

## 10.1.12 Editors - Compositor Editor - Header - Add Menu - Matte

### Table of content

Detailed table of content.....	1
Add menu - Matte.....	5
Box Mask.....	5
Channel Key.....	7
Chroma Key.....	8
Color Key.....	8
Color Spill.....	9
Cryptomatte.....	10
Cryptomatte (Legacy).....	12
Difference Key.....	13
Distance Key.....	14
Ellipse Mask.....	16
Keying.....	17
Keying Screen.....	20
Luminance Key.....	20

### Detailed table of content

#### Detailed table of content

Detailed table of content.....	1
Add menu - Matte.....	5
Box Mask.....	5
Inputs.....	5
Mask.....	5
Value.....	5
Properties.....	6
X, Y.....	6
Width.....	6
Height.....	6
Rotation.....	6
Mask Type.....	6
Add.....	6
Subtract.....	6
Not.....	6
Outputs.....	6
Mask.....	6
Channel Key.....	7
Inputs.....	7
Image.....	7
Properties.....	7
Color Space.....	7
Key channel.....	7
Algorithm.....	7
Limit.....	7
High.....	7

Low.....	7
Outputs.....	7
Image.....	7
Matte.....	7
Chroma Key.....	8
Inputs.....	8
Image.....	8
Key Color.....	8
Properties.....	8
Acceptance.....	8
Cutoff.....	8
Falloff.....	8
Outputs.....	8
Image.....	8
Matte.....	8
Color Key.....	8
Inputs.....	9
Image.....	9
Properties.....	9
Color.....	9
Outputs.....	9
Image.....	9
Matte.....	9
Color Spill.....	9
Inputs.....	9
Image.....	9
Factor.....	9
Properties.....	10
Despill Channel.....	10
Algorithm.....	10
Limiting Channel.....	10
R, G, B.....	10
Ratio.....	10
Unspill.....	10
R, G, B.....	10
Outputs.....	10
Image.....	10
Cryptomatte.....	10
Usage.....	11
Inputs.....	11
Image.....	11
Properties.....	11
Render / Image.....	11
Render.....	11
Scene Property.....	11
Cryptomatte Layer.....	11
Image.....	11
Image Property.....	11
Cryptomatte Layer.....	11
Matte ID.....	11
Outputs.....	12
Image.....	12
Matte.....	12

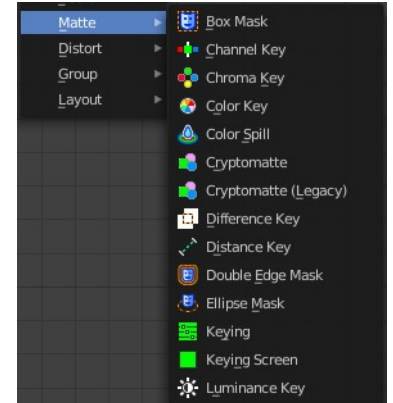
Pick.....	12
Cryptomatte (Legacy).....	12
Usage.....	12
Adding/Removing Layers.....	12
Inputs.....	13
Image.....	13
Crypto Passes.....	13
Properties.....	13
Add/Remove.....	13
Matte ID.....	13
Outputs.....	13
Image.....	13
Matte.....	13
Pick.....	13
Difference Key.....	13
Inputs.....	13
Image.....	13
Image.....	13
Properties.....	14
Tolerance.....	14
Falloff.....	14
Outputs.....	14
Image.....	14
Matte.....	14
Distance Key.....	14
Inputs.....	14
Image.....	14
Key Color.....	14
Properties.....	14
Tolerance.....	14
Falloff.....	14
Color Space.....	15
Outputs.....	15
Image.....	15
Matte.....	15
Double Edge Mask.....	15
Inputs.....	15
Inner Mask.....	15
Outer Mask.....	15
Properties.....	15
Inner Edge.....	15
All.....	15
Adjacent Only.....	15
Buffer Edge.....	15
Keep In.....	15
Bleed Out.....	15
Outputs.....	16
Mask.....	16
Ellipse Mask.....	16
Inputs.....	16
Mask.....	16
Value.....	16
Properties.....	16

