Tutorial: Baking Ambient Occlusion textures with Blender

Table Of Contents

1 What we want to achieve

First let us look at the initial, untextured version of the 3D object

Automatic couplers, often in the middle of the carriage, like this famous Scharfenberg coupler very often have a very detailed mesh (because of all the handles, cables and cogs) but have gotten brown-grey all over over time. If you would use a simple, 'naked' texture it would look like this:

That looks pretty modest, of course. If the sun shines in LOTUS, it looks a bit fancier:

Nevertheless - the sun does not always shine and furthermore there are even places that are unnaturally

bright (see red arrows).

It would look much better if we would use the Ambient Occlusion technique to add shadows that are caused because the ambient light cannot reach all places:

Note that the sun is not shining in this picture!

2 How to achieve this

The AO-information is stored in a separate texture, that is why it is called Texture AO. The AO texture itself is in greyscale: white if fully lit and black is fully dark.

2.1 Create the AO texture yourself?

The simplest case would be where you would create this black and white texture yourself. You can simply use your virtual pen to color in bright and dark tones of grey on a new texture corresponding to spots in the original texture that should be lit more or less brightly.

A suitable case for this would be a not particularly bright carriage depot.

After finishing the texture you use it in LOTUS and look at it in cloudy weather. You will find that all surfaces, whether inside or outside, are brightly lit because of the weather conditions, only ambient light is present, and therefore no shadows can be seen.

"OK," you think, just darken the interior walls in texture. Now it looks good again in overcast weather: bright outside, dark inside.

However if you now look at it during the night: While a light is pointed directly at an interior wall it will remain quite dark, even if actually painted white. With sunlight shining in through an overhead window the interior walls still remain very dark, even with direct sunlight.

The solution here is of course an AO texture. However, you do not feel like messing around with texture coordinates or generating AO textures, so you just paint a very simple AO texture: where the exterior walls of the carriage depot are located on the original texture, leave paint them white on the AO texture, the areas where the interior of the carriage depot are located, you paint dark gray. Interior areas that are closer to the windows, are colored with a nice gradient and / or slightly brightened. It works pretty fast and can look good right away.

As a result, the carriage depot will be darker inside than outside - but if the sun's light or an artificial source of light falls on an interior surface, it will still shine in its original brightness.

2.2 Letting Blender calculate the AO texture

Using the previous approach for the coupler isn't a good idea. The shape is fairly complex and we don't want to spend a lot of time messing around to make it good. For the coupler we will let Blender calculate a realistic AO texture.

Now there are two cases: Either the mapping of the AO texture is *identical* to the mapping of the normal texture or the AO texture uses a completely *different* mapping compared to the original texture. While using another mapping for AO texture coordinates would occupy us with more work, we ask ourselves why anyone would ever want to use this method. For this we need to look at the very strict rules that we must follow to be able to generate an AO texture:

- No part of the texture may be used more than once! Otherwise Blender doesn't know what instance of use of this texture part it is lighting now, the first instance, second instance, ... ?
- Every part of the object must be mapped sensibly.
- No 1-pixel mapping, always reasonably orthogonal mapping and reasonably consistent resolution - or at least each part should be mapped with a minimum resolution.

Because of these rules we see that either the main texture also has to follow these rules or one needs to use separate AO texture coordinates.

