkyGolay*](#)

Savitzky-Golay first quadratic derivative operator of size 23

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quad25Bases: [SavitzkyGolay](#)

Savitzky-Golay first quadratic derivative operator of size 25

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | *NoReturn*

Returns

Mask clip

class `vsmask.edge.SavitzkyGolayDeriv1Cubic5`

Bases: [*SavitzkyGolay*](#)

Savitzky-Golay first cubic/quartic operator of size 5

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Cubic7Bases: [SavitzkyGolay](#)

Savitzky-Golay first cubic/quartic derivative operator of size 7

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Cubic9Bases: [SavitzkyGolay](#)

Savitzky-Golay first cubic/quartic operator of size 9

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Cubic11

Bases: [SavitzkyGolay](#)

Savitzky-Golay first cubic/quartic operator of size 11

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Cubic13

Bases: SavitzkyGolayNormalise

Savitzky-Golay first cubic/quartic operator of size 13

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Cubic15

Bases: SavitzkyGolayNormalise

Savitzky-Golay first cubic/quartic operator of size 15

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Cubic17Bases: [SavitzkyGolay](#)

Savitzky-Golay first cubic/quartic operator of size 17

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Cubic19

Bases: SavitzkyGolayNormalise

Savitzky-Golay first cubic/quartic operator of size 19

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Cubic21

Bases: SavitzkyGolayNormalise

Savitzky-Golay first cubic/quartic operator of size 21

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip

- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Cubic23

Bases: SavitzkyGolayNormalise

Savitzky-Golay first cubic/quartic operator of size 23

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type*vs.VideoNode* | NoReturn**Returns**

Mask clip

class *vs*.mask.edge.SavitzkyGolayDeriv1Cubic25Bases: *SavitzkyGolayNormalise*

Savitzky-Golay first cubic/quartic operator of size 25

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type*vs.VideoNode***Returns**

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0

- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quint7Bases: [SavitzkyGolay](#)

Savitzky-Golay first quintic/sextic derivative operator of size 7

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quint9

Bases: SavitzkyGolayNormalise

Savitzky-Golay first quintic/sextic derivative operator of size 9

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quint11

Bases: SavitzkyGolayNormalise

Savitzky-Golay first quintic/sextic derivative operator of size 11

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.**SavitzkyGolayDeriv1Quint13**

Bases: SavitzkyGolayNormalise

Savitzky-Golay first quintic/sextic derivative operator of size 13

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quint15

Bases: SavitzkyGolayNormalise

Savitzky-Golay first quintic/sextic derivative operator of size 15

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quint17

Bases: SavitzkyGolayNormalise

Savitzky-Golay first quintic/sextic derivative operator of size 17

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quint19

Bases: SavitzkyGolayNormalise

Savitzky-Golay first quintic/sextic derivative operator of size 19

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quint21

Bases: SavitzkyGolayNormalise

Savitzky-Golay first quintic/sextic derivative operator of size 21

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quint23

Bases: SavitzkyGolayNormalise

Savitzky-Golay first quintic/sextic derivative operator of size 23

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv1Quint25

Bases: SavitzkyGolayNormalise

Savitzky-Golay first quintic/sextic derivative operator of size 25

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class *vsmask.edge.SavitzkyGolayDeriv2Quad5*

Bases: *SavitzkyGolay*

Savitzky-Golay second quadratic/cubic derivative operator of size 5

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quad7

Bases: [SavitzkyGolay](#)

Savitzky-Golay second quadratic/cubic derivative operator of size 7

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quad9Bases: [SavitzkyGolay](#)

Savitzky-Golay second quadratic/cubic derivative operator of size 9

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quad11Bases: [SavitzkyGolay](#)

Savitzky-Golay second quadratic/cubic derivative operator of size 11

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip

- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr=0.0*, *hthr=None*, *multi=1.0*, *clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quad13Bases: [SavitzkyGolay](#)

Savitzky-Golay second quadratic/cubic derivative operator of size 13

edgemask(*clip*, *lthr=0.0*, *hthr=None*, *multi=1.0*, *clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quad15Bases: [SavitzkyGolay](#)

Savitzky-Golay second quadratic/cubic derivative operator of size 15

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0

- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quad17

Bases: [SavitzkyGolay](#)

Savitzky-Golay second quadratic/cubic derivative operator of size 17

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quad19Bases: [SavitzkyGolay](#)

Savitzky-Golay second quadratic/cubic derivative operator of size 19

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quad21Bases: [SavitzkyGolay](#)

Savitzky-Golay second quadratic/cubic derivative operator of size 21

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class `vsmask.edge.SavitzkyGolayDeriv2Quad23`

Bases: [*SavitzkyGolay*](#)

Savitzky-Golay second quadratic/cubic derivative operator of size 23

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quad25Bases: [SavitzkyGolay](#)

Savitzky-Golay second quadratic/cubic derivative operator of size 25

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class `vs.mask.edge.SavitzkyGolayDeriv2Quart7`

Bases: [*SavitzkyGolay*](#)

Savitzky-Golay second quartic/quintic derivative operator of size 7

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quart9Bases: [SavitzkyGolay](#)

Savitzky-Golay second quartic/quintic derivative operator of size 9

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quart11Bases: [SavitzkyGolay](#)

Savitzky-Golay second quartic/quintic derivative operator of size 11

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quart13

Bases: SavitzkyGolayNormalise

Savitzky-Golay second quartic/quintic derivative operator of size 13

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quart15

Bases: SavitzkyGolayNormalise

Savitzky-Golay second quartic/quintic derivative operator of size 15

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class *vs.mask.edge.SavitzkyGolayDeriv2Quart17*

Bases: *SavitzkyGolayNormalise*

Savitzky-Golay second quartic/quintic derivative operator of size 17

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quart19

Bases: SavitzkyGolayNormalise

Savitzky-Golay second quartic/quintic derivative operator of size 19

edgemask(*clip*, *lthr=0.0*, *hthr=None*, *multi=1.0*, *clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr=0.0*, *hthr=None*, *multi=1.0*, *clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quart21

Bases: SavitzkyGolayNormalise

Savitzky-Golay second quartic/quintic derivative operator of size 21

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quart23

Bases: SavitzkyGolayNormalise

Savitzky-Golay second quartic/quintic derivative operator of size 23

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip

- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv2Quart25

Bases: SavitzkyGolayNormalise

Savitzky-Golay second quartic/quintic derivativeoperator of size 25

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Cub5Bases: [SavitzkyGolay](#)

Savitzky-Golay third cubic/quartic derivative operator of size 5

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0

- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Cub7Bases: [SavitzkyGolay](#)

Savitzky-Golay third cubic/quartic derivative operator of size 7

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Cub9Bases: [SavitzkyGolay](#)

Savitzky-Golay third cubic/quartic derivative operator of size 9

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Cub11Bases: [SavitzkyGolay](#)

Savitzky-Golay third cubic/quartic derivative operator of size 11

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Cub13Bases: [SavitzkyGolay](#)