X, Y.....	16
Width.....	16
Height.....	16
Rotation.....	16
Mask Type.....	16
Add.....	16
Subtract.....	17
Multiply.....	17
Not.....	17
Outputs.....	17
Mask.....	17
Keying.....	17
Inputs.....	17
Image.....	17
Key Color.....	17
Garbage Matte.....	17
Core Matte.....	17
Properties.....	18
Pre Blur.....	18
Screen Balance.....	18
Despill Factor.....	18
Despill Balance.....	18
Edge Kernel Radius.....	18
Edge Kernel Tolerance.....	18
Clip Black.....	18
Clip White.....	18
Dilate/Erode.....	19
Feather Falloff.....	19
Feather Distance.....	19
Post Blur.....	19
Outputs.....	19
Image.....	19
Matte.....	19
Edges.....	19
Keying Screen.....	20
Properties.....	20
Movie Clip.....	20
Tracking Object.....	20
Outputs.....	20
Screen.....	20
Luminance Key.....	20
Inputs.....	21
Image.....	21
Properties.....	21
Limit.....	21
High.....	21
Low.....	21
Outputs.....	21
Image.....	21
Matte.....	21

## Add menu - Matte

These nodes give you the essential tools for creating a Matte for images that do not already have their own Alpha Channel. One usage scenario is blue-screen or green-screen footage, where live action is shot in front of a blue or green backdrop for replacement by a matte painting or virtual background.

In general, hook up these nodes to a viewer, set your Image Editor to show the Viewer node, and play with the sliders in real-time using a sample image from the footage, to get the settings right. In some cases, small adjustments can eliminate artifacts or foreground image degradation. Taking out too much green can result in foreground actors looking flat or blueish/purplish.



You can and should chain these nodes together, improving your masking and color correction in successive refinements, using each node's strengths to operate on the previous node's output. Keying Node is the closest to a "does-it-all" node for green screens, but the best results stem from a combination of techniques.

### Note!

Garbage Matte is not a node, but a technique selecting what to exclude from an image. It is a Mask used to identify content to be removed from an image that cannot be removed by an automatic process like chroma keying. It is used either to select specific content to be removed, or it is the inverse of a rough selection of the subject; removing everything else.

Some nodes accept a garbage matte directly. For those that don't, you can still apply one by subtracting the garbage matte from the matte generated by the node.

Simple garbage mattes can be created with the Box Mask or the Ellipse Mask. More complicated matte shapes using a Double Edge Mask or using a Mask.

## Box Mask

The Box Mask node creates an image suitable for use as a simple matte.

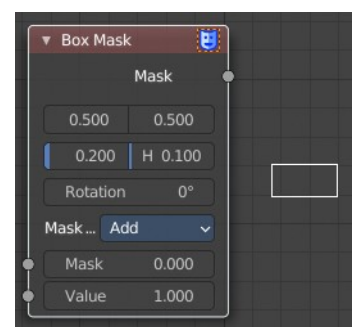
### Inputs

#### Mask

An optional mask to use as the base for mask operations.

#### Value

Intensity of the generated mask.



## Properties

### ***X, Y***

Position of the center of the box as a fraction of the total width or height. (0.5, 0.5 creates a centered box; 0.0, 0.0 creates a box in the lower left.)

### ***Width***

Width of the box as a fraction of the total image width.

### ***Height***

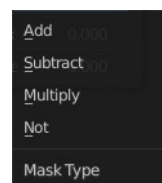
Height of the box as a fraction of the total image width, not height.

### ***Rotation***

Rotation of the box around its center point.

### ***Mask Type***

Operation to use against the input mask.



### **Add**

This yields the union of the input mask and the generated mask: Areas covered by the generated mask are set to the specified Value. Other parts of the input masked are passed through unchanged, or set to black if there is no input mask.

### **Subtract**

Values of the input mask have the specified Value subtracted from them.

### **Multiply**

This yields the intersection of this generated mask and the input mask: Values of the input mask are multiplied by the specified Value for the area covered by the generated mask. All other areas become black.

### **Not**

Any area covered by both the input mask and the generated mask becomes black. Areas covered by the generated mask that are black on the input mask become the specified Value. Areas uncovered by the generated mask remain unchanged.

