Material Properties

Table Of Contents

1 Material Type

The most important material property is the **material type**. This is the first property you must define. The material type defines the basic character of the material. Every material type has a different cost for rendering. Also what settings are available is also defined by the material type.

Internally every material type corresponds to a shader pack.

At the moment LOTUS offers the following material types:

- *Simple*: This material is very efficient to render, however it only supports transparency (hard/soft), glossiness (also using the Alpha channel), snow and night textures. That makes this type mainly suitable for normal scenery objects.
- *Direct Texture, no light:* This material ONLY renders a texture, no effects are used at all.
- *Terrain*: The material is mainly intended for the materials for surface polygons in the Map-Editor. In contrast to *Simple* it does not allow snow textures (these are handled in another way for the surface) and night textures. It does provide a Detail texture, which gives the material an additional structure when in close proximity.
- *Complex*: This material is a universally applicable material, especially for normal vehicle surfaces. In addition to the effects of the *Simple* type, a detail texture and ambient occlusion can be used, as well as the intensity of the night texture.
- *Complex with normal map*: Similar to *Complex*, additionally allows for a normal texture and for Relief Mapping.
- *Raindrops:* Special material for windows that are wet with raindrops. Discussed in another article.
- *Human:* Will later be used for AI people. Please do not use now.
- *Complex+Normal+Simple Meshanim:* Extension to *Complex with normal map:* Using this material you can create simple Mesh animations. See Mesh animations
- *User Vehicle Windows:* Special material. **MUST** be used for the inside of the front window of a vehicle. This material is connected to the graphical settings "Enlight windows". This provides, among others, a significant performance boost!
- *Roller Blind Display:* Special material for roller blind displays, see Tutorial: Rollbandanzeige .

2 Effects 101: Simple effects

To motivate you for a quick start, while also laying a basis for the parameters we will discuss later, we show you some typical effects and how to set them up:

2.1 Material without transparency

- Blend mode *Normal rendering*
- *Min. Alpha* = 0

2.2 Material with "soft" transparency

- Standard texture must have an Alpha channel (black = transparent, white = opaque)
- Blend mode *Alpha transparency*
- *Min. Alpha* = 0 or, to reduce possible sequence failures 0.1 - 0.5

2.3 Material with "hard" transparency

- Blend mode *Normal rendering*
- *Min. Alpha* = 0.5 (0.1 - 0.9, depending on what parts need to remain

2.4 Matte material

- *Shininess factor* = 0
- *Use alpha value for shininess* off

2.5 Constant semi-shiny material

- *Shininess factor* = 0.25
- *Brightness of the highlight* = 1
- *Use alpha value for shininess* off
- *More semi-matte characteristic* on

2.6 Constant high gloss material

- *Shininess factor* = 1
- *Brightness of the highlight* = 1
- *Use alpha value for shininess* off

2.7 Material with varying shininess

- Standard texture must have an Alpha channel (black = matte, white = high gloss)
- *Shininess factor* = 0.25
- *Brightness of the highlight* = 1
- *Use alpha value for shininess* on
- *More semi-matte characteristic* off

2.8 Material with a night texture, controlled automatically (globally)

- Set *Building night variable* to the variable *NightTex*
- Add the prepared texture for *Night texture*
- For *Complex* or *Complex with normal map*: *Brightness of the night texture* = 1

2.9 Material with an always active night texture

- *Building night variable* = 1
- Add the prepared texture for *Night texture*
- For *Complex* or *Complex with normal map*: *Brightness of the night texture* = 1

2.10 Snow texture, completely covering

- *Use alpha value for snow* off
- *Snow texture* = *snow.dds* (or choose your own)

2.11 Snow texture with varying covering

- Standard texture must have an Alpha channel (black = no snow, white = completely covered with snow)
- *Use alpha value for snow* on

3 Clarifying Alpha channel

For some settings, the term *Alpha channel* is crucial

3.1 General

Even though the alpha channel is often understood as transparency, this is not necessarily true!

