Efficient UV Mapping of Complex Models

There you are at your computer, tumbling a nicely designed 3D polygonal model that needs texturing. Maybe it's a highly detailed architectural structure or some organic alien plant life created in your favorite 3D package. The assignment now is to texture-map this intricate model for a real-time game and then create the actual custom images you will assign to it. Aesthetically speaking, your texture-mapping work should not only be well detailed and designed but also appear seamless on your model and have a relatively even distribution of pixel resolution throughout your model. Technically speaking, your mapping should make efficient use of your texture sheet by using the maximum possible area of the texture space. Sound easy?

The process for assigning mapping coordinates is quite the same whether you are creating 3D geometry for a PC title, a console title, or a web-based game. The actual steps you take for setting up and mapping texture coordinates onto complex geometry are equally laborious for each of these platforms. The focus here is to get you through this less creative process efficiently and with great accuracy so that you can have more fun creating the bitmap images afterward. The more intuitive and interactive this creative process becomes, the better control you'll have and the more convincing your final result will be. In your efforts, you may find a plug-in or two that will help expedite a few steps in this process, but the basic steps are always the same. The "make it so" one-click feature that creates the perfect mapping hasn't been written yet for any 3D program. Until that happens, here are some proven basic steps you can take that will help you prepare your models for skinning and produce great mapping results every time. (For the rest of my column, I will assume you are experienced with 3DS Max 3 and know your way around Adobe Photoshop.)

Analyze Your Model's Geometry

If you are the original modeler of the 3D object you are about to texture-map, then you're already intimately familiar with every detail and polygon that makes up this geometry. If another modeler handed off the model for you to texture, take the time to analyze the design and look for ways you can begin subdividing the

TITO PAGÁN | *Tito is a seasoned 3D artist/animator working at WildTangent and teaching at DigiPen in Seattle. His e-mail address is tpagan@w-link.net, or visit his web site at www.titopagan.com.*

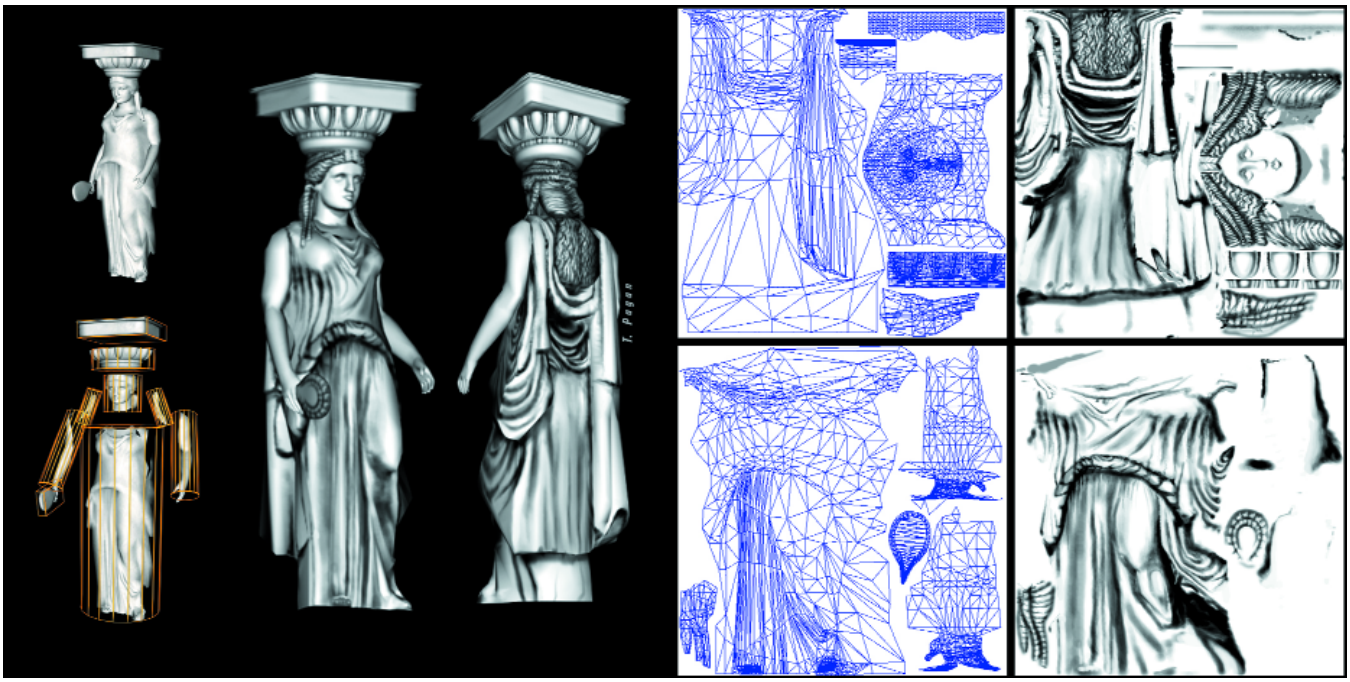


FIGURE 1. A decorative structural column (14,419 polygons) with precise UV mapping.