## Outputs

### ***Mask***

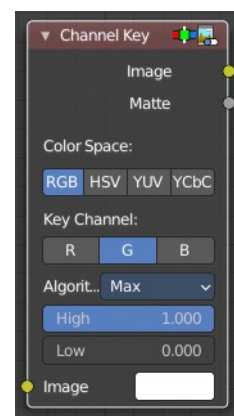
A generated rectangular mask merged with the input mask. The created mask is the size of the current scene render dimensions.

Tip. For soft edges, pass the output mask through a slight Blur node.

## Channel Key

The Channel Key node determines background objects from foreground objects by the difference in the selected channel's levels.

For example in YUV color space, this is useful when compositing stock footage of explosions (very bright) which are normally shot against a solid, dark background.



### Inputs

#### *Image*

Standard image input.

### Properties

#### *Color Space*

This button selects what color space the channels will represent.

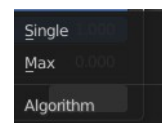
RGB, HSV, YUV, YCbCr

#### *Key channel*

This button selects the channel, defined by the Color Space, to use to determine the matte.

#### *Algorithm*

Max or Single.



#### *Limit*

It is possible to have a separation between the two values to allow for a gradient of transparency between foreground and background objects.

#### **High**

Determines the lowest values that are considered foreground. (Which is supposed to be – relatively – height values: from this value to 1.0.)

#### **Low**

Determines the highest values that are considered to be background objects. (Which is supposed to be – relatively – low values: from 0.0 to this value.)

### Outputs

#### *Image*

Image with an alpha channel adjusted for the keyed selection.

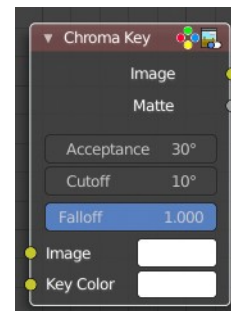
#### *Matte*

A black-and-white alpha mask of the key.

## Chroma Key

The Chroma Key node determines if a pixel is a foreground or background (and thereby should be transparent) based on its chroma values.

Use this, for example, to composite images that have been shot in front of a green or blue screen.



### Inputs

#### *Image*

Standard image input.

#### *Key Color*

The background color usually selected using the color picker and the original image.

### Properties

#### *Acceptance*

An angle on the color wheel that represents how tolerant the keying color is. Larger angles allow for larger variation in the keying color to be considered background pixels.

#### *Cutoff*

Controls the level that is considered the pure background. Higher cutoff levels mean more pixels will be 100% transparent if they are within the angle tolerance.

#### *Falloff*

Increase to make nearby pixels partially transparent producing a smoother blend along the edges.

### Outputs

#### *Image*

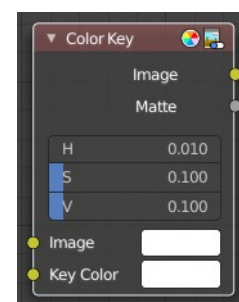
Image with its alpha channel adjusted for the keyed selection.

#### *Matte*

A black-and-white alpha mask of the key.

## Color Key

The Color Key node creates a matte based on a specified color of the input image.





## Inputs

### *Image*

Standard image input.

## Properties

### *Color*

The sliders represent threshold values. Higher values in this node's context mean a wider range of colors from the specified will be added to the matte.

Hue, Saturation, Value

## Outputs

### *Image*

Image with its alpha channel adjusted for the keyed selection.

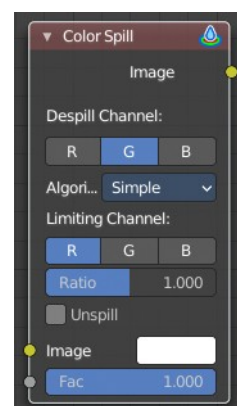
### *Matte*

A black-and-white alpha mask of the key.

## Color Spill

The Color Spill node reduces one of the RGB channels so that it is not greater than any of the others.

