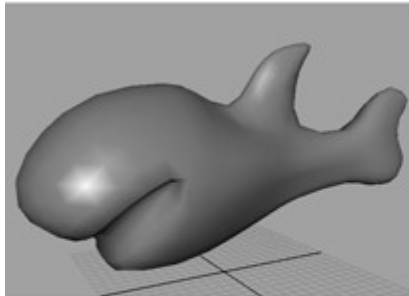
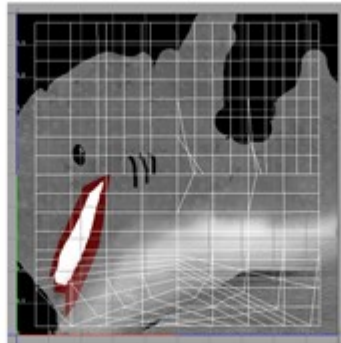


Understanding Mapping Coordinates

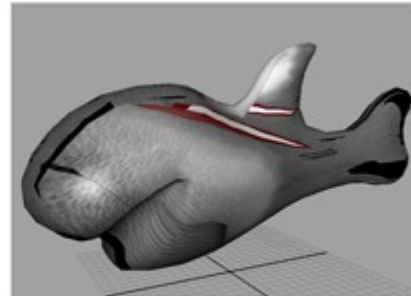
Mapping coordinates, otherwise known as UV coordinates, determine where images (maps) will show upon 3-D geometry. UV's bridge the gap from the 3d world (vertex) to the 2d world (pixel).



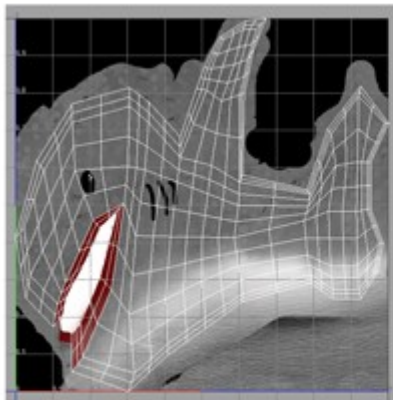
Fish model



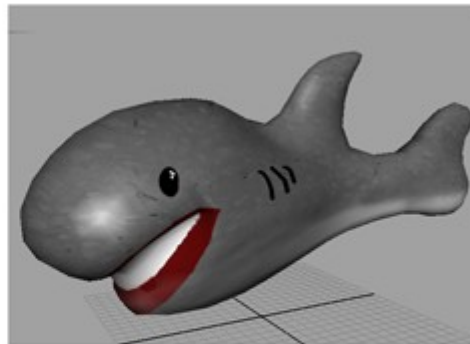
Default UVs for fish model



Default texture appearance



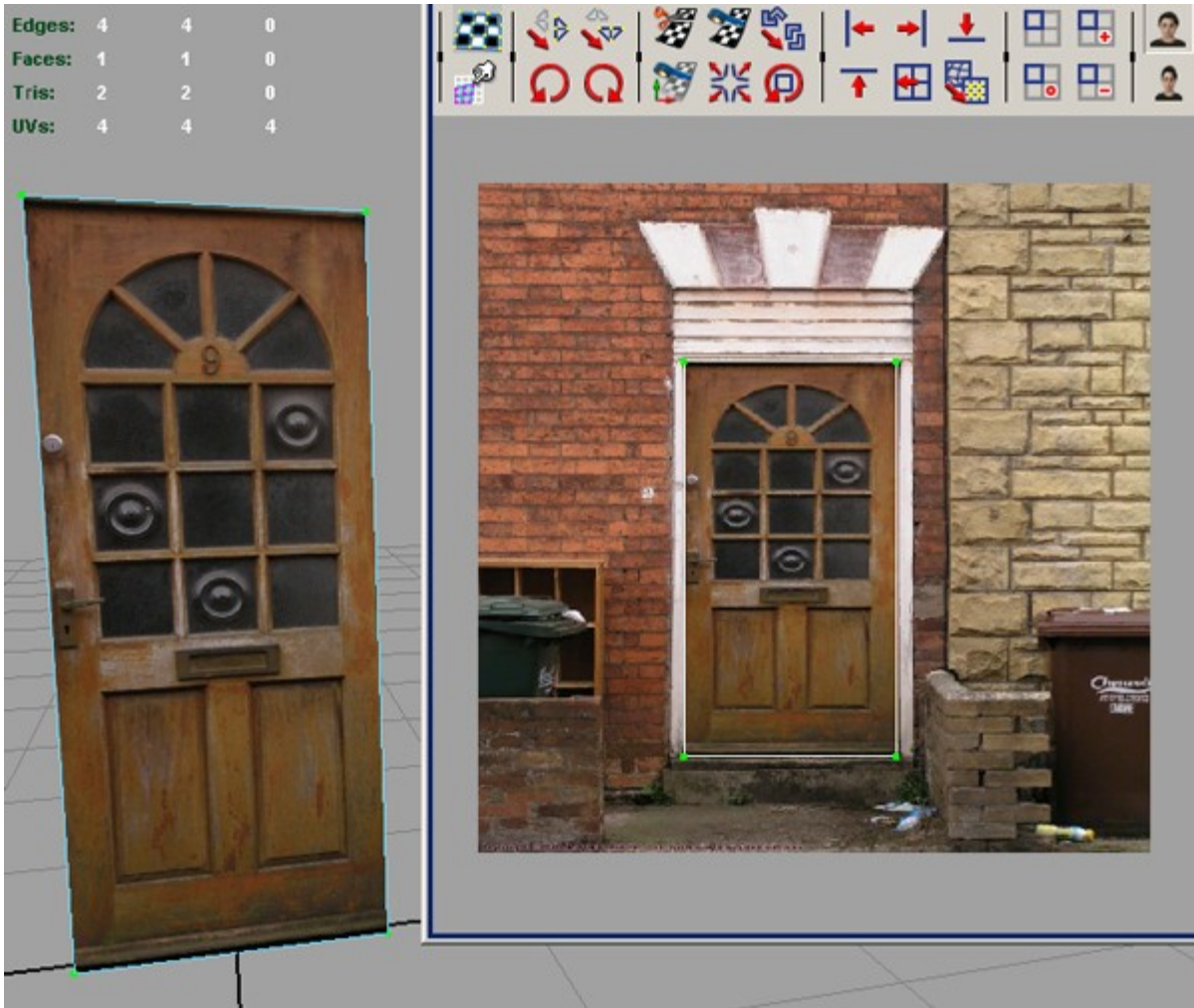
UVs rearranged for model



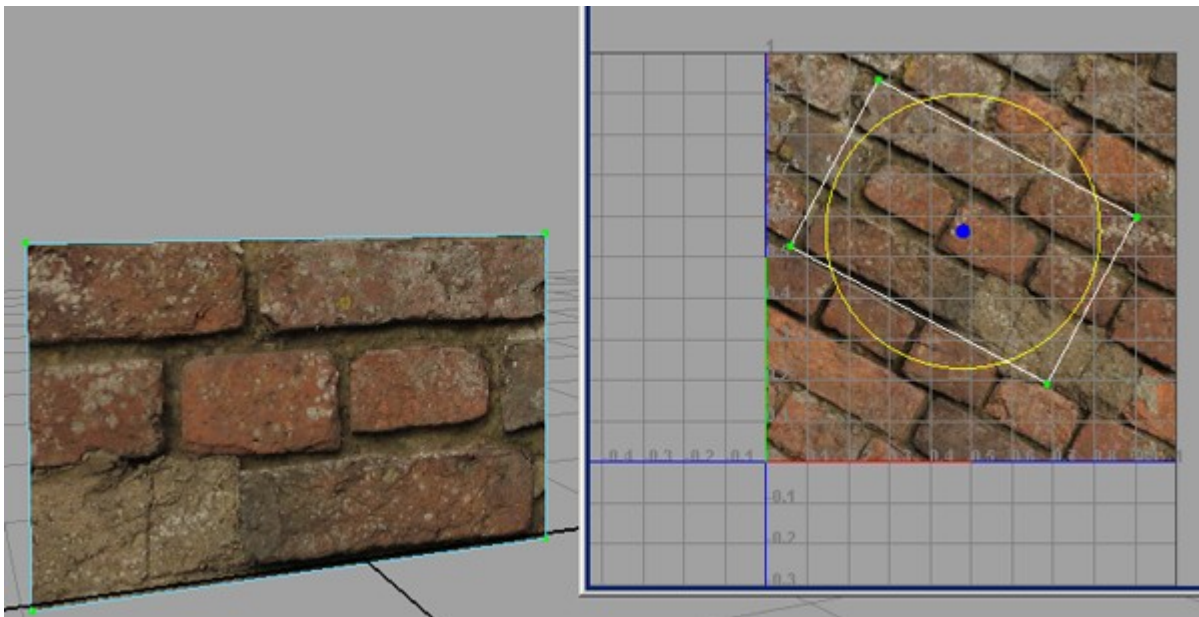
New texture appearance

As discussed earlier, each vertex on an object has an X,Y, and Z location, which is in visible 3D space. To determine mapping coordinates, each vertex receives another location in texture space. By texture space it is meant that each vertex receives a location somewhere on the picture. This location in texture space is often described by u,v and w coordinates (although w coordinates are rarely used since pictures are almost always 2D).

By giving polygons UV coordinates we are essentially placing them in picture space. If we have a image, but only want to show one part of it on a polygon, that polygon's UVs can be moved to that part of the picture.



UV coordinates are in 3D and can be rotated in any direction. Because of this, the up direction in your images will not always be the up direction in 3D software.

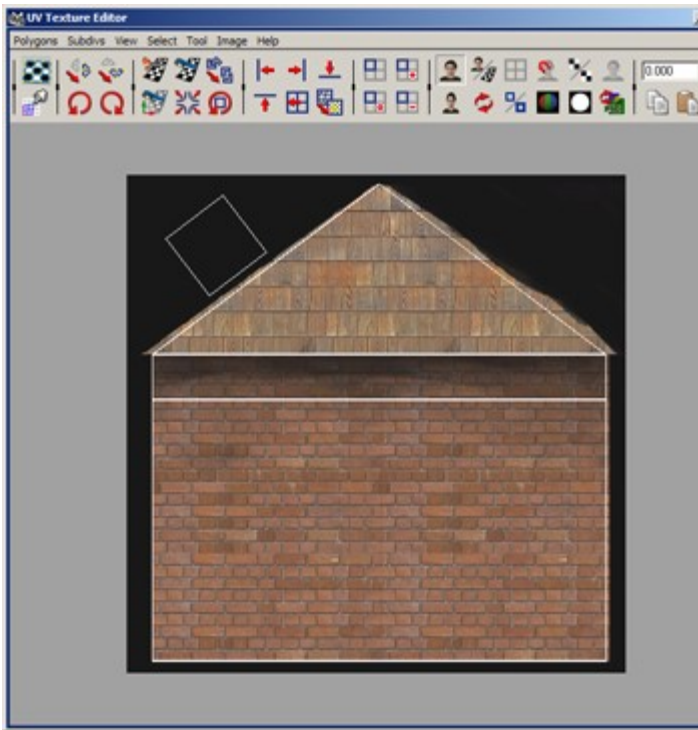
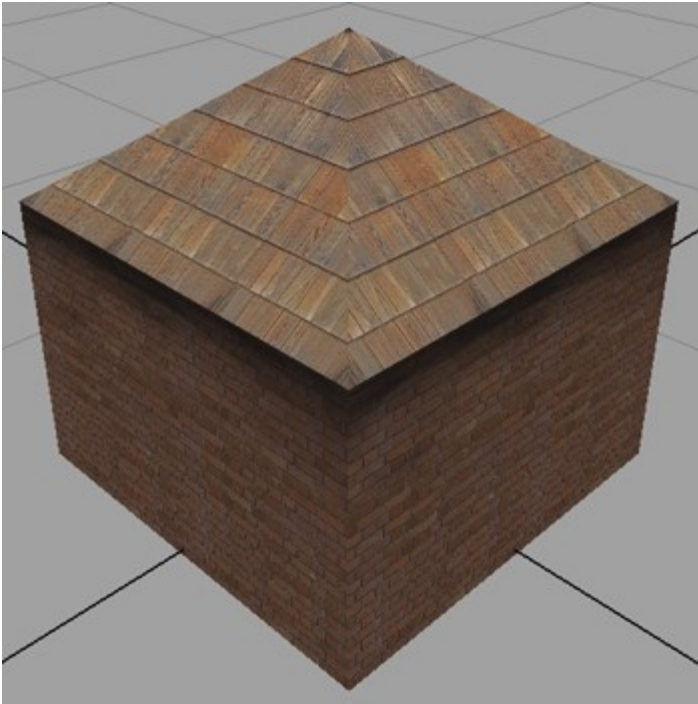


