Subject: Re: an envi question- extracting the pixel values of seveal points with lat-lon values
Posted by MarioIncandenza on Tue, 31 Oct 2006 17:06:19 GMT
View Forum Message <> Reply to Message

Jeff,

This problem should be ridiculously easy to solve in ENVI, but when I had to solve it, I had to do it the hard way. At bottom, the text of ENVI_POINT_PRO.PRO, which is a mini-library of routines for solving your point extraction problem.

How to use this:
1) Get your XY points into an IDL variable. At worst, just type then in at the command line like:
ENVI> XDATA = [n.nn , n.nn , n.nn....]
ENVI> YDATA = [n.nn , n.nn , n.nn...]
2) Save the text below as ENVI_POINT_PRO.PRO, just copy it into notepad or download it from ftp://ftp.nrlmry.navy.mil/pub/receive/hyer/idl/.
3) Compile it with
ENVI> .COMPILE <FULL PATH>ENVI_POINT_PRO.PRO
4) Pull data from one file, to see if it works:
4a) Load the file into ENVI and display it in Display #1
4b) from the command prompt, type:
ENVI> MYROI = ENVI_POINT_ROIGEN(0,XDATA,YDATA,NAME="My ROI", /GETDATA, DATA=MYROI_DATA)
ENVI> HELP, MYROI, MYROI_DATA
4c) MYROI should be a single number greater than 0, and MYROI_DATA should be an array of the file values for each of your points.
5) If your other image files have different areas and/or geographic extents, you will have to automate or repeat Step 4. If they are all the same, you can just do Steps 6-10:
6) Load as many files into ENVI as you can without crashing. If you do crash ENVI, just start back at Step 1 (not a big deal, really).
7) Get the file IDS:
ENVI> ENVI_LIST_FILEIDS, FIDS=MYFIDS
8) Hopefully, all of these IDS are files you are interested in. If not, modify Steps 9-10.
9) Create an array to hold your results:
ENVI> MY_ALLFILES_DATA = FLTARR(N_ELEMENTS(MYFIDS),N_ELEMENTS(XDATA))
10) Loop through loading in results:
ENVI> FOR i = 0,N_ELEMENTS(MYFIDS)-1 DO MY_ALLFILES_DATA[i,*]=ENVI_POINT_ROIMATCH(MYROI,MYFIDS[i])

An annoying and difficult solution to a very common problem.

Here's the program text:
; ENVI_POINT_PRO.PRO

```
;This is a set of routines for handling XY point data in ENVI. It
;provides the following functions:
; ENVI_get_ddbounds: given a display number or file ID, returns the
NSEW
; edges of the file (or Image window) in decimal degrees.
; ENVI_point_subsets(): given XY data in decimal degrees and a display
; number, returns the indices of points within the area of the file,
; (and optionally within the area of the Image window)
; ENVI_point_xyconvert(): given XY data in decimal degrees and a
display
; number, returns file coordinates (or image window coordinates)
; ENVI_point_roigen: given XY data in decimal degrees and a display ID,
; generates a point ROI for that display. Optionally, returns data
; from the file.
; ENVI_point_roimatch: given an ROI ID and a file ID, returns file
; data
; ENVI_point_ddplot: use PLOTS to put XY data directly onto Image or
; Scroll windows
function envi_get_ddbounds,dn,image=image
; ENVI_get_ddbounds: given a display number or file ID, returns the
NSEW
; edges of the file (or Image window) in decimal degrees.
if(n_elements(dn) eq 0) then dn=0; Display #1 default
ENVI_DISP_QUERY, dn,w1=dwin,fid=fid,x0=dx0_p,y0=dy0_p
 ,xds=dxdim_p,yds=dydim_p,nl=fydim_p,ns=fxdim_p,rebin=scrolls cale
map=envi_get_map_info(dn=dn); rather than fid=fid[0])
proj_u=map.proj; extract projection from structure $MAP
proj_dd=envi_proj_create(/GEOGRAPHIC);projection structure for lat/lon
data
; Establish file coordinate system
fwest_p=0; NW corner of file is 0,0
fnorth_p=0; NW corner of file is 0,0
feast_p=fwest_p+fxdim_p
fsouth_p=fnorth_p+fydim_p; file and display count up when all others
count down.
