10.2.2.6 Render - Blender Render Engine - Textures - Influence

Influence

- Material Textures Influence
  - Surface and Wire materials
  - Volume materials
  - Halo materials
- Texture Blending Modes
- Bump and Normal Maps
  - Description
  - Workflow
- Displacement Maps
  - Description
  - Options
  - Hints
- World
- Particles

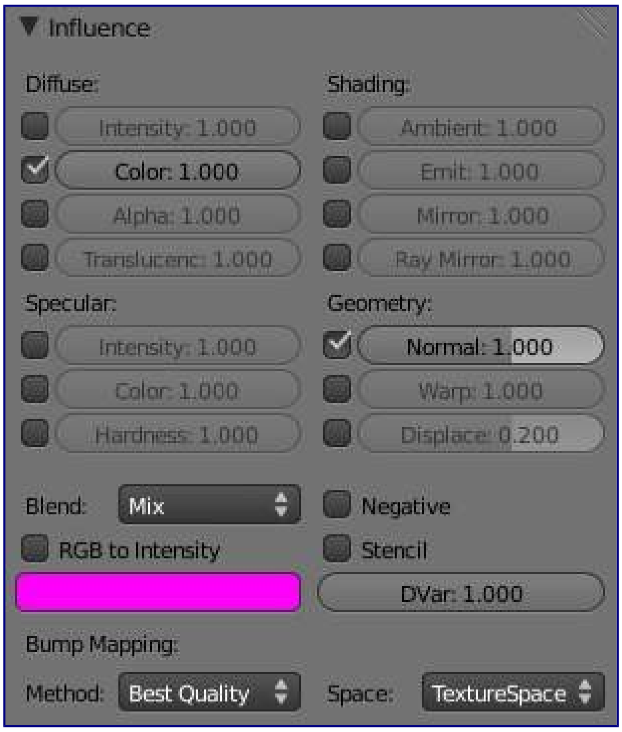
Material Textures Influence

Not only can textures affect the color of a material, they can also affect many of the other properties of a material. The different aspects of a material that a texture influences are controlled in the *Influence* panel.

Note

Texture options for *Surface* and *Wire* materials and in some cases also for *Volume* and *Halo* materials.

Surface and Wire materials



Texture Influence panel for a Surface material

Diffuse

Intensity
: Amount texture affects affects diffuse reflectivity

Color
: Amount texture affect the basic color or RGB value of the material

Alpha
: Influences the opacity of the material. Also use *Z Transparency* for light and if combining multiple channels.

Translucency
: Influences the Translucency amount.

Specular

Intensity
: Amount texture affect specular reflectivity

Color
: Influences the *Specular* color, the color of the reflections created by the lamps on a glossy material.

Hardness
: Influences the specular hardness amount. A DVar of 1 is equivalent to a Hardness of 130, a DVar of 0.5 is equivalent to a Hardness of 65.

Shading

Ambient
Influences the amount of Ambient light the material receives.
Emit
Influences the amount of light Emitted by the material.
Mirror
Influences the mirror color. This works with environment maps and raytraced reflection.
Ray Mirror
Influences the strength of raytraced mirror reflection.

Geometry

Normal
Commonly called bump mapping, this alters the direction of the surface normal. This is used to fake surface imperfections or unevenness via bump mapping, or to create reliefs.
Warp
Warp allows textures to influence/distort the texture coordinates of a next texture channel. The distortion remains active over all subsequent channels, until a new Warp has been set. Setting the factor at zero cancels out the effect.
Displace
Influences the Displacement of vertices, for using *Displacement Maps*.

Other Controls

Blend
Blending operation to perform. See *Texture Blending Modes* for details.
RGB to intensity
With this option enabled, an RGB texture (affects color) is used as an intensity texture (affects a value).
Blend Color
If the texture is mapped to Col, what color is blended in according to the intensity of the texture? Click on the swatch or set the RGB sliders.
Negative
The effect of the Texture is negated. Normally white means on, black means off, *Negative* reverses that.
Stencil
The active texture is used as a mask for all following textures. This is useful for semitransparent textures and “Dirt Maps”. Black sets the pixel to “untexturable”. The *Stencil* mode works similar to a layer mask in a 2D program. The effect of a stencil texture can not be overridden, only extended. You need an intensity map as input.
DVar
Destination Value (not for RGB). The value with which the Intensity texture blends with the current value. Two examples:

