
Subject: Re: ENVI batch ROI

Posted by [Peter Scarth](#) on Tue, 25 Jun 2002 05:36:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi John,

I can now see what the problem is - ENVI_DEFINE_ROI does not test for multipart polygons. You have to work this out for yourself by adding in the following lines. There may be a simpler way, but this method seems to work with complex vegetation multipart polygons.

This raises an interesting point - how can you detect multipart lines in a EVF?

Cheers,
Peter

```
> FOR i=0,num-1 DO BEGIN
>
> ; read the record
> ;
> vec = ENVI_EVF_READ_RECORD(evf_id, i)
> ; convert from map coordinates to file coordinates, again
> ; this assumes that the map proj for the vector data is the
> ; same as that of the image
> ;
> ENVI_MAP_CONVERT, xpts, ypts, vec(0,*), vec(1,*), h_map=h_map,
> /to_file
> ; make an ROI in memory
> roi_id = ENVI_CREATE_ROI(ns=ns, nl=nl)
>
```

```
; -----Start new ENVI_DEFINE_ROI section -----
```

```
num_vert=n_elements(xpts)-1 ; How many vertices in the record?
```

```
sv=0 ; Set the starting vertex index to 0
```

```
WHILE (sv lt num_vert-1) DO BEGIN ; Loop until all polygons processed
```

```
 ; Find the place where the polygon closes
```

```
idx=where((xpts[sv] eq xpts[sv+1:num_vert])$
  AND (ypts[sv] eq ypts[sv+1:num_vert]),match)
```

```
IF match eq 0 THEN BREAK ; Exit if it is not a polygon
```

```

; If there is closure start processing
ed=sv+1+idx[0] ; Find the array index of the endpoint vertex

; Write the polygon
ENVI_DEFINE_ROI, roi_id, xpts=REFORM(xpts[sv:ed], /over),$
  ypts=REFORM(ypts[sv:ed], /over), /polygon

sv=ed+1 ; Set the new start vertex index

```

```
ENDWHILE
```

```
; -----End new ENVI_DEFINE_ROI section -----
```

```

>
> ; extract the image data associated w/the ROI
> ; XXX CHANGE J=0,8 below to represent the appropriate number of
> bands
> FOR j=0,0 DO BEGIN
> print, roi_id
> roi_data = ENVI_GET_ROI_DATA( roi_id, fid=fid, pos = [j])
> printf, lun, strtrim(i,2), total(roi_data), MEAN(roi_data)
> ENDFOR
> ENDFOR

```

"John" <jdvona@yahoo.com> wrote in message

news:844e4579.0206050713.67b2d0df@posting.google.com...

```

> I am not an IDL programmer and need help with a routine I developed.
> The routine batch processes a GIS vector file with multiple polygons
> by creating a single ROI for each individual polygon, which then
> outputs the mean, total and standard deviation for the pixels within
> the ROI.

```

```

>
> The program has worked well for very simple polygons with few
> verticies, but I found a bug, which occurs with fairly complex (eg
> real-world) vegetation polygons. After the routine completes if I
> look at the ROI's in ENVI they have missing parts of the polygons,
> weird bisections, and filled in holes that were not part of the
> original polygon. The code which creates the ROI is below:
> I think 3.5 has this capability but I am still running 3.2 (long
> story).
> Thanks for the help.

```

```

>
> FOR i=0,num-1 DO BEGIN
>
> ; read the record

```

```
> ;
> vec = ENVI_EVF_READ_RECORD(evf_id, i)
> ; convert from map coordinates to file coordinates, again
> ; this assumes that the map proj for the vector data is the
> ; same as that of the image
> ;
> ENVI_MAP_CONVERT, xpts, ypts, vec(0,*), vec(1,*), h_map=h_map,
> /to_file
> ; make an ROI in memory
> roi_id = ENVI_CREATE_ROI(ns=ns, nl=nl)
>
> ; add the vector record as an ROI
> ENVI_DEFINE_ROI, roi_id, xpts=REFORM(xpts, /over), $
>   ypts=REFORM(ypts, /over), /polygon
>
> ; extract the image data associated w/the ROI
> ; XXX CHANGE J=0,8 below to represent the appropriate number of
> bands
> FOR j=0,0 DO BEGIN
>   print, roi_id
>   roi_data = ENVI_GET_ROI_DATA( roi_id, fid=fid, pos = [j])
>   printf, lun, strtrim(i,2), total(roi_data), MEAN(roi_data)
>   ENDFOR
> ENDFOR
```
