Subject: Re: texture map irregularities OR pimento problems
Posted by Rick Towler on Fri, 22 Jun 2001 20:35:00 GMT
View Forum Message <> Reply to Message

"Karl Schultz" <kschultz@researchsystems.com> wrote in message news:9gvt3olkk1@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? 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. 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.)

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.

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.

Thanks.

-Rick

> Now drawing these transparent
> polygons also sets the Z buffer in this part of the screen. So, as we
draw
> the rest of
> the green orb, the bottom or -Z part, the Z buffer test fails and these
> polys do not
> draw. The part of the screen where the transparent polys are does not
> change, and
> you end up seeing background and red orb where you instead
> wanted to see the inside of the green orb.
>
> In the second case:
>
> - The transparent part of the green orb is drawn last, because you moved
it
> from
> +z to -z, in terms of the orb coords. (Assuming that you've spun it
around
> 180 in xobjview)
> So, the side opposite the transparent "hole" draws first since it is in
the
> +z
> part of the orb. This parts makes it onto the screen and then the
> transparent polygons are drawn on top, letting you see the inside.
>
> Hope this helps. The general point is that you really have to be very
> careful in
> layering primitives when drawing with alpha. The layering may need to
take
> into account the viewing angle and the geometry of the primitives.
>
> For example, in this case, you may have to rotate the orb so that it
always
> draws
> back-to-front and move the transparent part so that it is always "in
front",
> with
> respect to the viewing direction.
>

> Karl
>
> "Rick Towler" <rtowler@u.washington.edu> wrote in message

> news:9gtvnl$q08$1@nntp6.u.washington.edu...
>> I have been experimenting using the alpha channel to represent
confidence
> in
>> a data set. I produce a polygon object representing the data and then
>> texture map the polygon accordingly. But, I have run into an issue that
I
>> can't resolve.
>>
>> I have attached a simple example (requires at least IDL 5 something and
24
>> bit display). I create 2 orb objects, one green (the olive) and one red
>> (the pimento).
>>
>> At first I place the olive in front of the pimento, then I make the near
>> surface of the olive transparent. Rendering in hardware for the most
part
>> works except between the last opaque vertices and the first transparent
>> vertices I get a white (background color) ring where you can actually
see
>> thru the entire olive to the pimento behind it. If your imagination is
>> failing I'll remind you that we are just supposed to be looking into the
>> inside of the olive, we shouldn't actually see the pimento thru the
olive.
>> Rendering in software, things get even more wacky. I don't see the
inside
>> of my olive at all. Xray vision I guess. What is interesting to note
is
>> that this ring I get when rendering in hardware renders just like the
all
> of
>> the transparent vertices in software.
>>
>>
>> I then flip the pimento and the hole in my olive. The pimento now is in
>> front of the olive and the "hole" is created on the back side of the
green
>> orb. Now when I rotate the olive around so I can see in my hole, things
>> look pretty good. The astute observer will notice that the ring is
still
>> there (you will see a little red pimple if viewed from the right angle)
> but
>> otherwise things look good. In software mode, things seem to work.
>>
>>
>> What is going on in the first case? Why does software rendering fail to
>> render the inside of the olive? What am I missing?
>>

>> I would be happy to provide images upon request.
>>
>> -Rick Towler
>
>
>