UV coordinates are normalized. Normalized means that primary range is between 0 and 1. At the left side of the image, $U=0$, and at the right side of the image, $U=1$. At the bottom of the image $V=0$, and at the top of the image $V=1$. This means that for example, at the top left-hand side of the picture, the coordinate is $(0,1)$. at the very center of any picture the coordinate is $(0.5,0.5)$

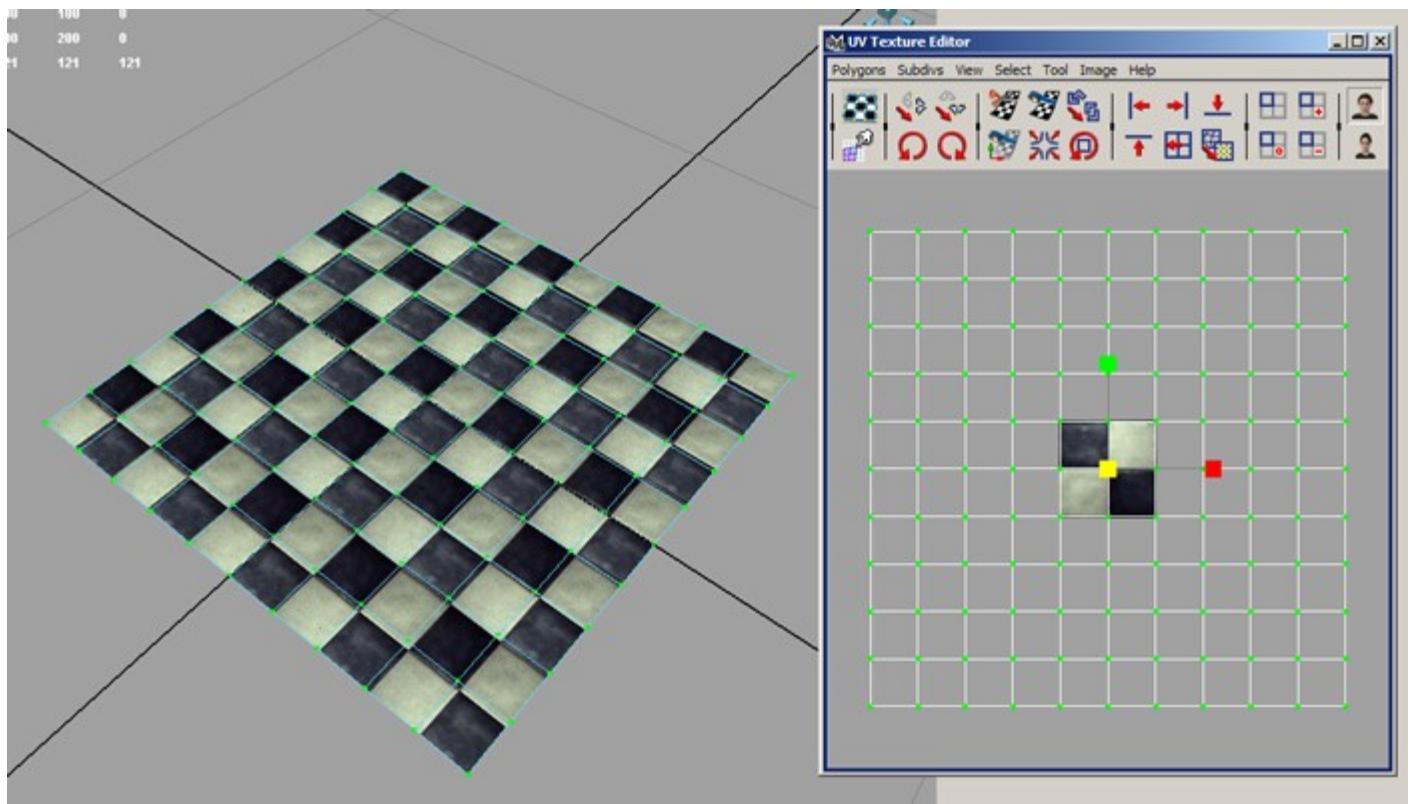
Because UV coordinates are normalized, it makes no difference what resolution the texture map is. Whatever picture you use will always be stretched/squished into a 1×1 square. This is very useful because it means you can change the resolution of the textures without changing the UV coordinates.



By utilizing UVs, objects can share different areas of the same texture.



Another interesting property of UV coordinates is that they loop continuously outside the range of 0 to 1. In other words, although the texture looks like it extends only to its border - it actually tiles to infinity. For example, 0.3, 1.3, 2.3 would all return the same color from the picture. -0.7 and -1.7 would return the same color as well. This means that by scaling up vertices in UV space, you can cause your texture to repeat (tile). Shaders often contain a control to enable or disable this repeating effect, to provide for situations where you don't want the texture to repeat.



Utility Textures

A utility texture helps you visualize the mapping coordinates on your model, and see how textures will conform to the geometry. In any situation requiring complex mapping coordinates, you should always use a utility texture. It will help you find and correct stretching and seams. A good utility texture should have colorization, as well as readable coordinates in the texture itself. It should also contain both curvature and square section.



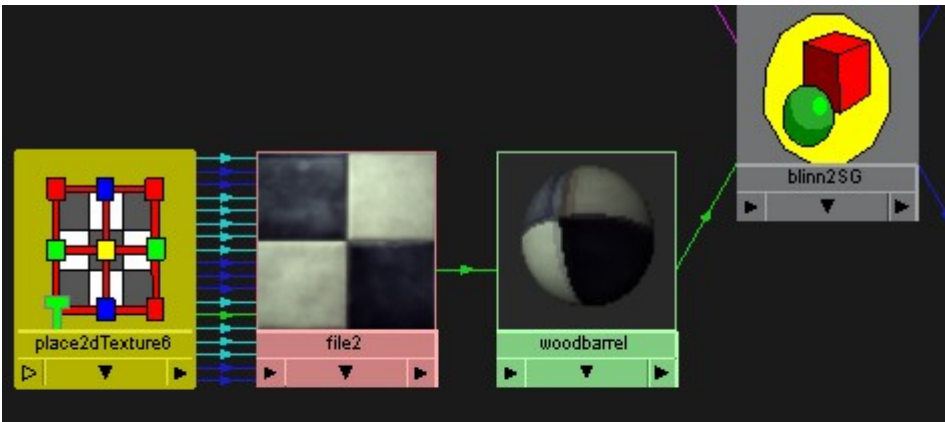
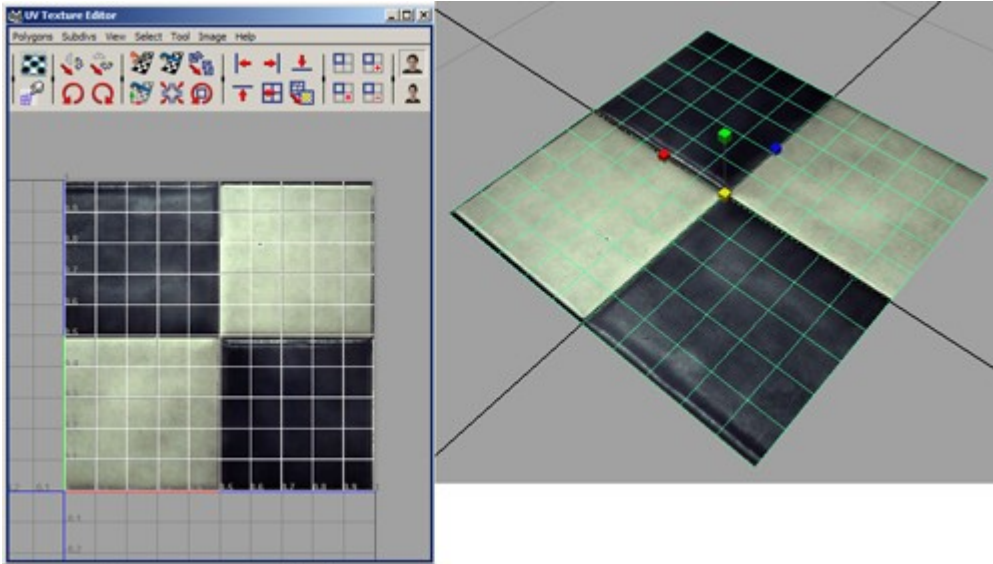
As you can see, this head model definitely has some texture stretching. Problem areas include nose, chin and his ears (to name a few).

No utility texture is ever perfect, and it is probably best to use a variety for different situations. Sometimes people simply use a checker pattern for a utility texture, but generally you want something that provides better feedback. Good utility textures have colors, numbers, and different shapes, such as square and circles. Circles and square really help to visualize whether or not your UV are stretching the texture.

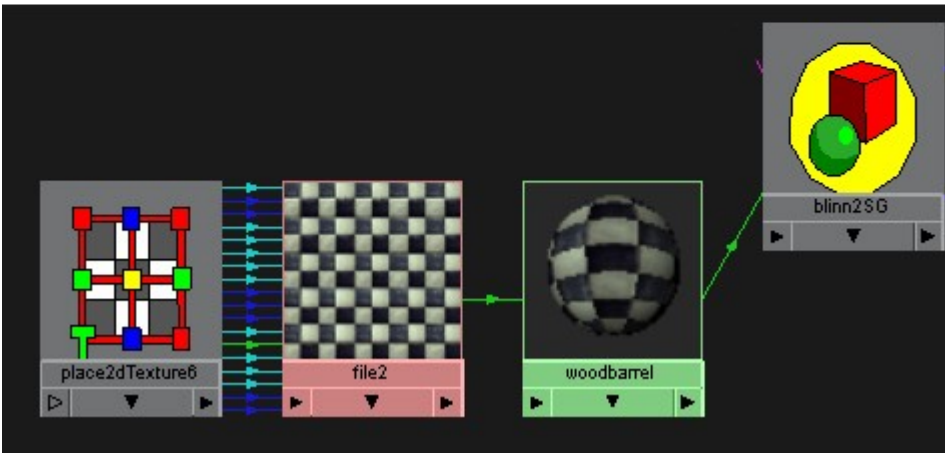
Texture placement in materials/shaders/textures

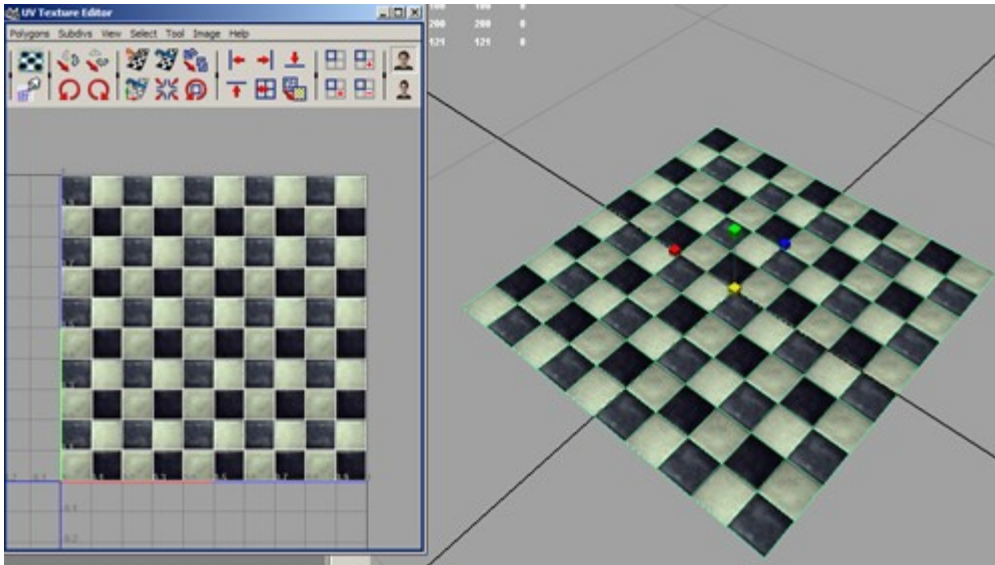
Often, aspects of materials/shaders/textures can contain information which modifies the UV coordinates for the texture. For example, a texture could have a repeat UV values of 2,2. Meaning that if that texture was assigned to a square polygon that fit perfectly inside UV space, with corners at 0,0 and 1,1, we would see the texture repeat even though the UVs did not exceed the range of 0 to 1. What we would see would be a 2x2 tile, for a total of 4 instances of the texture on the object.

Cropping function may also exist in texture controls, so that UV space 0 to 1 only represents a section of the image used, and not the entire image. In this case, the UV's behave as if you cropped the image in an image editor and simply loaded a smaller image as a texture.



