
Subject: texture map irregularities OR pimento problems
Posted by [Rick Towler](#) on Thu, 21 Jun 2001 23:29:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

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

;pimento example

```
container = obj_new('IDL_Container')
```

```
;create a rainbow texture map with varying alpha  
palette = obj_new('IDLgrPalette')
```

```
palette -> LoadCT, 13
container->add, palette
```

```
imagedata = bytarr(4, 256, 256, /nozero)
for a = 0, 255 do begin
  for nc=0,255 do begin
    imagedata[0:2,a,nc] = palette -> GetRGB(nc)
    imagedata[3,a,nc] = a
  endfor
endfor
texmap = obj_new('IDLgrImage', imagedata, interleave=0, $
  blend_function=[3,4], /interpolate)
container-> add, texmap
```

```
;a model to stick things in
model = obj_new('idlgrmodel')
container -> add, model
```

```
;some reference orbs
pimento = obj_new('Orb', pos=[0,0,-3.0], radius=1.0, color=[255,0,0],
density=2.0)
container-> add, pimento
```

```
;something to texmap
olive = obj_new('Orb', pos=[0,0,0], radius=2.0, color=[255,255,255],
density=2.0)
container-> add, olive
```

```
;add atoms from back to front
model -> add, [pimento, olive]
```

```
;get the olive's vertex data
olive -> getproperty, data=vtex
```

```
;calculate an array of latitudes for each vertex point
vtex_dims = size(vtex, /dimensions)
latitude = fltarr(vtex_dims[1])
nf = sqrt(total(vtex^2,1))
for n = 0, vtex_dims[1] - 1 do begin
  vtex_norm = vtex[* ,n] / nf[n]
  latitude[n] = acos(vtex_norm[2] / sqrt(total(vtex_norm^2))) * !RADEG
endfor
```

```

;set the texture coordinates
;create a "bowl" by making the positive z verticies with latitudes
;less than 45 deg transparent. The result is an orb with a hole in
;the side.
texcoords = fltarr(2,vtex_dims[1], /nozero)
texcoords[1,*] = .65 ;pick a nice green color for our olive
texcoords[0,*] = 0.99
;set the opacities to ~0.0 at lats lt 45.
i = where(latitude lt 45.)
texcoords[0,i] = 0.01

;set some properties
;note that you have to set reject=0 since by default orbs set it to 1
olive -> setproperty, texture_map=texmap, texture_coord=texcoords, $
    /texture_interp, /zero_opacity_skip, reject=0

;take a look at what we got
xobjview, model, /block

;now flip this whole deal in the z direction

;move our pimento
pimento -> setproperty, pos=[0,0,3.0]

;cut the top off of the other side of the bowl where
;latitude is gt 180. - 45.
texcoords[0,*] = 0.99
i = where(latitude gt 180. - 45.)
texcoords[0,i] = 0.01

olive -> setproperty, texture_coord=texcoords

xobjview, model, /block

;cleanup
obj_destroy, container

end

```
