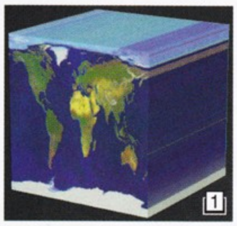
Unwrapping for IFW

Cylindrical mapping with an unwrapper utility

By R. Cory Collins • Images by the Author

There are three ways to apply bitmaps to an object in Imagine for Windows.



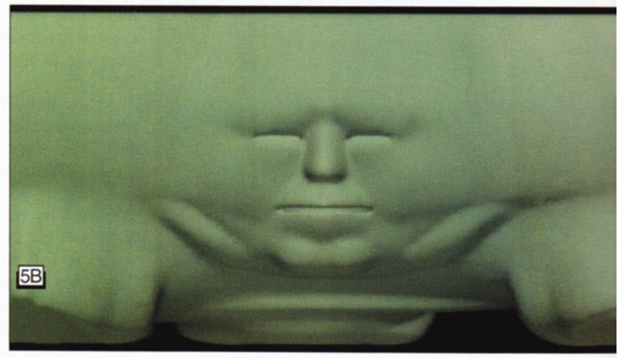
Planar, or "flat" mapping will apply a map with its width along the X-axis and its height along the Z-axis, and will stretch the depth to the length of the Y-axis (Figure 1). **Spherical** mapping wraps the map spherically around the object's Z-axis (Figure 2). **Cylindrical** mapping wraps the bitmap as a cylinder around your choice of the Z- or X-axis, much the same as a label is wrapped around a bottle or can of soda (Figure 3). This article covers cylindrical mapping, with the aid of a special technique.

Imagine will wrap cylindrical maps for you, but creating the right size map, with the features in the correct position, can be tricky. One of the best aids is Alex Chouls' unwrapper/scanner utility, an ingeniously simple way to create cylindrical maps of your objects, with nearly any size or resolution output you may need. The Scan.zip archive is available at Chouls' Web site at **pages.prodigy.net/ivorengine/index.html**.

The archive contains a specially mapped plane object, a stage file, and a short text file with basic instructions and a few tips for unwrapping your objects. When you unzip the file to your hard drive, a `\Scan.imp` directory is created, and a stage file (scan.ist), object file (unwrap.iob), and text file (unwrap.txt) are placed in this directory. A `\render` subdirectory is also created for storing the maps that will result. (You also will notice a few other objects and another stage file. These are for creating a spherically stretched image of a rendered scene. This method is a whole other ball of wax, so just ignore them for now.)

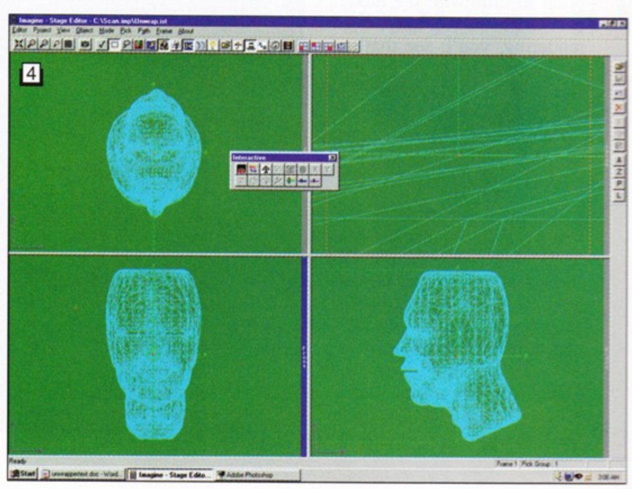
The unwrapping process works like a slit camera that scans a one-degree increment of your object in each frame of a 360-frame rotation. The images generated from these frames are stored in the render subdirectory, and then applied to the plane object in frame 361 of the stage file. The plane is specifically sized at a 9:5 ratio and already contains the proper maps, in the proper sequence, at the proper location. All you have to do is quick render. It may sound complicated, but it is actually ingeniously simple, once you run through the process a couple of times.

