Subject: Re: mesh clipping
Posted by Rick Towler on Wed, 28 Aug 2002 16:24:25 GMT
View Forum Message <> Reply to Message

"lyubo" <lzagorch@cs.wright.edu> wrote

> Rick, you were right. I really want to slice the mesh up interactively
> and that's why I was trying to clip it to a plane. I guess alpha blending
> will be faster but the question that I have here is how can I use alpha
> blending with a mesh? I thought that I can apply alpha blending only to
> texture mapped polygons, by using an alpha image as texture. With
> the mesh I don't have any texture. I will try to find examples on the net,
> I just wanted to thank you for your reply.

Ahh, you have a wire mesh....

You are *mostly* correct in thinking that you need to work with texture mapped solid polygons to use alpha blending. In IDL 5.5 there is a bug that allows you to texture wireframe models. But, before we go there, you need to texture your polygon first...

For now, work with a solid polygon. Let's assume you want to draw your polygon in grey. Create a instance of IDLgrImage with this texture data:

imagedat = [[180,180,180,255],[180,180,180,0]]

Use this image object to texture your polygon.

The trick will be setting up the texture coordinates. Your texcoords array will be a 2xn array where n is the number of verticies in your mesh and each coordinate pair maps a pixel in your image to a vertex in your mesh. So, for verticies you want "on" you will give it a texcoord of [0,0] and for verts you want off, [0,1] (or is it [1,0]? Well, you get the idea).

There are a few things to watch out for. One is that if I remember correctly, I don't actually use texcoords of 0 or 1 to assign pixels at the edge of my texture. I ended up using 0.001 and 0.999. Unfortunatly I can't remember why...

A second issue will be that you will not have a cleanly defined edge along your slices. IDL will blend from opaque to transparent giving you a "soft" edge. This may be a result of the type of shading used though..

And then there is the order in which the polygon is drawn. It has to be drawn back to front. And if you rotate it 180 degrees you draw it back to front, which turns out to be front to back. I usually end up slicing my mesh into a +z portion and -z portion and then keep track of where the camera is and flip the two objects in my model when the camera crosses the

xy plane.

Ahh, the wire mesh... Like I said, IDL 5.5 has a bug where wire mesh polygons can be textured. It just doesn't work as expected. But you should be able to get it to work. Start with the solid and get that working...

> As far as my graphics adapter, I use Nvidia GeForce3 on a P4 2.0GHz
> dual processor with 512Mb Ram platform. Which graphic adapters
> support rendering of volumes?

That I can't answer. We don't do volumes so I haven't ever investigated this. I can tell you that the high end consumer cards like your GF3 are optimized for gaming. They concentrate on fill rate first, then polygon count. If there is any support for volumes it is WAY down the list.

-Rick