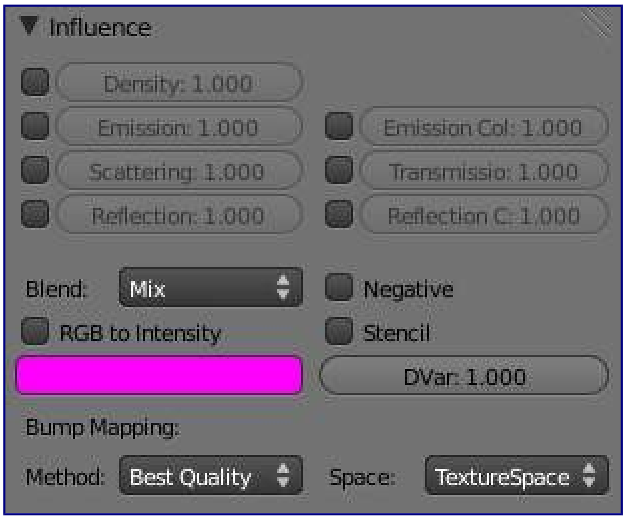
- The *Emit* value is normally 0. With a texture mapped to *Emit* you will get maximal effect, because *DVar* is 1 by default. If you set *DVar* to 0 no texture will have any effect.
- If you want transparent material, and use a texture mapped to *Alpha*, nothing happens with the default settings, because the *Alpha* value in the *Material* panel is 1. So you have to set *DVar* to 0 to get transparent material (and of course *Z Transparency* also). This is a common problem for beginners. Or do it the other way round - set *Alpha* to 0 and leave *Dvar* on 1. Of course the texture is used inverted then.

Bump Mapping

Settings for bump mapping. *Method Best Quality*, *Default*, *Compatible*, *Original Space*

Texture Space, *Object Space*, *View Space*

Volume materials



Texture Influence panel for Volume material

Special texture options for *Volume* materials

Density
: Causes the texture to affect the volume's density.

Emission
: Causes the texture to affect the volume's emission.

Scattering
: Amount the texture affects scattering.

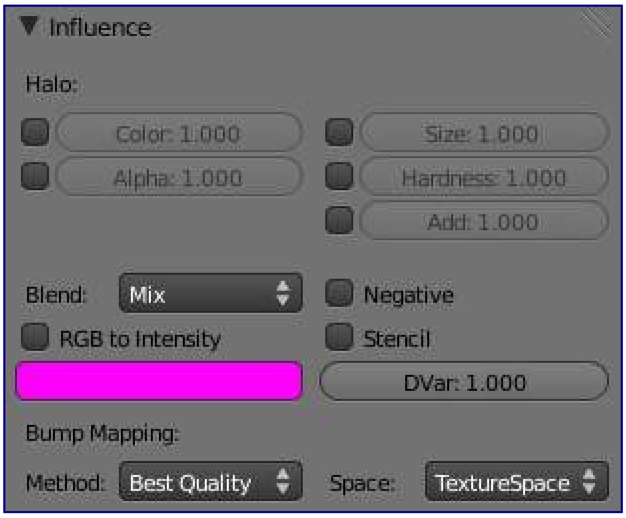
Reflection
: Amount the texture affects brightness of out-scattered light

Emission Color
: Amount the texture affects emission color.

Transmission
: Amount the texture affects result color after light has been scattered/absorbed.

Reflection Color
: Amount the texture affects color of out-scattered light.

Halo materials



Texture Influence panel for a Halo material

Special texture options for *Halo* materials

Size
: Amount the texture affects ray mirror.

Hardness
: Amount the texture affects hardness.

Add
: Amount the texture affects translucency.

Texture Blending Modes

Blending Modes are different methods of controlling how the texture influences material properties. While a blending mode defines the specific operation performed, blending factor controls the amount, the overall “strength” of this operation. For textures such blending factor is set via sliders in the Influence panel. Throughout this section, the term *base layer* refers to the base material color being manipulated (as defined by texture’s Influence) and *blend layer* refers to the texture. Following is a list of available texture blending modes:

See also

Color Blend Modes for details on each blending mode.

Bump and Normal Maps

Description

Normal Maps and *Bump Maps* both serve the same purpose: they simulate the impression of a detailed 3D surface, by modifying the shading as if the surface had lots of small angles, rather than being completely flat. Because it’s just modifying the shading of each pixel, this will not cast any shadows and will not obstruct other objects. If the camera angle is too flat to the surface, you will notice that the surface is not really shaped.

Both *Bump Maps* and *Normal Maps* work by modifying the normal angle (the direction pointing perpendicular from a face), which influences how a pixel is shaded. Although the terms *Normal Map* and *Bump Map* are often used synonymously, there are certain differences.

Bump maps

These are textures that store an **intensity**, the relative height of pixels from the viewpoint of the camera. The pixels seem to be moved by the required distance in the direction of the face normals. (The “bump” consists only of a displacement, which takes place along the existing, and unchanged, normal-vector of the face.) You may either use greyscale pictures or the intensity values of a RGB-Texture (including images).

Normal maps

These are images that store a **direction**, the direction of normals directly in the RGB values of an image. They are much more accurate, as rather than only simulating the pixel being away from the face along a line, they can simulate that pixel being moved at any direction, in an arbitrary way. The drawbacks to normal maps are that unlike bump maps, which can easily be painted by hand, normal maps usually have to be generated in some way, often from higher resolution geometry than the geometry you’re applying the map to.