; Establish map coordinate system
fxcent_p=map.mc[0]; UTM coordinates defined from center
fycent_p=map.mc[1]; UTM coordinates  def. from center
fxcent_u=map.mc[2]
fycent_u=map.mc[3]
xpxsz_u=map.ps[0]
ypxsz_u=map.ps[1]
fwest_u=fxcent_u+(long(fwest_p-fxcent_p)*xpxsz_u)
feast_u=fxcent_u+(long(feast_p-fxcent_p)*xpxsz_u)
fnorth_u=fycent_u+(long(fycent_p-fnorth_p)*ypxsz_u)
fsouth_u=fycent_u+(long(fycent_p-fsouth_p)*ypxsz_u)
; convert map coordinate boundaries to DD
 envi_convert_projection_coordinates,[fwest_u,feast_u,fxcent_
```

```
u,fxcent_u],[fycent_u,fycent_u,fnorth_u,fsouth_u],proj_u,fco rnerx_dd,fcornery_dd,proj_dd
 fwest_dd=fcornerx_dd[0]
 feast_dd=fcornerx_dd[1]
 fnorth_dd=fcornery_dd[2]
 fsouth_dd=fcornery_dd[3]
nsew_out=[fnorth_dd,fsouth_dd,feast_dd,fwest_dd]; NSEW for FILE
if(keyword_set(image)) then begin; get bounds for Image window
dwest_p=dx0_p
deast_p=dwest_p+dxdim_p[0]
dnorth_p=dy0_p
dsouth_p=dnorth_p+dydim_p[0]
dwest_u=fxcent_u+(long(dwest_p-fxcent_p)*xpxsz_u)
deast_u=fxcent_u+(long(deast_p-fxcent_p)*xpxsz_u)
dnorth_u=fycent_u+(long(fycent_p-dnorth_p)*ypxsz_u)
dsouth_u=fycent_u+(long(fycent_p-dsouth_p)*ypxsz_u)
 envi_convert_projection_coordinates,[dwest_u,deast_u],[dnort
h_u,dsouth_u],proj_u,dcornerx_dd,dcornery_dd,proj_dd
 dnorth_dd=dcornery_dd[0]
 dsouth_dd=dcornery_dd[1]
 dwest_dd=dcornerx_dd[0]
 deast_dd=dcornerx_dd[1]
nsew_out=[dnorth_dd,dsouth_dd,deast_dd,dwest_dd]; NSEW for IMAGE WINDOW

return,nsew_out; return NSEW for IMAGE WINDOW
endif
return,nsew_out; return NSEW for FILE
end; end FUNCTION ENVI_GET_DDBOUNDS()

function envi_point_subsets,dn,xdata,ydata,image=image
; ENVI_point_subsets(): given XY data in decimal degrees and a display
; number, returns the indices of points within the area of the file,
if(keyword_set(image)) then $
  nsew=envi_get_ddbounds(dn,/image) else $
  nsew=envi_get_ddbounds(dn)
sub_x=(xdata gt nsew[3] and xdata le nsew[2])
sub_y=(ydata gt nsew[1] and ydata le nsew[0])
sub=where(sub_x*sub_y,subc)
if(subc eq 0) then print,'no points within bounds!'
return,sub
end; end function ENVI_POINT_SUBSETS()

function
 envi_point_xyconvert,dn,xdata,ydata,image=image,scroll=scrol l,nosub=nosub
; ENVI_point_xyconvert(): given XY data in decimal degrees and a
display
; number, returns file coordinates (or image window coordinates)

if(keyword_set(nosub)) then $
```

```
  sub=lindgen(n_elements(xdata)) else $
  if(keyword_set(image)) then $
  sub=envi_point_subsets(dn,xdata,ydata,/image) else $
  sub=envi_point_subsets(dn,xdata,ydata)
if(sub[0] eq -1) then message, "No data within bounds." ; bail if no
data
subc=n_elements(sub)
out=lonarr(2,subc); XY output array
; get projection information
ENVI_DISP_QUERY, dn,w1=dwin,fid=fid,x0=dx0_p,y0=dy0_p
 ,xds=dxdim_p,yds=dydim_p,nl=fydim_p,ns=fxdim_p,rebin=scrolls cale
map=envi_get_map_info(dn=dn); rather than fid=fid[0])
proj_u=map.proj; extract projection from structure $MAP
proj_dd=envi_proj_create(/GEOGRAPHIC);projection structure for lat/lon
data

; convert XY data to map projection, then to file coords
 envi_convert_projection_coordinates,xdata[sub],ydata[sub],pr oj_dd,x_u,y_u,proj_u;
convert lat/long points to map projection
envi_convert_file_coordinates,fid[0],x_fp,y_fp,x_u,y_u; convert map
points to file coordinates
if(mean(y_fp) lt 0) then begin
    print,'dealing with negative y-values'
    y_fp=y_fp+fydim_p          ; deal with different UTM setup
endif
; some SH UTM have a high false northing which means you deal with
; positive numbers. Without this false northing, the equation gives
; negative results in file coordinates.