Savitzky-Golay third cubic/quartic derivative operator of size 13

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Cub15Bases: [SavitzkyGolay](#)

Savitzky-Golay third cubic/quartic derivative operator of size 15

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Cub17Bases: [SavitzkyGolay](#)

Savitzky-Golay third cubic/quartic derivative operator of size 17

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Cub19Bases: [SavitzkyGolay](#)

Savitzky-Golay third cubic/quartic derivative operator of size 19

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Cub21Bases: [SavitzkyGolay](#)

Savitzky-Golay third cubic/quartic derivative operator of size 21

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Cub23

Bases: [SavitzkyGolay](#)

Savitzky-Golay third cubic/quartic derivative operator of size 23

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Cub25Bases: [SavitzkyGolay](#)

Savitzky-Golay third cubic/quartic derivative operator of size 25

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class *vsmask.edge.SavitzkyGolayDeriv3Quint7*

Bases: *SavitzkyGolay*

Savitzky-Golay third quintic/sexic derivative operator of size 7

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Quint9

Bases: [SavitzkyGolay](#)

Savitzky-Golay third quintic/sexic derivative operator of size 9

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Quint11Bases: [SavitzkyGolay](#)

Savitzky-Golay third quintic/sexic derivative operator of size 11

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Quint13Bases: [SavitzkyGolayNormalise](#)

Savitzky-Golay third quintic/sexic derivative operator of size 13

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip

- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr=0.0*, *hthr=None*, *multi=1.0*, *clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Quint15

Bases: SavitzkyGolayNormalise

Savitzky-Golay third quintic/sexic derivative operator of size 15

edgemask(*clip*, *lthr=0.0*, *hthr=None*, *multi=1.0*, *clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Quint17

Bases: SavitzkyGolayNormalise

Savitzky-Golay third quintic/sextic derivative operator of size 17

edgemark(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0

- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Quint19

Bases: SavitzkyGolayNormalise

Savitzky-Golay third quintic/sexic derivative operator of size 19

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Quint21

Bases: SavitzkyGolayNormalise

Savitzky-Golay third quintic/sexic derivative operator of size 21

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv3Quint23

Bases: SavitzkyGolayNormalise

Savitzky-Golay third quintic/sexic derivative operator of size 23

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class *vsmask.edge.SavitzkyGolayDeriv3Quint25*

Bases: *SavitzkyGolayNormalise*

Savitzky-Golay third quintic/sexic derivative operator of size 25

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv4Quart7Bases: [SavitzkyGolay](#)

Savitzky-Golay fourth quartic/quintic derivative operator of size 7

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class `vs.mask.edge.SavitzkyGolayDeriv4Quart9`

Bases: [*SavitzkyGolay*](#)

Savitzky-Golay fourth quartic/quintic derivative operator of size 9

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv4Quart11Bases: [SavitzkyGolay](#)

Savitzky-Golay fourth quartic/quintic derivative operator of size 11

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv4Quart13Bases: [SavitzkyGolay](#)

Savitzky-Golay fourth quartic/quintic derivative operator of size 13

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv4Quart15

Bases: [SavitzkyGolay](#)

Savitzky-Golay fourth quartic/quintic derivative operator of size 15

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv4Quart17Bases: [SavitzkyGolay](#)

Savitzky-Golay fourth quartic/quintic derivative operator of size 17

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class *vsmask.edge.SavitzkyGolayDeriv4Quart19*

Bases: *SavitzkyGolay*

Savitzky-Golay fourth quartic/quintic derivative operator of size 19

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv4Quart21Bases: [SavitzkyGolay](#)

Savitzky-Golay fourth quartic/quintic derivative operator of size 21

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv4Quart23

Bases: SavitzkyGolayNormalise

Savitzky-Golay fourth quartic/quintic derivative operator of size 23

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv4Quart25

Bases: SavitzkyGolayNormalise

Savitzky-Golay fourth quartic/quintic derivative operator of size 25

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip

- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv5Quint7Bases: [SavitzkyGolay](#)

Savitzky-Golay fifth quintic/sexic derivative operator of size 7

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type*vs.VideoNode* | NoReturn**Returns**

Mask clip

class *vs*.mask.edge.SavitzkyGolayDeriv5Quint9Bases: [*SavitzkyGolay*](#)

Savitzky-Golay fifth quintic/sextic derivative operator of size 9

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type*vs.VideoNode***Returns**

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0

- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv5Quint11Bases: [SavitzkyGolay](#)

Savitzky-Golay fifth quintic/sexic derivative operator of size 11

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv5Quint13Bases: [SavitzkyGolay](#)

Savitzky-Golay fifth quintic/sexic derivative operator of size 13

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv5Quint15Bases: [SavitzkyGolayNormalise](#)

Savitzky-Golay fifth quintic/sexic derivative operator of size 15

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.**SavitzkyGolayDeriv5Quint17**Bases: [SavitzkyGolay](#)

Savitzky-Golay fifth quintic/sexic derivative operator of size 17

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv5Quint19Bases: [SavitzkyGolay](#)

Savitzky-Golay fifth quintic/sexic derivative operator of size 19

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv5Quint21

Bases: SavitzkyGolayNormalise

Savitzky-Golay fifth quintic/sexic derivative operator of size 21

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv5Quint23Bases: [SavitzkyGolay](#)

Savitzky-Golay fifth quintic/sexic derivative operator of size 23

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SavitzkyGolayDeriv5Quint25Bases: [SavitzkyGolay](#)

Savitzky-Golay fifth quintic/sexic derivative operator of size 25

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.Matrix2x2

Bases: [EdgeDetect](#), ABC

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding

- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.Roberts

Bases: *RidgeDetect*, *EuclidianDistance*, *Matrix2x2*

Lawrence Roberts operator. 2x2 matrices computed in 3x3 matrices.

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

class *vs**mask*.*edge*.**Matrix3x3**

Bases: *EdgeDetect*, *ABC*

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding

- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.Laplacian1

Bases: [SingleMatrix](#), [Matrix3x3](#)

Pierre-Simon de Laplace operator 1st implementation.

edgemask(*clip*, *lthr=0.0*, *hthr=None*, *multi=1.0*, *clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr=0.0*, *hthr=None*, *multi=1.0*, *clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.Laplacian2Bases: [SingleMatrix](#), [Matrix3x3](#)

Pierre-Simon de Laplace operator 2nd implementation.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.Laplacian3Bases: [SingleMatrix](#), [Matrix3x3](#)

Pierre-Simon de Laplace operator 3rd implementation.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip

- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr=0.0*, *hthr=None*, *multi=1.0*, *clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.Laplacian4Bases: [SingleMatrix](#), [Matrix3x3](#)

Pierre-Simon de Laplace operator 4th implementation.

edgemask(*clip*, *lthr=0.0*, *hthr=None*, *multi=1.0*, *clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.KayyaliBases: *SingleMatrix*, *Matrix3x3*

Kayyali operator.

edgemark(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0

- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.Tritical

Bases: *RidgeDetect, EuclidianDistance, Matrix3x3*

Operator used in Tritical's original TCanny filter. Plain and simple orthogonal first order derivative.