To represent a texture, only one red, one green and one blue value is needed for each pixel. These are called channels. However, for technical reasons each pixel also provides memory for a fourth value, we call this the alpha value. Every texture has four channels: the red, the green, the blue and the *Alpha* channel.

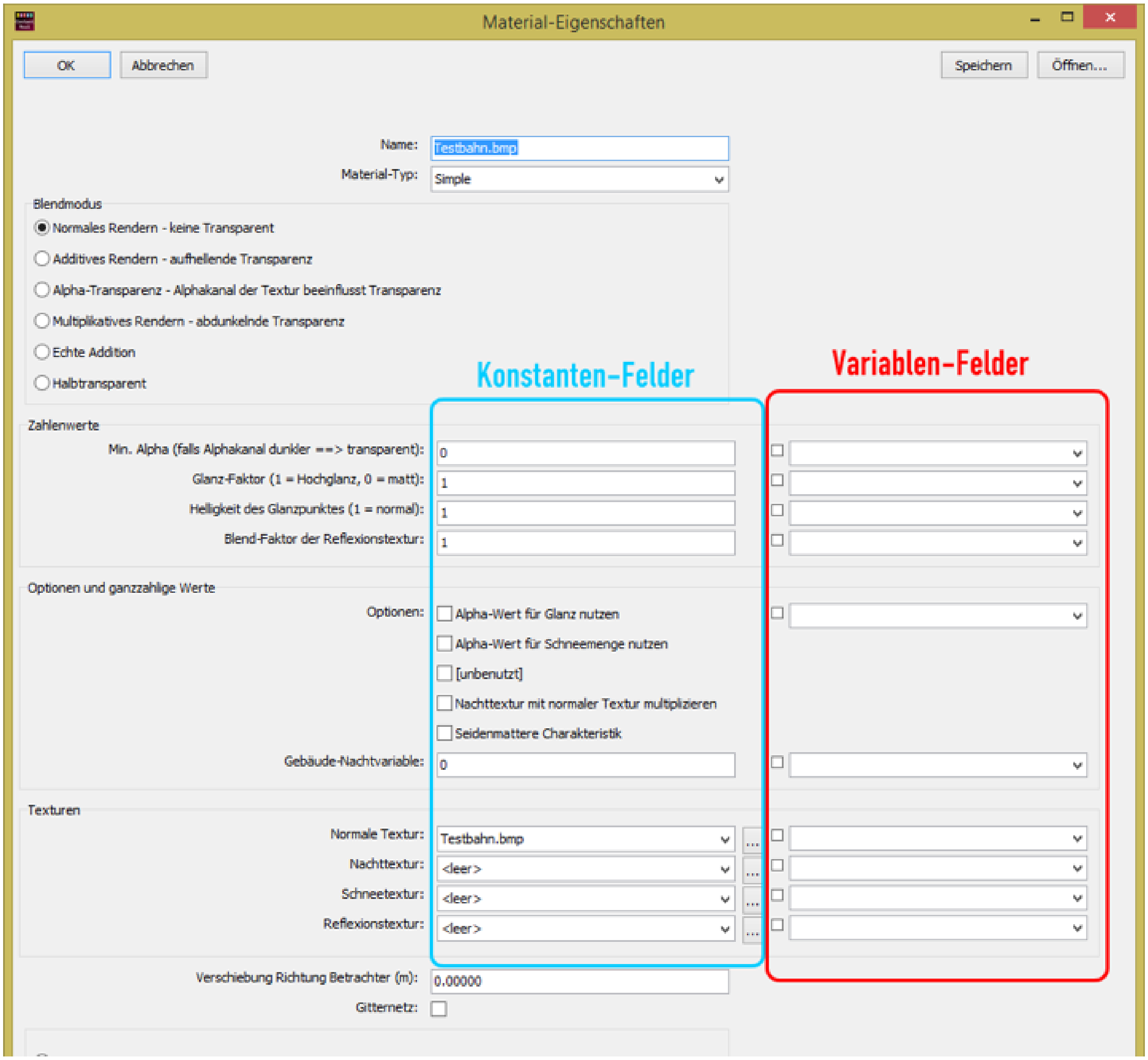
3.2 Texture and resulting alpha channel

Rendering a pixel with a specific material can be thought of as a "Computation task": it goes through various texture information and parameters to find the color and alpha value for the pixel.

