Subject: Objects in ENVI ROI files
Posted by Bradley Wallet on Mon, 22 May 2006 12:22:07 GMT
View Forum Message <> Reply to Message

Does anyone know how to extract polygon information from an ENVI ROI file? I have ROI files that contain multiple objects, each as its own polygon. I have an application I wrote, and it needs to extract each of these objects separately. Currently, I do a region growing algorithm, but I want something cleaner and faster. Tech support told me the following:

> You can do this in a round about way by exporting the ROI to an EVF (each point
> as separate record) and then exporting the evfs back to ROIs (each record of EVF to
> new ROI). This can be used to 'separate' the different entities in one ROI into multiple
> ROIs.

Surely there is a cleaner way to do this that would be transparent to the user. Any suggestions? If I have to read the ROI file myself, that's fine, but I haven't found any documentation on the format.

Brad

-- ------------------------------------
Brad Wallet
President and Chief Scientist
Automated Decisions, LLC
consulting@automateddecisions.com

Subject: Re: Objects in ENVI ROI files
Posted by zhour.najoui on Thu, 16 May 2013 12:23:28 GMT
View Forum Message <> Reply to Message

Mr. Bradley
I am facing the same problem that you were confronted six years ago.
My goal is to qualify the training parcels used in classification, I try to study each polygon in my ROIs file.
Have you found the soltion to it using the ENVI/IDL routines?

Best Regardes

Subject: Re: Objects in ENVI ROI files
Posted by Josh Sixsmith on Tue, 21 May 2013 05:08:47 GMT
View Forum Message <> Reply to Message

Considering that the original post was back in 2006, it has probably already been solved.

Anyway, here is one method that should achieve what you're after.

Rather than use region_grow to find all of the values. Use a dummy array of the same x/y dimensions as to what the ROI is based off, and label it. This will be similar (if not the same as) to the region grow method, without the iterations.

You can use the roi addresses to index the dummy array and set the value to one (all other pixels should be zero).
Then use label_region. This will give each of your roi polygons a unique identifier.
The next step would be to find all the unique labels, using a combination of uniq and sort.
Use the histogram function and reverse indices. This can then be used to index the original array and do whatever you want, such as assign a new value, or calculate some stats.

The bins of interest are defined by the unique labels, which are also sorted. We know that the background is zero so ignore it when looping over the bins.

Eg:

```
dummy = bytarr(samples,lines)
dummy[roi_addr] = 1
label_arr = label_region(dummy)
labels = label_arr[uniq(label_arr, sort(label_arr))]
hist = histogram(label_arr, min=0, max=max(labels), reverse_indices=ri)

for i=1,n_elements(labels)-1 do begin
   if hist[labels[i]] eq 0 then continue
   ;retrieve the data from the original array
   polygon = orig_data[ri[ri[labels[i]]:ri[labels[i]+1]-1]]
   ;do something
endfor
```

That should get the individual polygons within each ROI that you’re after.

Cheers
Josh

Subject: Re: Objects in ENVI ROI files
Posted by zhour.najoui on Wed, 22 May 2013 13:40:08 GMT

View Forum Message <> Reply to Message

I tried the code, but the number of polygons that I got is different from the number of polygons in the file .roi (I Verife using ENVI ==> basic tools ==> region of interest ==> ROI tools).
I thought to do this step on ENVI (ROI TO EVF) and I need to know how to get the number of polygons in each region ROI from my program.
I can get the number of points in each region = n_elements(roi_addr)
 but how to get the number of polygons in each region?
here is part of my code :

```
;---------------------------------
; Load the region_of_interest file
;--------------------------------
envi_restore_rois, region_of_interest      ; Restore the ground truth ROIs
if (fid eq -1) then return
roi_ids = envi_get_roi_ids(fid=fid, roi_names=roi_names)
class_ptr = lindgen(n_elements(roi_ids)) + 1
if (roi_ids[0] eq -1) then begin
  print, ' ROIs not found'
  return
endif

for i=0, n_elements(roi_ids)-1 do begin
  roi_addr = envi_get_roi(roi_ids[i], roi_name=name)
  ENVI_GET_ROI_INFORMATION,roi_ids, ns=ns, nl=nl, npts=npts

endfor
```

thanks

Subject: Re: Objects in ENVI ROI files
Posted by Josh Sixsmith on Fri, 24 May 2013 04:22:52 GMT

