Subject: Re: Shapefile 'parts' woes
Posted by Gaurav on Thu, 20 Mar 2008 06:38:45 GMT
View Forum Message <> Reply to Message

Dear Dr. Fanning,

I was able to find out the simple working of "cuts" after reading the Globe display program "d_map.pro" from the selection of demo programs and your program too discusses as much. Maybe it is my shortcoming that I fail to implement it in my program but here is what I want:

The shapefiles that I have as input are those obtained by Raster to Vector conversion program and have jagged edges. I want to smoothen the edges by replacing each of the points in the polygon by an average value of the previous point, the current point and the next point. In my simple and ideal shapefile in which each polygon had one single part, I could do so simply using the following code:

```
myshape=OBJ_NEW('IDLffShape', 'G:\r2v\inputshp.shp')
mynewshape=OBJ_NEW('IDLffShape','G:\r2v\outshp.shp' , /UPDATE,
ENTITY_TYPE=5)
myshape->IDLffShape::GetProperty, N_ENTITIES=num_ent

FOR x=0, (num_ent-1) DO BEGIN
     ent=myshape->IDLffShape::GetEntity(x)
     numVertices = n_elements((*ent.vertices)[0,*])
     newVertices = ptr_new(dblarr(2,numVertices))
  (*newVertices)[0,0] = (*ent.vertices)[0,0];The first vertex is
simply copied
  (*newVertices)[1,0] = (*ent.vertices)[1,0]
     for y = 1, numVertices-2 do begin
 (*newVertices)[0,y] = ((*ent.vertices)[0,y-1] + (*ent.vertices)[0,y]
+ (*ent.vertices)$ [0,y+1])/3.0
 (*newVertices)[1,y] = ((*ent.vertices)[1,y-1] + (*ent.vertices)[1,y]
+ (*ent.vertices)$ [1,y+1])/3.0
    endfor
          ;The last vertex is the same as the first
  (*newVertices)[0,numVertices -1] = (*ent.vertices)[0,0]
  (*newVertices)[1,numVertices -1] = (*ent.vertices)[1,0]
Endfor ;for each of the polygon entities
```

But in any ordinary shapefile many of the entities consist of many parts and I am having trouble smoothening out each of these parts separately. I am very much at sea as to how to go about it. What I tried is following, perhaps you could point out where I am going wrong:

```
FOR x=0, (num_ent-1) DO BEGIN
      ent=myshape->IDLffShape::GetEntity(x)
      numVertices = n_elements((*ent.vertices)[0,*])
      newVertices = ptr_new(dblarr(2,numVertices)); for the vertices
in old and new polygons are same in number
      if(ptr_valid(ent.parts) ne 0)then begin
      cuts = [*ent.parts, ent.n_vertices]
        for j=0, ent.n_parts-1 do begin
          ;the first vertex of each cut is copied as it is:
  ((*newVertices)[0,cuts[j]]) = ((*ent.vertices)[0,cuts[j]])
   ((*newVertices)[1,cuts[j]]) = ((*ent.vertices)[1,cuts[j]])
  ;find the number of remaining vertices in each cut leaving the first
and the last
   numb = n_elements((*ent.Vertices)[0,cuts[j]+1:cuts[j+1]-2])
  for y = 1, numb-2 do begin
    ;calculating the average values of the vertex of the "cuts"
    ((*newVertices)[0,cuts[j]+y]) = (((*ent.vertices)[0,cuts[j]+y-1])
+ ((*ent.vertices)[0,cuts[j]+y]) + ((*ent.vertices)[0,cuts[j]+y+1]))/3
    ((*newVertices)[1,cuts[j]+y]) = (((*ent.vertices)[1,cuts[j]+y-1])
+ ((*ent.vertices)[1,cuts[j]+y]) + ((*ent.vertices)[1,cuts[j]+y+1]))/3
          Endfor;for the vertices
   ;the value of last vertex of each "cut" is set to be same as the
first vertex
    ((*newVertices)[0,cuts[j+1]-1]) = ((*ent.vertices)[0,cuts[j]])
    ((*newVertices)[1,cuts[j+1]-1]) = ((*ent.vertices)[1,cuts[j]])
 endfor;for the "cuts"
   endif
endfor; for the entities
```

Regards
Gaurav