Henceforth a distinction is made between the alpha channel of a texture and the *resulting alpha channel*, i.e. the calculated resulting alpha value for a pixel.

4 Script and variable connections

Most material properties can be set, as you prefer, to a constant or to the value of a variable



Name: Testbahn.bmp

Material-Typ: Simple

Blendmodus

● Normales Rendern - keine Transparent

○ Additives Rendern - aufhellende Transparenz

○ Alpha-Transparenz - Alphakanal der Textur beeinflusst Transparenz

○ Multiplikatives Rendern - abdunkelnde Transparenz

○ Echte Addition

○ Halbtransparent

Zahlenwerte

Min. Alpha (falls Alphakanal dunkler ==> transparent): 0 □

Glanz-Faktor (1 = Hochglanz, 0 = matt): 1 □

Helligkeit des Glanzpunktes (1 = normal): 1 □

Blend-Faktor der Reflexionstextur: 1 □

Optionen und ganzzahlige Werte

Optionen: □ Alpha-Wert für Glanz nutzen □

□ Alpha-Wert für Schneemenge nutzen

□ [unbenutzt]

□ Nachttextur mit normaler Textur multiplizieren

□ Seidenmattere Charakteristik

Gebäude-Nachtvariable: 0 □

Texturen

Normale Textur: Testbahn.bmp ... □

Nachttextur: <leer> ... □

Schneetextur: <leer> ... □

Reflexionstextur: <leer> ... □

Verschiebung Richtung Betrachter (m): 0.00000

Gitternetz: □

These parameters do not only have a entry field for the relevant value, they also have, after that field a checkbox and dropdown selection. Using the dropdown you can select variable that is global or defined in the included script. This marks the checkbox automatically.

Now the value of the variable will control this property. It is important to know, especially for floating point numbers, that the value of the variable is multiplied with the constant value in the first field. If you want to directly use the variable value, you have to enter a 1 there.

To remove the coupling with the variable, simple uncheck the checkbox.

5 Material properties

Here the material properties are described in detail. The parameters are discussed grouped by their effect. In the properties dialog screen these settings can show up in different locations and orderings, because there they are shown based on the data type.

5.1 Transparency

There are four options for transparency: Hard, soft, additive and multiplicative.

5.1.1 Hard

For this the material is either completely visible or completely hidden, with hard edges and without the option for semi-transparency. The standard texture must have an alpha channel where 0/black = transparent and 1/white = not transparent.

For this the value *Min. alpha* should be set to 0.5. You can also use other values > 0 and < 1 which moves the visibility cutoff value.

5.1.2 Soft

Here the transition between not transparent, semi-transparency and fully transparent is smooth. A downside is that the ordering of the draw calls can influence the result and this can lead to issues: sometimes the objects behind the transparent surface are not visible, leading to a "Röntgen view". It is strongly recommended to, where possible, select the "Hard" transparency; of course avoiding this soft transparency is not possible for semi-transparent objects like glass.

There are two options: Either a constant semi-transparency - no alpha channel is needed. For this the *Blend mode* is set to *Half transparent*.

Or the alpha channel controls the transparency: for this the standard texture should obviously have an alpha channel, where 0/black = transparent and 1/white = not transparent. The *Blend mode* should be set to *Alpha transparency*.

In this case you can also use the special *Alpha scale* to further manipulate the transparency. If you attach a script variable there then the script can control the transparency by setting a value between 0 and 1.

5.1.3 Additive

This type of transparency does not use an alpha channel. It uses the actual color of the surface to be drawn to light the background. The brighter the color the more the background will be lighted - if the surface color is black, nothing will happen. To do this, set the *Blend mode* to *Additive rendering* or *Real addition*. *Additive rendering* has be corrected a bit to work more naturally whereas *Real addition* uses the exact mathematics.