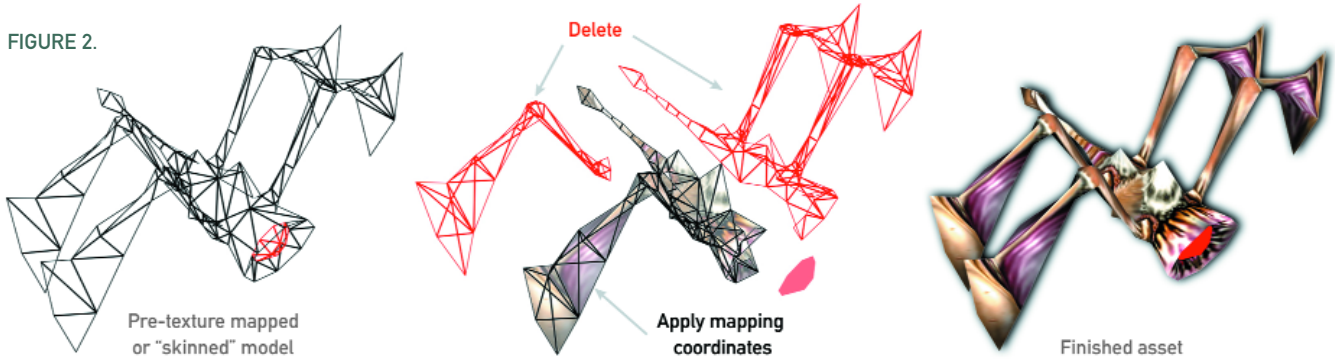
entire model into smaller groups of more primitive-like shapes (see Figure 1). Doing so will make your next step easier by breaking down your model into manageable parts to which you can assign separate mapping coordinates. For example, a spider's body can be thought of as narrow cylinders attached to a couple of spheres or a capsule-like shape. Look for elements in the design of the geometry that are identical. A spider's eight legs can be the same identical pieces without compromising much of the design. This is probably the way it was created to begin with — one leg was created and then cloned seven more times for a full set.

Subdivide and Consolidate

For this step you will dissect and delete parts of the model. If you work in a production environment with other contributing artists like I do, this may feel uncomfortable at first because you may be taking apart someone else's model. This is a necessary and practical step that can't be avoided. If the identical or repeating mesh detail is to have the same bitmap assigned to it, there is no point in defining mapping coordinates for this detail twice. Nor would it be efficient to use valuable texture space to represent the same texture detail on the model's similar geometry more than once. This conservation of time and consolidation of texture space can be handled in two ways.

The first approach you can take is to subdivide your model and delete anything that is represented in your model more than once (see Figure 2). This is the approach I often use. Focus on doing a good job applying mapping coordinates to only one of these similar parts. Later you will duplicate and attach the newly mapped parts back to the model in their original positions. Again, fewer pieces of geometry representing your entire model will equate to fewer polygons, less work in mapping these subparts of your model, and therefore more texture space real estate you can devote to these fewer parts.

FIGURE 2.



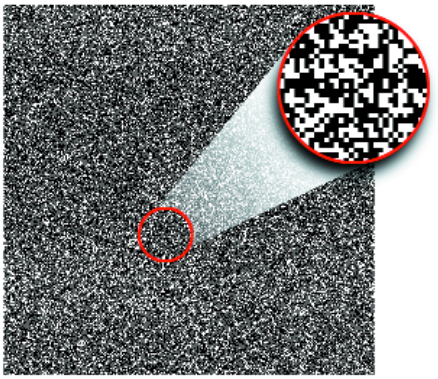
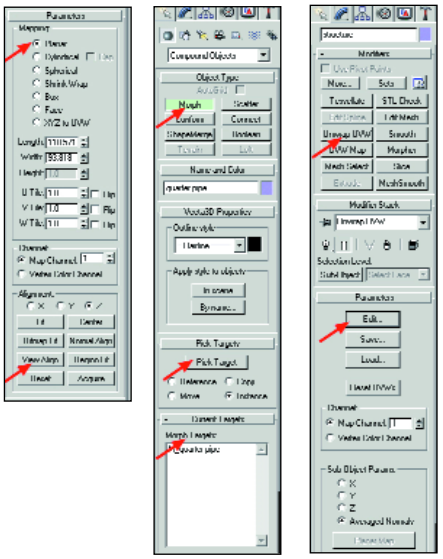


FIGURE 3 (above). Create a temporary bitmap and assign it to your model. FIGURES 4, 5, and 6 (below, left to right). The editing modes of the UVW Map and Unwrap UVW modifiers.



