
Subject: Re: object graphics - transparent surfaces
Posted by [Michael Galloy](#) on Tue, 30 May 2006 16:16:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Presumably, you rotate or translate each surface some small amount to represent the change between eyes? Doesn't this cause peaks of the surface stick out? The problem with transparency in object graphics is the order the atoms are added to their parent matters. So if you have two atoms that intersect -- i.e. neither is strictly in front or behind the other, sometimes one is in front, sometimes the other is -- then you can't choose an order that makes the transparency work.

My suggestion would be to rotate and render each surface separately in a buffer, read the buffer, and then blend the images (either yourself or using ALPHA_CHANNEL).

I wrote some code that does this, but I have to re-implement this in order to give it away.

Mike

--

www.michaelgalloy.com

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=olmage2,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

Subject: Re: object graphics - transparent surfaces
Posted by [greg michael](#) on Tue, 06 Jun 2006 12:44:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Rick,

Thanks very much for your detailed and very useful reply. I was hoping for a nice keyword that would sort that out... it looks to be not quite so simple!

Certainly I can add my request to RSI - I suppose, from the wording of the documentation, they already think it's something to be fixed.

I'd be very interested to take a look at your meshing routines, and your camera object, too.

kind regards,
Greg

Subject: Re: object graphics - transparent surfaces
Posted by [greg michael](#) on Tue, 06 Jun 2006 12:53:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks, Mike. Yes, that's what I do. The peaks do stick out, but what you would expect would be a blue or red shadow on topographic edges. What I get (when things go badly) are blocks of pure red, where the underlying (from behind the ridge) blue information is obliterated. A pity there's no image posting facility here - I could show you.

I agree that your solution should work - in the short term, I shall probably do that.

many greetings,
Greg

mgalloy@gmail.com wrote:

> Presumably, you rotate or translate each surface some small amount to
> represent the change between eyes? Doesn't this cause peaks of the
> surface stick out? The problem with transparency in object graphics is
> the order the atoms are added to their parent matters. So if you have
> two atoms that intersect -- i.e. neither is strictly in front or behind
> the other, sometimes one is in front, sometimes the other is -- then
> you can't choose an order that makes the transparency work.
>
> My suggestion would be to rotate and render each surface separately in
> a buffer, read the buffer, and then blend the images (either yourself
> or using ALPHA_CHANNEL).
>
> I wrote some code that does this, but I have to re-implement this in
> order to give it away.
>
> Mike
> --
> www.michaelgalloy.com

Subject: Re: object graphics - transparent surfaces
Posted by [Michael Galloy](#) on Fri, 16 Jun 2006 15:52:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

greg michael wrote:

> Thanks, Mike. Yes, that's what I do. The peaks do stick out, but what
> you would expect would be a blue or red shadow on topographic edges.
> What I get (when things go badly) are blocks of pure red, where the
> underlying (from behind the ridge) blue information is obliterated. A
> pity there's no image posting facility here - I could show you.
>
> I agree that your solution should work - in the short term, I shall
> probably do that.
>
> many greetings,
> Greg
>

OK, check out

<http://michaelgalloy.com/2006/06/16/anaglyphs-mggr3dconvert-r-and-mggrwindow3d.html>

and see if that works for you. You should be able to make your normal object graphics hierarchy and then make a MGgrWindow3D as your destination:

```
owindow = obj_new('MGgrWindow3d')  
owindow->draw, oview
```

Mike

--

www.michaelgalloy.com

Subject: Re: object graphics - transparent surfaces

Posted by [greg michael](#) on Wed, 28 Jun 2006 13:46:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Mike,

That works very nicely. I like the replacement window object, too. It wouldn't have occurred to me to make a separate object for the converter - I'd have done it in the window object. What's the advantage to splitting them? (I probably don't get as much out of objects as I could...)

By the way, your code uses `pref_get`, which isn't included on your page.

kind regards,
Greg