Before and after changing the repeat attribute found within the "place2dTexture" node



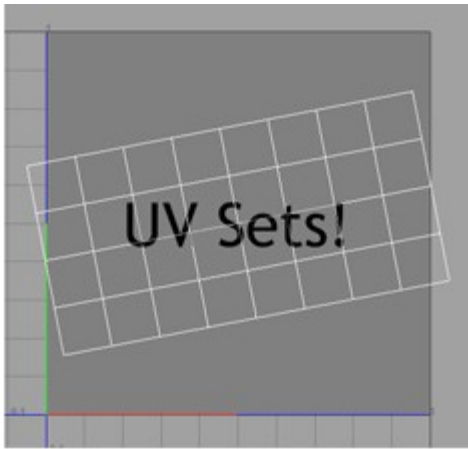


Using placement controls in the material or texture is usually not as useful as using editing the UV coordinates, because things you change in the material can't generally be edited on each individual object. Also, it is less standardized than using mapping coordinates to get the same effect. Mapping coordinates usually work the same way in different software packages, but material controls over placement of textures tends to be completely different across software.

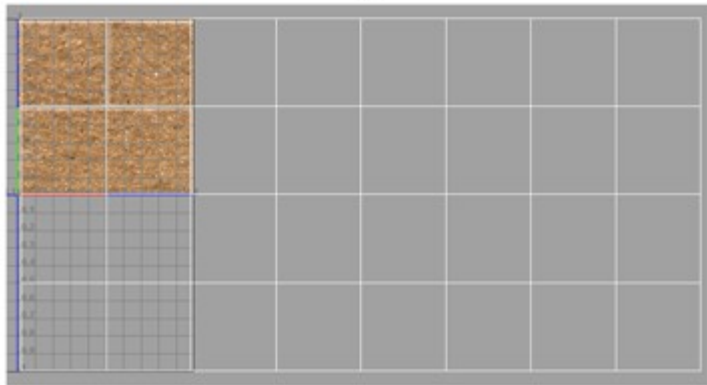
UV Sets/Channels

UV sets allow you to have multiple UV coordinates per polygon. You can assign or connect different textures to different UV sets on the same geometry.

In the following example, each texture uses different UV coordinates applied to the same polygons, utilizing UV sets.



+



=



Important Definitions

Overlap

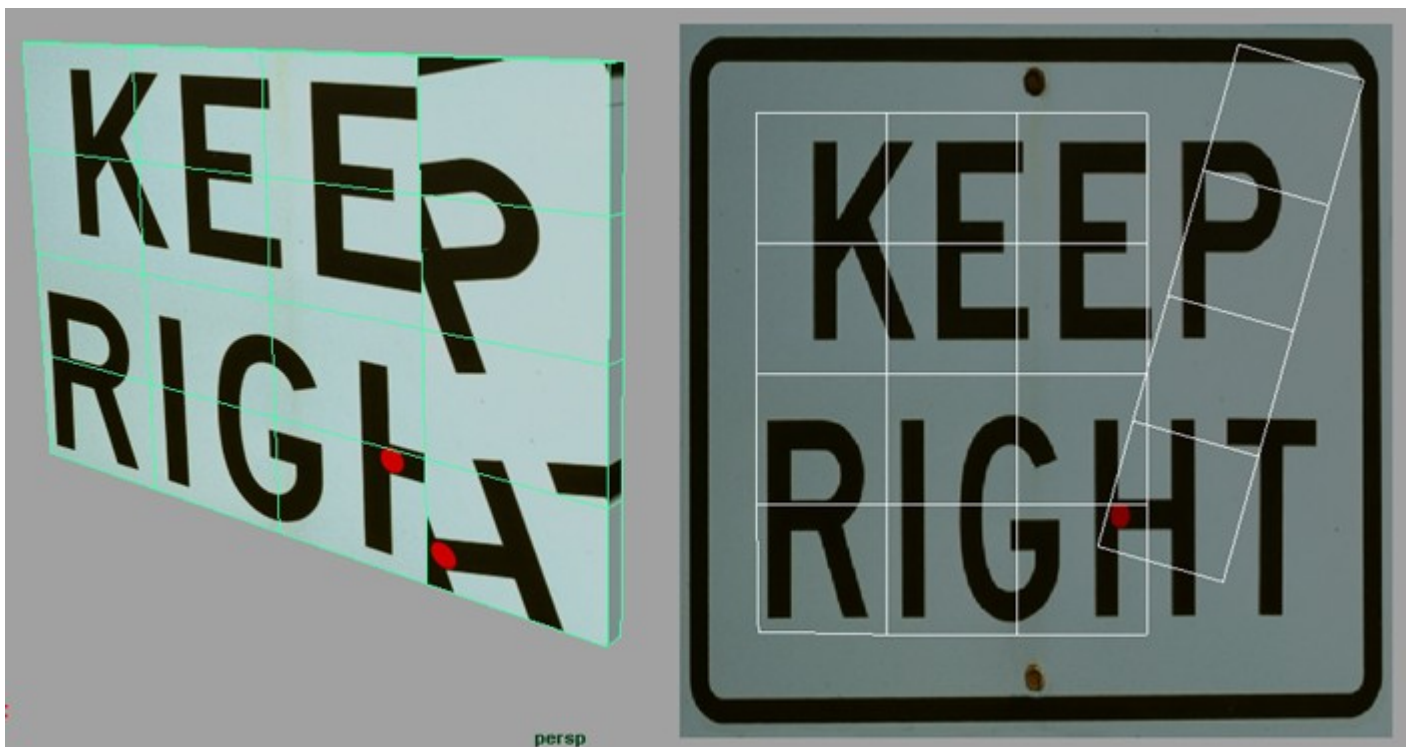
Overlap occurs when different parts of the geometry are placed over top of one another in UV space. This means that both sections would end up showing the same texture.

Overlap can be useful when trying to save texture memory, because different geometry can share/repeat the same part of the texture. However, the disadvantage to overlap is as follows: **On two pieces of geometry that overlap in texture space, it is impossible to paint the texture differently.** If you paint one of those sections, the other section will always match, because both sections reference the same part of a the texture.

Generally, when UV mapping, the first goal is to get rid of overlap, and get all the sections of the model seperated in UV space.

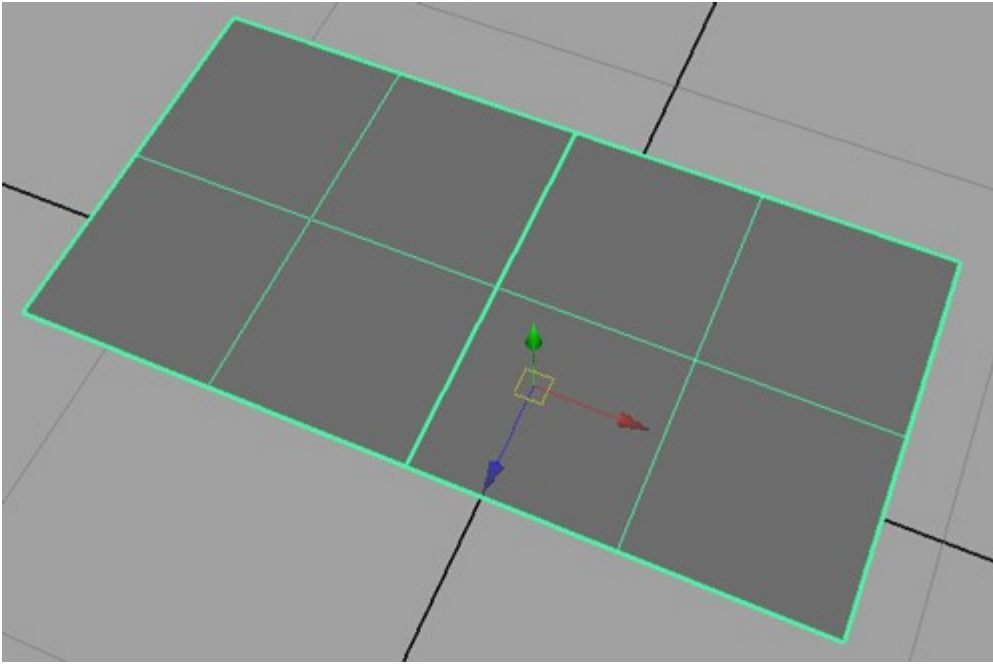
Baked lighting and normal maps are unlikely to work with overlapping UV coordinates. This is because these maps are generally less likely to repeat. Lighting especially, never really repeats on objects.

In the example below notice how the single red dot shows up in two places on the 3D geometry. This UV layout makes it impossible to change the colour of a single dot.



Seams

This eight polygon plane has had its UVs separated along its center resulting in a seam.



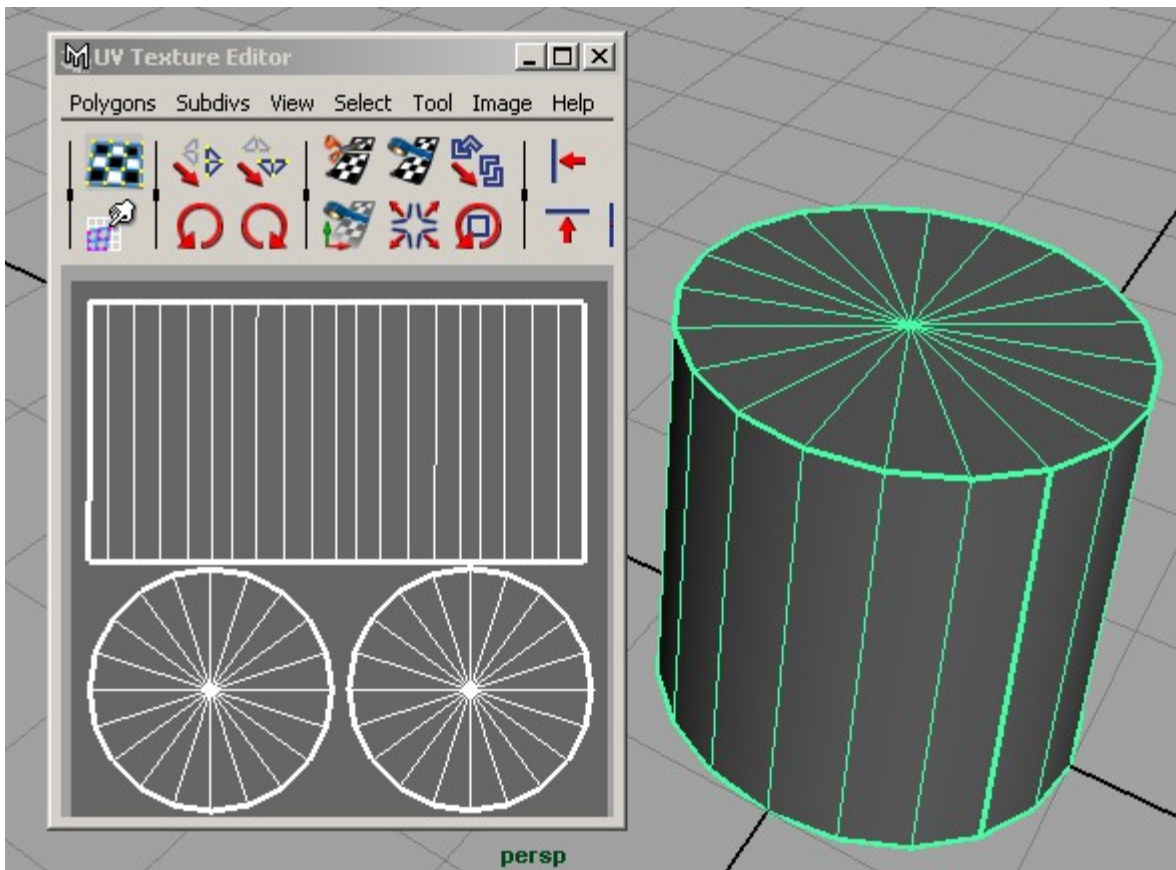
Even where the geometry is continuous in 3D space, it can be broken/split in UV space. For example, each face on a cube can be separated from the others in UV space. They would probably all be placed side by side, but all their vertices couldn't be welded/merged, and (for no overlap to occur) some edges would have to be open in UV space.

An edge belonging to only one polygon in a given space can be considered open in that space. Open edges in UV space are called **seams**.

Another more intuitive example would be the seams found on clothing. Clothing is actually created from flat pieces of fabric, but in 3D they are put together. The places where the different flat sections (in the 2D world) join together in 3D are called seams.

Islands

The cylinder below has 3 islands contained within its UV layout.



An island is a section of a model which is sewn together, and not completely separated by any seams. A UV island is the same concept as a polygon element, shell, or continuous-mesh but applies in UV space. It means that all the polygons in the UV island are connected in UV space somehow. The vertices are welded/merged in UV space. Two polygons are considered to be part of the same island if the edge between them is sewn, creating a "winged" edge in UV space.

Just because UVs overlap does not mean they form parts of the same island. In general, different islands should never overlap unless they are going to have different materials/shaders, or each island is on a different uv-set/map-channel.