This is common when compositing images that were shot in front of a green or blue screen. In some cases, if the foreground object is reflective, it will show the green or blue color; that color has “spilled” onto the foreground object. If there is light from the side or back, and the foreground actor is wearing white, it is possible to get “spill” green (or blue) light from the background onto the foreground objects, coloring them with a tinge of green or blue. To remove the green (or blue) light, you use this fancy node.



## Inputs

### *Image*

Standard image input.

### *Factor*

Standard Factor.

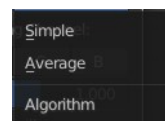
## Properties

### ***Despill Channel***

R, G, B

### ***Algorithm***

Simple or Average.



### ***Limiting Channel***

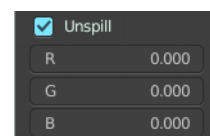
R, G, B

### ***Ratio***

Scale limit by value.

### ***Unspill***

Allows you to reduce the selected channel's input to the image greater than the color spill algorithm normally allows. This is useful for exceptionally high amounts of the color spill.



R, G, B

## Outputs

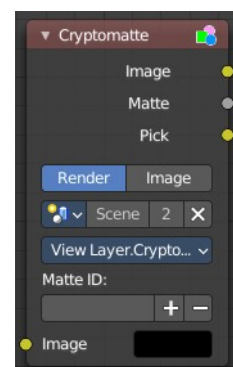
### ***Image***

The image with the corrected channels.

## Cryptomatte

The Cryptomatte node uses the Cryptomatte standard to efficiently create mattes for compositing. Cycles outputs the required render passes, which can then be used in the Compositor or another Compositor with Cryptomatte support to create masks for specified objects.

Unlike the Material and Object Index passes, the objects to isolate are selected in compositing, and mattes will be anti-aliased and take into account effects like motion blur and transparency.



## Usage

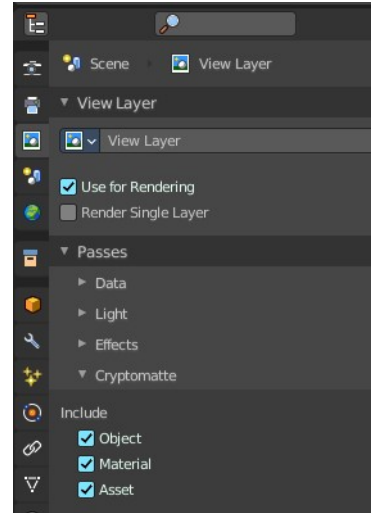
Enable Cryptomatte Object render pass in the Passes panel, and render.

In the compositing nodes, create a Cryptomatte node and link the Render Layer matching Image and Cryptomatte passes to it.

Attach a Viewer node to the Pick output of the Cryptomatte node.

Use the Cryptomatte Add/Remove button to sample objects in the Pick Viewer node.

Use the Matte output of the Cryptomatte node to get the alpha mask.



## Inputs

### Image

Standard image input.

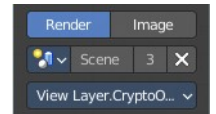
## Properties

### Render / Image

Use the render result or an external image as the base image for cryptomatte.

### Render

Use Cryptomatte data that are stored as part of the render.

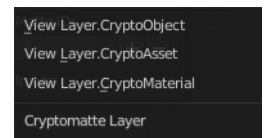


### Scene Property

Pick the scene that you want to use.

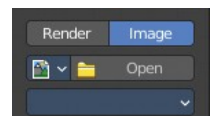
### Cryptomatte Layer

Pick the cryptomatte layer that you want to use.



### Image

Use Cryptomatte data that are stored inside a multilayered OpenEXR image.

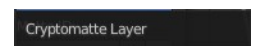


### Image Property

Pick the image that you want to use.

### Cryptomatte Layer

Pick the cryptomatte layer that you want to use.



## Matte ID

List of object and material crypto IDs to include in matte. This can be used for example to quickly clear all mattes by deleting the text or used to copy-and-paste crypto IDs from other software.

## Outputs

### **Image**

A colored output of the input image with the matte applied to only include selected layers.

### **Matte**

A black-and-white alpha mask of the all the selected crypto layers.

### **Pick**

A colored representation of the Cryptomatte pass which can be used with a Viewer node to select which crypto passes are used to create the matte image.

## Cryptomatte (Legacy)

This is the legacy node version.