if(keyword_set(scroll)) then begin
scrollx=x_fp/float(scrollscale)
scrolly=(fydim_p[0]-y_fp)/float(scrollscale)
out[0,*]=scrollx
out[1,*]=scrolly
return,out
endif
if(keyword_set(image)) then begin
; establish display coordinate system
dwest_p=dx0_p
deast_p=dwest_p+dxdim_p[0]
dnorth_p=dy0_p
dsouth_p=dnorth_p+dydim_p[0]
; convert file coords to display
x_dp=x_fp-dwest_p
y_dp=dydim_p[0]-(y_fp-dnorth_p);
out[0,*]=x_dp
out[1,*]=y_dp
return,out
endif
```

```
;default setting is to return file coords
out[0,*]=x_fp
out[1,*]=y_fp
return,out
end; end FUNCTION ENVI_POINT_XYCONVERT

function envi_get_wid,dn,image=image,scroll=scroll
ENVI_DISP_QUERY, dn,w1=dwin,fid=fid,x0=dx0_p,y0=dy0_p
 ,xds=dxdim_p,yds=dydim_p,nl=fydim_p,ns=fxdim_p,rebin=scrolls cale
if(keyword_set(image)) then return,dwin[0]
if(keyword_set(scroll)) then return,dwin[2]
return,dwin; default is to return 3-element window array
end                          ; end function ENVI_GET_WID


function envi_point_roimatch,roi_id,fid,channels=channels
; ENVI_point_roimatch: given an ROI ID and a file ID, returns file
; data
; set channels to query
if(n_elements(channels) eq 0) then pos=0 else pos=channels
data=envi_get_roi_data(roi_id,fid=fid,pos=pos)
return,data
end; end function ENVI_POINT_ROIMATCH


function
 envi_point_roigen,dn,xdata,ydata,image=image,sub=sub,name=na
me,getdata=getdata,data=data,channels=channels,nosub=nosub
; ENVI_point_roigen: given XY data in decimal degrees and a display ID,
; generates a point ROI for that display. Optionally, returns data
; from the file.
if(n_elements(name) eq 0) then name='New ROI' ;
; Get subset and check for points in bounds
if(keyword_set(nosub)) then $
  sub=lindgen(n_elements(xdata)) else $
if(keyword_set(image)) then $
  sub=envi_point_subsets(dn,xdata,ydata,/image) else $
  sub=envi_point_subsets(dn,xdata,ydata)
if(sub[0] eq -1) then message, "No data within bounds." ; bail if no
data
subc=n_elements(sub)
; Convert points within bounds to file coords
  filecoords= envi_point_xyconvert(dn,xdata[sub],ydata[sub],/nosub)
; create an ROI to hold the points
ENVI_DISP_QUERY, dn,w1=dwin,fid=fid,x0=dx0_p,y0=dy0_p
 ,xds=dxdim_p,yds=dydim_p,nl=fydim_p,ns=fxdim_p,rebin=scrolls cale
roi_id=envi_create_roi(name=name,nl=fydim_p,ns=fxdim_p)
; add points to the ROI
```

```
 envi_define_roi,roi_id,/point,xpts=filecoords[0,*],ypts=file coords[1,*]
; ***KLOOGE***: Save, Delete, and Restore ROI for proper function
envi_save_rois,'temp.roi',roi_id
envi_delete_rois,roi_id
envi_restore_rois,'temp.roi'
;find your ROI again
ids=envi_get_roi_ids(dn=dn)
nids=n_elements(ids)
roi_id=ids[nids-1]; last element = newest ROI = yours
;*** end KLOOGE***

; If requested, get ROI data from the file
if(keyword_set(getdata)) then begin $
data=envi_point_roimatch(roi_id,fid[0],channels=channels)
endif; end GETDATA sub
return,roi_id; give the ROI ID back to the USER
end; end function ENVI_POINT_ROIGEN


pro
 envi_point_ddplot,dn,xdata,ydata,scroll=scroll,_extra=extra_ keywords
; ENVI_point_ddplot: use PLOTS to put XY data directly onto Image or
; Scroll windows
if(keyword_set(scroll)) then begin
xy=envi_point_xyconvert(dn,xdata,ydata,/scroll)
w=envi_get_wid(dn,/scroll)
endif else begin $
xy=envi_point_xyconvert(dn,xdata,ydata,/image)
w=envi_get_wid(dn,/image)
endelse
wset,w
plots,/device,xy[0,*],xy[1,*],_extra=extra_keywords
end; end pro ENVI_POINTS_DDPLOT

function
 envi_boundingbox_roigen,dnsmall,dnbig,name=name,getdata=getd
ata,data=data,channels=channels
; generates an ROI corresponding to the area of overlap between two
; images. The ROI is generated to be displayed and manipulated in the
; larger image.
