Subject: Re: Shapefile 'parts' woes
Posted by David Fanning on Wed, 19 Mar 2008 12:33:46 GMT
View Forum Message <> Reply to Message

Gaurav writes:

> I have a shapefile with many polygons. Most of these polygons have
> holes and hence these have multiple parts. I wish to create a new
> shapefile by modifying points in some 'parts' of the polygons. How do
> I acess the points of a particular 'part' of a polygon? I went through
> the two examples: the one in help and the other in demo programs-
> showing the globe, but could not deduce how to solve my problem.

You might be able to deduce how to access these parts, which I
call "cuts" in DRAWSTATES, a program to try shapefiles:

http://www.dfanning.com/programs/drawstates.pro

> Would anyone be kind enough to bail me out of the rut or point me to
> some link where I might be shone upon with the divine light?

Don't look for divine light from me. I remember it as
a confusing slog, with more perspiration than inspiration.
What you see in that program is the sum total of my knowledge
of shapefiles, if not more.

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.")

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

Subject: Re: Shapefile 'parts' woes
Posted by David Fanning on Thu, 20 Mar 2008 14:01:13 GMT
View Forum Message <> Reply to Message

Gaurav writes:

> 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:

Uuhhg! Wandering through a sea of shapefile code is *not*
on my list of fun things to do today. :-)

I do have a suggestion, however. I have a program named
ARCSAMPLE that resamples a closed curve, such as your
polygons. One of the effects it has is to smooth the
curve. If you can read the polygon and apply ArcSample
to it and put it back, it may be just what you are
looking for. But the pointer code is just too tedious
for me today, sorry.

  http://www.dfanning.com/programs/arcsample.pro

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.")

Subject: Re: Shapefile 'parts' woes
Posted by cgpadwick on Thu, 27 Mar 2008 13:38:45 GMT
View Forum Message <> Reply to Message

Here's a routine that might help you understand how to do what you
want. IDLffShape has a nasty gotcha that can trip people up : it only
defines the parts pointer if the polygon contains multiple contours.
So I usually handle that by checking the n_parts and if it is zero
then I set parts = [0]. That way I don't have to have a bunch of
nasty if (nparts eq 0) logic in my code.

```
pro printPoly, verts, partsPtr

  nParts = N_Elements(partsPtr)
  nVerts = N_Elements(verts[0,*])

  for ndx=0L, nParts-1L do begin
    sIdx = partsPtr[ndx]
    eIdx = ( (ndx+1) gt nParts-1L) ? nVerts - 1L : partsPtr[ndx+1L]-1L

    Print, '=============================='
    Print, 'NDX: ', String(StrTrim(ndx, 2))
    Print, 'VERTICES: ', verts[*,sIdx:eIdx]

  endfor

end

pro testShape

  filename = Dialog_Pickfile(TITLE='Pick the shapefile...')

  oShape = Obj_New('IDLffShape', filename)
  oShape->GetProperty, N_ENTITIES=nEnt

  window, 1

  for ndx=0L, nEnt-1L do begin
```

```
    ent = oShape->GetEntity(ndx)

    verts = *ent.vertices
    parts = *ent.parts
    nParts = ent.n_parts
    if (nParts eq 0) then parts = [0]

    printPoly, verts, parts
    oShape->DestroyEntity, ent


  endfor

  Obj_Destroy, oShape

end
```

Subject: Re: Shapefile 'parts' woes
Posted by Gaurav on Fri, 28 Mar 2008 08:16:35 GMT
View Forum Message <> Reply to Message

Thanks indeed for the attention. Creation of shapefiles is a very nifty utility that IDL provides and a fair understanding of it saves a lot of work. I was able to tide over my current problem by trial and error. Apparently, I was not calculating the number of points that I needed to modify (stored in the variable 'numb'), properly.

The output of the code is very proper and it beautifully removes the jagged, staircase effect in the vector files that are very common when one converts a raster image to vector.

Thanks indeed.
Gaurav