View Forum Message <> Reply to Message

Using a toy example, I have a single ROI file containing 3 groups; Group A has 5 polygons, Group B has 4 polygons and Group C has 3 polygons.

roi_ids = envi_get_roi_ids(fid=fid) ; should give me an array of length 3, indicating 3 distinct groups.

roi_addr_A = envi_get_roi(roi_ids[0]) ; will give me the pixels contained within Group A
roi_addr_B = envi_get_roi(roi_ids[1]) ; will give me the pixels contained within Group B
roi_addr_C = envi_get_roi(roi_ids[2]) ; will give me the pixels contained within Group C

I make an assumption here that each polygon within a single group shares no common border with another polygon. If two polygons within a group share a border then the label_region() function will not distinguish between the tow and lump them together. It makes no sense (to me) for polygons of the same group to share a border, so they should be merged. An alternative would be to put one of those polygons into a different group.

Anyway, we now have the addresses (indices) of all the pixels contained within Group A. If there are no borders are shared between the polygons contained within Group A, then the label_region() function should label the image with unique ids for each polygon, eg 1,2,3. A value of 0 will be the background.

labels = label_arr[uniq(label_arr, sort(label_arr))]

In this example, Group A has 5 polygons, the length of labels should be 6, taking into account the background value of 0.

```
num_polygons = n_elements(labels) -1
print, num_polygons
 5
```

The rest of the code i posted previously will then visit each polygon and retrieve the data contained within.

In your code example:

```
for i=0, n_elements(roi_ids)-1 do begin
   roi_addr = envi_get_roi(roi_ids[i], roi_name=name)
   ENVI_GET_ROI_INFORMATION,roi_ids, ns=ns, nl=nl, npts=npts
   dummy = bytarr(ns,nl)
   dummy[roi_addr] = 1
   label_arr = label_region(dummy)
   labels = label_arr[uniq(label_arr, sort(label_arr))]
   num_polygons = n_elements(labels) -1 ;ignoring background value of 0
   print, format='(%"Polygon %s contains %i polygons ")', name, num_polygons
endfor
```

I hope that clarifies things. If it still doesn't work, then you might have polygons that share a common border. If so, then consider putting one of those polygons into a different group.

Cheers
Josh

Subject: Re: Objects in ENVI ROI files
Posted by samsammurphy on Thu, 05 Mar 2015 18:02:46 GMT

View Forum Message <> Reply to Message

On Tuesday, May 21, 2013 at 2:08:47 AM UTC-3, Josh Sixsmith wrote:
> Considering that the original post was back in 2006, it has probably already been solved.
>
> Anyway, here is one method that should achieve what you're after.
>
> Rather than use region_grow to find all of the values. Use a dummy array of the same x/y dimensions as to what the ROI is based off, and label it. This will be similar (if not the same as) to the region grow method, without the iterations.
>
> You can use the roi addresses to index the dummy array and set the value to one (all other pixels should be zero).

> Then use label_region. This will give each of your roi polygons a unique identifier.
> The next step would be to find all the unique labels, using a combination of uniq and sort.
> Use the histogram function and reverse indices. This can then be used to index the original array and do whatever you want, such as assign a new value, or calculate some stats.
>
> The bins of interest are defined by the unique labels, which are also sorted. We know that the background is zero so ignore it when looping over the bins.
>
> Eg:

```
>  dummy = bytarr(samples,lines)
>  dummy[roi_addr] = 1
>  label_arr = label_region(dummy)
>  labels = label_arr[uniq(label_arr, sort(label_arr))]
>  hist = histogram(label_arr, min=0, max=max(labels), reverse_indices=ri)
>
>  for i=1,n_elements(labels)-1 do begin
>     if hist[labels[i]] eq 0 then continue
>     ;retrieve the data from the original array
>     polygon = orig_data[ri[ri[labels[i]]:ri[labels[i]+1]-1]]
>     ;do something
>  endfor
```

>
> That should get the individual polygons within each ROI that you're after.
>
> Cheers
> Josh

Thanks for posting this. It is what I was looking for - looks like its a subject that comes around every few years :)