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.TriticalTCannyBases: *Matrix3x3*, *EdgeDetect*

Operator used in Tritical's original TCanny filter. Plain and simple orthogonal first order derivative.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.CrossBases: *RidgeDetect*, *EuclidianDistance*, *Matrix3x3*

“HotDoG” Operator from AVS ExTools by Dogway. Plain and simple cross first order derivative.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

class *vsmask.edge.Prewitt*

Bases: *RidgeDetect, EuclidianDistance, Matrix3x3*

Judith M. S. Prewitt operator.

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.PrewittStdBases: [Matrix3x3](#), [EdgeDetect](#)

Judith M. S. Prewitt Vapoursynth plugin operator.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class `vsmask.edge.PrewittTCanny`

Bases: [*Matrix3x3*](#), [*EdgeDetect*](#)

Judith M. S. Prewitt TCanny plugin operator.

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SobelBases: *RidgeDetect*, *EuclidianDistance*, *Matrix3x3*

Sobel–Feldman operator.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.SobelStdBases: *Matrix3x3*, *EdgeDetect*

Sobel–Feldman Vapoursynth plugin operator.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.SobelTCanny

Bases: [Matrix3x3](#), [EdgeDetect](#)

Sobel–Feldman Vapoursynth plugin operator.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.ASobelBases: *Matrix3x3*, *EdgeDetect*

Modified Sobel–Feldman operator from AWarpSharp.

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class *vsmask.edge.Scharr*

Bases: *RidgeDetect*, *EuclidianDistance*, *Matrix3x3*

Original H. Scharr optimised operator which attempts to achieve the perfect rotational symmetry with coefficients 3 and 10.

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0

- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.RScharBases: *RidgeDetect, EuclidianDistance, Matrix3x3*

Refined H. Scharr operator to more accurately calculate 1st derivatives for a 3x3 kernel with coeffs 47 and 162.

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.ScharrTCannyBases: *Matrix3x3*, *EdgeDetect*

H. Scharr optimised TCanny Vapoursynth plugin operator.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.KroonBases: *RidgeDetect*, *EuclidianDistance*, *Matrix3x3*

Dirk-Jan Kroon operator.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.KroonTCannyBases: *Matrix3x3*, *EdgeDetect*

Dirk-Jan Kroon TCanny Vapoursynth plugin operator.

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.FreyChenG41Bases: [RidgeDetect](#), [EuclidianDistance](#), [Matrix3x3](#)

“Chen Frei” operator. 3x3 matrices from G41Fun.

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.FreyChenBases: [MatrixEdgeDetect](#)

Chen Frei operator. 3x3 matrices properly implemented.

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.Robinson3Bases: *Max, Matrix3x3*

Robinson compass operator level 3.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.Robinson5Bases: *Max, Matrix3x3*

Robinson compass operator level 5.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.TheToof

Bases: [Max](#), [Matrix3x3](#)

TheToof compass operator from SharpAAMCmod.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.KirschBases: *Max*, *Matrix3x3*

Russell Kirsch compass operator.

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class *vsmask.edge.KirschTCanny*

Bases: *Matrix3x3*, *EdgeDetect*

Russell Kirsch compass TCanny Vapoursynth plugin operator.

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class `vs.mask.edge.MinMax`(*radx=2, racy=0*)

Bases: [*EdgeDetect*](#)

Min/max mask with separate luma/chroma radii.

Parameters

- **radx** (*int*) – Luma radius
- **racy** (*int*) – Chroma radius

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.**Matrix5x5**Bases: [EdgeDetect](#), ABC**edgemask**(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.**ExLaplacian1**Bases: [SingleMatrix](#), [Matrix5x5](#)

Extended Pierre-Simon de Laplace operator, 1st implementation.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class `vsmask.edge.ExLaplacian2`

Bases: [*SingleMatrix*](#), [*Matrix5x5*](#)

Extended Pierre-Simon de Laplace operator, 2nd implementation.

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.ExLaplacian3Bases: [SingleMatrix](#), [Matrix5x5](#)

Extended Pierre-Simon de Laplace operator, 3rd implementation.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | *NoReturn*

Returns

Mask clip

class *vs*.mask.edge.**ExLaplacian4**

Bases: [*SingleMatrix*](#), [*Matrix5x5*](#)

Extended Pierre-Simon de Laplace operator, 4th implementation.

edgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip, lthr=0.0, hthr=None, multi=1.0, clamp=False*)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.LoGBases: *SingleMatrix*, *Matrix5x5*

Laplacian of Gaussian operator.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.ExPrewittBases: *RidgeDetect*, *EuclidianDistance*, *Matrix5x5*

Extended Judith M. S. Prewitt operator.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.ExSobel

Bases: [RidgeDetect](#), [EuclidianDistance](#), [Matrix5x5](#)

Extended Sobel–Feldman operator.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.FDoGBases: [RidgeDetect](#), [EuclidianDistance](#), [Matrix5x5](#)

Flow-based Difference of Gaussian

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple[float, float]* | *List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

class *vsmask.edge.FDoGTCanny*

Bases: *Matrix5x5*, *EdgeDetect*

Flow-based Difference of Gaussian TCanny Vapoursynth plugin.

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional[float]*) – High threshold. Anything above *hthr* will be set to the range *max*

- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.DoGBases: *EuclidianDistance*, *Matrix5x5*

Zero-cross (of the 2nd derivative) of a Difference of Gaussians

edgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with *lthr*, *hthr* and multiplied with *multi*.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below *lthr* will be set to 0
- **hthr** (*Optional*[*float*]) – High threshold. Anything above *hthr* will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool* | *Tuple*[*float*, *float*] | *List*[*Tuple*[*float*, *float*]]) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

class vsmask.edge.FaridBases: [RidgeDetect](#), [EuclidianDistance](#), [Matrix5x5](#)

Farid & Simoncelli operator.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip
- **lthr** (float) – Low threshold. Anything below lthr will be set to 0
- **hthr** (Optional[float]) – High threshold. Anything above hthr will be set to the range max
- **multi** (float) – Multiply all pixels by this before thresholding
- **clamp** (bool | Tuple[float, float] | List[Tuple[float, float]]) – Clamp to TV or full range if True or specified range (low, high)

Return type

vs.VideoNode

Returns

Mask clip

class vsmask.edge.ExKirschBases: [Max](#)

Extended Russell Kirsch compass operator. 5x5 matrices.

edgemask(clip, lthr=0.0, hthr=None, multi=1.0, clamp=False)

Makes edge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (vs.VideoNode) – Source clip

- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode

Returns

Mask clip

ridgemask(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Makes ridge mask based on convolution kernel. The resulting mask can be thresholded with lthr, hthr and multiplied with multi.

