The Blended Planet
www.blendedplanet.com

Texture Mapping 101

Prerequisites

This tutor assumes you understand the difference between materials and textures, and that you know how to load an image into Blender to use as a texture. In this tutor you will learn how image texture are mapped onto your models, and how you can control this process in order to make your textures appear on your models the way you want them to.

Why Not Use the Hot New UV Editor?

Before you dive into editing texture mapping at the microlevel, which is what Blender's new UV editing does for you, you should first make sure that you get a good handle on the basics of texture mapping in Blender. You'll discover that the UV editor is often overkill, and skillful use of basic texture mapping will solve the vast majority of mapping problems you encounter

Texture Mapping?

Texture mapping seems to be a great source of confusion, and a topic that is seldom discussed. But it's a critical thing to understand if you want to make your models look the way you expect when they render. This is particularly important if you're trying to map a logo, a sign, or the face of a building onto a model in a way that looks convincingly photographic. The good news is, Blender's texture mapping implementation is functionally identical to what you'll find in Lightwave, SoftImage, and 3D Studio Max, and adheres to well-established conventions in computer graphics.

The Challenge of Texture Mapping

The basic principle behind texture mapping is that it's virtually impossible to make a detailed texture map wrap itself around a detailed model in a way that makes sense. That shouldn't surprise us. Most of the textures that you'll use come from image files which are rectangular. One of the classic puzzles of computer graphics, is how to use a rectangular image map onto a complex three-dimensional surface in predictable fashion without an unacceptable degree of stretching of the image. Stop and think about this for a moment, and you'll see why this is so.

The Solution

The classic solution to this problem comes in the form of what we call mapping primitives. The idea is that we can take our rectangular image and map it in predictable fashion onto a very simple primitive form such as the sphere, cube, cylinder, or flat plane. The final projection of the image onto the model becomes much more predictable, and once the image has been mapped onto a simple geometric shape, a certain amount of stretching can be tolerated in the process of mapping the image from the primitive form to more complex model.

Just so we're clear, these images that show how an image with the words "Dave Was Here" project onto the four basic mapping primitives. The black lines are to help make it very clear how the image is mapping on to the geometry.

Here's the original image :

Cylindrical Mapping (Tube)

Cylindrical mapping wraps the image around a model like a label on can. The top and bottom of the team simply inherit the color of the top and bottom row of pixels in the image. When Blender cylindrically maps the image, it assumes that the model is in a standard up-and-down position, with the top and bottom of the can facing the sky and the floor. The pole, therefore, runs through the y axis.

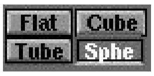
These primitives are all cylindrically mapped

Hmmm...
As you can see, cylindrical mapping onto a plane, (or any other object that is basically flat) looks pretty stupid. You can see why. As far as the mapping is concerned, the flat surface of the plane represents the top, sides, and bottom of the tin can!

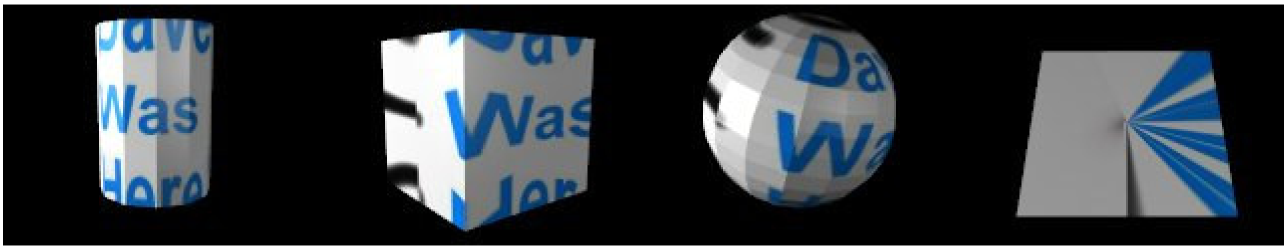
TIP
Blender does not seem to care whether you create mesh primitives in the front, top, or side view. Even though the primitive is created with its top facing the z axis of the active viewport, (ie, pointing at you, the viewer), the default mapping still respects the local Y Axis

of the model. In other words, the tin can mapping looks how you expect no matter what the initial orientation of the model is.

Spherical Mapping (Sphe)



Spherical mapping projects the image in such a way that the top and bottom of the image are constricted so that they meet at the polar extremes of the model. Once again the default mapping places the poles on the Y Axis.



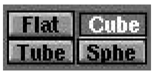
(By the way, the sphere was deliberately tilted so you could see how the image was mapping to the sphere better.)