The second approach is to leave your mesh object intact and do a face-level subobject selection as many times as you need to and apply separate mapping coordinates for each selection. I don't recommend this approach because it is more work than it's worth, takes more time to execute, and doesn't yield better results.

Assign a Texture

You are now at a good place to assign a bitmap to your model. Create a texture file using your final asset name and place it in the directory for your game's final bitmap assets. Assign it to the model from within Max's Material Editor. This is usually a .BMP or .JPG image file that is 512×512 pixels in size or smaller. I like using the highest possible resolution for texture images that the game will support while I create them. This enables me to design and evaluate my textures at their most optimal level. It also gives me the flexibility of easily downsampling this size later to 256×256 or 128×128. This will depend on resolution limitations imposed by low-end graphics cards or file optimization issues that often arise toward the end of many projects. Another common practice is to create a "test purpose" temporary bitmap that has a consistent pattern throughout and assign it to your model. A checker pattern of small black and white squares is what I have found works well (see Figure 3). Other artists use numbers and colored squares in their test texture.

Applying UVs

Applying your initial mapping coordinates is a relatively simple process. 3DS Max offers many options for specifying how bitmaps are projected onto the surface of an object. Mapping coordinate types such as Planar, Cylindrical, or Spherical can be found using the UVW Map modifier (see Figure 4). Here again is a step in the process that can be handled in several ways. A high-production 3D artist will typically develop proficiency in Max for at least two or more approaches. I will discuss two of my personal favorites.

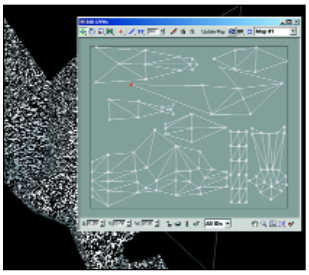
The first approach is to take your ready-to-map pieces of geometry and break them down into a "flat pattern" to which you can then apply a Planar-type mapping coordinate. This approach is like breaking down a simple cardboard box. A more complicated cardboard box, like a closet shoe storage box with cubbyholes, would of course present a bigger challenge to flatten out, but it can still be done. When you think about it, even an automobile's exterior body once started out as a series of flat-patterned sheets of metal that were formed and welded into a unique shape. This idea of reverse engineering the construction of your model's geometry requires you to know your way around Max's vertex manipulation and translation tools comfortably. You can find third-party plug-ins to help you through this process. Just remember always to clone a copy of this shape before you deform it into something suitable for mapping, because you will need it to morph (a compound object type for creating geometry) your newly deformed object back to its original shape. Use the copy as a morph target after you are done with your map assignments (see Figure 5). Avoid deleting or creating new vertices in either copy of this object. This is important to remember if you want to guarantee you'll get back to the original target shape during the simple morphing process. You can temporarily hide this copy to get it out of your way while you work on the other.

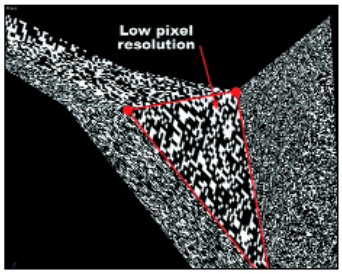
A second common approach to applying initial mapping coordinates is to leave your object's mesh intact and select a small group of faces you wish to planar-map. Make sure that these selected polygons all face the camera as if you were looking down their normals. This puts your selected polygons perpendicular to your camera. Assign a UVW Map modifier and set the alignment to View Align, thus reorienting the mapping icon to face the active viewport, which is the camera. This will be good enough for a first pass at applying mapping coordinates for these faces.

You could have easily rotated your object to put these selected faces perpendicular to an active orthographic viewport. Oftentimes, however, translating an object in world space presents problems with alignment to other objects. It can also throw off orientation during integration into your game. I don't recommend it.