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – Low threshold. Anything below lthr will be set to 0
- **hthr** (*Optional[*float*]*) – High threshold. Anything above hthr will be set to the range max
- **multi** (*float*) – Multiply all pixels by this before thresholding
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

vs.VideoNode | NoReturn

Returns

Mask clip

vsmask.edge.get_all_edge_detects(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Returns all the EdgeDetect subclasses

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – See EdgeDetect.get_mask()
- **hthr** (*Optional[*float*]*) – See EdgeDetect.get_mask()
- **multi** (*float*) – See EdgeDetect.get_mask()
- **clamp** (*bool | Tuple[*float*, *float*] | List[Tuple[*float*, *float*]]*) – Clamp to TV or full range if True or specified range (*low*, *high*)

Return type

List[vs.VideoNode]

Returns

A list edge masks

vsmask.edge.get_all_ridge_detect(*clip*, *lthr*=0.0, *hthr*=None, *multi*=1.0, *clamp*=False)

Returns all the RidgeDetect subclasses

Parameters

- **clip** (*vs.VideoNode*) – Source clip
- **lthr** (*float*) – See `EdgeDetect.get_mask()`
- **hthr** (*Optional[float]*) – See `EdgeDetect.get_mask()`
- **multi** (*float*) – See `EdgeDetect.get_mask()`
- **clamp** (*bool | Tuple[float, float] | List[Tuple[float, float]]*) – Clamp to TV or full range if True or specified range (*low, high*)

Return type

List[vs.VideoNode]

Returns

A list edge masks

1.2 Utility functions

`vsmask.util.max_expr(n)`

Dynamic variable max string to be integrated in `std.Expr`.

Parameters

n (*int*) – Number of elements.

Return type

str

Returns

Expression

class `vsmask.util.XxpandMode(value)`

Expand/inpand mode

RECTANGLE = <object object>

Rectangular shape

ELLIPSE = <object object>

Elliptical shape

LOSANGE = <object object>

Diamond shape

`vsmask.util.morpho_transfo(clip, func, sw, sh=None, mode=XxpandMode.RECTANGLE, thr=None, planes=None)`

Calls a morphological function in order to grow or shrink a clip from the desired width and height.

Parameters

- **clip** (*vs.VideoNode*) – Source clip.
- **func** (*MorphoFunc*) – Morphological function.
- **sw** (*int*) – Growing/shrinking shape width.
- **sh** (*Optional[int]*) – Growing/shrinking shape height. If not specified, default to `sw`.
- **mode** (*XxpandMode*) – Shape form. Ellipses are combinations of rectangles and losanges and look more like octogons. Losanges are truncated (not scaled) when `sw` and `sh` are not equal.

- **thr** (*Optional[int]*) – Allows to limit how much pixels are changed. Output pixels will not become less than `input - threshold`. The default is no limit.
- **planes** (*int | Sequence[int] | None*) – Specifies which planes will be processed. Any unprocessed planes will be simply copied.

Return type

vs.VideoNode

Returns

Transformed clip

`vsmask.util.expand(clip, sw, sh=None, mode=XpandMode.RECTANGLE, thr=None, planes=None)`

Calls `std.Maximum` in order to grow each pixel with the largest value in its 3x3 neighbourhood from the desired width and height.

Parameters

- **clip** (*vs.VideoNode*) – Source clip.
- **sw** (*int*) – Growing shape width.
- **sh** (*Optional[int]*) – Growing shape height. If not specified, default to `sw`.
- **mode** (*XpandMode*) – Shape form. Ellipses are combinations of rectangles and losanges and look more like octogons. Losanges are truncated (not scaled) when `sw` and `sh` are not equal.
- **thr** (*Optional[int]*) – Allows to limit how much pixels are changed. Output pixels will not become less than `input - threshold`. The default is no limit.
- **planes** (*int | Sequence[int] | None*) – Specifies which planes will be processed. Any unprocessed planes will be simply copied.

Return type

vs.VideoNode

Returns

Transformed clip

`vsmask.util.inpand(clip, sw, sh=None, mode=XpandMode.RECTANGLE, thr=None, planes=None)`

Calls `std.Minimum` in order to shrink each pixel with the smallest value in its 3x3 neighbourhood from the desired width and height.

Parameters

- **clip** (*vs.VideoNode*) – Source clip.
- **sw** (*int*) – Shrinking shape width.
- **sh** (*Optional[int]*) – Shrinking shape height. If not specified, default to `sw`.
- **mode** (*XpandMode*) – Shape form. Ellipses are combinations of rectangles and losanges and look more like octogons. Losanges are truncated (not scaled) when `sw` and `sh` are not equal.
- **thr** (*Optional[int]*) – Allows to limit how much pixels are changed. Output pixels will not become less than `input - threshold`. The default is no limit.
- **planes** (*int | Sequence[int] | None*) – Specifies which planes will be processed. Any unprocessed planes will be simply copied.

Return type

vs.VideoNode

Returns

Transformed clip

`vsmask.util.max_planes(*clips, resizer=core.resize.Bilinear)`

Set max value of all the planes of all the clips

Output clip format is a GRAY clip with the same bitdepth as the first clip

Parameters

- **clips** (VideoNode) – Source clips.
- **resizer** (ZResizer) – Resizer used for converting the clips to the same width, height and to 444.

Return type

VideoNode

Returns

Maxed clip

`vsmask.util.region_mask(clip, left=0, right=0, top=0, bottom=0)`

Alias for [region_rel_mask\(\)](#)

Region relatively the clip with the desired numbers of pixels

Parameters

- **clip** (VideoNode) – Source clip
- **left** (int) – Left side
- **right** (int) – Right side
- **top** (int) – Top side
- **bottom** (int) – Bottom side

Return type

VideoNode

Returns

Regionned mask

`vsmask.util.region_rel_mask(clip, left=0, right=0, top=0, bottom=0)`

Region relatively the clip with the desired numbers of pixels

Parameters

- **clip** (VideoNode) – Source clip
- **left** (int) – Left side
- **right** (int) – Right side
- **top** (int) – Top side
- **bottom** (int) – Bottom side

Return type

VideoNode

Returns

Regionned mask

`vsmask.util.region_abs_mask(clip, width, height, left=0, top=0)`

Region the clip with absolute desired dimensions

Parameters

- **clip** (VideoNode) – Source clip
- **width** (int) – Width of the box
- **height** (int) – Height of the box
- **left** (int) – Shift from the left, AKA x parameter
- **top** (int) – Shift from the top, AKA y parameter

Return type

VideoNode

Returns

Regionned mask

LICENSE

vs-mask is under the MIT License. See the LICENSE file.

2.1 Conditions for Contributors

By contributing to this software project, you are agreeing to the following terms and conditions for your contributions: First, you agree your contributions are submitted under the MIT license. Second, you represent you are authorized to make the contributions and grant the license. If your employer has rights to intellectual property that includes your contributions, you represent that you have received permission to make contributions and grant the required license on behalf of that employer.