First, you need to set up your object for the best scanning results. This method works best on individual parts rather than the whole. It's better, for instance, to break down a character object into pieces like the head, arms, legs, and torso, and scan each individual part to create its own map. This will minimize distortion and maximize the ability to cylinder-map the cylinder-shaped objects that comprise a complex model like that of a human being. The central axis of each of these objects should be located as close to the center of mass as possible, with the Z-axis pointed toward the top of the object. Keep in mind that the object will be rotating around the Z-axis as if it were on a turntable.



Once the objects are broken down and set up, you can move to the Stage Editor and load the scan.ist stage file. Load the object to be scanned and, in the object info requester that pops up, set the frame numbers to 1 for the start frame, and 360 for the ending frame. Select your object and transform it to the absolute position of 0,0,0.

Turn on the camera lines and you will be able to see the -100/+100 unit scanning area. Scale the object to fit within the camera lines as seen in the right view. You might have to move the object up or down, and scale a bit more or less, to get it to fit. You may notice that you are zoomed extremely close up in the perspective view (Figure 4). Do *not* zoom out or adjust the camera in any way. If you want a better look in the perspective view of what is going on, use the View/Render Frame/Set command and change the aspect to custom, 1 in the horizontal and 200 in the vertical.



Go to the Action Editor and find your object's actor settings bars (it should be the last object in the list). Add the Rotate2 effect using 1 for the start frame, 360 for the end frame, Z-axis rotation, and 360 degrees. Return to the Stage Editor and render frames 1 to 360 with a 1:200 render size. One pixel width to 200 pixel height works well, but two width to 400 height will be twice as accurate.

Save the individual image files in the render subfolder with the name "strip" and TIFF as the file type. When the render is finished, you should have strip.0001.tif through strip.0360.tif in your render directory. These strips are the images that are mapped onto the unwrap.iob that is loaded in frame 361 of the project.

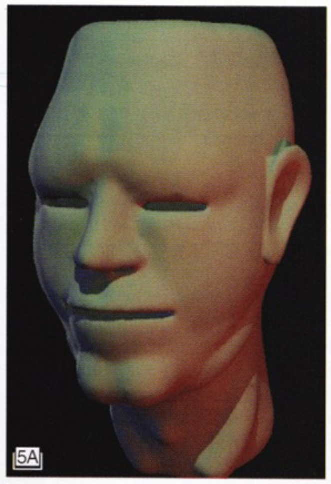
Now go to frame 361. If you set the render size to 1:200 earlier, or if it is still 4:3 (the default aspect), you should now use View/Render Frame/Set to set the size to 9:5. Quick Render frame 361 in scanline mode and you should now have a cylindrical map of your object. Save this image as basemap.bmp or something similar to load into your favorite image editing software to create custom color, bump, or specular maps for your objects (Figures 5A-E).

In the examples shown, I used Photoshop, loaded the basemap, and added a layer with opacity set to around 75% so I could see the features underneath. I also cropped the finished maps to where the map starts and ends at the very top and bottom edge of color. This aids in alignment of the map on the object within

Imagine. When you apply the map in Imagine with the attributes requester, you will see that the map usually is in very close alignment, once you have selected the cylinder wrap option. If the map needs some adjustment try rotating a degree or two on the Z-axis before doing any scaling. This usually does the trick.

The above example shows a head object, but this method also works well for mapping arms and legs as well (Figure 6).

In some cases, especially with arms and legs, it is hard to tell which features are on a particular side of an object. Look at the base cylinder map of the arm (Figure 7). It's tough to tell which bump is the protruding portion of the elbow and shoulder, and which ridge is the crook of