The Cryptomatte node uses the Cryptomatte standard to efficiently create mattes for compositing. Cycles outputs the required render passes, which can then be used in the Compositor or another Compositor with Cryptomatte support to create masks for specified objects.

Unlike the Material and Object Index passes, the objects to isolate are selected in compositing, and mattes will be anti-aliased and take into account effects like motion blur and transparency.



## Usage

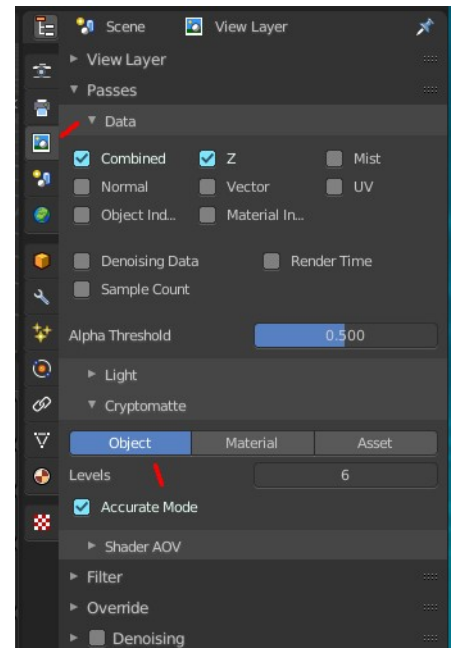
Enable Cryptomatte Object render pass in the Passes panel, and render.

In the compositing nodes, create a Cryptomatte node and link the Render Layer matching Image and Cryptomatte passes to it.

Attach a Viewer node to the Pick output of the Cryptomatte node.

Use the Cryptomatte Add/Remove button to sample objects in the Pick Viewer node.

Use the Matte output of the Cryptomatte node to get the alpha mask.



## Adding/Removing Layers

By default there are only four crypto layers available as inputs to the Cryptomatte node. You can add or remove layer inputs through Sidebar > Item > Properties > Add/Remove Crypto Layer. These operators will add/remove layers from the bottom of the pass inputs.

## Inputs

### *Image*

Standard image input.

### *Crypto Passes*

Each crypto layer will be given its own render pass; each of these render passes must be connected to one of these crypto layer inputs. By default there are only four layers, see [Adding/Removing Layers](#) to add more.

## Properties

### *Add/Remove*

Adds/Removes an object or material from matte, by picking a color from the Pick output.

### *Matte ID*

List of object and material crypto IDs to include in matte. This can be used for example to quickly clear all mattes by deleting the text or used to copy-and-paste crypto IDs from other software.

## Outputs

### *Image*

A colored output of the input image with the matte applied to only include selected layers.

### *Matte*

A black-and-white alpha mask of the all the selected crypto layers.

### *Pick*

A colored representation of the Cryptomatte pass which can be used with a Viewer node to select which crypto passes are used to create the matte image.

## Difference Key

This node produces a matte that isolates foreground content by comparing it with a reference background image.

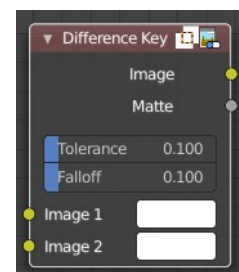
### Inputs

#### *Image*

Contains foreground content against the background that is to be removed.

#### *Image*

The reference background image.



## Properties

### **Tolerance**

Where pixels match the reference background to within the specified threshold, the matte is made transparent.

### **Falloff**

Increase to make nearby pixels partially transparent producing a smoother blend along the edges.

## Outputs

### **Image**

Image with its alpha channel adjusted for the keyed selection.

### **Matte**

A black-and-white alpha mask of the key.

## Distance Key

The Distance Key node determines a pixel's alpha value based on the three-dimensional distance between the image pixel color and the key color in a 3D color space.

This key works well when trying to single out a specific color in a background (not necessarily green).

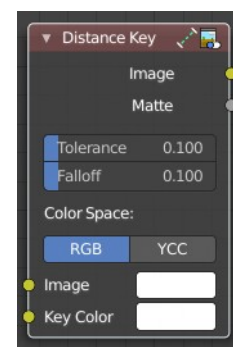
## Inputs

### **Image**

Standard image input.

