
Subject: Re: Polygon Problems
Posted by [ronn](#) on Wed, 29 Mar 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Struan,

I agree that what you describe is a problem. I pulled something together that is included below that at least solves part of the line problem. What you have to do is to create a polyline object in another model, but offset the model by a small amount in the z direction. However, the polyline object suffers from the same problem as the surface object. They just aren't set up for cases with more faces than vertices. This is really obvious when you run the example below.

I looked at the teapot demo source code, it is part of the normal distribution. Look under `rsi/idl53/examples/demo/demosrc`. It appears that the way RSI solved this with the teapot is to add points in the center of a polygon, In your octahedron example you would have to add point in the center of the face and thereby artificially create more triangular regions.

I hope this helps a little.

-Ronn

--

Ronn Kling

Ronn Kling Consulting

Application Development with IDL book at : <http://www.rkling.com>

UPDATED FOR IDL 5.3!

email: ronn@rkling.com

Shareware and Freeware at: <http://www.rkling.com/>

----- CUT HERE -----

```
function octahedron
```

```
rt2 = sqrt(2.0)
```

```
vertex_array = [ $  
  [0,0,rt2], $  
  [rt2,0,0], $  
  [0,rt2,0], $  
  [-rt2,0,0], $  
  [0,-rt2,0], $  
  [0,0,-rt2] $
```

```

]

poly_array = [ $
  [3,0,1,2], $
  [3,2,3,0], $
  [3,3,4,0], $
  [3,4,1,0], $
  [3,5,1,4], $
  [3,1,5,2], $
  [3,5,3,2], $
  [3,5,4,3] $
]

s = OBJ_NEW("IDLgrPolygon", data=vertex_array, $
  SHADING=0, $
  POLY=poly_array, COLOR=[200,200,200])

return,s
end

;{{{::{{{::{{{::{{{::{{{::{{{::{{{::{{{::{{{::{{{::}}}

pro octahedron_test

myview = [-0.5,-0.5,1,1]
; Create view.
oView = OBJ_NEW('idlgrview', PROJECTION=2, EYE=3, ZCLIP=[1.5,-1.5],$
  VIEWPLANE_RECT=myview, COLOR=[0,0,0])

; Create model for the geometric objects.
;
oModelTop = OBJ_NEW('IDLgrModel')
oModelSurface = OBJ_NEW('IDLgrModel')

oSurface = octahedron()
oModelTop->add, oModelSurface
oModelSurface->add,oSurface

;giving this a slight offset makes the lines visible.
oModelOffset = OBJ_NEW('IDLgrModel')
oModelOffset->translate, 0, 0, 0.005 ;Offset Z to make visible
oModelEdges = OBJ_NEW('IDLgrModel')
oModelOffset->add, oModelEdges
oSurface->GetProperty, POLY=pmesh

oLine = OBJ_NEW('IDLgrPolyline', SHARE_DATA=oSurface, POLY=pmesh, $
  COLOR=[255,255,255],thick=1)

```

```

oModelEdges->add, oLine           ;Add the edging data
oModelTop->add, oModelOffset

;make object smaller
scs = 0.3
oModelTop->Scale, scs, scs, scs

; Create the vertex colors and make the 3-D objects
; to have these color.
;
vc = BYTARR(3, 8, /NOZERO)
sat = 1.0
val = 1.0
ic = 0
for i = 45, 360, 45 do begin
    angle = i
    Color_convert, angle, sat, val, red, green, blue, /HSV_RGB
    vc(0, ic) = red
    vc(1, ic) = green
    vc(2, ic) = blue
    ic = ic + 1
endfor

oSurface -> SetProperty, VERT_COLORS=vc

; Create a light
;
oLight3 = OBJ_NEW('IDLgrLight', LOCATION=[0,0,5], TYPE=0, $
    COLOR=[255,255,255])
oModelTop->Add, oLight3

; Place the model in the view.
oView->Add, oModelTop
;make a window
oWindow = OBJ_NEW('IDLgrWindow', dim=[400,400])
;rotate the object to see the problem
;Note that the lines have the same problem as the color.
;They only show up along some edges.
for i=0,360,5 do begin
    oModelTop->rotate,[1,0,0],5
    oWindow->Draw, oView
    wait,.1
endfor

for i=0,360,5 do begin
    oModelTop->rotate,[0,1,0],5
    oWindow->Draw, oView

```

```
wait,.1  
endfor
```

```
return  
end
```

Sent via Deja.com <http://www.deja.com/>
Before you buy.

Subject: Re: Polygon Problems
Posted by [Struan Gray](#) on Thu, 30 Mar 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ronn Kling, ronn@rkl Kling.com writes:

```
> I agree that what you describe is a problem. I pulled  
> something together that is included below that at least  
> solves part of the line problem. What you have to do is  
> to create a polyline object in another model, but offset  
> the model by a small amount in the z direction.
```

This works until the user rotates the model. I've also tried expanding the polyline object slightly, which works until you construct a polyhedron with no well-defined centre.

I'm working on adding a cylindrical linestyle to the polygon edges, which will sidestep the problem at the expense of longer calculation times.

As an aside: I would love to be able to modify the Draw method for various graphics primitives. Not by actually modifying it, but by augmenting it in the same way that Init and Cleanup methods can be built upon in sub-classes. In the above case, this would allow the Draw method to work out where 'z' was in the current view and displace the polygon that way before handing off to the usual Draw method. It would also allow you to easily implement useful 'on glass'-type objects for labelling plots and diagrams.

```
> However, the polyline object suffers from the same  
> problem as the surface object. They just aren't set  
> up for cases with more faces than vertices. This is  
> really obvious when you run the example below.
```

Aarrgh!

- > It appears that the way RSI solved this with the teapot
- > is to add points in the center of a polygon, In your
- > octahedron example you would have to add point in the center
- > of the face and thereby artificially create more triangular
- > regions.

This is another way of doing it. Because I plan to add methods to my polyhedra objects which will work on a polygon-by-polygon basis (and not a vertex-by-vertex one) I prefer to duplicate vertices: the octagon can be plotted if you just add two more vertices identical to any of the existing ones.

It's probably worth pointing out that with any shape which is predictable you can always come up with a hack that works well enough.

With objects you can even confine that hack to a superclass which can be altered if and when RSI address the problem. The difficulty is that none of the hacks are general, so when I stop playing with nice cubic lattices and head into the wild badlands of monoclinic crystals or amorphous materials, I can't be sure that my programs will work.

- > I hope this helps a little.

It puts my mind at rest that I'm not the stupid one. Thanks for looking into it.

As far as I can tell, neither of these effects are caused by any limitation in OpenGL. I'll submit a request to RSI for a composite style and a `polygon_normals` keyword and see what happens.

Struan