5.1.4 Multiplicative rendering

This blend mode works like a (color) filter. This can be used e.g. realize a tinted window, where the color value of the material corresponds to the tinting color.

5.1.5 Hide with increasing distance

If the material type "Complex with Normal Map" is selected and the alpha channel controlled, soft transparency is activated, it is possible to gradually make the mesh more transparent with increasing distance until finally it is completely invisible. For this purpose, the value of *"Hide with increasing distance"* must be set to a value greater than 0 and usually less than 1. The smaller the value, the further the object is visible. The range in which an object is visible is only one meter for a value of "1", it increases to 10m for "0.1" and 100m for "0.01".

5.2 Glossiness

5.2.1 Lighting and highlight

One part of the gloss is the question of how the object reacts being lighted by the surrounding light sources. Objects can be rendered dull, silky or glossy (or any characteristic in between) and the highlight may be lighter or darker.

First of all, it has to be distinguished whether the alpha channel of the standard texture should be used for the gloss or the parameters should be constant. Depending on this, the option *Use alpha value for gloss* can be switched on or off.

The gloss is controlled by the parameters *Shininess factor* and *Brightness of the highlight*. If the gloss is constant, *Shininess factor* can be used to set the object to be matte (0), semi-gloss (0.1 or 0.25, see below) or high-gloss (1).

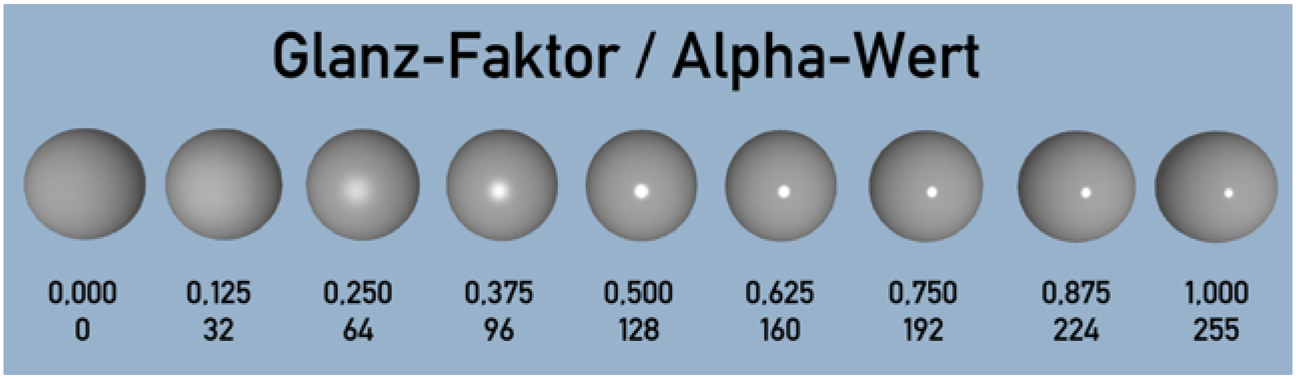
If the alpha channel is enabled, then it is multiplied by the parameter *Shininess Factor* so that it represents the maximum value. As a result, when the alpha channel is enabled, the *Shininess factor* value should normally be 1.

Brightness of the highlight should normally be 1; when needed it can be used to darken or brighten the highlight.

Also important is the checkbox *More semi-matt characteristic*. For objects where the alpha channel is mainly used to distinguish high-gloss and matte surfaces, such in buildings, the checkbox should be off. The mapping alpha value ==> Shininess is then optimized to represent the High Gloss / Matte transition with as few errors as possible.

However, if you want to use the alpha value (or even without using the alpha channel) to create a variety of different gloss levels, then it is best to turn on the *More semi-matte characteristic*, which will allow for better control.

Here are some examples for the shininess factor with applied *More semi-matte characteristic* applied to a grey ball:



5.2.2 Reflection texture

A technically completely different effect is reflection textures. Reflection textures create seemingly more or less reflective surfaces which reflect the surroundings of the object. This can either be an image of the environment generated by LOTUS itself in real time ("real-time reflections") or a constant texture that has already been generated in advance.