PYTHON MODULE INDEX

V

`vsmask`, [1](#)
`vsmask.edge`, [1](#)
`vsmask.util`, [122](#)

A

ASobel (class in *vsmask.edge*), 99

C

Cross (class in *vsmask.edge*), 93

D

DoG (class in *vsmask.edge*), 119

E

EdgeDetect (class in *vsmask.edge*), 1
 edgemask() (*vsmask.edge.ASobel* method), 99
 edgemask() (*vsmask.edge.Cross* method), 93
 edgemask() (*vsmask.edge.DoG* method), 119
 edgemask() (*vsmask.edge.EdgeDetect* method), 1
 edgemask() (*vsmask.edge.EuclidianDistance* method), 4
 edgemask() (*vsmask.edge.ExKirsch* method), 120
 edgemask() (*vsmask.edge.ExLaplacian1* method), 111
 edgemask() (*vsmask.edge.ExLaplacian2* method), 112
 edgemask() (*vsmask.edge.ExLaplacian3* method), 113
 edgemask() (*vsmask.edge.ExLaplacian4* method), 114
 edgemask() (*vsmask.edge.ExPrewitt* method), 115
 edgemask() (*vsmask.edge.ExSobel* method), 116
 edgemask() (*vsmask.edge.Farid* method), 120
 edgemask() (*vsmask.edge.FDoG* method), 117
 edgemask() (*vsmask.edge.FDoGTCanny* method), 118
 edgemask() (*vsmask.edge.FreyChen* method), 105
 edgemask() (*vsmask.edge.FreyChenG41* method), 104
 edgemask() (*vsmask.edge.Kayyali* method), 91
 edgemask() (*vsmask.edge.Kirsch* method), 108
 edgemask() (*vsmask.edge.KirschTCanny* method), 109
 edgemask() (*vsmask.edge.Kroon* method), 102
 edgemask() (*vsmask.edge.KroonTCanny* method), 103
 edgemask() (*vsmask.edge.Laplacian1* method), 88
 edgemask() (*vsmask.edge.Laplacian2* method), 89
 edgemask() (*vsmask.edge.Laplacian3* method), 89
 edgemask() (*vsmask.edge.Laplacian4* method), 90
 edgemask() (*vsmask.edge.LoG* method), 115
 edgemask() (*vsmask.edge.Matrix1D* method), 6
 edgemask() (*vsmask.edge.Matrix2x2* method), 85
 edgemask() (*vsmask.edge.Matrix3x3* method), 87

edgemask() (*vsmask.edge.Matrix5x5* method), 111
 edgemask() (*vsmask.edge.MatrixEdgeDetect* method), 2
 edgemask() (*vsmask.edge.Max* method), 4
 edgemask() (*vsmask.edge.MinMax* method), 110
 edgemask() (*vsmask.edge.Prewitt* method), 94
 edgemask() (*vsmask.edge.PrewittStd* method), 95
 edgemask() (*vsmask.edge.PrewittTCanny* method), 96
 edgemask() (*vsmask.edge.RidgeDetect* method), 5
 edgemask() (*vsmask.edge.Roberts* method), 86
 edgemask() (*vsmask.edge.Robinson3* method), 106
 edgemask() (*vsmask.edge.Robinson5* method), 106
 edgemask() (*vsmask.edge.RSchar* method), 101
 edgemask() (*vsmask.edge.SavitzkyGolay* method), 8
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic11* method), 20
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic13* method), 21
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic15* method), 22
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic17* method), 23
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic19* method), 24
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic21* method), 24
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic23* method), 25
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic25* method), 26
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic5* method), 18
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic7* method), 19
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic9* method), 19
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad11* method), 11
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad13* method), 12
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad15* method), 13
 edgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad17*

