

[Skip to content](#)

Image or Movie

The term *Image Texture* simply means that a graphic image, which is a pixel grid composed of R, G, B, and sometimes Alpha values. It is used as the input source to the texture. As with other types of textures, this information can be used in a number of ways, not only as a simple “decal”.

Video textures are a some kind of Image textures and based on movie file or sequence of successive numbered separate images. They are added in the same way that image textures are.

When the Texture Type *Image or Movie* is selected, three new panels present themselves allowing to control most aspects of how image textures are applied: *Image*, *Image Sampling*, and *Image Mapping*.

About Image-Based Texturing

Texture images take up precious memory space, often being loaded into a special video memory bank that is very fast and very expensive, so it is often very small. So, keep the images as small as possible. A 64×64 image takes up only one fourth the memory of a 128×128 image.

For photorealistic rendering of objects in animations, often larger image textures are used, because the object might be zoomed in on in camera moves. In general, you want to use a texture sized proportionally to the number of pixels that it will occupy in the final render. Ultimately, you only have a certain amount of physical RAM to hold an image texture and the model and to provide workspace when rendering your image.

For the most efficient memory usage, image textures should be square, with dimensions as powers of 2, such as 32×32, 64×64, 128×128, 256×256, 1024×1024, 2048×2048, and 4096×4096.

If you can reuse images across different meshes, this greatly reduces memory requirements. You can reuse images if you map those areas of the meshes that “look alike” to a layout that uses the common image.

When using file textures, it is very important that you have [Mapped the UVs](#) of the mesh, and they are laid out appropriately.

You do not have to UV map the *entire* mesh. The sphere above on the left has some faces mapped, but other faces use procedural materials and textures. Only use UV textures for those portions of your mesh where you want very graphic, precise detail. For example, a model of a vase only needs UV textures for the rim where decorative artwork is incorporated. A throw pillow does not need a different image for the back as the front; in fact many throw pillows have a fabric (procedural material) back.

As another example, you should UV map both eyes of a head to the same image (unless you want one bloodshot and the other clear). Mapping both sides of a face to the same image might not be advisable, because the location of freckles and skin defects are not symmetrical. You could of course change the UV map for one side of the face to slightly offset, but it might be noticeable. Ears are another example where images or section of an image can be mapped to similar faces.

Options

Image

The Image [Data-Block Menu](#).

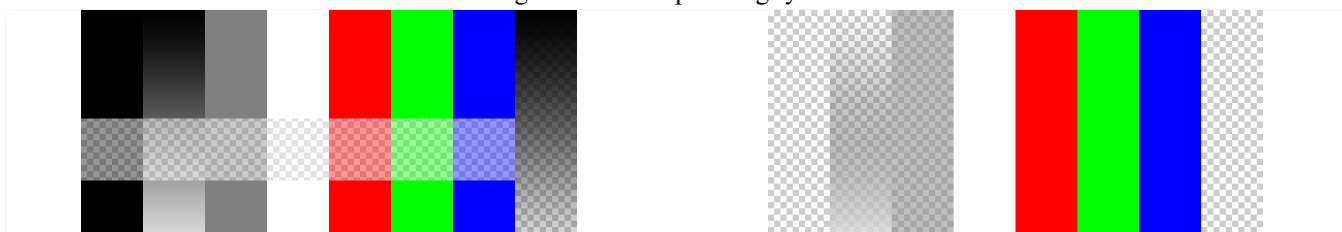
Alpha

Use the alpha channel information stored in the image. Where the alpha value in the image is less than 1.0, the object will be partially transparent and thin behind it will be visible. Works with [image formats](#) that store transparency information.

Calculate

Calculate an alpha based on the RGB values of the Image. Black (0, 0, 0) is transparent, white (1, 1, 1) opaque. Enable this option if the image texture is a mask. Note that mask images can use shades of gray that result in semi-transparency, like ghosts, flames, and smoke/fog.

The image with various alpha and gray-scale values.





Invert

Reverses the alpha value. Use this option if the mask image has white where you want it transparent and vice versa.

Mapping

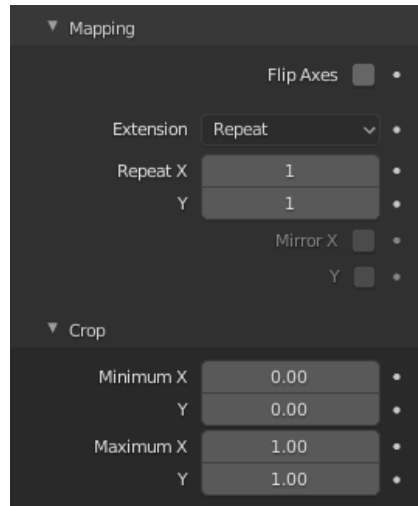


Image Mapping panel.

In the *Mapping* panel, you can control how the image is mapped or projected onto the 3D model.

Flip Axes

Rotates the image 90 degrees counterclockwise when rendered.

Extension

How the image is extrapolated beyond its original bounds.

Extend:

Outside the image the colors of the edges are extended.

Clip:

Clip to image size and set exterior pixels as transparent. Outside the image, an alpha value of 0.0 is returned. This allows you to 'paste' a small logo on a large object.

Clip Cube:

Clips to cubic-shaped area around the images and sets exterior pixels as transparent. The same as Clip, but now the 'Z' coordinate is calculated as well. An alpha value of 0.0 is returned outside a cube-shaped area around the image.