### **Key Color**

The color that is to be keyed.



## Properties

### **Tolerance**

A threshold what the node considers a match between the key color and the foreground pixel. The tolerance affects how close a pixel needs to be to the background pixel to be considered an absolute match.

### **Falloff**

When the Falloff value is high, pixels that are close to the Key Color are more transparent than pixels that are not as close to the Key Color (but still considered close enough to be keyed). When the Falloff value is low, it does not matter how close the pixel color (Image) is to the Key Color, it is transparent.

## Color Space

RGB, YCC

It is also possible to work with YCbCr color space, but only the Cb and Cr channels are taken into consideration for determining the distance between the foreground and background pixels.

## Outputs

### Image

The image with an alpha channel adjusted for the keyed selection.

### Matte

A black-and-white alpha mask of the key.

## Double Edge Mask

The Double Edge Mask node creates a gradient between two masks.

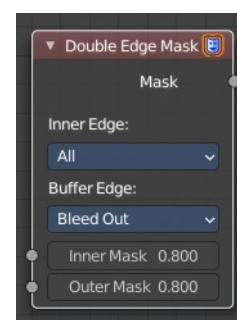
## Inputs

### Inner Mask

A mask representing the inside shape, which will be fully white.

### Outer Mask

A mask representing the outside shape, which will fade from black at its edges to white at the Inner Mask.



## Properties

### Inner Edge

#### All

All shapes in the Inner Mask contribute to the gradient, even ones that do not touch the Outer Mask shape.

#### Adjacent Only

Only shapes in the Inner Mask that overlap with the Outer Mask contribute to the gradient.

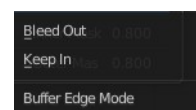
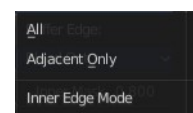
### Buffer Edge

#### Keep In

Parts of the Outer Mask that touch the edge of the image are treated as if they stop at the edge.

#### Bleed Out

Parts of the Outer Mask that touch the edge of the image are extended beyond the boundary of the image.



## Outputs

### **Mask**

Standard mask output.

## Ellipse Mask

The Ellipse Mask node creates an image suitable for use as a simple matte or vignette mask.

## Inputs

### **Mask**

An optional mask to use as the base for mask operations.

### **Value**

Intensity of the generated mask.

## Properties

### **X, Y**

Position of the center of the ellipse as a fraction of the total width or height. (0.5, 0.5 creates a centered ellipse; 0.0, 0.0 creates an ellipse with its center in the lower left.)

### **Width**

Width of the ellipse as a fraction of the total image width.

### **Height**

Height of the ellipse as a fraction of the total image width, not height. Equal Width and Height values with produce a circle.

### **Rotation**

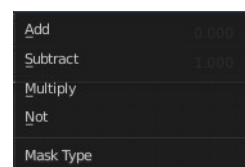
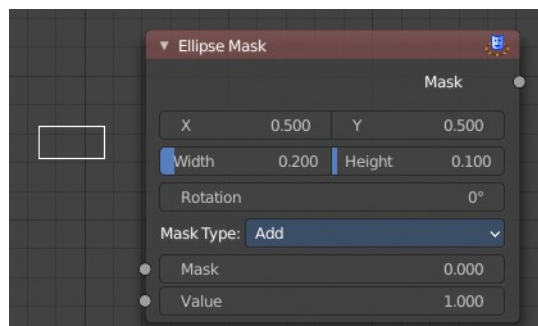
Rotation of the ellipse around its center point.

### **Mask Type**

What operation to use against the input mask.

### **Add**

This yields the union of the input mask and the generated mask: Areas covered by the generated mask are set to the specified Value. Other parts of the input masked are passed through unchanged, or set to black if there is no input mask.





## Subtract

Values of the input mask have the specified Value subtracted from them.

## Multiply

This yields the intersection of this generated mask and the input mask: Values of the input mask are multiplied by the specified Value for the area covered by the generated mask. All other areas become black.

## Not

Any area covered by both the input mask and the generated mask becomes black. Areas covered by the generated mask that are black on the input mask become the specified Value. Areas uncovered by the generated mask remain unchanged.