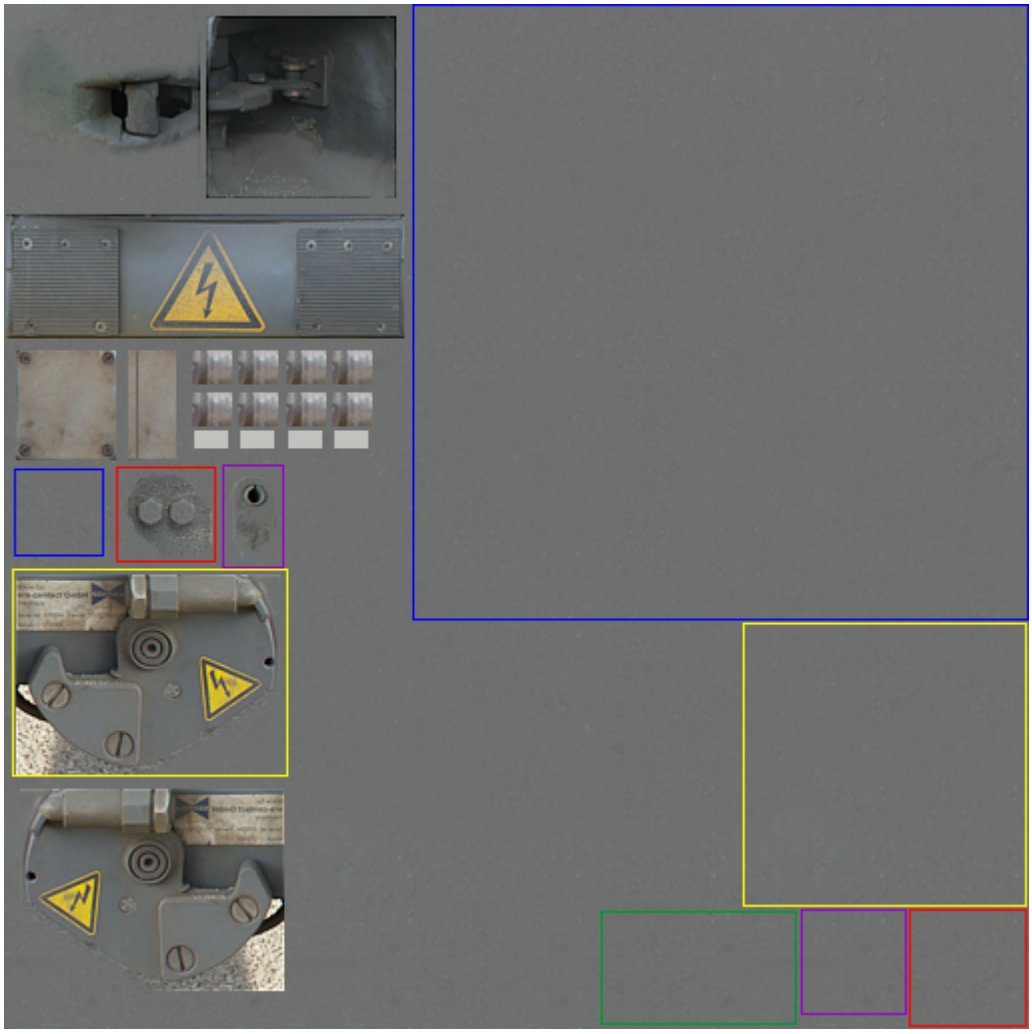
As there are very little "real" textured parts we want to follow the rules above with our main texture.

2.3 Separate texture coordinates: How will the texture look?

Luckily we can tell Blender to automatically map an object following the rules above using a specific algorithm. We do not need to mess around to get all polygons of the coupler in.

This automatic mapping results in a compact square mapping. Important info for later... 

This is our original texture:



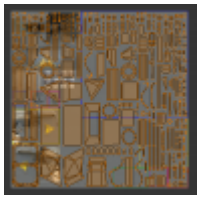
On the one hand you can see the subtextures for all parts that shouldn't be just grey. Furthermore there is a big grey area. All parts that can simply remain grey will end up in here. The large area is needed because every part of the model is mapped individually. Using the colored squares we divide the texture into different mapping areas while this texture will also be used on some sub objects. For this tutorial we focus on the big blue square. In this square all not individually textured parts will go. Remember that we cannot overlap the mapping! That is also the reason that both sides of the coupler are provided separately as well as the four sockets, all looking the same but because there are four in the reality we also need four here.

2.4 Mapping

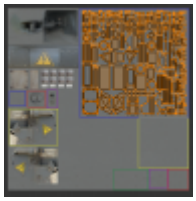
The parts with individual texture are mapped as usual, while taking into account the rules:

Now we select all other polygons (remember you can inverse the selection with [Ctrl]+[I]). These are the polygons we want to Blender to map following the rules. For this we press [U] and select "Smart UV Project". There are now four settings for which the defaults are good. Only for "Island Margin" we have to select something larger than 0, like 0.01. This represents the distance between each mapping. A couple of pixels here can help us avoid messy edges.