The reflection is controlled by the *Blend factor of the reflection texture.* This indicates how much of the reflection texture should be visible. In addition, the already known *Use alpha value for gloss* option causes the blend factor to be multiplied by the alpha channel.

Of course, a reflection texture must be selected. By default, LOTUS includes the two textures *sharp_envir.bmp* and *smooth_envir.bmp*. You can also use your own textures - these have to be so-called Cube Maps (https://en.wikipedia.org/wiki/Cube_mapping). LOTUS offers the possibility to create Cube Maps as a screenshot; but that is a topic for another article.

The special feature of *sharp_envir.bmp* and *smooth_envir.bmp* is that they are automatically replaced if the user enables the option for Realtime reflections. The difference is that the first shows the surroundings sharp (in focus) (for chrome-plated/mirrored surfaces or glass) whereas the latter shows the surroundings blurred (for normal metals etc.).

Important: Although the reflection texture can be adjusted with the options depending on the alpha channel, it is always independent of the *Shininess factor* and *Brightness of the highlight*!

5.3 Night texture

The night texture is placed on top of the standard texture (additive) and is always "bright", it is independent from the position of the sun or other illuminations of the object.

You can select the texture itself under "Night texture".

Normally the night texture is simply placed on top of the standard texture. It is however possible to have a night texture that does not show the structure but that instead brightens the standard texture. For this you select the option *Multiplicate night text with standard tex*.

5.3.1 Automatic mode / night textures for buildings.

To let the night texture be controlled automatically, you must select the variable "NightTex" in *Building night variable*. To have the night texture always active simply enter a 1 there. When the night texture is controlled automatically an alpha channel in the night texture can be used to determine which areas light up at what times. Details can be found elsewhere, if the article is already available.

Manual mode / script operated

It is possible to control the night texture directly through your script. For this enter a 0 in *Building night variable* and make sure no variable is selected. Then you can control the texture directly using the *Brightness of the night texture*.

When the night texture is manually controlled, it should not have an alpha channel.

5.4 Translumination

That is, light can shine through an object. A typical example is leaves on a tree; when you look at a backlit tree you see the leaves are bright because the sun shines through them from the other side.

The translumination texture is set under *Night texture*, the option *Translumination* must also be set.

5.5 Retroreflexion

Retroreflection is the technical term for the light effect known from reflectors / "cat's eyes" and traffic signs and the like: The incident light is reflected exactly in the direction it comes from, so that the object is very bright to look at, especially when it is viewed from approximately the same position as the light source, e.g. when looking at traffic signs etc. in the dark with the headlights on from the front of the car.

This effect is only available to the "Complex" material type and is controlled by two retroreflection checkboxes in the "Options and Integer Values" section. The first one activates the retroreflection and is initially valid for the whole surface. With the second one you can make sure that the intensity is controlled depending on the alpha channel of the main texture.

5.6 Snow texture

LOTUS has an automatic snow effect. For this purpose, any snow texture - "snow.dds" should normally be used - is projected vertically from above onto the object. The visibility varies depending on the amount of snow fallen and the inclination angle of the respective surface, so that steep roof surfaces automatically get far less snow than flat roofs.

The snow texture is selected in the parameter with the same name.

For particularly structured surfaces, such as roof tiles, the alpha channel can also be added to influence the intensity; as a result, the original structure can be seen despite the snow. All you have to do is tick the *Use alpha value for snow*.

5.7 Detail Texture (only *Terrain*, *Complex*, *Complex with normal map, Complex+Normal+Simple Meshanim* and *Roller Blind Display*)

In order for surfaces to have a sufficient level of detail both far and near, it is possible to provide the "large-area" standard texture with a "small-scale" detail texture that is mixed in. On closer inspection the detail texture dominates the structure, while with remote viewing the standard texture dominates.

The detail texture can be activated via the parameter of the same name. The factor the detail texture is smaller is set in *Scaling of detail texture*.

The operation differs depending on whether or not the hook "Alpha controlled detail texture" is set.