Another way to think about an island is as a section whose topology is contained entirely within the border of a UV seam. This does not imply that all the vertices must be within this border in UV space. (Sometimes bad UVs will occur where UVs from the inside of an island stretch out past the UV border.) Assuming there is no bad overlap however, all the UVs will be contained within the space inside the border.

Stretching

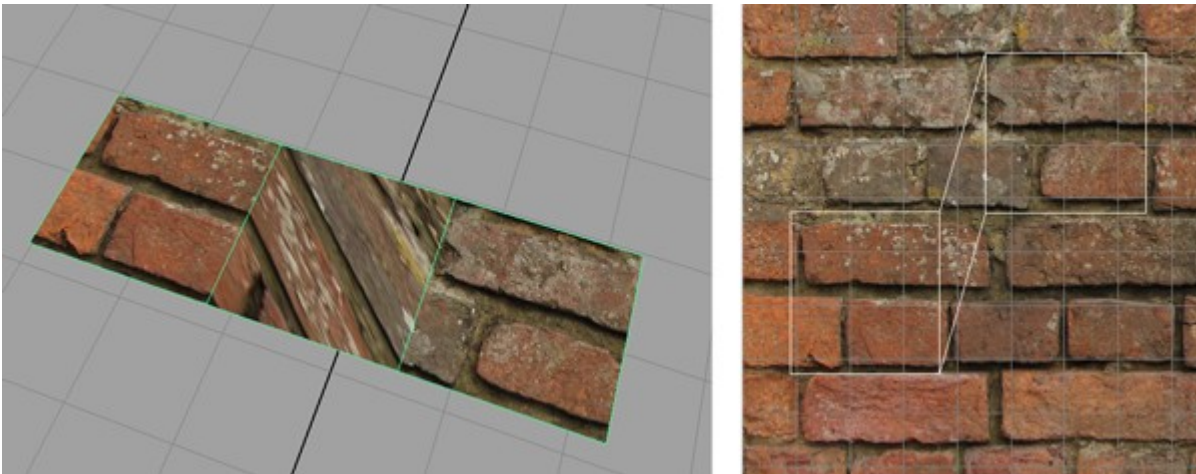
Stretching occurs when the shape of polygons in 3D space differs from their shape in UV space. As a common example, when an object has an even mesh density and an unevenly spaced UV layout, the texture will stretch. Planar mapping on a curved surface will also cause texture stretching.

On most models (especially organic ones... anything with enough complex curves), the only way to completely eliminate stretching would be to create seams around every polygon. Seams usually present problems, so we generally have to accept a certain amount of stretching, and balance it with the right

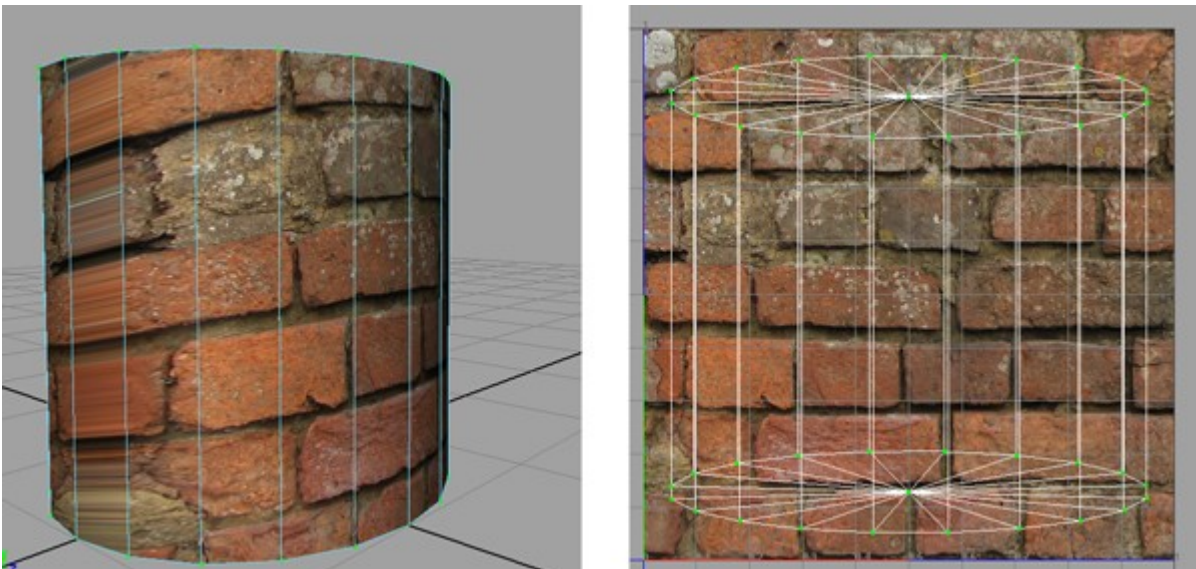
number of seams.

Sometimes, stretching can be useful. One such example is for dealing with non-square textures. If the image used as a texture is not square then the UV's will generally be squished to line up with the non square image. Many programs have features that will draw U and V at different scales so that UV space does not appear as a square, and thus, images contained within it can be displayed correctly.

The image below demonstrates texture stretching due to the middle polygon being small compared to the other two polygons in texture space. Since the geometry's mesh density is evenly spaced, the UVs should also be evenly spaced to properly display this specific texture.



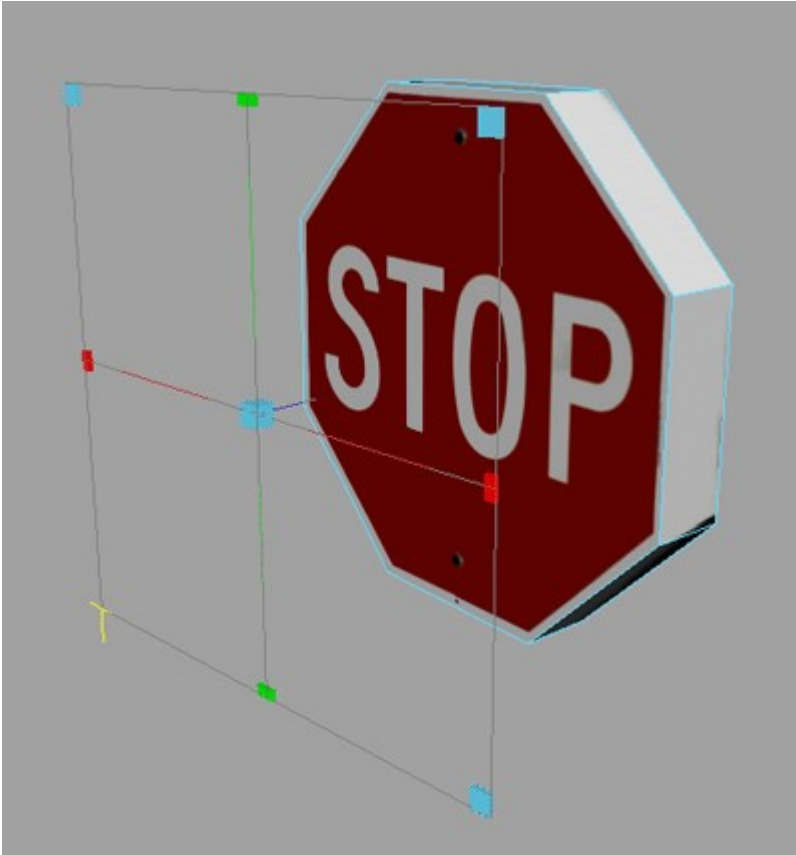
The image below shows a planar mapping result (from a camera). Notice how the texture is stretched on the side due to the UVs being really tight.



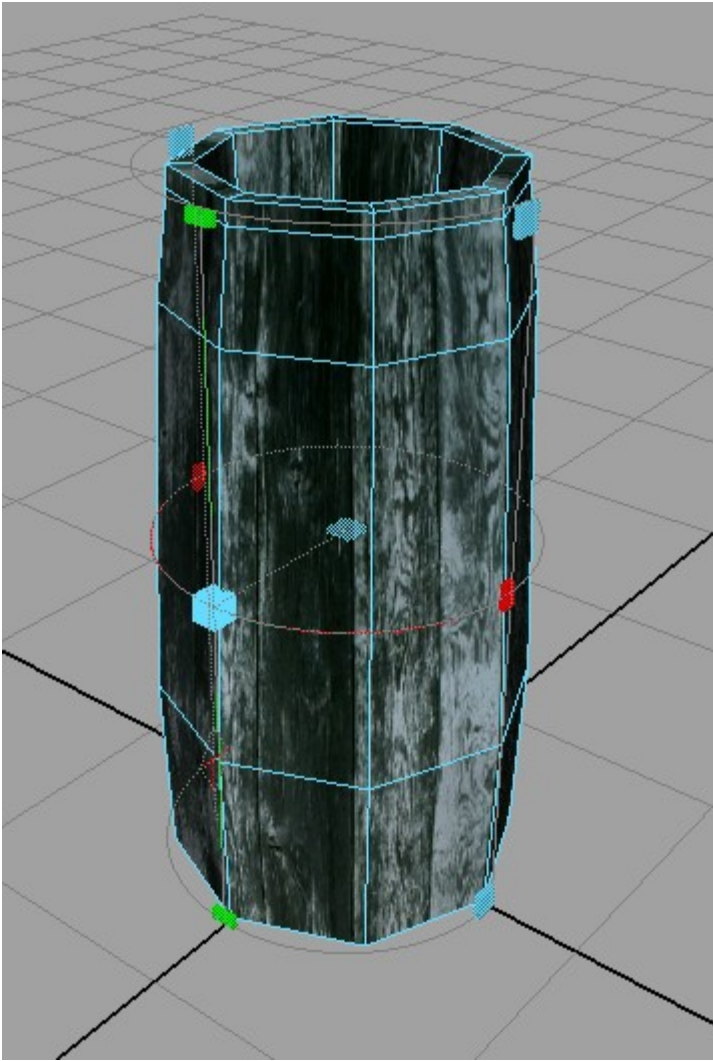
Ways of Generating UV Mapping

Projection Mapping

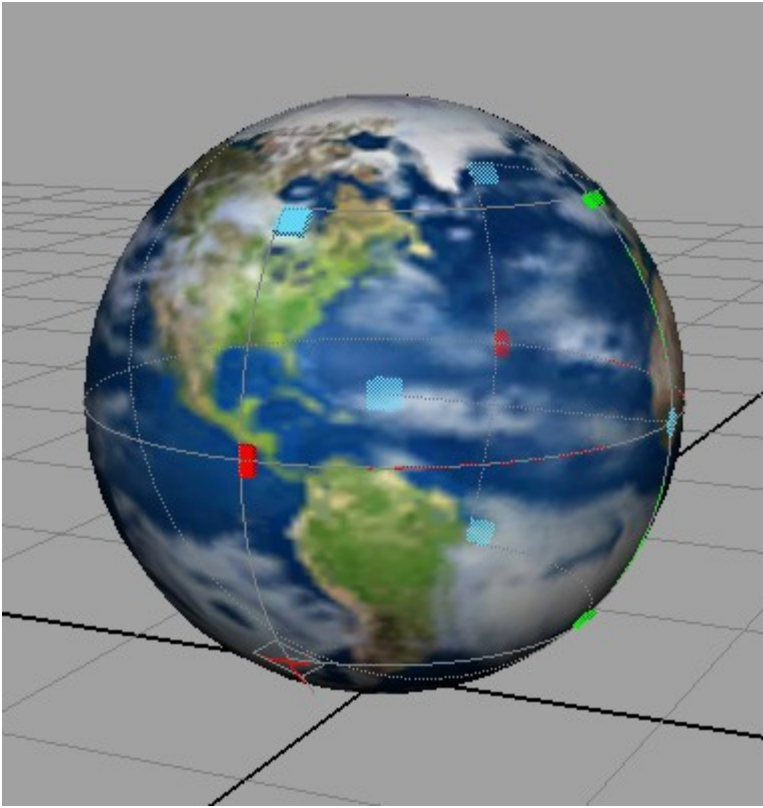
Planar - UVs are projected onto the selected mesh by projecting them along from one axis, similar to a slide projector shining light onto an object. This mapping method is mainly used for flat surfaces. This method is incorporated in another mapping method described later on called automatic mapping. Note that this method is known for stretches textures on surfaces that are not flat (see image directly above).



