Subject: Re: object graphics - transparent surfaces
Posted by Rick Towler on Tue, 30 May 2006 16:49:31 GMT
View Forum Message <> Reply to Message

greg michael wrote:
> I'm trying to make some 3d red/blue anaglyphs using object graphics. I
> manage to blend the red surface onto the blue using the alpha channel
> and /depth_test_disable on the second surface, something like:
>
> oSurface2 = OBJ_NEW('IDLgrSurface', data,x,y, style=2, alpha=.5, $
> color=[255,255,255],texture_map=oImage2,shading=1)
> oModel2 = OBJ_NEW('IDLgrModel',/depth_test_disable)
> oModel2->add,oSurface2
>
> In most cases it works fine, but occasionally I get ugly pure-red bands
> on the tops of steep ridges. Looking more closely at the idlgrsurface
> alpha_channel documentation, it seems to be saying this technique is
> not recommended... some facets may be rendered in the wrong order for
> transparency. So is there a better way?

Yeah, this is a problem. What might happening is that the front of your peak is rendered before the back so it isn't blended with anything that lies behind it (because at the time it is drawn nothing is behind it). Viewing at certain angles you lose part of your surface behind your peak. This is an intra-object transparency issue.

There are also inter-object transparency problems where an entire object is draw out of order which the docs address in the first sentence of that 2nd paragraph.

> I could render one, read it back, then the other, read it back, blend
> them myself, and then display. But that wouldn't be a nice way to make
> an interactive object. By the way, I'm using 6.1, just in case anyone
> happens to know whether it's been changed since.

There are ways around these limitations, but none of them are particularly elegant. The best thing to do would be to send an email off to RSI requesting order independent transparency rendering for the OG engine. You can double the number of people requesting this feature ;)

The simplest fix would be to order your objects in their container depending on viewing angle. Your problem arises when you most likely rotate your objects 180 degrees and the back surface is now in front of the front surface and there is an inter-object issue. The more sophisticated you get, the better the results. For more complex objects you will get better results rendering your surface as a collection of smaller surfaces all of which are ordered dynamically as you manipulate

them. This does not address intra-object rendering issues but it minimizes them by limiting the number of self-overlapping areas. It would be best to subset your data but if you can't you can use MESH_CLIP. It's a little hairy but it works.

This approach will take longer to set up but will render at full speed since all you will be doing will be changing the order of your objects in the parent container.

Another approach would be to control the order in which the individual polygons of each surface are rendered depending on viewing angle. This addresses the intra-object issue. A crude approach would be to create maybe 4 polygon connectivity arrays, one for each cardinal direction, where the surface polys are rendered from back to front. You would then change the polygons property of your surface depending on the viewing angle. You wouldn't use IDLgrSurface for this, instead you would write your own meshing algo and use IDLgrPolygon (I have a simple meshing routine for surfaces if you want it). This too would render quite fast.

A more complete solution would be to reorder on every draw. Karl Schultz worked up a dlm which does this using binary space partition trees called bsp_poly. This was a few years ago and it was a bit raw. I can't find it on the code contrib site but Karl may be willing to post it or send it. This would be by far the slowest to render but it would yield the "truest" results.

You may want to try my camera object. It will simplify tracking the viewing angle and you may want to try its built in (beta) 3d features. It supports shutter glasses connected thru the serial port (requires some simple hardware hacking, schematic included) and it also supports rendering to single/dual screens or dual projectors for viewing with polarized glasses. If you are interested in trying this let me know.

-Rick