The difference between cylindrical and spherical mapping is slight. Basically, instead of leaving the polar extremes of the model unmapped, the way the tin can is treated, spherical mapping stretches the image all the way to the top and bottom of the model.

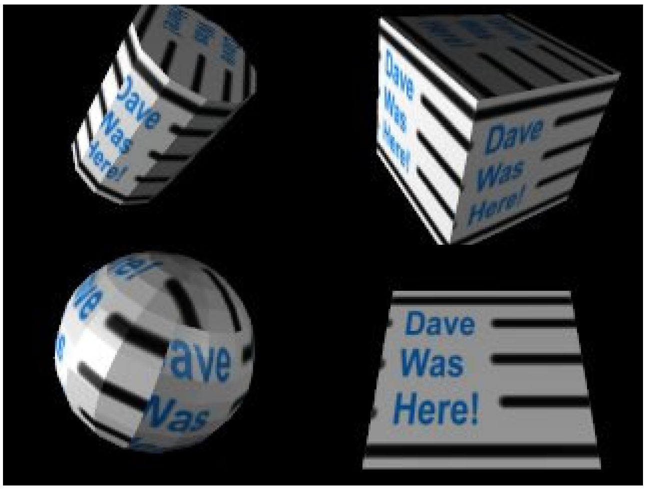
As a result, the cylinder and cube choke a bit, and just as before, the mapping to the plane is just about useless. Obviously, the best use of spherical mapping is to map...er, a sphere! Or a model that is basically derived from a sphere. The GIMP (a popular free image editing application for LINUX and Windows) will allow you to run a filter on an image that will convert it to a polar mapped image :

You can use image editing tools like smudge and clone to hide the seams, then run the filter in reverse to convert the image back to a flat plane. At this point the image will map seamlessly onto a sphere.

Cubic mapping



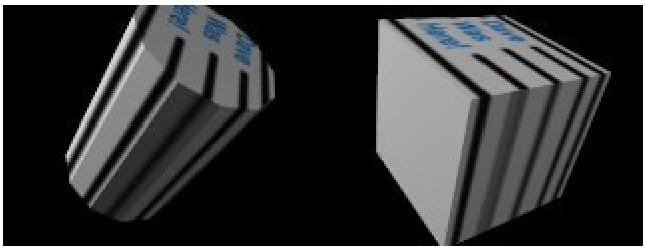
Cubic mapping, on the other hand, actually projects the entire image onto the model in each of six different directions -- front, back, left, right, top, and bottom. Obviously, cubic mapping has no poles, because the projection is the same from every direction.



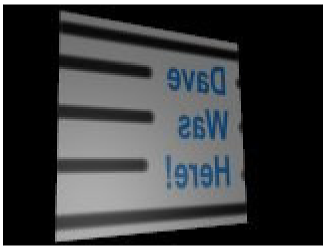
Obviously, it maps best to the cube. Any model derived from a cube, such as an office building, can benefit from cubic mapping. Especially if you want the same image projected in the same way to each side of the model. You could achieve a similar result by using cylindrical mapping, but tiling the map 4 times. In a way this could be better, because with cubic mapping the image of an apartment complex onto a tall box, you have to somehow hide the top.

Planar Mapping (Flat)

Finally, planar mapping, the simplest form of mapping, simply projects the image onto the model as though it were a flat plane. My personal opinion is that you will get the most use out of cylindrical and planar mapping, although the others certainly have their uses.



Planar mapping is more or less as if a projector was projecting a slide onto the model from a single vantage point. You'll notice that the negative image does appear on the back of the model, because the projection shoots right through it.



You can see how each of the mapping types is different, and they each translate from their respective primitives to complex models with varying degrees of success.

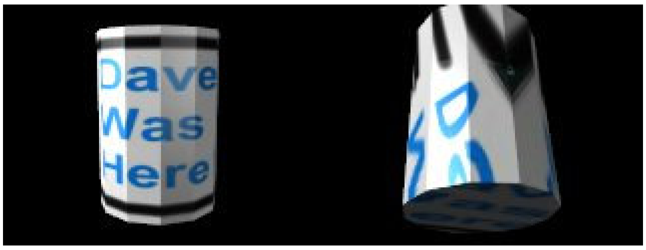
Herein lies the first secret of successful texture mapping. Approach your modeling the same way an artist learns to draw basic forms. Try to reduce your model to its primitive components, and have the mapping in mind before you model. For example, an arm will generally be cylindrically mapped.