## Outputs

### Mask

A generated elliptical mask merged with the input mask. The created mask is the size of the current scene render dimensions.

Tip. For soft edges, pass the output mask through a slight Blur node. For a vignette, pass the output of this through a heavy blur.

## Keying

The Keying node is a one-stop-shop for “green screen” / “blue screen” removal. It performs both chroma keying to remove the backdrop and despill to correct color cast from the backdrop. Additionally, you can perform common operations used to tweak the resulting matte.

## Inputs

### Image

Standard image input.

### Key Color

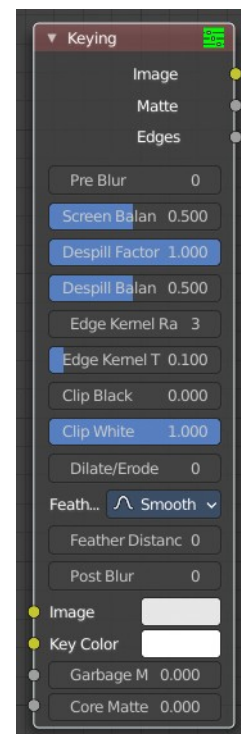
The color of content to be removed. This may be a single color, or a reference image such as generated by the Keying Screen Node.

### Garbage Matte

An optional mask of area(s) to always exclude from the output. This is removed from the chroma key generated matte.

### Core Matte

An optional mask of area(s) to always include in the output. This is merged with the chroma key generated



matte.

## Properties

### ***Pre Blur***

Reduce the effects of color noise in the image by blurring only color by the given amount, leaving luminosity intact. This will affect matte calculation only, not the result image.

### ***Screen Balance***

This is the balance between color channels compared with the key color. 0.5 will average the other channels (red and blue in the case of a green screen).

This may be tweaked in tandem with Clip Black and Clip White while checking the Matte output to create a mask with optimal separation.

### ***Despill Factor***

Controls how much color bleed from the key color is removed from the input image: 0 means no despill, 1 means all possible spilling will be removed. The underlying implementation is the same as adjusting the Unspill amount of the Color Spill Node.

### ***Despill Balance***

This controls how the color channels are compared when computing spill, affecting the hue and shade of the corrected colors. It is similar to setting the Limiting Channel in the Color Spill Node.

### ***Edge Kernel Radius***

Defines the radius in pixel used to detect an edge.

### ***Edge Kernel Tolerance***

Defines threshold used to check if pixels in radius are the same as current pixel: if the difference between pixel colors is higher than this threshold then the point will be considered an edge.

### ***Clip Black***

This sets the threshold for what becomes fully transparent in the output (black in the matte). It should be set as low as possible. Uneven backdrops will require this value to be increased. Use of the Keying Screen Node can help keep this value low. You may also use a Garbage Matte to exclude problematic areas.

This value does not impact areas detected as edges to ensure edge detail is preserved.

### ***Clip White***

This sets the threshold for what becomes fully opaque in the output (white in the matte). It should be set as high as possible. Colors close to green in the foreground may require reducing this value and/or adjusting the Screen Balance. Particularly problematic parts can be fixed with a Core Matte instead of a low Clip White.

This value does not impact areas detected as edges to ensure edge detail is preserved.

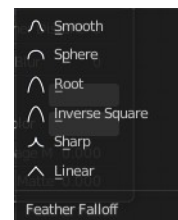
## ***Dilate/Erode***

Enlarge (positive numbers) or shrink (negative numbers) the matte by the specified number of pixels. This is similar to using the Dilate/Erode Node on the matte.

This a simple way to include more or less along the edges of the matte, particularly combined with Post Blur.

## ***Feather Falloff***

The rate of the falloff at the edges of the matte when feathering, to manage edge detail.



## ***Feather Distance***

Controls how much the matte is feathered inwards (negative number) or outwards (positive number).

## ***Post Blur***

Make the matte less sharp, for smoother transitions to the background and noise reduction.

## **Outputs**

### ***Image***

Processed image with the Matte applied to the images' alpha channel.

### ***Matte***

Output matte to use for checking the quality of the key, or to manually apply using a Set Alpha Node or Mix Node.