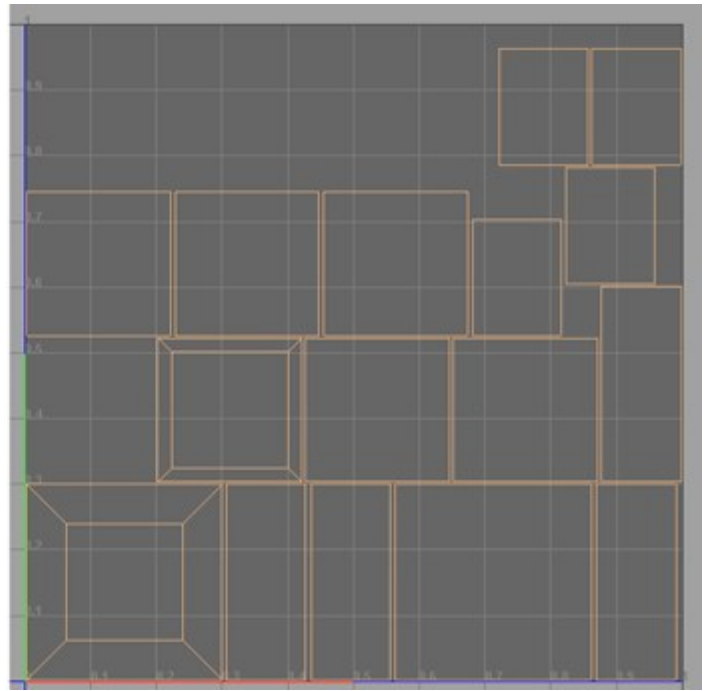
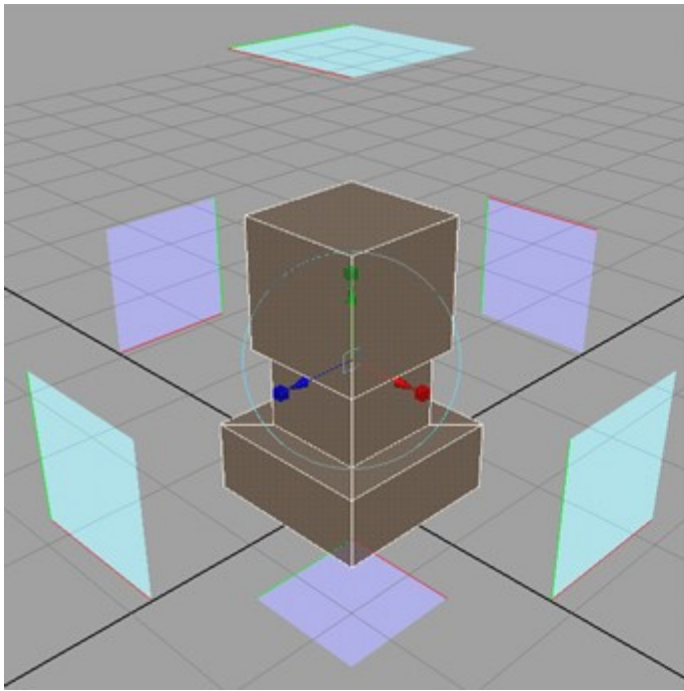
Cylindrical - UVs are projected onto an object as if the image were rolled into a tube and projected inwards towards the object.



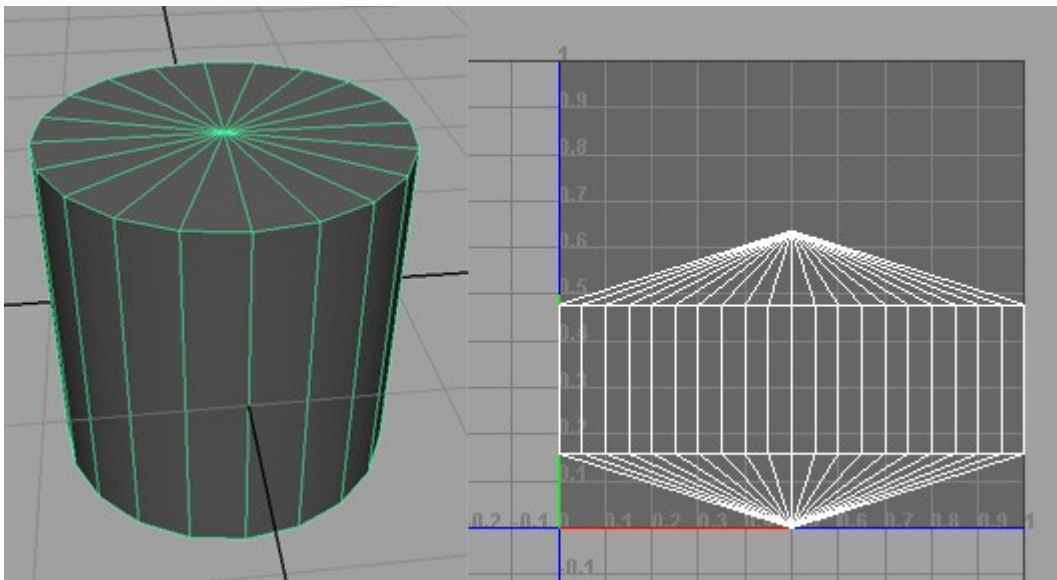
Spherical - UVs are projected onto the surface from an imaginary sphere that surrounds the surface.

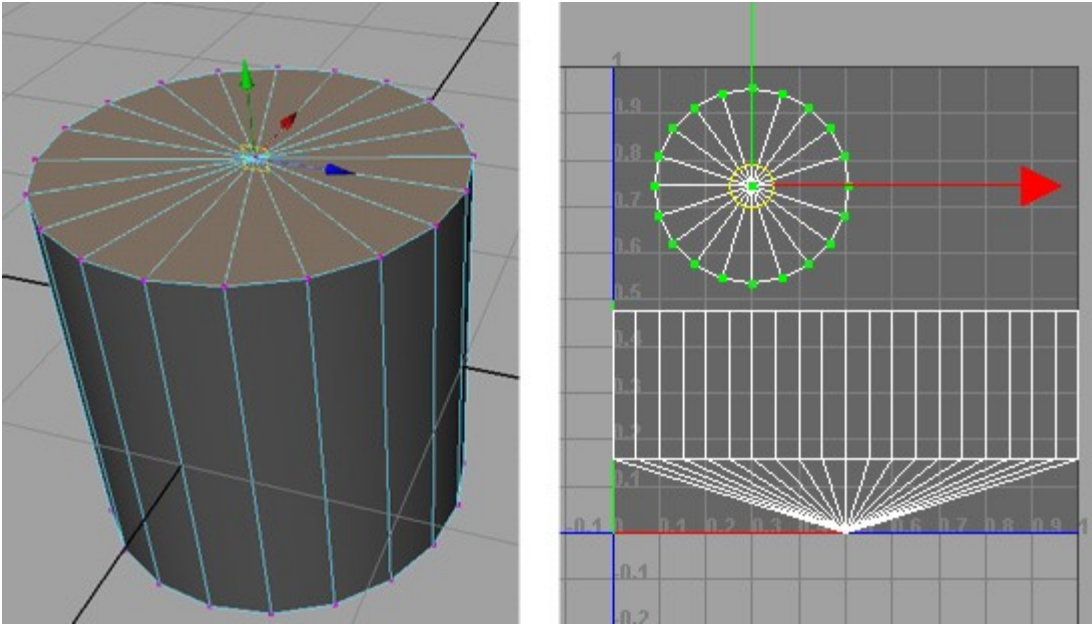


Automatic/Flatten/Atlas - These are systems that take the normals of your mesh into account and separate the result into many pieces, each laid out in UV space and not overlapping. Though often useful as a start, this sort of mapping is far less "automatic" than it sounds. It takes a huge amount of time to fix, and to turn into something usable. Generally, default "automatic" mapping has far too many seams to be useful. A lot of stitching is likely to be necessary if using normal based automatic mapping.



Applying any of the previous projection mapping techniques on a selection will detach the UVs contained within the selection to their own island.





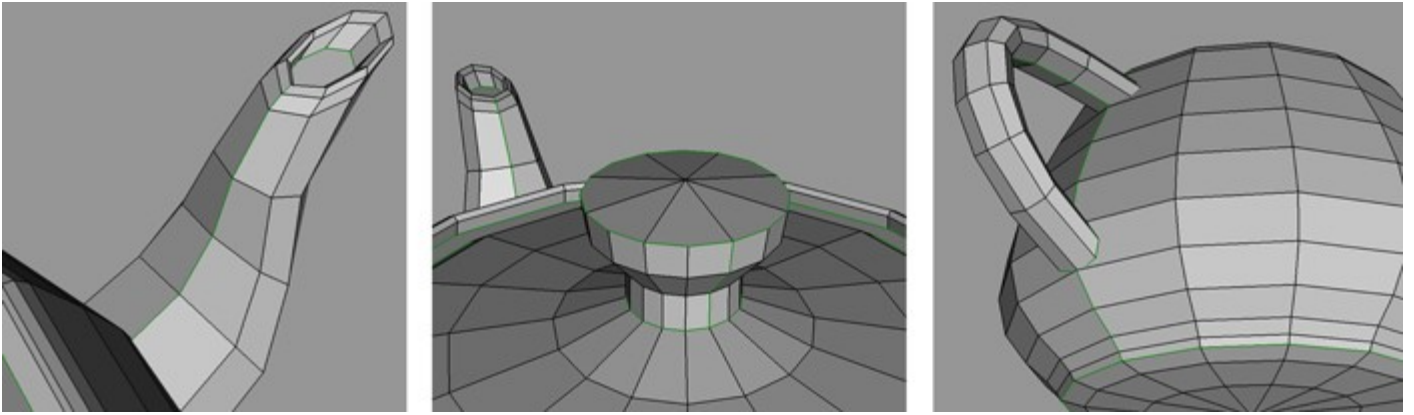
Pelt Mapping

- Also called unfolding, or LSCM mapping.

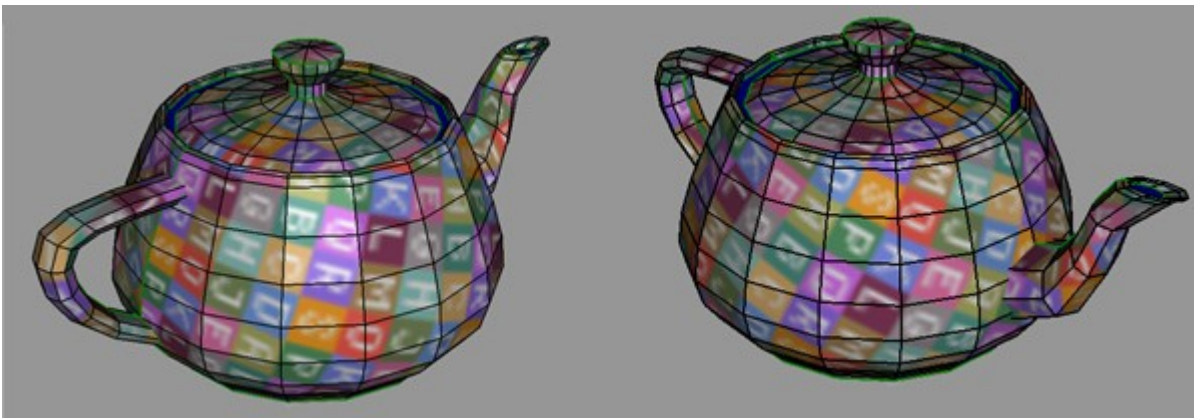
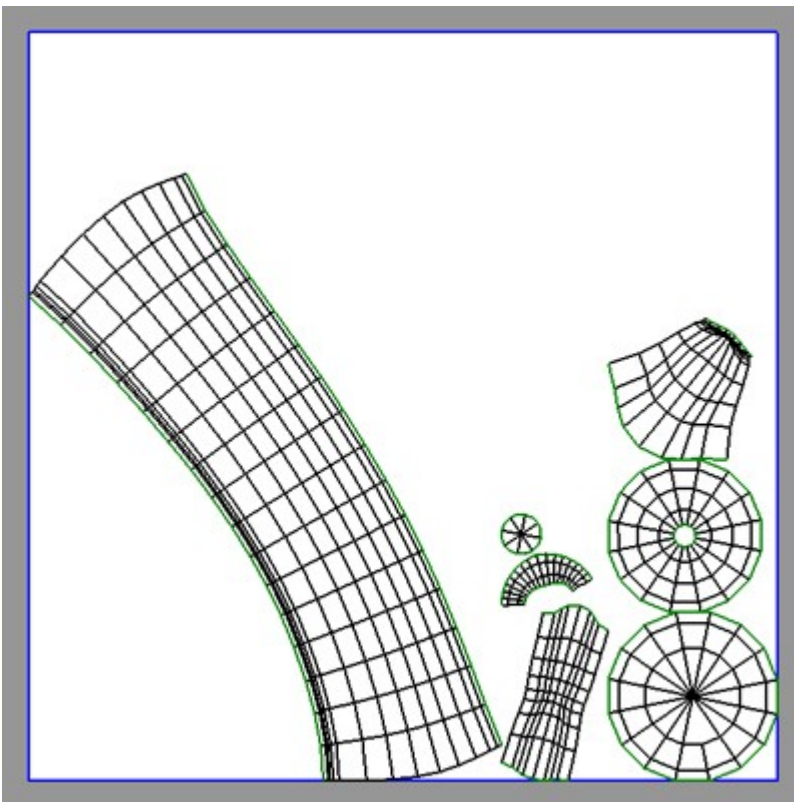
Unfolding in UV space:

LSCM means Least square conformal mapping. This is an advanced mathematical method to automatically create UV mapping while keeping texture stretch and deformations minimal. LSCM mapping generates mapping coordinates for the entire object at once. The artist marks seams on the object and the LSCM system determines a method of mapping the results in the least amount of stretching given those seams

- This is a modern and excellent method of generating UV coordinates extremely quickly even for highly complex objects. It will usually require some sort of editing afterwards. But gives a huge head start towards the finished product. It usually takes far less time than projection mapping.
- 3D Studio Max 8 includes pelt mapping.
- Many open source packages contain implementations of pelt mapping systems. Wings 3D has an excellent pelt based mapping system (highly recommended). It is free open source software which works well in combination with commercial packages. Blender 3D also has an excellent pelt based mapping system, though the software is generally harder to use than wings 3D.
- The biggest key to conceptualizing mapping on complex objects is to think of the object as cloth. Understand the object in the same way that a seamstress would. Many clothes can be looked at as examples of good layouts.
- Generally seams should be put where they are least likely to be noticed. Very often inside of creases or wrinkles or areas of the mesh that are usually covered up by other geometry.