A stop sign, on the other hand, will generally be planar mapped. To map a face onto a model of the head generally calls for cylindrical or spherical mapping. We already mentioned using cubic mapping for a building. And on and on. This first step, simply deciding in advance how your models, or their individual pieces will be mapped, will at least get you in the ballpark.

Fine Tuning
However, Blender does have a few tricks up its sleeve to help you tweak the way images are mapped onto your models. With all of these techniques, you are not changing the basic mapping method. You're pretty much stuck with the four basic mapping primitives of plane, cylinder, cube, and sphere. But that's okay, because this is the exact same technique that is used by every major 3-D applications on the planet! So you know it works. However there are several ways to size or rotate a texture on your model in order to position it perfectly.

Translate the vertice
First of all, let's make a distinction between translating an object, versus translating its vertices. If you take a model that is cylindrically mapped and rotate it, a the mapping will stick to the object just as you would expect.

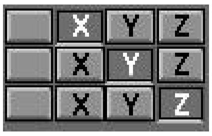
However, if you enter edit mode by pressing your Tab key, then press the AKEY to select all the vertices, and rotate *that*, the mapping *does not rotate* *with the object*! It's as if the object slides underneath the map...



That's pretty handy. If, for example, I have a model that needs to be cylindrically mapped, but for some reason the poles of the cylinder are in the wrong place, I can simply jump into edit mode and rotate the mesh until the texture maps correctly. The is a disadvantage with this technique is that there is no direct feedback in the materials editor window. That is, you can't see the change until you do a test render.

Change the mapping axes

Another way to accomplish the same thing in Blender is to use these keys



By simply changing what Blender considers to be the X, Y, and Z axes of the mapping, you because the image to be mapped onto the model differently. Pressing hte "blank" buttons causes that axis to be ignored. Play around with these and see how changing from XYZ to YXZ or ZXY affect how the image is mapped onto the model. You'll get a basic idea by looking at the sphere material preview, which will show you how you are affecting the mapping.

Scale or offset the map

These buttons do pretty much what you might think. ofsX offsets the map on the X axis the amount you specify. To get a feel for it, just click it and see how the mapping changes in the preview window. You can offset your map on the X, Y or Z Axis. Similarly, you can also resize the map to determine how many times it is tiled in each dimension. Mess with these a bit and you'll get a feel for it.

Use an empty as a mapping coordinates gizmo

This technique arguably gives you the most control. You simply create an empty in your scene, give it a meaningful name, and use it to change the scaling, rotation, or position of the map by scaling, rotating, or moving the empty in relation to the model.

Create the empty by pressing your spacebar and selecting **Add >> Empty**. I strongly suggest that for the most predictable results, you create the empty in the center of your model, then parent the empty to the model. This way when you move your model, the empty will move along with it, and the mapping will stay the same.

You parent the empty to the model by first selecting the empty, then, while holding down to **SHIFT KEY**, select the model. Then press **Ctrl P**, and confirm that you want to make parent.

Then press this button :

and type here :

to change the name of the empty. I called mine empMAP. Then go back to your materials editor window, and select your model in the 3-D window so you can see its mapping set up.

Instead of using ORCO mapping, which is the default, you want to choose "object". In the space next to the "object" button, type the name of the empty.

Now the image mapping will be determined by the position, rotation, and scaling of the empty. Do a test render, then move, rotate, or size the empty. Be careful to only move the empty a little bit, so that you are able to see how the results of changing the position of the empty affect the mapping.

TIP
In fact, you can animate the empty and thus animate the way the image projects on to the model! Plug ins to create similar effects in some other applications cost hundreds of dollars. CHA-CHING! In Blender, it's free!

Teacher Recommendations...
I recommend linking your mapping to an empty only if you need to animate the map with precise control, for two reasons. First of all, the material preview will not show you how the translation of hte empty is affecting your mapping, so you'll have to test render frequently. Also, the benefits of linking to an empty can be achieved in other ways discussed above, and will allow you to see how changes affect the material preview. By the way, mapping parameters are fully animatable!

Wrap-Up ***(or should I say "Map-Up"?)***
Well, that's texture mapping 101! We could certainly dig much, much, MUCH deeper, because Blender offers a variety of powerful and flexible ways to influence how textures mapped onto your models, and effects the surface characteristics.

However, what we've learned here in this tutorial encompasses the key functionality that is common to most high-end 3-D applications. In other words, the techniques here are more than adequate to address most of the mapping problems you'll need to solve. Put these techniques to work, and you may find that it's quite some time before you need to go beyond them.

Happy Blendering!

Dave