### ***Edges***

Shows what edges were detected on the matte. Useful for adjusting the Edge Kernel Radius and Edge Kernel Tolerance.

#### Tip

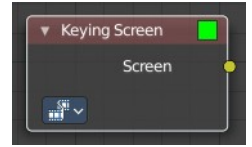
If there are problems with the edges of the matte, it may help to start with adjusting the Edge Kernel parameters before adjusting feathering. Detected edges are not subject to Clip Black / Clip White thresholds to preserve fine edge detail. You can check edge detection by connecting a Viewer Node to the Edges output.

Sharper detected edges (smaller Edge Kernel Radius, like 2 / larger Edge Kernel Tolerance, like 0.4) will create a sharper matte, but may loose some detail like stray hairs. A sharp matte is good, but disappearing or flickering hairs are distracting.

Fat edges (larger Edge Kernel Radius, like 8 / smaller Edge Kernel Tolerance, like 0.05) will capture more edge detail, but may also produce a halo around the subject. The halo can be adjusted with Feather controls along with Dilate/Erode.

## Keying Screen

The Keying Screen node creates plates for use as a color reference for keying nodes. It generates gradients from sampled colors on motion tracking points on movie clips. It can be used to deal with uneven colors of green screens.

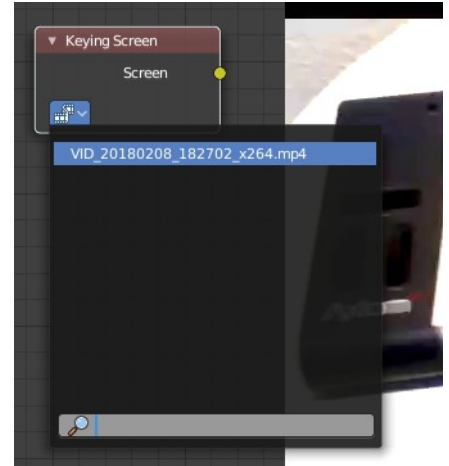


## Properties

### Movie Clip

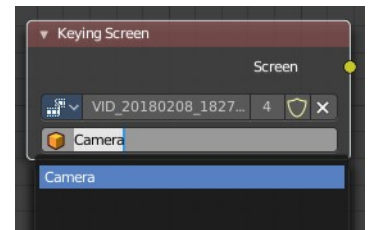
The selectable clip data-block used as input for the gradient colors.

The movie needs to be loaded already. You cannot load it from here, just choose.



### Tracking Object

Tracking Object to generate the gradient. You will probably want to create new a tracking object in the Object panel, because tracks used for gradients can not actually be used for camera/object tracking. After this tracks might be placed in places where gradient colors should be sampled. These tracks could be tracked or moved manually, so gradients would be updating automatically along the movie. Tracks might have an offset for easier tracking of feature-less screens.



## Outputs

### Screen

Gradient image output.

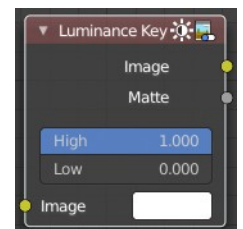
## Luminance Key

The Luminance Key node determines background objects from foreground objects by the difference in the luminance (brightness) levels.

Stock footage of explosions, smoke or debris are normally shot against a solid, dark background rather than a green screen. This node can separate the foreground effect from the background. It can also be used for sky replacement for overexposed or gray skies that aren't suitable for chroma keying.

### Tip

When compositing footage of something that emits light and has a dark background, like fire, a Mix Node using a Screen or Add operator will produce better results.



## Inputs

### *Image*

Standard image input.

## Properties

### *Limit*

#### **High**

Determines the lowest values that are considered foreground. (Which is supposed to be – relatively – light: from this value to 1.0.)

#### **Low**

Determines the highest values that are considered to be background objects. (Which is supposed to be – relatively – dark: from 0.0 to this value.)

Note. Brightness levels between the two values form a gradient of transparency between foreground and background objects.

## Outputs

### *Image*

Image with an alpha channel adjusted for the keyed selection.

### *Matte*

A black-and-white alpha mask of the key.