In the image above you can see the green lines marked by the artist which represent seams. The seams get torn and unfolded which is shown in Wings3D's UV editor below.

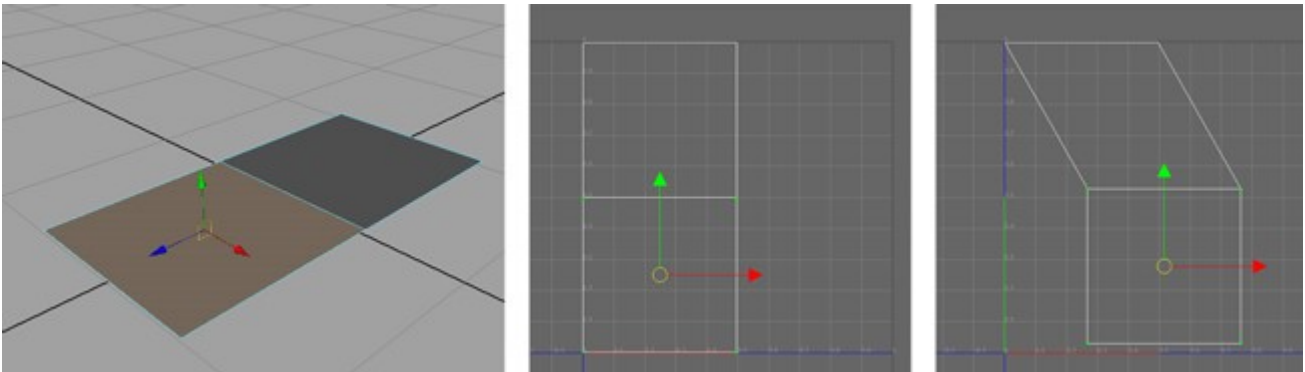


Look no stretching!

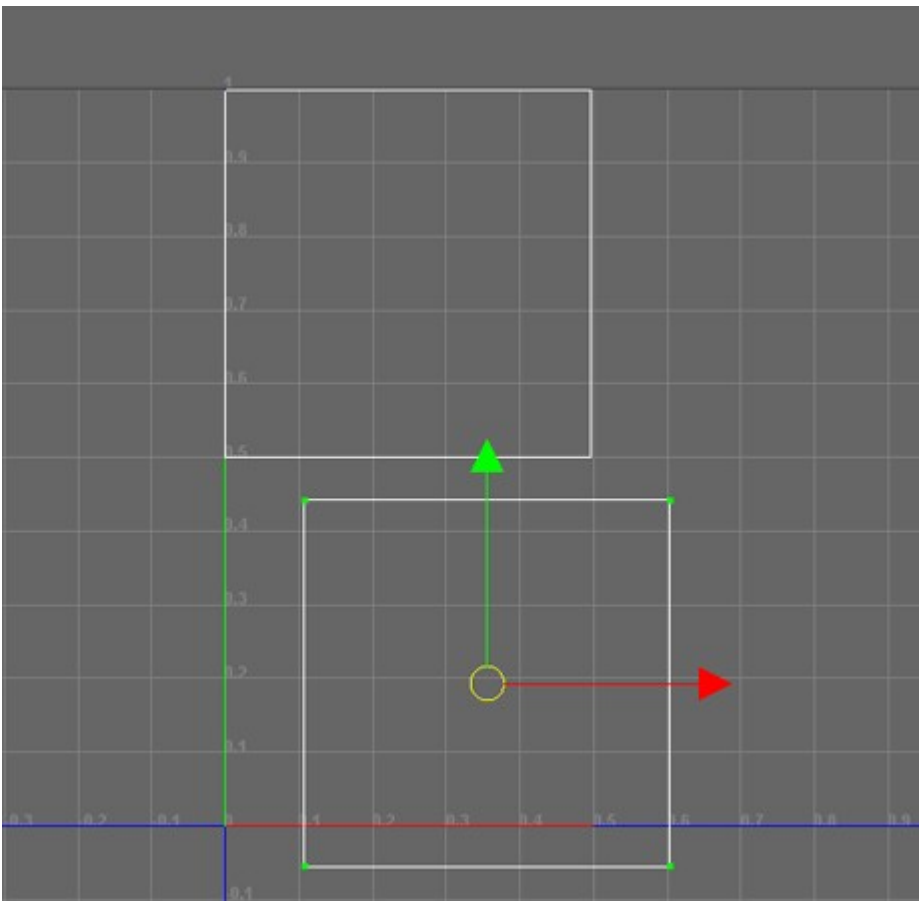
Editing UV coordinates

Cutting

In the example below, a face selection is converted to a UV selection. The face's UV selection is then moved demonstrating that the faces are connected.

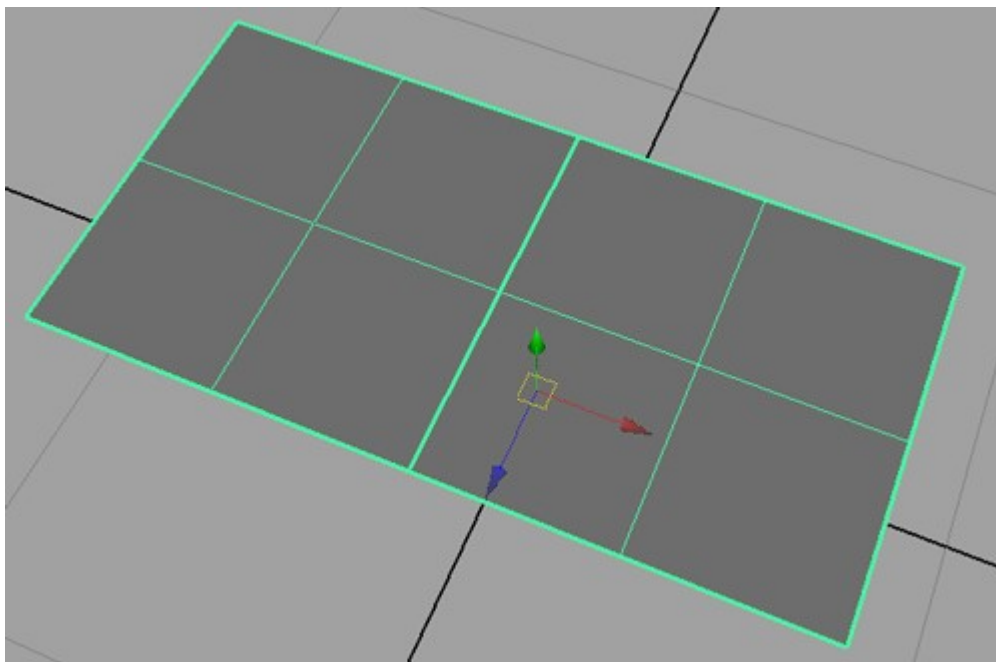


If you wanted to move the top face without effecting the bottom face, you would have to cut the connected UVs to separate them. The example below shows moving the face's UVs in a similar fashion after executing a cut command on the selection of the connected UVs.



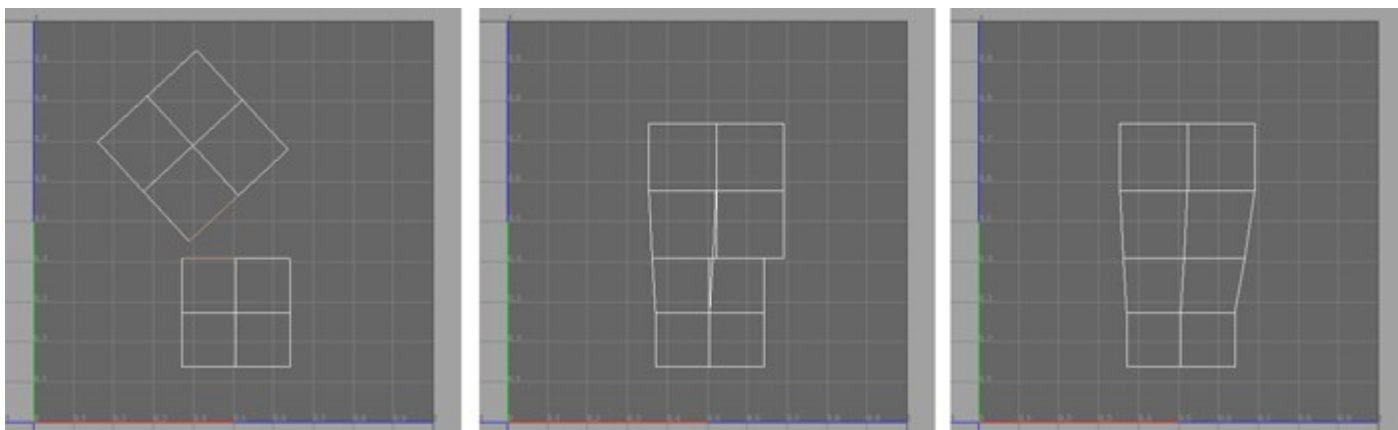
Sewing

Notice the seam found on the piece of geometry below, it is common to find within results from projection mapping a seam that shouldn't be there.



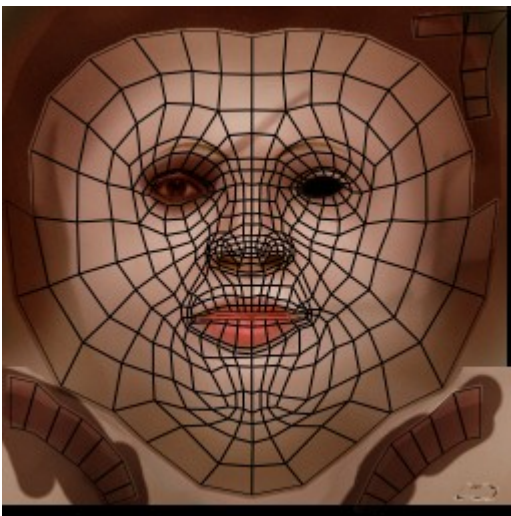
Selecting an edge that makes up a seam in a UV editor (first image below) will highlight the seam's connecting edge.

The majority of 3D software packages have a "move and sew"(Maya) command or a "stitch" (3DSMax & Wings 3D) command. When a selection is made that contains seams and a move and sew command is used, the UV island will snap to the corresponding island:



Layouts

A proper UV layout wastes as little texture space as possible.