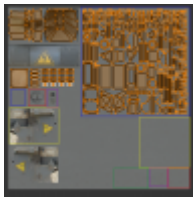
The resulting automatic mapping looks like this:



Now we scale and move the mapping to fit inside the reserved area:

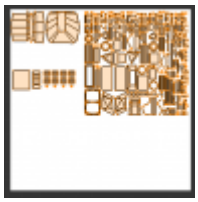
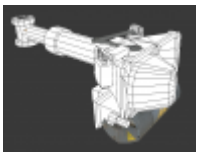


And then we are done:



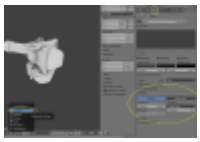
2.5 Generating AO

We now want to generate the AO. For this we create a white texture with the resolution we want (this resolution can be different from the original texture) and load this into Blender onto the just mapped polygons. As a result the coupler will be white:

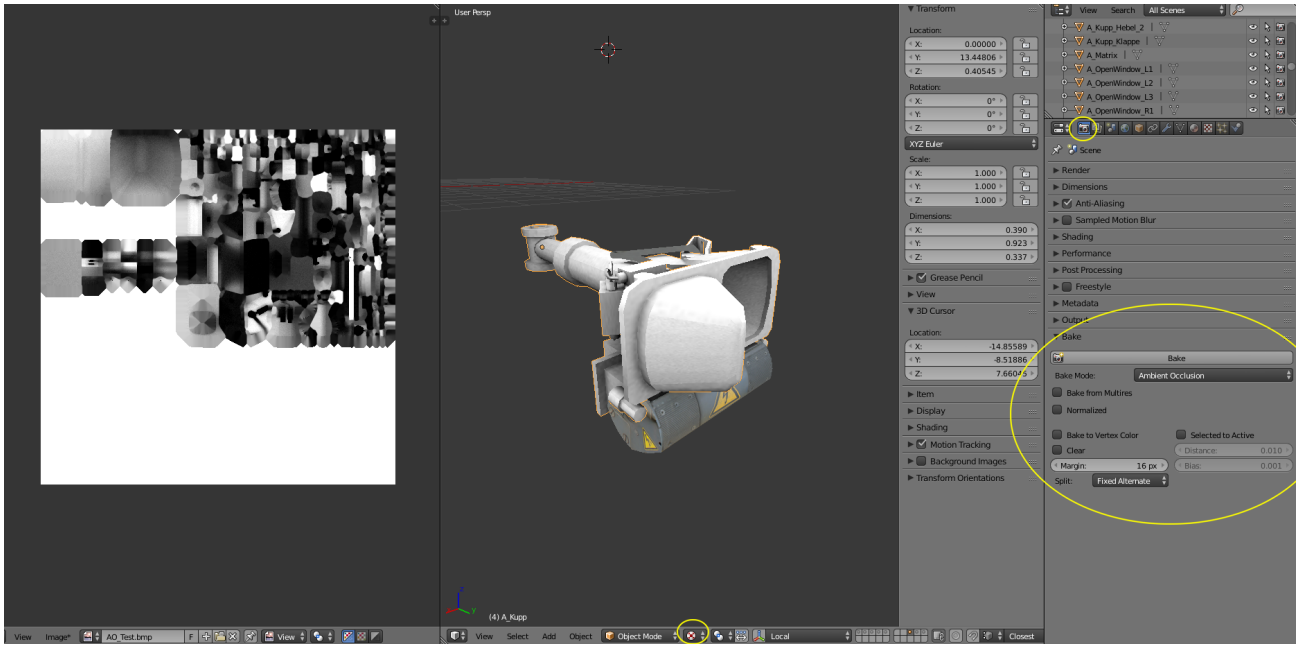
Then we are all set!

We change into Object Mode and click on the globe in the Properties section. We can then *configure* the Ambient Occlusion in the section "Gather". To test our settings we can choose "Rendered" as "Viewport Shading". In the screenshot you can find the settings we used.



Now it is time to get serious. First set the "Viewport Shading" to "Texture". The coupler should be completely white again. Then in properties we select the camera in the section "Bake". Then we can tweak some further settings, see screenshot. Then we click on "Bake" and wait.....

After some time the process will be done and we can see the result:



In the texture window you can now see the finished AO texture. You can enjoy the improvements in the 3D window. To finish we need three more steps:

- Export the texture from the texture window: "Image" -> "Save Image".
- Before exporting to x3d file set the original texture for the object again
- Configuration in the Content-Tool:
    - Material type: "Complex"
    - "Intensity of the AO Texture": start with 1, change to match your taste
    - Set "Activate AO Texture" and "AO uses original mapping" in "Options and integer values"
    - At the bottom, texture, select the AO texture

Done!

2.6 I can't see it

"No worries, in the Content-Tool it is (nearly) impossible to see"

Remember that the AO is only active with ambient light. Before you start doubting yourself go to "Test Environment" on the left and set Direct and Indirect light to 0. Then increase the third slider for Ambient Light and enjoy your work.