method), 14

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quad19*
method), 15

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quad21*
method), 15

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quad23*
method), 16

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quad25*
method), 17

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quad5*
method), 9

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quad7*
method), 10

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quad9*
method), 11

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quint11*
method), 28

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quint13*
method), 29

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quint15*
method), 30

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quint17*
method), 31

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quint19*
method), 32

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quint21*
method), 32

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quint23*
method), 33

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quint25*
method), 34

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quint7*
method), 27

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv1Quint9*
method), 28

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quad11*
method), 37

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quad13*
method), 38

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quad15*
method), 39

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quad17*
method), 40

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quad19*
method), 41

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quad21*
method), 41

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quad23*
method), 42

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quad25*
method), 43

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quad5*
method), 35

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quad7*
method), 36

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quad9*
method), 37

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quart11*
method), 45

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quart13*
method), 46

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quart15*
method), 47

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quart17*
method), 48

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quart19*
method), 49

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quart21*
method), 50

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quart23*
method), 50

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quart25*
method), 51

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quart7*
method), 44

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv2Quart9*
method), 45

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Cub11*
method), 54

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Cub13*
method), 55

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Cub15*
method), 56

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Cub17*
method), 57

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Cub19*
method), 58

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Cub21*
method), 58

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Cub23*
method), 59

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Cub25*
method), 60

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Cub5*
method), 52

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Cub7*
method), 53

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Cub9*
method), 54

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Quint11*
method), 63

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Quint13*
method), 63

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Quint15*
method), 64

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Quint17*
method), 65

`edgemask()` (*vsmask.edge.SavitzkyGolayDeriv3Quint19*
method), 65

method), 66

edgemask() (vsmask.edge.SavitzkyGolayDeriv3Quint21 method), 67

edgemask() (vsmask.edge.SavitzkyGolayDeriv3Quint23 method), 67

edgemask() (vsmask.edge.SavitzkyGolayDeriv3Quint25 method), 68

edgemask() (vsmask.edge.SavitzkyGolayDeriv3Quint7 method), 61

edgemask() (vsmask.edge.SavitzkyGolayDeriv3Quint9 method), 62

edgemask() (vsmask.edge.SavitzkyGolayDeriv4Quart11 method), 71

edgemask() (vsmask.edge.SavitzkyGolayDeriv4Quart13 method), 71

edgemask() (vsmask.edge.SavitzkyGolayDeriv4Quart15 method), 72

edgemask() (vsmask.edge.SavitzkyGolayDeriv4Quart17 method), 73

edgemask() (vsmask.edge.SavitzkyGolayDeriv4Quart19 method), 74

edgemask() (vsmask.edge.SavitzkyGolayDeriv4Quart21 method), 75

edgemask() (vsmask.edge.SavitzkyGolayDeriv4Quart23 method), 76

edgemask() (vsmask.edge.SavitzkyGolayDeriv4Quart25 method), 76

edgemask() (vsmask.edge.SavitzkyGolayDeriv4Quart7 method), 69

edgemask() (vsmask.edge.SavitzkyGolayDeriv4Quart9 method), 70

edgemask() (vsmask.edge.SavitzkyGolayDeriv5Quint11 method), 79

edgemask() (vsmask.edge.SavitzkyGolayDeriv5Quint13 method), 80

edgemask() (vsmask.edge.SavitzkyGolayDeriv5Quint15 method), 80

edgemask() (vsmask.edge.SavitzkyGolayDeriv5Quint17 method), 81

edgemask() (vsmask.edge.SavitzkyGolayDeriv5Quint19 method), 82

edgemask() (vsmask.edge.SavitzkyGolayDeriv5Quint21 method), 83

edgemask() (vsmask.edge.SavitzkyGolayDeriv5Quint23 method), 84

edgemask() (vsmask.edge.SavitzkyGolayDeriv5Quint25 method), 84

edgemask() (vsmask.edge.SavitzkyGolayDeriv5Quint7 method), 77

edgemask() (vsmask.edge.SavitzkyGolayDeriv5Quint9 method), 78

edgemask() (vsmask.edge.Scharr method), 100

edgemask() (vsmask.edge.ScharrTCanny method), 102

edgemask() (vsmask.edge.SingleMatrix method), 3

edgemask() (vsmask.edge.Sobel method), 97

edgemask() (vsmask.edge.SobelStd method), 97

edgemask() (vsmask.edge.SobelTCanny method), 98

edgemask() (vsmask.edge.TEdge method), 6

edgemask() (vsmask.edge.TEdgeTedge mask method), 7

edgemask() (vsmask.edge.TheToof method), 107

edgemask() (vsmask.edge.Tritical method), 92

edgemask() (vsmask.edge.TriticalTCanny method), 93

ELLIPSE (vsmask.util.XpandMode attribute), 122

EuclidianDistance (class in vsmask.edge), 3

ExKirsch (class in vsmask.edge), 120

ExLaplacian1 (class in vsmask.edge), 111

ExLaplacian2 (class in vsmask.edge), 112

ExLaplacian3 (class in vsmask.edge), 113

ExLaplacian4 (class in vsmask.edge), 114

expand() (in module vsmask.util), 123

ExPrewitt (class in vsmask.edge), 115

ExSobel (class in vsmask.edge), 116

F

Farid (class in vsmask.edge), 119

FDoG (class in vsmask.edge), 117

FDoGTCanny (class in vsmask.edge), 118

FreyChen (class in vsmask.edge), 105

FreyChenG41 (class in vsmask.edge), 104

G

get_all_edge_detects() (in module vsmask.edge), 121

get_all_ridge_detect() (in module vsmask.edge), 121

I

in pand() (in module vsmask.util), 123

K

Kayyali (class in vsmask.edge), 91

Kirsch (class in vsmask.edge), 108

KirschTCanny (class in vsmask.edge), 109

Kroon (class in vsmask.edge), 102

KroonTCanny (class in vsmask.edge), 103

L

Laplacian1 (class in vsmask.edge), 88

Laplacian2 (class in vsmask.edge), 88

Laplacian3 (class in vsmask.edge), 89

Laplacian4 (class in vsmask.edge), 90

LoG (class in vsmask.edge), 115

LOSANGE (vsmask.util.XpandMode attribute), 122

M

Matrix1D (class in vsmask.edge), 6

Matrix2x2 (class in vsmask.edge), 85

Matrix3x3 (class in *vsmask.edge*), 87
 Matrix5x5 (class in *vsmask.edge*), 111
 MatrixEdgeDetect (class in *vsmask.edge*), 2
 Max (class in *vsmask.edge*), 4
 max_expr() (in module *vsmask.util*), 122
 max_planes() (in module *vsmask.util*), 124
 MinMax (class in *vsmask.edge*), 110
 module
 vsmask, 1
 vsmask.edge, 1
 vsmask.util, 122
 morpho_transfo() (in module *vsmask.util*), 122

P

Prewitt (class in *vsmask.edge*), 94
 PrewittStd (class in *vsmask.edge*), 95
 PrewittTCanny (class in *vsmask.edge*), 96

R

RECTANGLE (*vsmask.util.XpandMode* attribute), 122
 region_abs_mask() (in module *vsmask.util*), 124
 region_mask() (in module *vsmask.util*), 124
 region_rel_mask() (in module *vsmask.util*), 124
 RidgeDetect (class in *vsmask.edge*), 5
 ridgemask() (*vsmask.edge.ASobel* method), 99
 ridgemask() (*vsmask.edge.Cross* method), 94
 ridgemask() (*vsmask.edge.DoG* method), 119
 ridgemask() (*vsmask.edge.EdgeDetect* method), 1
 ridgemask() (*vsmask.edge.EuclidianDistance* method), 3
 ridgemask() (*vsmask.edge.ExKirsch* method), 121
 ridgemask() (*vsmask.edge.ExLaplacian1* method), 112
 ridgemask() (*vsmask.edge.ExLaplacian2* method), 113
 ridgemask() (*vsmask.edge.ExLaplacian3* method), 113
 ridgemask() (*vsmask.edge.ExLaplacian4* method), 114
 ridgemask() (*vsmask.edge.ExPrewitt* method), 116
 ridgemask() (*vsmask.edge.ExSobel* method), 117
 ridgemask() (*vsmask.edge.Farid* method), 120
 ridgemask() (*vsmask.edge.FDoG* method), 117
 ridgemask() (*vsmask.edge.FDoGTCanny* method), 118
 ridgemask() (*vsmask.edge.FreyChen* method), 105
 ridgemask() (*vsmask.edge.FreyChenG41* method), 104
 ridgemask() (*vsmask.edge.Kayyali* method), 91
 ridgemask() (*vsmask.edge.Kirsch* method), 108
 ridgemask() (*vsmask.edge.KirschTCanny* method), 109
 ridgemask() (*vsmask.edge.Kroon* method), 103
 ridgemask() (*vsmask.edge.KroonTCanny* method), 104
 ridgemask() (*vsmask.edge.Laplacian1* method), 88
 ridgemask() (*vsmask.edge.Laplacian2* method), 89
 ridgemask() (*vsmask.edge.Laplacian3* method), 90
 ridgemask() (*vsmask.edge.Laplacian4* method), 91
 ridgemask() (*vsmask.edge.LoG* method), 115
 ridgemask() (*vsmask.edge.Matrix1D* method), 6
 ridgemask() (*vsmask.edge.Matrix2x2* method), 86

ridgemask() (*vsmask.edge.Matrix3x3* method), 87
 ridgemask() (*vsmask.edge.Matrix5x5* method), 111
 ridgemask() (*vsmask.edge.MatrixEdgeDetect* method), 2
 ridgemask() (*vsmask.edge.Max* method), 4
 ridgemask() (*vsmask.edge.MinMax* method), 110
 ridgemask() (*vsmask.edge.Prewitt* method), 95
 ridgemask() (*vsmask.edge.PrewittStd* method), 95
 ridgemask() (*vsmask.edge.PrewittTCanny* method), 96
 ridgemask() (*vsmask.edge.RidgeDetect* method), 5
 ridgemask() (*vsmask.edge.Roberts* method), 86
 ridgemask() (*vsmask.edge.Robinson3* method), 106
 ridgemask() (*vsmask.edge.Robinson5* method), 107
 ridgemask() (*vsmask.edge.RSchar* method), 101
 ridgemask() (*vsmask.edge.SavitzkyGolay* method), 8
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic11* method), 21
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic13* method), 21
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic15* method), 22
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic17* method), 23
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic19* method), 24
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic21* method), 25
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic23* method), 26
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic25* method), 26
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic5* method), 18
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic7* method), 19
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Cubic9* method), 20
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad11* method), 12
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad13* method), 13
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad15* method), 13
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad17* method), 14
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad19* method), 15
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad21* method), 16
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad23* method), 17
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad25* method), 17
 ridgemask() (*vsmask.edge.SavitzkyGolayDeriv1Quad5*