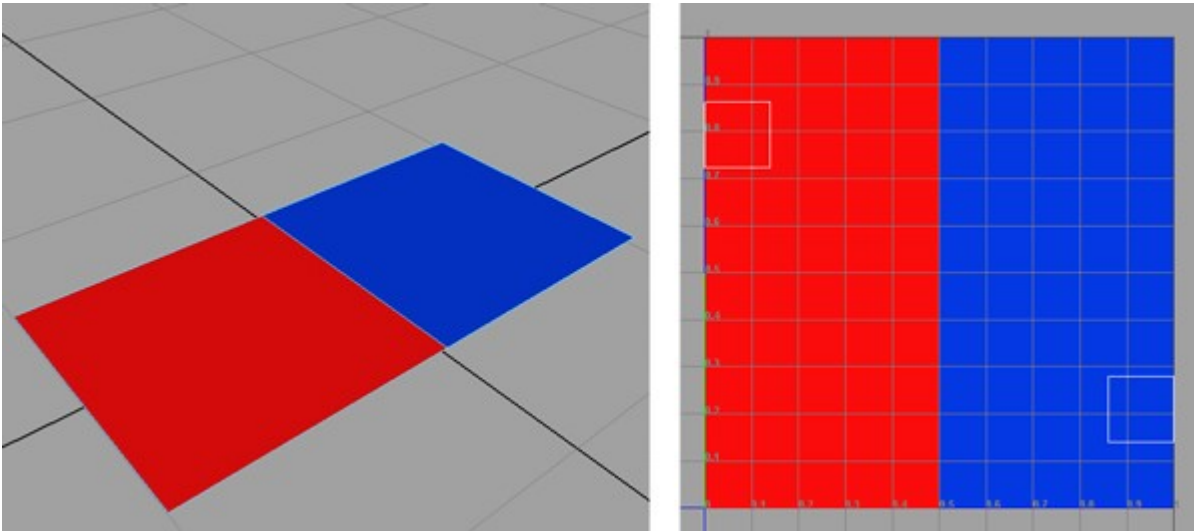
As you make good coordinates for everything, the sizing and layout will likely be all wrong. You should decide what sizes things should be in relation to each other, based on how important each part of the map is going to be. Once you have scaled the pieces to the appropriate sizes, sit and think for a while on the most efficient layout. How can the pieces fit together so that the least amount of space is wasted. Look for gaps in one piece that can be filled in by another piece. If there are holes in one piece, fill them with other pieces. It's rare that you can make perfect use of the space, but you should be able to make it very efficient in most circumstances. Again, see the example above to understand this concept.

There may be a tool to automatically take care of the layout and place everything inside of the 0 to 1 range in UV space. Something called "layout UV's" or "pack UVs". Although these kinds of tools may provide a good start, chances are you can get much better results if you really think about it yourself and lay them out manually. The computer is unlikely to know the relative importance of each UV island, and will likely not use it. If you are in a hurry, use pack UVs. If you want really good and efficient results, do it yourself.

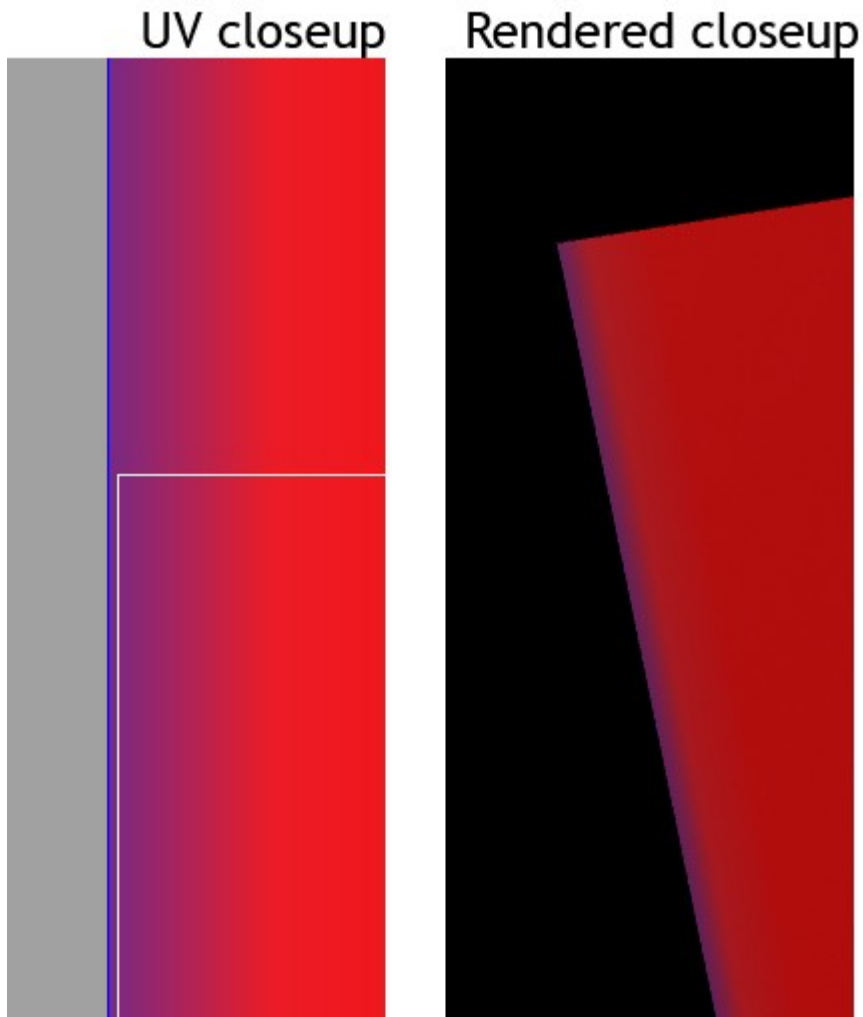
The majority of the above layout was created using a Automatic/Flatten/Atlas mapping command in conjunction with the move and sew technique. Some parts were pelt mapped with the help of Wings3D such as her face, main body and jacket.

Utilizing an island (element) selection feature can speed up such a process. Results from a mapping technique, such as projection, will sometimes include isolated parts of the mesh due to lacing their UV's into separate islands.

UV Padding

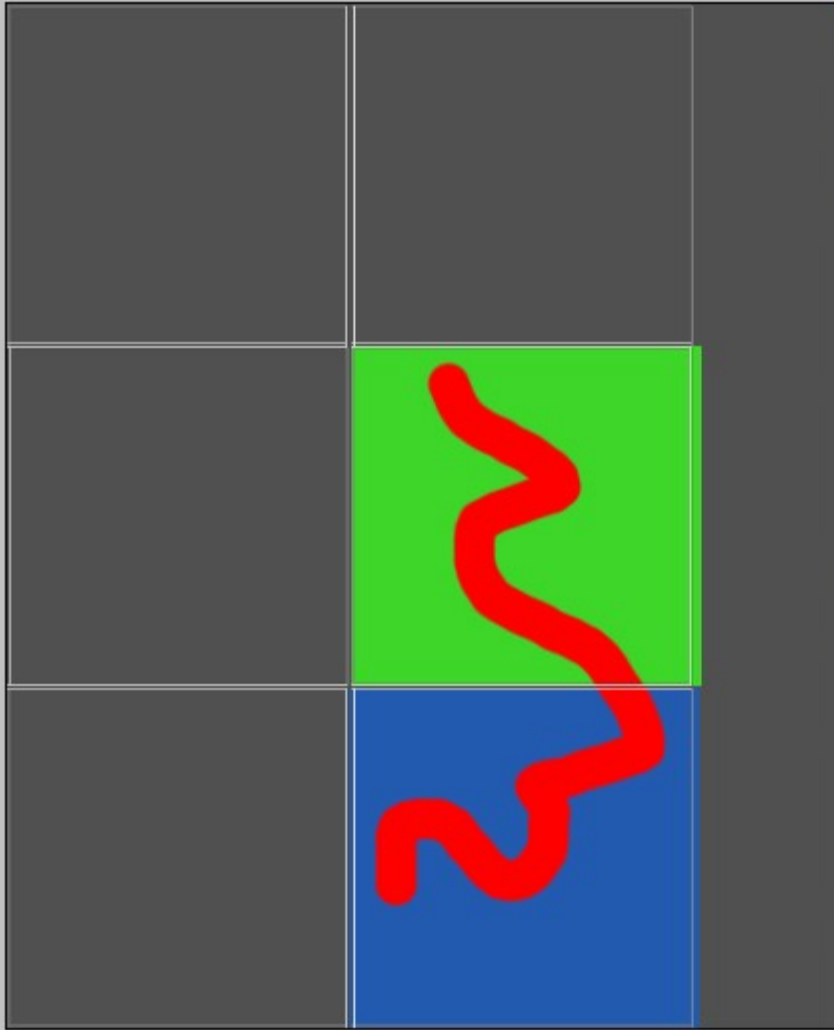


The 2 polygon plane shown above has its UVs layed out very close to the texture's edge. 3D software generally filters the textures which results in blurrier texture edges; colours from one area will blur into adjacents areas. The image below shows this clearly. Although it is important not to waste texture space, it is also very important to give texture space between islands and texture edges.



Snapshots






Once a UV layout is complete, 3D software packages include a UV snapshot function which will take a UV layout and export it as an image. Exporting your UVs allows you to use them as guides when paint textures with 2D programs such as Photoshop. Typically, PNG imaging format works best for texture painting since a "portable network graphics" image supports transparency. When using such an image in photoshop, the UV layout can be placed on top of your painting as demonstrated below:

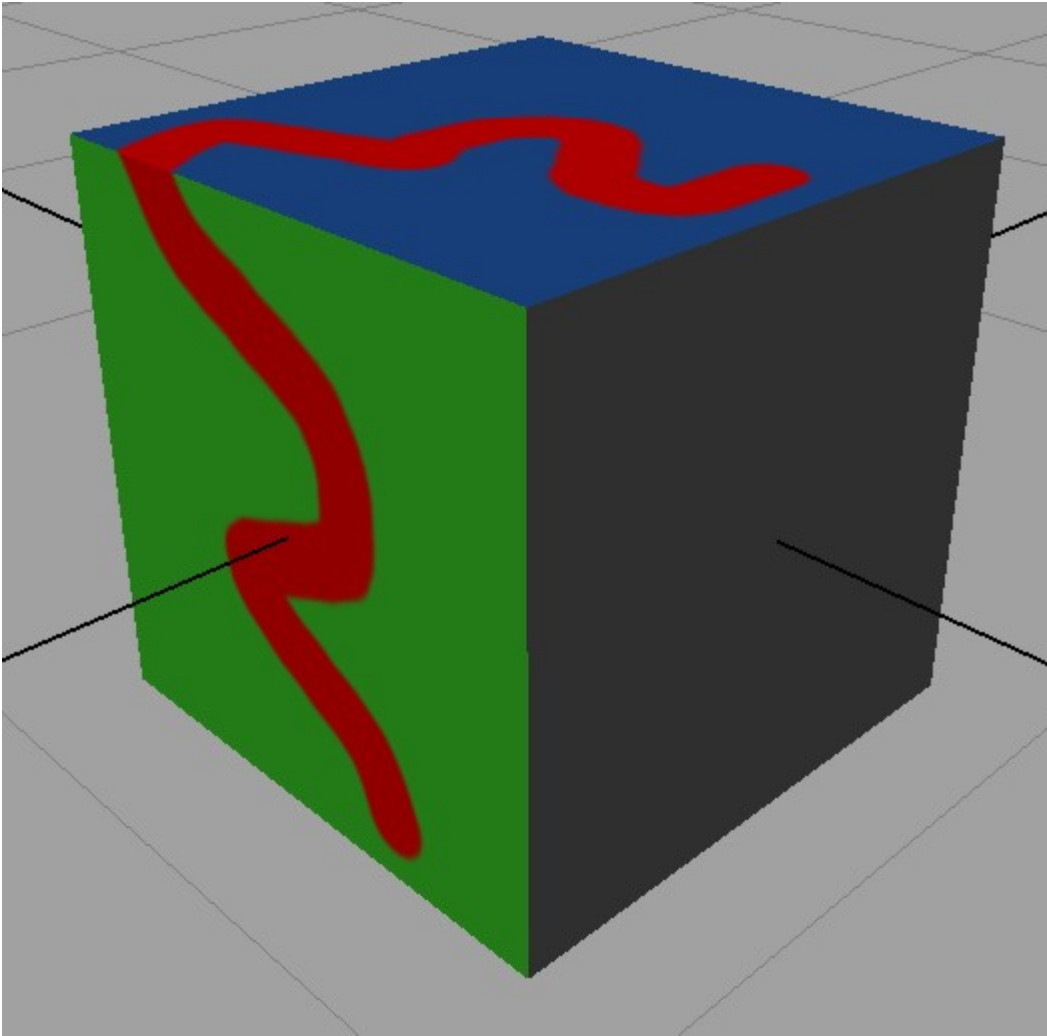


Layers Channels Paths

Normal ▾ Opacity: 100% ▾

Lock: Fill: 100% ▾

| | | |
|--------------------------|---|-------------|
| <input type="checkbox"/> |  | UV SNAPSHOT |
| <input type="checkbox"/> |  | RED LINE |
| <input type="checkbox"/> |  | BLUE FILL |
| <input type="checkbox"/> |  | GREEN FILL |
| <input type="checkbox"/> |  | BACKGROUND |



Texture Map File Resolutions and Sizes

If you're editing Uvs for a game model, please note that the resolution size of your texture must be in powers of 2, such as 512x512. These sizes are also useful because some 3D software can also have display errors when using textures that aren't square or powers of two. This is particularly noticeable with cheaper video cards, which tend to show a lot of problems with non square, non power of 2, textures.

Here are several values that are powers of two:

- 1
- 2
- 4
- 8
- 16
- 32
- 64
- 128
- 256
- 512
- 1024

2048
4096

Combinations of the above texture sizes are also commonly used. An example of a combination sized texture would be 256x128, a texture wider than it is tall. You should test these types of combination textures with your hardware to see if they are compatible with your hardware, target platform, and production requirements.