Evaluate Your 2D Image Space

The Unwrap UVW modifier is used to assign planar maps to subobject selections of polygons and to edit the UVW coordinates of these selections (see Figure 6). I strongly suggest that you become very familiar with the Unwrap UVW modifier. We will use it as a UVW coordinate editor to unwrap and edit the existing UVW coordinates which were created earlier using the





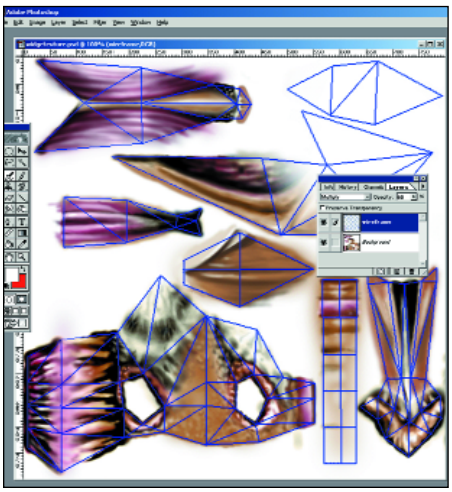


FIGURE 7 (above left). View your textures in the UVW window that displays the faces of your entire model. FIGURE 8 (above right). Scale your UVW vertices for consistent pixel resolution. FIGURE 9 (left). A dark line wireframe reference image with its layer-blending mode set to Multiply and a low Opacity setting.

UVW Map modifier. When the Unwrap UVW modifier is applied, it takes the current UVWs applied to the object and stores them in the modifier. If the incoming data on the modifier stack (from bottom of the stack upwards) is a face-level subobject selection, then only the UVWs for the previously selected faces are brought into this modifier. These same selected faces will display in the UVW window as UVW faces and UVW vertices overlaying a 2D image space of the map. (Technically, I should refer to bitmap coordinates as "UVs" instead of "UVWs" because the "W" represents the axis that is generally only used for procedural maps.)

You will want to scale and adjust your UVW vertices to ensure you are getting a consistent pixel resolution and distribution throughout your model. As you move the vertices around, you will see the texture move about the surface of your model as it updates in your viewport (see Figure 7). I suggest you turn on Texture Correction at this time by right-clicking the label of that viewport. Look at the checkered squares on the surface of your model. If areas appear to have rectangular checkers instead of square checkers then you have stretching going on in the direction of the longer side of your black and white rectangle. To correct this, select and scale the UVW vertices that define those faces in Unwrap UVW using the Scale Vertical or Scale Horizontal tools. If the checkers in some areas appear to be larger or smaller than most (see Figure 8), adjust in Unwrap UVWs using the uniform Scale tool until they better resemble the other checkers.

Don't be overly concerned about trying to use all of your 2D image space. Depending on your texture detail, contrast, and complexity, maintaining a consistent pixel resolution and clear surface detail may be more important than using up every possible area in your 2D image space. With time and experience you will get better at deciding which is more important as you adjust your UVW lattice. When you are done with all of your mapping you can weld your vertices back to quickly re-create the single object with which you originally started.

Create Your Bitmap Iteratively

Since a correctly named texture is already referenced to your model and the mapping coordinates are also properly applied, you are now ready to "improve" your existing bitmap. If you're like many 3D artists I know, the excitement is building up knowing the fun part is only moments away. You've just worked hard to prepare your model and you're about to slap down some cool colors and rich detail that will take this content to the next level of believability. I recommend that you do a couple of things before starting to make this last iterative step feel more interactive. Open and view your texture space in the UVW window that displays the UVW faces of your entire model. Take a screenshot to capture the 2D image space of the map that shows all of the UVW lattice lines. This will put a copy of what you see on your screen into your system's clipboard. Now launch Photoshop, if you don't already have it open, and create a new image by clicking on Control-N. Hit Enter when the options box comes up to accept the new image file with the default dimensions. Paste your clipboard image into it by clicking Control-V. Now you have a wireframe print of your UVW lattice for your mapped model as a reference. Crop your image to include only the 2D image space area in the image. Resize this image to 512×512 pixels using Image Size in Photoshop, making certain that the Constrain Proportions option is not checked. I like having a dark line wireframe reference image in a top layer with its layer-blending mode set to Multiply and a low Opacity setting (see Figure 9).

Finally, to help automate the iterative process of saving an updated copy of your image over the existing bitmap referenced in Max's Material Editor, we'll create a Photoshop Action. By doing so you can duplicate your image, hide or delete your wireframe reference layer, merge your other layers, save a copy of this image file over your existing one, and close the file, all with a single click of a button. You can now jump back into Max by hitting Alt-Tab. With the Material Editor open and your Bitmap Parameters>Reload button showing for your objects material, you're a click away from seeing how your new texture is shaping up, all in the context of your model or scene. Can you want more than that without the expense or learning curve of other third-party software?

These steps are all you will need to know to improve your skinning process and get up to speed with industry veterans. By eliminating tons of guesswork during the mapping stage, you can save yourself lots of time and headaches. Good art alone requires enough of your time already — these basic steps should buy you more time to create just that.