Normal maps in Blender store a normal as follows:

- Red maps from (0-255) to X (-1.0 - 1.0)
- Green maps from (0-255) to Y (-1.0 - 1.0)
- Blue maps from (0-255) to Z (0.0 - 1.0)

Since normals all point towards a viewer, negative Z-values are not stored (they would be invisible anyway). In Blender we store a full blue range, although some other implementations also map blue colors (128-255) to (0.0 - 1.0). The latter convention is used in “Doom 3” for example.

Workflow

The steps involved in making and using Bump and Normal Maps is:

- Model a highly detailed (“hi-poly”) model
- Bake the Bump and/or Normal maps
- Make a low-poly, less detailed model
- Map the map to the low-poly model using a common coordinate system

Consult the Modeling section for how to model a highly detailed model using the Mesh tools. How much detail you put in is totally up to you. The more ridges and details (knobs, creases, protrusions) you put in, the more detailed your map will be.

Baking a map, simply put, is to take the detail of a high polygon mesh, and apply it to a similar object. The similar object is identical to the high-poly mesh except with less vertices. Use the *Render Bake* feature in Blender to accomplish this.

Modeling a low-poly using Blender’s Mesh editing tools. In general, the same or similar faces should exist that reflect the model. For example, a highly detailed ear may have 1000 faces in the high-poly model. In the low-poly model, this may be replaced with a single plane, oriented in the same direction as the detailed ear mesh. *(Tip:* Blender’s *multi-resolution mesh* modeling feature can be used to good effect here.)

Mapping is the process of applying a texture to the low-poly mesh. Consult the *Textures Mapping section* for more information on applying a texture to a mesh’s material.Special considerations for Bump and Normal Maps is:

- When using a Bump map, map the texture to *Normal* and enable *No RGB*.
- When using a Normal map, map the texture to *Normal*.

The coordinate systems of the two objects must match. For example, if you bake using a UV map of the high-poly model, you must UV map the low poly model and line up its UV coordinates to match the outline of the high-poly image (see UV unwrapping to line up with the high-poly map edges.

Displacement Maps

Description

Displacement mapping allows a texture input to manipulate the position of vertices on rendered geometry. Unlike *Normal or Bump mapping*, where the shading is distorted to give an illusion of a bump (discussed on the previous page), Displacement Maps create real bumps, creases, ridges, etc in the actual mesh. Thus, the mesh deformations can cast shadows, occlude other objects, and do everything that changes in real geometry can do, but, on the other hand, requires a lot more vertices to work.

Options

In the *Influence panel*, the strength of the displacement is controlled by the *Displace* and *Normal* sliders.

- If a texture provides only normal information (e.g. *Stucci*), vertices move according to the texture's normal data. The normal displacement is controlled by the *Normal* slider.
- If a texture provides only intensity information (e.g. *Magic*, derived from color), vertices move along the directions of their normals (a vertex has no normal itself, it's the resulting vector of the adjacent faces). White pixels move outward in the direction of the normal, black pixels move in the opposite direction. The amount of displacement is controlled with the *Displace* slider.

The two modes are not exclusive. Many texture types provide both information (*Clouds*, *Wood*, *Marble*, *Image*). The amount of each type can be mixed using the respective sliders. Intensity displacement gives a smoother, more continuous surface, since the vertices are displaced only outward. Normal displacement gives a more aggregated surface, since the vertices are displaced in multiple directions.

The depth of the displacement is scaled with an object's scale, but not with the relative size of the data. This means if you double the size of an object in object mode, the depth of the displacement is also doubled, so the relative displacement appears the same. If you scale inside *Edit Mode*, the displacement depth is not changed, and thus the relative depth appears smaller.

Hints

Displacement maps move the rendered faces, not the physical mesh faces. So, in 3D View the surface may appear smooth, but render bumpy. To give a detailed surface, there has to be faces to displace and have to be very small. This creates the trade-off between using memory and CPU time versus render quality.

From best to worst, displacement works with these object types using the methods listed to control the render face size:

***Subdivision Surface* Meshes**
: Rendered face size is controlled with render subsurf level. Displacement really likes smooth normals.

Manually (*Edit Mode*) *subdivided* meshes
: Control render faces with number of subdivides. (This can be combined with the above methods). Displaces exactly the same Simple Subsurf, however the overhead of drawing extra faces can slow down editing.

Meta Objects
: Control render faces with render wiresize. Small wire == more faces.

The following are available, but currently don't work well. It is recommended that you convert these to meshes before rendering.

Open *NURBS Surfaces*
: Control render faces with U/V *Surface Resolution*. Higher numbers give more faces. (Note normal errors).

Closed NURBS Surfaces
: Control with *Surface Resolution* controls. (Note the normal errors, and how implicit seam shows).

Curves* and *Text
: Control with *Surface Resolution* controls. Higher gives more render faces. (Note that the large flat surfaces have few render faces to displace).

Note
Displace Modifier If you want more control over your displacement, you'll probably want to use the *Displace Modifier.* This feature has lots of different options so that you can customize the displacement exactly to your

liking.

World

TODO: https://developer.blender.org/T46363

Particles

TODO: https://developer.blender.org/T46363