Of the sizes in the list, all of the larger ones are common texture sizes in games. The really small ones aren't often used, but many games actually do use textures as small as 32x32.

To figure out these numbers yourself, and understand how we get these numbers, take a calculator and press 2 times 2 times 2 times etc... and just keep repeating yourself, taking the answer and then multiplying it by 2. All it really does is double every time, starting at one. Its quite simple really.

General Concepts and Workflow Tips:

- No mapping will ever be perfect on a complex object. Many objects will require either seams or stretching to map in a useful manner. Generally either too many seams or too much stretching will ruin the mapping coordinates and make the texture difficult to paint. Thus, a proper balance between seams and stretching needs to be found. Generally a bit of touch up will be required to fix seams/stretching later. Examples of fixes include painting over them in a 3D painting program, or using some kind of mapping coordinate blending method and baking the result in a map.
- See the ideal hand layout example, on the ninja by Bobo The Seal. (<http://www.bobotheseal.com/>)
- Whenever possible, use image resolutions that are square, and a power of 2. There are a host of good reasons for doing this. Video cards will display images more clearly, the computer will be able to organize memory more efficiently, UV space will line up to your image and not be stretched, your art will more readily work in game engines, etc, etc...

Examples of such image resolutions are:

16x16,32x32,64x64,128x128,256x256,512x512,1024x1024,2048x2048,4096x4096, etc.....

Standard Projection Techniques VS The Painstaking art of UV Mapping

By: Leigh Van Der Byl
<http://leigh.cgcommunity.com/>

What on Earth is UV Mapping?

"UVs" is a way of saying u,v texture coordinates (as opposed to the X, Y, and Z axis that you construct your meshes on), which are points which define 1-by-1 positions within an image. These obviously connect to points in your 3D model, to position an image texture onto it's surface. Kind of like virtual "thumb tacks", what they do is pin an exact spot on an image that you wish to use to texture your model to a specific point on an object's surface. Between these points, your software will stretch the image smoothly. This is what is referred to as UV mapping.

So why use UV coordinates instead of standard projections? Well, once you have made your model, and are ready to texture it, the simplest way to apply your texture map is by using a standard planar, cylindrical, spherical or cubic projection.

Planar projection is a method whereby an image is projected straight through the object along either the x, y, or z axis. This method is especially useful for items such as sheets of paper, posters, book covers, etc - in other words, flat objects. The problem with planar projections is that if the texture is projected along an uneven surface, or if the image reaches a side that curves away from the projection's plane, it results in unsightly lines such as in Figure A. When this happens, you then have to create lots of alpha-channel enabled images to cover up seams between adjacent planar projections and invariably ends up becoming a huge amount of annoying work. So never project a single image through an entire object if it has depth, like the box in the image, or if it has a very non-planar or irregular surface. Rather, as in the case of this box, create separate projections for the x and y axis as well, making sure that their edges will blend together properly. Alternatively, ensure the image to be used is tileable (seamless), and use a cubic projection (which I will discuss in a moment).

Most software has a bitmap fit/automatic sizing option that will stretch the image to fit the surface properly. Obviously, if your image is not the same shape as the surface onto which you are projecting it, this fit option will stretch it until it does. This usually doesn't look too fantastic, so ensure that you measure the size of your object before making your image map.



Figure A - Planar Projections

Cylindrical projection is pretty self-explanatory. Basically what happens is your image is wrapped in a cylindrical fashion around your model along one of it's axis. This is really useful for one kind of object only - cylindrical objects. Please don't try and use this for anything else. When making an image that will be used for cylindrical projection, ensure that it's sides will meet correctly - in other words, where the two sides of the image wrap around and come together, make sure that there is no visible seam.

Tip: A useful way of ensuring that the sides of an image will meet and merge properly is by using the Offset filter in Photoshop (listed under Filters/Other/Offset), and offsetting the image by however many pixels you choose, and using the Wrap Around option.



Figure B - Cylindrical Projection

Spherical projection is when the image is stretched from one pole to the other along the axis you choose, and then wrapped around the sides from the back meridian. Make sense? Basically what I mean, is that, say for instance, you do a spherical projection along the Y axis, then the image goes straight from the top point down to the bottom point, and wraps itself around so that the two sides of the image meet along a straight line down the side of the sphere - this line is known as a meridian. Once again, only ever use spherical projection for spheres. Use it on any other kind of shape, and, well, it's not going to look very nice... Also, be sure to use the Offset filter (described previously) to check that your two end will meet properly along the meridian.

Another useful way of checking whether a spherical map will work nicely is to use the Polar Coordinates filter (listed under Filters/Distort/Polar Coordinates) and use the Rectangular to Polar option. (However - **ONLY USE THIS FILTER TO CHECK YOUR IMAGE. No NOT** save your image that you want to use with this filter applied - it is not going to work!). This can give you an idea or whether there will be any seams when the image is wrapped around the sphere. This method is of course extremely popular for newbies to map their images onto the planets they make for those sci-fi scenes that just about everyone makes when they are starting out in 3D. Come on, admit it, you have all made sci-fi scenes, haven't you? Use the methods I've described here to ensure that your planet doesn't have any unsightly seams tearing across it's surface.

Cubic projection just repeats a single image on each side of a box model it is applied to. Cubic mapping is basically a planar projection from 6 sides. Once again, ensure that the edges of the image will not form seams. Cubic projection is really only limited to perfectly square models, because if you try and use it on a rectangular shaped box, it will stretch the image on the long sides, and squash it on the shorter sides. Which looks pretty awful.

So, back to the original question - why use UV mapping instead of these options? Well, as you have seen, these projection methods are very limited. It's pretty obvious that they are not going to suffice for extremely complex models. But here is something bizarre - these projection techniques are actually, technically speaking, more accurate than UV mapping. This is because texture images will be more accurately mapped using standard projections, which have some exact, continuous values over the entire surface, whereas a UV map has accurate samples of the projection only at specific points (where the polygons join, basically), between which it then uses a linear interpolation on the surface between those points. For non-English speaking people, interpolation is, basically speaking, an estimation of values which go together to form a continuous series - this being a series of colour/tonal values being applied to your model, and the interpolation being the application of the parts of the image, which are not "tacked" down at those specific points, to the areas between them. The cool thing about this, however, is that once you have applied to UV coordinates to the model, pulling these points on your UV map around will pull the image with it. Pretty useful, hey?

UV Unwrapping

Once your model is complete (and I cannot stress enough how important it is to only ever start texturing your model once it is completely finished. Never begin texturing until your modelling is 100% finished. This just makes the entire process run much smoother), you are ready to decide whether your model requires UV mapping or not. And seeing as this chapter is about UV mapping, I'm going to assume your model does need it.

Firstly, you need to decide what method of unwrapping you are going to use (unwrapping being the term most commonly used for the actual process whereby the X, Y and Z information is translated into the flat UV template). Yes, I know it can be so tiresome that everything has to have so many options, but it would all be pretty boring and not much fun at all if we didn't have all these different methods, now would it?

Now, seeing as I personally am a Lightwave user, I am only really familiar with Lightwave's unwrapping techniques. I am sure, however, that these different methods are common to all the major 3D applications.

So, here are the different UV unwrapping options...

Planar Once again, we have a planar option. Yes, it's basically the same as before. The resulting UV coordinates are basically a flattened out straight-on projection through the surface along the desired axis. One thing to note, however, is that UV templates are always perfectly square (remember what I

said about 1-by-1 image proportions?), so what often happens is that your resulting UV map can look somewhat squashed. Don't worry about that, you can change it by polling your UV points with the UV map itself without actually altering the actual geometry of the object. But more about editing UV's later on. I personally use the Planar unwrapping more than any other method, as it generally produces the simplest maps with which to work.

Cylindrical Yes, it's the cylindrical option again. And yes it's also basically the same as before. One thing to watch out for, is that if your cylinder object has a top that you wish to include in the unwrap, be sure to unwrap along the X or Z axis only, as unwrapping along the Y axis will leave you with a completely flattened top.

Spherical Ok, you guessed it. It's also the same as before. You are probably wondering why you should bother using UV unwrapping if what they do is basically the same as the standard projection types, but as I said, the power behind Uvs is in the way that they "stick" to the points of your model and can be edited without affecting your models geometry.

Atlas (also known as automatic mapping or flatten mapping) This method of unwrapping may go by a different name in other applications, or may not be an option at all in some programs. Basically, Atlas unwrapping produces a UV template that translates the surface information into a UV map that represents the models polygons in a projection whereby once it is painted onto, will produce an image that will remain constantly perpendicular to the face normals of the surface polygons. Simply speaking, it is like taking a ball of paper that you have bunched up into an extremely irregular ball, and flattening it out again. The problem with Atlas unwrapping is that the resulting map is often a terrifying and confusing mess of disjointed polygons all over the show, as in Figure C. This kind of projection, although ideally useful, often ends up needing a lot of editing to get it into a state from which to work. I try and avoid it.



Figure C - Atlas Unwrapping. This is going to take loads of editing before you can export it to Photoshop and begin painting!

As I said before, I generally tend to use planar unwrapping the most. The problem occurring from this is that obviously when texturing, say for example, a character's body, multiple planar unwraps are going to be required (for instance, just the upper part of my Anubis character has 11 different planar UV maps applied to it), and because I make these all in separate files, what often happens is I end up with visible seams where the polygons which are using the different UV maps join. One way of covering them up is to make seam images which blend the different UV mapped surfaces into each other. There is another, easier method of avoiding this: When doing multiple unwraps for a single surface (as is the norm), choose a base colour that will be used as the base for all the different UV map images. Then, when painting onto the individual maps, just ensure that the detail you add lies a couple of pixels within the seam, that way, where all the UV maps meet, there is only that base colour between them so the seams will not be visible.

The same goes for bump, spec, etc maps - make sure that no detail "breaks" the borders of the polygons within the UV map, that would then become noticeable when they do not continue on the polygons using different UV maps surrounding them.

So what do you do now that you have your UV map?

Firstly, if any editing of the UV map is required, then do so. Your 3D program will have a bunch of

tools that you can use to edit the map, and as I said before, remember that altering this map is not going to affect your geometry in any way. Most UV unwraps will need some editing as fragmentation often occurs, resulting in polys that actually lie adjacent to each other being displayed on opposite ends of the template. Edit your map until you are satisfied that you can work well with it, and then you are ready to apply an image to it. You two options now...

The first option you use if you have already created an image map that you now wish to apply to the UV map. Change one of your viewports to display the UV map. Now, just stick the image you have made into the background behind the UV map, and pull the points of the map until they sit in the correct positions in correlation to the image. This is a rather bizarre way of doing it though, I must say that I personally have never used this method.

The second method is to export this template to Photoshop (or whatever painting program you use). There are two ways to do this - some Unwrap plugins will create an image for you, which you can then open up in Photoshop, or you can just get a screengrab (using the PrintScr button next to your Scroll Lock key), go to Photoshop, go to New Image (the image size will already be there, defaulted to the resolution your OS is running in), and Paste. Then just trim the square UV template (make sure you do this!! Lots of people have come to me asking why their UV maps aren't working, and it often ends up that they didn't trim the image down to the square size) and you are ready to begin painting.

As I discussed in Part 1, image size is important when making texture maps. To refresh your memory: In order to determine what size you should make it, you need to know what the final rendered frame size of the animation is going to be. Once you know what the final size is, then you can work out the size of your texture map as follows...

Take the maximum width that the texture map can appear on the screen (in pixels) and multiply it by two. Use this pixel size as the size of your image map, if you want to ensure that your textures do not become blurry or pixellated when viewed up close.

For example, the most common frame size when rendering for television is 720 pixels X 576 pixels (PAL D1), so if you were to make a texture image for an object that will be viewed right up close in the frame, then the width of your texture map should not be less than 1440 pixels. I usually work with square images, as I almost always work with UV Unwrapping, so my images are generally at least 1440 x 1440 pixels. It's usually safe to leave the image at 72 Dpi, as this is the resolution that monitors and televisions display at. Of course, the drawback to this is that these kinds of file sizes for image maps do slow down the rendering process quite a bit, but on the other hand, you know they will hold close-up. So if you have a job that is going to end up on IMAX, you had better make sure your computer can handle 4000 x 4000 pixel images....

Well, hopefully you have a clear idea of UVs now, as well as a good understanding of projections in general. You'll find that once you start practising this, and experimenting with them more, you'll wonder why they ever seemed confusing in the first place ;)

The thing to keep in mind is that planning projections and unwrapping and editing UVs is a long and sometimes painful process. But, unfortunately, it is a necessary process, and one that you will have to get used to doing if you wish to become a texturing master! Good luck!

---End of section by Leigh Var Der Byl---