Subject: Re: texture map irregularities OR pimento problems
Posted by Karl Schultz on Fri, 22 Jun 2001 23:11:01 GMT
View Forum Message <> Reply to Message

"Rick Towler" <rtowler@u.washington.edu> wrote in message
news:UZNY6.226424$p33.4540384@news1.sttls1.wa.home.com...
>
> "Karl Schultz" <kschultz@researchsystems.com> wrote in message
> news:9gvt3o$lkk$1@news.rsinc.com...
>> The order in which the polygons are drawn is extremely important when
> using
>> alpha blending.
>>
>> In the first case:
>>
>> 1) The red orb is drawn first without alpha blending. So far, so good.
>>
>> 2) The green orb is then drawn with alpha blending. You don't know
>> beforehand
>> which polygons in the green orb draw first. What is happening is that
the
>> top
>> (pos z) part of the orb is drawing first. This is the transparent part,
> and
>> after we draw
>> this part, we see the red orb and parts of the background, as we expect.
>>
>
> So the order in which the polygon VERTICIES are drawn per the polygon
> connectivity matters as well as order of the objects in the view?

Yes, but it is really more the order in which the POLYGONS (in the mesh
generated by Orb) are drawn according to the connectivity list. (Many
vertices are shared between polygons, so it is really the polygons we are
worried about, but I think we're saying the same thing) And the drawing
order really only matters if you are using alpha.

> So in the
> first case, if the polygon connectivity array was "flipped" so that the
orb
> polys would be drawn back to front the image would render correctly.

Yes.

> I
> would test this but it looks like the orb is comprised of a top and bottom
> drawn from triangles and a body from rectangles so I can't easily flip the
> polygon array. (this also explains the green and red pimples you see with

my
> example.)

We ship the source code for Orb, so you can see what is what is going on if
you want. But I wouldn't code anything that relied on an orb internal. (OO
programming)

But I think that you could also "flip" any arbitrary polygon connectivity
list pretty easily without knowing anything about its shape or what created
it. Just walk the connectivity list and copy each polygon descriptor to a
new descriptor list where you append each descriptor to the front of the new
list.

A descriptor looks like: (list of LONG)
n v0 v1 v2 ... vn-1

where n is the number of verts in that poly.

Anyway, in this case, it might be better to just rotate the orb with a
grModel. Luckily, orbs look the same (disregarding tesselation details)
after a rotation, so you could just rotate it so that the top (+z) is
pointing away from the viewer. Since we know (by observation) that the orb
draws from + to - z, that will make the back part of the orb draw first.

In your first example, applying a 180 degree rotation about X or Y would do
the trick.

This still relies on Orb internals - Orb could change in such a way that
this assumption on drawing order would no longer be valid. To be safe, you'd
have to control the generation of the mesh itself.

> A single polygon that is transparent on opposing sides can not be rendered
> correctly. In order to render this correctly you would have to break it
up
> into parts and order them correctly in the view according to the camera
> position in z.

Do you mean a single polygon *mesh*? If so, right. Unless you can rotate
the object so that the drawing order of the polygons in the mesh gives you
the ordering you want. For more general and more complex objects than a
sphere, this could be hard or impossible. In the end, you'd have to control
the order somehow, and breaking it up would work.

> If I understand you then this all makes sense. I never thought about it
and
> assumed that as long as objects were ordered correctly in the view then
they
> would be drawn correctly (which is true until you start alpha blending).

> This actually sheds some light on a number of subtle issues we have been
> having with alpha blending.

Right. I think of this sort of thing as more of "compositing", instead of just drawing z-buffered polygons. You have to control your scene much more carefully.