Repeat:

The image is repeated horizontally and vertically.

Repeat X, Y

X/Y repetition multiplier.

Mirror X, Y

Mirror on X/Y axes. These buttons allow you to map the texture as a mirror, or automatic flip of the image, in the corresponding X and/or Y direction.

Checker:

Checkerboards quickly made. You can use the option *size* on the *Mapping* panel as well to create the desired number of checkers.

Tiles Even/Odd

Set even/odd tiles.

Distance

Governs the distance between the checkers in parts of the texture size.

Crop

Minimum X, Y / Maximum X, Y

The offset and the size of the texture in relation to the texture space. Pixels outside this space are ignored. Use these to crop, or choose a portion of a larger image to use as the texture.

Sampling

In the *Sampling* panel you can control how the information is retrieved from the image.

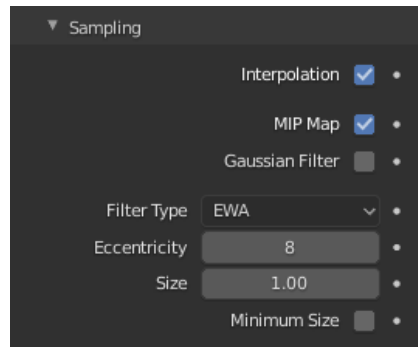
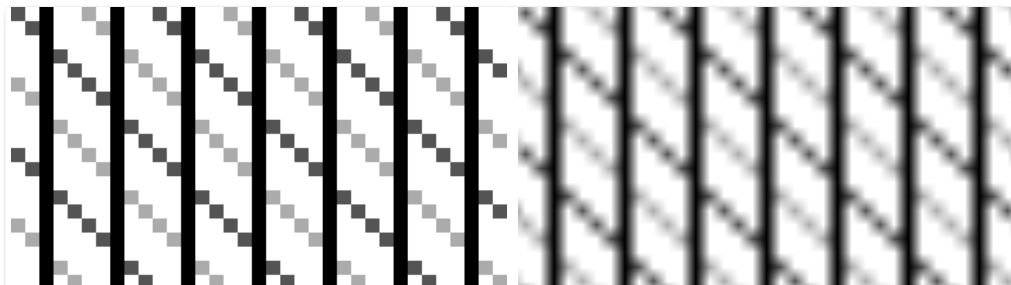


Image Sampling panel.

Interpolation

This option interpolates the pixels of an image. This becomes visible when you enlarge the picture. By default, this option is on. Turn this option off to keep the individual pixels visible and if they are correctly anti-aliased. This last feature is useful for regular patterns, such as lines and tiles; they remain 'sharp' even when enlarged considerably. Turn this image off if you are using digital photos to preserve crispness.



Enlarged Image texture without *Interpolation*.

Enlarged Image texture with *Interpolation*.

MIP Map

Mip-maps are precalculated, smaller, filtered textures for a certain size. A series of pictures is generated, each half the size of the former one. This optimizes the filtering process. By default, this option is enabled and speeds up rendering. When this option is off, you generally get a sharper image but this can significantly increase calculation time if the filter dimension (see below) becomes large. Without mip-maps you may get varying pictures from slightly different camera angles, when the textures become very small. This would be noticeable in an animation.

Gaussian Filter

Used in conjunction with mip-mapping, it enables the mip-map to be made smaller based on color similarities. In game engines, you want your textures, especially your mip-map textures, to be as small as possible to increase rendering speed and frame rate.

Filter Type

Texture filter to use for image sampling. Just like a *pixel* represents a *picture element*, a *texel* represents a *texture element*. When a texture (2D texture space) is mapped onto a 3D model (3D model space), different algorithms can be used to compute a value for each pixel based on samples from several texels.

Box:

A fast and simple nearest-neighbor interpolation known as Monte Carlo integration.

Box / Gaussian / EWA / Linear / Anisotropic

EWA (Empirical weighted Average):

One of the most efficient direct convolution algorithms developed by Paul Heckbert and Ned Greene in the 1980s. For each texel, EWA samples, weights, and accumulates texels within an elliptical footprint and then divides the result by the sum of the weights.

Eccentricity

Maximum Eccentricity. Higher values give less blur at distant/oblique angles, but is slower.

FELINE (Fast Elliptical Lines):

Uses several isotropic probes at several points along a line in texture space to produce an anisotropic filter to reduce aliasing artifacts without considerably increasing rendering time.

Light Probes

Number of probes to use. An integer between 1 and 256. Further reading: McCormack, J; Farkas, KI; Perry, R; Jouppe, NP (1999) [Simple and Table Feline: Fast Elliptical Lines for Anisotropic Texture Mapping](#), WRL

Area:

Area filter to use for image sampling.

Eccentricity

Maximum Eccentricity. Higher values give less blur at distant/oblique angles, but is slower.

Size

The filter size used in rendering, and also by the options *Mip Map* and *Interpolation*. If you notice gray lines or outlines around the textured object particularly where the image is transparent, turn this value down from 1.0 to 0.1 or so.

Minimum Size

Use Filter Size as a minimal filter value in pixels.

[Previous](#)
[Distorted Noise](#)

[Copyright](#) © : This page is licensed under a CC-BY-SA 4.0 Int. License

Made with [Furo](#)

Last updated on 2025-05-10

[View Source](#)
[View Translation](#)
[Report issue on this page](#)

[No](#)
[Ma](#)