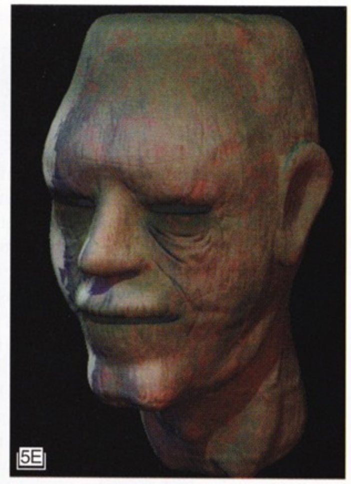
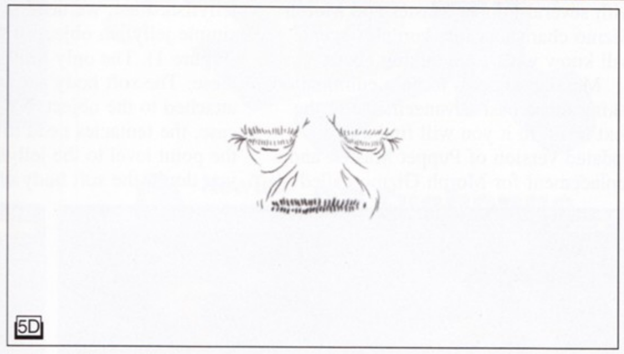
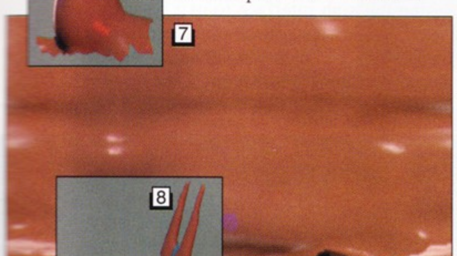
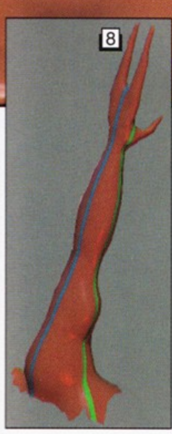


Fig. 5A. Head model with no maps or textures.
Fig. 5B. Head object "uwrapped."
Fig. 5C. Color map, derived from 5B.
Fig. 5D. Bump map, drawn over 5B for reference.
Fig. 5E. Head object with maps and textures applied.







the elbow. To remedy this situation, I applied a stripe texture with one color running along the top and bottom edge of the arm and another color along the front and back of the arm (Figure 8). I then scanned the arm again, and painted the new map accordingly (Figures 9A-C).

When I applied the maps in IFW, I used the striped map for placement and scaling, then substituted the color map by changing the name of the map file. This makes it much easier to line up features on the map with features on the object. Next, I duplicated the color map, changed its name to the bump map file, and changed its type to altitude. Figure 10 shows the result.

Cheers go out to Alex Chouls for developing this technique and sharing it with us. It is very powerful, easy to implement, and will complement any Imagine user's arsenal of tools for getting the best possible results.

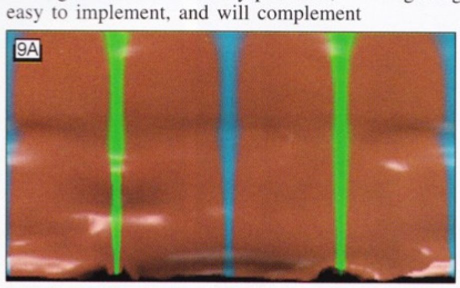
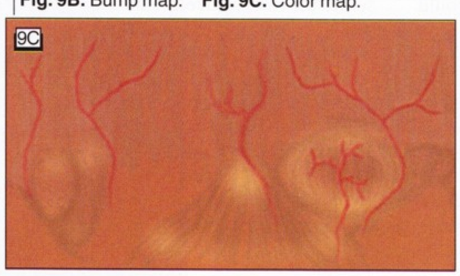


Fig. 9A. Unwrapped arm map with alignment stripes. **Fig. 9B.** Bump map. **Fig. 9C.** Color map.



Alex Chouls, on his Web site, also suggests using this utility for consolidating multiple maps and for converting procedural textures to simple maps.

***R. Cory Collins** is a freelance artist who has been using Imagine since its earliest days on the Amiga. In **3DA**#36, he wrote about facial animation and character design. In issue #37, he showed how he developed a six-legged walk cycle for a creature for Eagle Films. You can reach him at <rccollins@alltel.net>, or visit* **www.alltel.net/~rccollins**.