Subject: Re: Understanding IDLanROI
Posted by KRDean on Tue, 19 Oct 2010 20:19:09 GMT
View Forum Message <> Reply to Message

On Oct 19, 11:38 am, David Fanning <n...@dfanning.com> wrote:
> kBob writes:
>> To answer my own question ...
>
>> Yes, IDLanROI is NOT the IDL routine to perform this check.
>
>> After pondering on this for a couple of weeks, I figure what I needed
>> to do.
>
>> IDLgrROI is the prefer IDL routine to use.
>
>> It is a tessellation thing, especially when dealing with State and
>> Country Shapefiles that may contain gaps. You may get away with very
>> simple polygons using IDLanROI when dealing with a handful of
>> vertices, but IDLgrROI and IDLgrROIGroup are the routines you should
>> be using along with incorporating IDLgrTessellator in the code.
>
> Can you elaborate a bit on why this "is a tessellation thing"?
> I don't really understand what that means. And, supposing
> I did understand what it means, what would it have to do
> with choosing IDLgrROI over IDLanROI. Isn't IDLgrROI a
> subclass of IDLanROI?
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

In Object Graphics, polygons that are overlapping, disjointed, or self-intersecting are not handled very well by routines like IDLgrPolygon, thus IDLgrTessellator is used to convert polygons that are more suitable for IDLgrPolygon. From the IDL documentation, it converts the concave polygon into a convext polygon. Tessellation is the process of tiling a plane so there are no overlaps or gaps.

When I tried to use the US shapefile in IDLanROI (and IDLgrROI) there was a line running across the United States from the Northeast to Hawaii. Using IDLanROI, some States above this line were considered exterior polygons. See my original post. :(After tessellating, the

line from the Northeast to Hawaii is gone, so all the States that fall in the United States polygon are considered interior polygons. :)

Your right, IDLgrROI inherits IDLanROI, thus allowing IDLgrROI to use IDLanROI classes, like ->ContainsPoints().

IDLgrTesselator is an IDLgr object class, thus I recommend the use of the IDLgr classes IDLgrROI and IDLgrROIGroup. I haven't tried, but I may at a later date to see if just the IDLanROI and IDLanROIGroup will give the desired results.

Check out the example below that I modified from IDL Help, it tries to draw an intersecting polygon in Object Graphics, with and without tessellation.

```
PRO TessAux, DoTess = DoTess

  x = [0,1,0,1]
  y = [0,0,1,1]

  colors = [[0,255,0],[0,255,0],[0,64,0],[0,64,0]]

  IF ( KEYWORD_SET( DoTess ) ) THEN BEGIN

    oTess = OBJ_NEW('IDLgrTessellator')
    colors = [[0,255,0],[0,255,0],[0,64,0],[0,64,0]]
    oTess -> AddPolygon, x, y, AUXDATA=colors
    result = oTess -> Tessellate(v, c, AUXDATA=aux)

    oPoly = OBJ_NEW('IDLgrPolygon', v, POLYGONS=c, VERT_COLORS=aux,
SHADING=1)

  ENDIF ELSE BEGIN

    oPoly = OBJ_NEW('IDLgrPolygon', x, y )

  ENDELSE

  XOBJVIEW, oPoly, /BLOCK
  IF ( OBJ_VALID( oTess ) ) THEN OBJ_DESTROY, oTess
  IF ( OBJ_VALID( oPoly ) ) THEN OBJ_DESTROY, oPoly

END
;-------------------------------------------------------
```

Kelly Dean
Milliken, CO

```
;+
;
;
;-----------------------------------------------------
PRO TessAux, DoTess = DoTess

  x = [0,1,0,1]
  y = [0,0,1,1]

  IF ( KEYWORD_SET( DoTess ) ) THEN BEGIN

    oTess = OBJ_NEW('IDLgrTessellator')
    colors = [[0,255,0],[0,255,0],[0,64,0],[0,64,0]]
    oTess -> AddPolygon, x, y, AUXDATA=colors
    result = oTess -> Tessellate(v, c, AUXDATA=aux)

    oPoly = OBJ_NEW('IDLgrPolygon', v, POLYGONS=c, VERT_COLORS=aux,
SHADING=1)

  ENDIF ELSE BEGIN

    oPoly = OBJ_NEW('IDLgrPolygon', x, y )

  ENDELSE

  XOBJVIEW, oPoly, /BLOCK
  IF ( OBJ_VALID( oTess ) ) THEN OBJ_DESTROY, oTess
  IF ( OBJ_VALID( oPoly ) ) THEN OBJ_DESTROY, oPoly

END
```