`method`), 9
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quad7`
`method`), 10
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quad9`
`method`), 11
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quint11`
`method`), 29
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quint13`
`method`), 30
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quint15`
`method`), 30
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quint17`
`method`), 31
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quint19`
`method`), 32
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quint21`
`method`), 33
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quint23`
`method`), 34
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quint25`
`method`), 34
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quint7`
`method`), 27
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv1Quint9`
`method`), 28
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quad11`
`method`), 38
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quad13`
`method`), 39
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quad15`
`method`), 39
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quad17`
`method`), 40
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quad19`
`method`), 41
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quad21`
`method`), 42
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quad23`
`method`), 43
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quad25`
`method`), 43
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quad5`
`method`), 35
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quad7`
`method`), 36
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quad9`
`method`), 37
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quart11`
`method`), 46
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quart13`
`method`), 47
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quart15`
`method`), 47
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quart17`
`method`), 48
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quart19`
`method`), 49
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quart21`
`method`), 50
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quart23`
`method`), 51
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quart25`
`method`), 52
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quart7`
`method`), 44
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv2Quart9`
`method`), 45
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Cub11`
`method`), 55
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Cub13`
`method`), 56
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Cub15`
`method`), 56
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Cub17`
`method`), 57
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Cub19`
`method`), 58
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Cub21`
`method`), 59
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Cub23`
`method`), 60
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Cub25`
`method`), 60
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Cub5`
`method`), 52
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Cub7`
`method`), 53
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Cub9`
`method`), 54
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Quint11`
`method`), 63
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Quint13`
`method`), 64
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Quint15`
`method`), 65
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Quint17`
`method`), 65
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Quint19`
`method`), 66
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Quint21`
`method`), 67
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Quint23`
`method`), 68
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Quint25`
`method`), 69
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Quint7`
`method`), 61
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv3Quint9`

- `method`), 62
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv4Quart11`
`method`), 71
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv4Quart13`
`method`), 72
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv4Quart15`
`method`), 73
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv4Quart17`
`method`), 73
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv4Quart19`
`method`), 74
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv4Quart21`
`method`), 75
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv4Quart23`
`method`), 76
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv4Quart25`
`method`), 77
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv4Quart7`
`method`), 69
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv4Quart9`
`method`), 70
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv5Quint11`
`method`), 79
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv5Quint13`
`method`), 80
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv5Quint15`
`method`), 81
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv5Quint17`
`method`), 82
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv5Quint19`
`method`), 82
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv5Quint21`
`method`), 83
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv5Quint23`
`method`), 84
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv5Quint25`
`method`), 85
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv5Quint7`
`method`), 78
`ridgemask()` (`vsmask.edge.SavitzkyGolayDeriv5Quint9`
`method`), 78
`ridgemask()` (`vsmask.edge.Scharr` `method`), 100
`ridgemask()` (`vsmask.edge.ScharrTCanny` `method`), 102
`ridgemask()` (`vsmask.edge.SingleMatrix` `method`), 2
`ridgemask()` (`vsmask.edge.Sobel` `method`), 97
`ridgemask()` (`vsmask.edge.SobelStd` `method`), 98
`ridgemask()` (`vsmask.edge.SobelTCanny` `method`), 99
`ridgemask()` (`vsmask.edge.TEdge` `method`), 7
`ridgemask()` (`vsmask.edge.TEdgeTegemask` `method`), 8
`ridgemask()` (`vsmask.edge.TheToof` `method`), 108
`ridgemask()` (`vsmask.edge.Tritical` `method`), 92
`ridgemask()` (`vsmask.edge.TriticalTCanny` `method`), 93
Roberts (*class in* `vsmask.edge`), 86
Robinson3 (*class in* `vsmask.edge`), 106
Robinson5 (*class in* `vsmask.edge`), 106
RScharr (*class in* `vsmask.edge`), 101
- ## S
- SavitzkyGolay (*class in* `vsmask.edge`), 8
SavitzkyGolayDeriv1Cubic11 (*class in* `vsmask.edge`),
20
SavitzkyGolayDeriv1Cubic13 (*class in* `vsmask.edge`),
21
SavitzkyGolayDeriv1Cubic15 (*class in* `vsmask.edge`),
22
SavitzkyGolayDeriv1Cubic17 (*class in* `vsmask.edge`),
23
SavitzkyGolayDeriv1Cubic19 (*class in* `vsmask.edge`),
23
SavitzkyGolayDeriv1Cubic21 (*class in* `vsmask.edge`),
24
SavitzkyGolayDeriv1Cubic23 (*class in* `vsmask.edge`),
25
SavitzkyGolayDeriv1Cubic25 (*class in* `vsmask.edge`),
26
SavitzkyGolayDeriv1Cubic5 (*class in* `vsmask.edge`),
18
SavitzkyGolayDeriv1Cubic7 (*class in* `vsmask.edge`),
19
SavitzkyGolayDeriv1Cubic9 (*class in* `vsmask.edge`),
19
SavitzkyGolayDeriv1Quad11 (*class in* `vsmask.edge`),
11
SavitzkyGolayDeriv1Quad13 (*class in* `vsmask.edge`),
12
SavitzkyGolayDeriv1Quad15 (*class in* `vsmask.edge`),
13
SavitzkyGolayDeriv1Quad17 (*class in* `vsmask.edge`),
14
SavitzkyGolayDeriv1Quad19 (*class in* `vsmask.edge`),
15
SavitzkyGolayDeriv1Quad21 (*class in* `vsmask.edge`),
15
SavitzkyGolayDeriv1Quad23 (*class in* `vsmask.edge`),
16
SavitzkyGolayDeriv1Quad25 (*class in* `vsmask.edge`),
17
SavitzkyGolayDeriv1Quad5 (*class in* `vsmask.edge`), 9
SavitzkyGolayDeriv1Quad7 (*class in* `vsmask.edge`), 10
SavitzkyGolayDeriv1Quad9 (*class in* `vsmask.edge`), 10
SavitzkyGolayDeriv1Quint11 (*class in* `vsmask.edge`),
28
SavitzkyGolayDeriv1Quint13 (*class in* `vsmask.edge`),
29
SavitzkyGolayDeriv1Quint15 (*class in* `vsmask.edge`),
30
SavitzkyGolayDeriv1Quint17 (*class in* `vsmask.edge`),
31

