
Subject: Fed Up with IDL: image subset by shape or EVF file
Posted by [bulrushmower](#) on Mon, 23 Jun 2008 02:13:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

I have seen most of the postings about the topic and tried all of the suggestions to subset images by a shape file. I tired off and started to wonder IDL is so bad... My problem is:

1. Following two codes, one use EVF the other one shape file, works fine to create a ROI, but some parts of filled ROI is always missing. E.g., only Alaska was generated as a ROI when I used USA border shape file.
2. How can I subset image in the next step when I am able to create a ROI?
3. How can I display a vector file using IVECTOR?

Appreciate any help!

METHOD1 using a EVF file

=====

```
pro fill_polygon
compile_opt idl2
```

```
;open and query a file
envi_open_file,'testdata', r_fid=fid, /no_realize
```

```
envi_file_query, fid, dims=dims, nb=nb, ns=ns, nl=nl, $
fname=fname, data_type=data_type, interleave=interleave
```

```
.....
;.....
```

```
;open a EVF file
;this part is used if a EVF is used instead of shape file
evf_id = envi_evf_open(fPath_shp+Shapefile_.evf)
```

```
; Get the vector information
envi_evf_info, evf_id, num_recs=num_recs, $
data_type=data_type, projection=projection, $
layer_name=layer_name
;
;Print information about each record
;print, 'Number of Records: ',num_recs, data_type, projection
for i=0,num_recs-1 do begin          ; Loop through each
record
    record = envi_evf_read_record(evf_id, i) ;get the coord of each
nodes of the current polygon
    ;convert the projected coordinates of the nodes to a line/column
coordinate
```

```
envi_convert_file_coordinates,fid,X_polygon_pixel,Y_polygon_pixel,record[0,*],record[1,*]
```

```

;Get the 1D cell indices of the cells lying under the current
polygon.
    polygonIndices =
polyfillv(round(X_polygon_pixel),round(Y_polygon_pixel), ns, nl)

;print, 'Number of nodes in Record ' + $
;    strtrim(i+1,2) + ': ', n_elements(record[0,*])

endfor

; Close the EVF file
envi_evf_close, evf_id

Image = array_indices([ns, nl], polygonIndices, /dimensions)

;create a new ROI to confirm the polyfill worked
roi_id = envi_create_roi(color=4, name = 'test roi', ns=ns, nl=nl)
envi_define_roi, roi_id, /point, xpts=Image[0, *], ypts=Image[1, *]
roi_ids = envi_get_roi_ids()
envi_save_rois, fPath_out+'test.roi', roi_ids

; IVECTOR, Image[0, *], Image[1, *], xf, yf, AUTO_COLOR=1,
RGB_TABLE=39, SCALE ISOTROPIC=1

```

METHOD2 using a shape file

```
;open a shapefile
myshape=OBJ NEW('IDLffShape', fPath shp+'Shapefile.shp')
```

```
; Get the number of entities so we can parse through them.
myshape->GetProperty, N_ENTITIES=num ent
```

```
; Read all the entities
FOR i=0, (num_ent-1) DO BEGIN
    ;Read the entity i
    ent = myshape->GetEntity(i)
```

```
FOR i=0, (num_ent-1) DO BEGIN
```

```
ent = myshape->GetEntity(i)
```

```
xmap=(*ent.vertices)[0, *]  
ymap=(*ent.vertices)[1, *]
```

```
;get the file coordinates
envi_convert file coordinates, fid, xf, yf, xmap, ymap
```

```
;print, ent
;Clean-up of pointers
```

```
myshape->DestroyEntity, ent
ENDFOR
; Close the Shapefile.
OBJ_DESTROY, myshape

;fill the polygon
filled_poly = polyfillv(xf, yf, ns, nl)

ind = array_indices([ns, nl], filled_poly, /dimensions)

;create a new ROI to confirm the polyfill worked
roi_id = envi_create_roi(color=4, name = 'test1 roi', ns=ns, nl=nl)
envi_define_roi, roi_id, /point, xpts=ind[0, *], ypts=ind[1, *]
roi_ids = envi_get_roi_ids()
envi_save_rois, fPath_out+'test1.roi', roi_ids

;Display the vector file
; IVECTOR, ns, nl, xf, yf, AUTO_COLOR=1, RGB_TABLE=39,
SCALE_ISOTROPIC=1

Print, 'A successful Run!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'

end
```