If it is not set then the mixture is made as recommended for surface polygons. Also see Detailtexturen.

If itis set, then the alpha channel of the standard texture must be present and will be *exclusively* used for the mixing ratio: 0 / black = no detail texture, standard texture dominated, 1 / white = only detail texture, standard texture invisible. There is also a specialty: Although the alpha channel is mapped to the standard texture as the texture was originally mapped in the 3D program, the colors of the standard texture are mapped to the detail texture!

In fact, actually two detailed textures are mixed against each other, mixed over the "normal" mapped alpha channel.

5.8 Ambient Occlusion (only *Complex*, *Complex with normal map*, *Complex+Normal+Simple Meshanim* and *Roller Blind Display*)

Ambient occlusion is a technique in which the soft shadows that result from indirect ambient lighting are precalculated and stored in a separate texture. The reason for this is the very high computational effort required, which even today does not allow a real-time calculation and the fact that the ambient lighting is approximately constant.

There are three ways in which Ambient Occlusion (AO for short) can be realized in LOTUS:

- Per mesh vertex: This stores the brightness information in each vertex of the mesh. This can be done directly in the Content-Tool. Select the mesh, on the left in "Selected Mesh" all the way at the bottom "Per vertex Ambient Occlusion" choose the quality and click on "Generate". Do not select *Activate AO texture*

in the material properties.
- Using an AO texture, which is mapped exactly the same as the main texture: For this set *Activate AO texture* and *AO uses original mapping* and select the prepared AO texture for *Ambient Occlusion (AO) Texture*.
- Using an AO texture with different texture coordinates: For this set *Activate AO texture*, do not set *AO uses original mapping* and select the prepared AO texture for *Ambient Occlusion (AO) Texture*. The mesh must have secondary texture coordinates (see below).

In all cases the intensity of the AO can be adjusted with the *Intensity of AO* parameter.

Please see the main article about AO for more information: Ambient Occlusion einrichten

Normal Texture (only *Complex with normal map* and *Complex+Normal+Simple Meshanim*)

The position of a surface in the room is also described by the so-called normal. This is an imaginary line (vector) that is perpendicular to the surface. In particular, it influences how the calculation of light incidence and gloss takes place.