SavitzkyGolayDeriv1Quint19 (<i>class in vsmask.edge</i>), 32	SavitzkyGolayDeriv3Cub17 (<i>class in vsmask.edge</i>), 57
SavitzkyGolayDeriv1Quint21 (<i>class in vsmask.edge</i>), 32	SavitzkyGolayDeriv3Cub19 (<i>class in vsmask.edge</i>), 58
SavitzkyGolayDeriv1Quint23 (<i>class in vsmask.edge</i>), 33	SavitzkyGolayDeriv3Cub21 (<i>class in vsmask.edge</i>), 58
SavitzkyGolayDeriv1Quint25 (<i>class in vsmask.edge</i>), 34	SavitzkyGolayDeriv3Cub23 (<i>class in vsmask.edge</i>), 59
SavitzkyGolayDeriv1Quint7 (<i>class in vsmask.edge</i>), 27	SavitzkyGolayDeriv3Cub25 (<i>class in vsmask.edge</i>), 60
SavitzkyGolayDeriv1Quint9 (<i>class in vsmask.edge</i>), 28	SavitzkyGolayDeriv3Cub5 (<i>class in vsmask.edge</i>), 52
SavitzkyGolayDeriv2Quad11 (<i>class in vsmask.edge</i>), 37	SavitzkyGolayDeriv3Cub7 (<i>class in vsmask.edge</i>), 53
SavitzkyGolayDeriv2Quad13 (<i>class in vsmask.edge</i>), 38	SavitzkyGolayDeriv3Cub9 (<i>class in vsmask.edge</i>), 54
SavitzkyGolayDeriv2Quad15 (<i>class in vsmask.edge</i>), 39	SavitzkyGolayDeriv3Quint11 (<i>class in vsmask.edge</i>), 62
SavitzkyGolayDeriv2Quad17 (<i>class in vsmask.edge</i>), 40	SavitzkyGolayDeriv3Quint13 (<i>class in vsmask.edge</i>), 63
SavitzkyGolayDeriv2Quad19 (<i>class in vsmask.edge</i>), 41	SavitzkyGolayDeriv3Quint15 (<i>class in vsmask.edge</i>), 64
SavitzkyGolayDeriv2Quad21 (<i>class in vsmask.edge</i>), 41	SavitzkyGolayDeriv3Quint17 (<i>class in vsmask.edge</i>), 65
SavitzkyGolayDeriv2Quad23 (<i>class in vsmask.edge</i>), 42	SavitzkyGolayDeriv3Quint19 (<i>class in vsmask.edge</i>), 66
SavitzkyGolayDeriv2Quad25 (<i>class in vsmask.edge</i>), 43	SavitzkyGolayDeriv3Quint21 (<i>class in vsmask.edge</i>), 67
SavitzkyGolayDeriv2Quad5 (<i>class in vsmask.edge</i>), 35	SavitzkyGolayDeriv3Quint23 (<i>class in vsmask.edge</i>), 67
SavitzkyGolayDeriv2Quad7 (<i>class in vsmask.edge</i>), 36	SavitzkyGolayDeriv3Quint25 (<i>class in vsmask.edge</i>), 68
SavitzkyGolayDeriv2Quad9 (<i>class in vsmask.edge</i>), 36	SavitzkyGolayDeriv3Quint7 (<i>class in vsmask.edge</i>), 61
SavitzkyGolayDeriv2Quart11 (<i>class in vsmask.edge</i>), 45	SavitzkyGolayDeriv3Quint9 (<i>class in vsmask.edge</i>), 62
SavitzkyGolayDeriv2Quart13 (<i>class in vsmask.edge</i>), 46	SavitzkyGolayDeriv4Quart11 (<i>class in vsmask.edge</i>), 71
SavitzkyGolayDeriv2Quart15 (<i>class in vsmask.edge</i>), 47	SavitzkyGolayDeriv4Quart13 (<i>class in vsmask.edge</i>), 71
SavitzkyGolayDeriv2Quart17 (<i>class in vsmask.edge</i>), 48	SavitzkyGolayDeriv4Quart15 (<i>class in vsmask.edge</i>), 72
SavitzkyGolayDeriv2Quart19 (<i>class in vsmask.edge</i>), 49	SavitzkyGolayDeriv4Quart17 (<i>class in vsmask.edge</i>), 73
SavitzkyGolayDeriv2Quart21 (<i>class in vsmask.edge</i>), 49	SavitzkyGolayDeriv4Quart19 (<i>class in vsmask.edge</i>), 74
SavitzkyGolayDeriv2Quart23 (<i>class in vsmask.edge</i>), 50	SavitzkyGolayDeriv4Quart21 (<i>class in vsmask.edge</i>), 75
SavitzkyGolayDeriv2Quart25 (<i>class in vsmask.edge</i>), 51	SavitzkyGolayDeriv4Quart23 (<i>class in vsmask.edge</i>), 75
SavitzkyGolayDeriv2Quart7 (<i>class in vsmask.edge</i>), 44	SavitzkyGolayDeriv4Quart25 (<i>class in vsmask.edge</i>), 76
SavitzkyGolayDeriv2Quart9 (<i>class in vsmask.edge</i>), 45	SavitzkyGolayDeriv4Quart7 (<i>class in vsmask.edge</i>), 69
SavitzkyGolayDeriv3Cub11 (<i>class in vsmask.edge</i>), 54	SavitzkyGolayDeriv4Quart9 (<i>class in vsmask.edge</i>), 70
SavitzkyGolayDeriv3Cub13 (<i>class in vsmask.edge</i>), 55	SavitzkyGolayDeriv5Quint11 (<i>class in vsmask.edge</i>), 79
SavitzkyGolayDeriv3Cub15 (<i>class in vsmask.edge</i>), 56	SavitzkyGolayDeriv5Quint13 (<i>class in vsmask.edge</i>), 80
	SavitzkyGolayDeriv5Quint15 (<i>class in vsmask.edge</i>), 80

SavitzkyGolayDeriv5Quint17 (*class in vsmask.edge*),
81
SavitzkyGolayDeriv5Quint19 (*class in vsmask.edge*),
82
SavitzkyGolayDeriv5Quint21 (*class in vsmask.edge*),
83
SavitzkyGolayDeriv5Quint23 (*class in vsmask.edge*),
84
SavitzkyGolayDeriv5Quint25 (*class in vsmask.edge*),
84
SavitzkyGolayDeriv5Quint7 (*class in vsmask.edge*),
77
SavitzkyGolayDeriv5Quint9 (*class in vsmask.edge*),
78
Scharr (*class in vsmask.edge*), 100
ScharrTCanny (*class in vsmask.edge*), 102
SingleMatrix (*class in vsmask.edge*), 2
Sobel (*class in vsmask.edge*), 97
SobelStd (*class in vsmask.edge*), 97
SobelTCanny (*class in vsmask.edge*), 98

T

TEdge (*class in vsmask.edge*), 6
TEdgeTedgemask (*class in vsmask.edge*), 7
TheToof (*class in vsmask.edge*), 107
Tritical (*class in vsmask.edge*), 92
TriticalTCanny (*class in vsmask.edge*), 93

V

vsmask
 module, 1
vsmask.edge
 module, 1
vsmask.util
 module, 122

X

XxpanMode (*class in vsmask.util*), 122