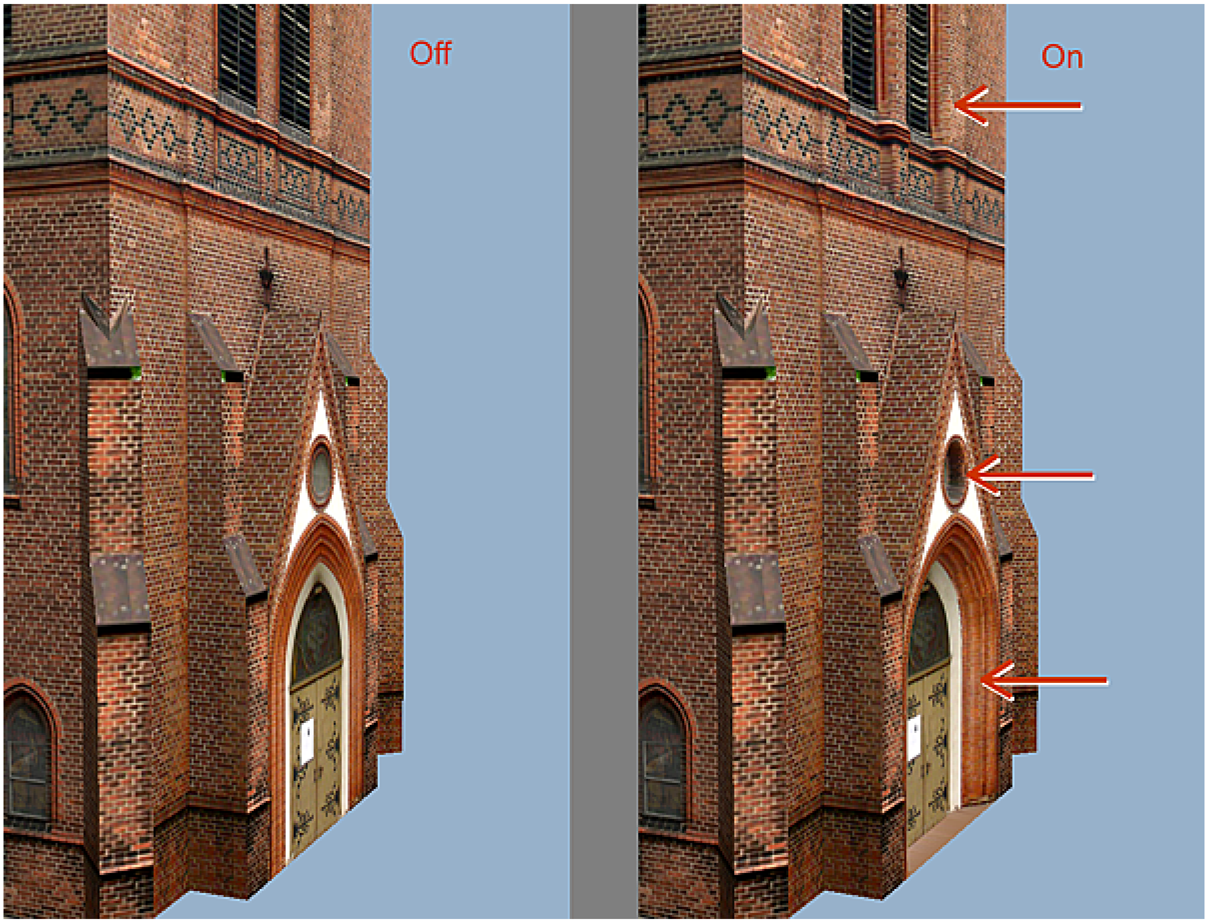
By means of a normal texture, this information can now be influenced using an extra texture. This can be used to simulate fine structures such as rivets or bumps in the material, which reflect the light differently than their surroundings. It is also commonly referred to as a *bump map*, but this is incorrect because a *bump map* contains other information than a normal texture (local height difference instead of surface normal).

In the dialog window, simply select the material type and add the texture for "Normal Texture". All applicable meshes must be provided with additional tangent information.

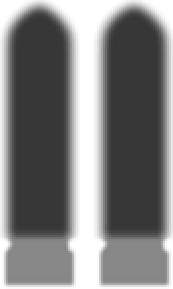
The intensity of the normal texture can be adjusted using the *Intensity of Normal texture* parameter.

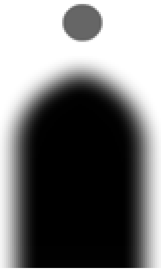
5.9 Parallax Mapping

This is a method in which a spatial depth is not only suggested to the observer by influencing the normal, but in the case of an oblique view can actually detect a vertical offset of the structure, as can be seen in the following example:



The configuration of the material is initially like a normal texture. In addition, the effect requires a normal texture with an alpha channel containing the height information. The associated normal map, for example, contains the following alpha channel:

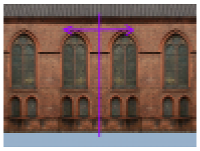




It can be seen that the deeper areas are darker than the higher areas. If areas are pure white, then they are in the actual polygon plane, so it looks like the parallax mapping is disabled.

Please note the following:

- Here, too, the tangent information must be added (see section Normal Texture).
- Lower levels require more performance than higher levels. The areas that are pure white need (compared to normal mapping) (almost) no higher computational effort.
- The depth is only simulated, the polygon still lies in the same place. Sometimes considerable display errors occur if the alpha channel is not white at the edges. In the example shown, the deeper windows are "modeled", which is basically possible without errors. In contrast, embossed structures require that the "base color" is no longer white, but a shade of gray. This leads to errors and artifacts at increasingly oblique viewing angles.
- There may be significant problems with certain polygon texture mappings. An example:

  Here, two polygons meet that have been mapped to the same texture area with one mirrored. In this way, the polygons in the middle access exactly the same vertex (both position and texture coordinate are identical). However, by mirroring, the polygons require different tangent information. Thus, when calculating the tangents, errors occur, which of course affect the representation. The solution is to slightly change the mapping of one of the polygons so that it is no longer completely identical at the affected points. Then each polygon gets its "own" point and the computational problem no longer

occurs.
- The steeper the transitions, that is, the more sharp-edged the alpha channel is, the messier the presentation becomes. As an example, I have constructed the circular window over the portal very sharply - the consequence is very well visible in the screenshot: On the one hand, creates a nearly 1-pixel mapping, on the other hand, it looks quickly as if the cut "slice by bit" milled out! 😉

Recommendation: If you are dealing with textures that are characterized by very soft transitions, consider giving up the Parallax Occlusion by leaving the alpha channel completely white, so that no steps can be created! An example would be the bodies of vehicles, which can easily be drawn "wobbly" using a normal map. Here Parallax Occlusion is inappropriate.

The depth is adjusted like with the normal texture using the *Intensity of Normal texture*.

5.10 Manipulating the Z-Buffer

A Z-buffer is information that tells you how far a given pixel of the finished image is from the viewer. In particular, this information is used in further drawing of the image to decide whether the new pixel should be drawn (if it is "in front of" the previously drawn image) or not (if it is "behind" the previously drawn image).

For special applications, the processing of the Z-buffer can be manipulated:

- *Shift towards the viewer* pretends that the pixel being drawn is closer to the viewer than it actually is. A typical application is, for example, a label applied to a previously drawn polygon. The so-called Z-test is then "on the cutting edge" and comes to different conclusions by chance - it comes to the typical flicker. The shift parameter now ensures that the new pixels are "allegedly" closer to the viewer and thus the test is positive in all cases.
- Deactivating *Write depth-buffer* ensures that the pixel to be redrawn remains "invisible" in the Z-buffer. Everything that is drawn afterwards does not consider this polygon. Here too, the example of the label is suitable: By the *shift towards the viewer* the wrong value would normally be written into the Z-buffer. However, since the correct value was already present (which is why the shift was actually necessary at all), the writing of the offset Z data can be dispensed with since it would only lead to misinformation.

6 Special texture coordinates

6.1 Creating secondary texture coordinates

Secondary texture coordinates must be created if the material is to contain AO texture with separate mapping or for mesh animations (see Mesh animations).

First, a copy of the 3D object to be animated is created. This gets the name of the original object and additionally the prefix "@2m_". If the original object is called "object", then the copy is called "@2m_object".

In the future, this copy has to be exported in EVERY export of the original object and - after changes to the original object - must be exactly adjusted to its original shape, ideally a new copy has to be created. The smallest deviations of the vertex data (position, normal, etc.) are ok, but more is not tolerated.

The technique is now the following: When importing the original object and the @2m_ copy into the Content-

Tool, the texture coordinates of the copy are copied into the original object and saved as a second set of texture coordinates.

6.2 Global texture coordinates

There are cases where splines should be allowed to overlap but not flicker ("z-fighting"). The classic case is the gravel bedding: In the switch/junction area, the two ballast beds of the two tracks continue to approach each other, then begin to overlap and do so exactly at the point of the turnout until the next track which again generates a single ballast bed. On the one hand, because the texture can be evenly mapped anyway and it is even of advantage if the repetitions of the texture do not necessarily follow the exact track, the possibility has been set up to map this texture globally in the coordinate system of the tile. While the spline is still "originally" mapped in the Content-Tool and during the installation in the map editor, the coordinates are automatically converted when "baking" the tile mesh. As a result, globally overlapping splines are mapped exactly identically and can no longer flicker each other.

To enable this feature, select the mesh in the Content-Tool. Then, in the "Selected Mesh" section, you can set the parameter *Glob. Tex size*. It can be set to a value greater than 0. This value then affects the mapping size of the texture relative to the tile, that is, how often this texture fits in a tile in x and y direction
[hl=1][/hl]