nsew_small=envi_get_ddbounds(dnsmall)
nsew_big=envi_get_ddbounds(dnbig)
; find overlap area
if( $; tests for no overlap
      nsew_small[0] lt nsew_big[1] or $;
      nsew_small[1] gt nsew_big[0] or $;
      nsew_small[2] lt nsew_big[3] or $;
      nsew_small[3] gt nsew_big[2] ) then $
```

```
  message, "No overlap between displays!"
overlap_north=min([nsew_small[0],nsew_big[0]])
overlap_south=max([nsew_small[1],nsew_big[1]])
overlap_east=min([nsew_small[2],nsew_big[2]])
overlap_west=max([nsew_small[3],nsew_big[3]])
;print,'NSEW of
 overlap',overlap_north,overlap_south,overlap_east,overlap_we st

; create ROI with overlap dims
ENVI_DISP_QUERY, dnbig,w1=dwin,fid=fid,x0=dx0_p,y0=dy0_p
 ,xds=dxdim_p,yds=dydim_p,nl=fydim_p,ns=fxdim_p,rebin=scrolls cale
roi_id=envi_create_roi(name=name,nl=fydim_p,ns=fxdim_p)
if(roi_id eq -1) then message, "nocando create_roi!"
 xbox=[overlap_west,overlap_east,overlap_east,overlap_west,ov erlap_west]
 ybox=[overlap_south,overlap_south,overlap_north,overlap_nort h,overlap_south]
box_fp=envi_point_xyconvert(dnbig,xbox,ybox,/nosub); convert to file
coords
;print,'Box in File coords',box_fp[0,*],box_fp[1,*]
;print,break
 envi_define_roi,roi_id,/polygon,xpts=reform(box_fp[0,*]),ypt s=reform(box_fp[1,*])
;; ***KLOOGE***: Save, Delete, and Restore ROI for proper function
;envi_save_rois,'temp.roi',roi_id
;envi_delete_rois,roi_id
;envi_restore_rois,'temp.roi'
;;find your ROI again
;ids=envi_get_roi_ids(dn=dn)
;nids=n_elements(ids)
;roi_id=ids[nids-1]; last element = newest ROI = yours
;;*** end KLOOGE***

; If requested, get ROI data from the big file
if(keyword_set(getdata)) then begin $
data=envi_point_roimatch(roi_id,fid[0],channels=channels)
endif; end GETDATA sub
return,roi_id; give the ROI ID back to the USER
end; end function ENVI_BOUNDINGBOX_ROIGEN

pro envi_list_fileids,fids=fids
fids = envi_get_file_ids()
if (fids[0] gt 0) then begin
for i = 0, n_elements(fids) - 1 do begin
   envi_file_query, fids[i], fname = fname
   print, fids[i], fname,format='(i5,"   ",a)'
endfor
endif
end; end PRO ENVI_LIST_FILEIDS

pro envi_roi_info
```

```
ids=envi_get_roi_ids(); get all valid ROI IDs
if(ids[0] eq -1) then return; don't run if no ROIs.

 envi_get_roi_information,ids,/short_name,nl=nl,ns=ns,npts=np
ts,roi_colors=roi_colors,roi_names=roi_names
print,["ID","NAME","COLOR","POINTS","DX","DY"],$
      format='(a-5,a-35,a6,a9,a7,a7)'
for iroi=0,n_elements(ids)-1 do $
         print, $
           ids[iroi],$
           roi_names[iroi],$
           roi_colors[iroi],$
           npts[iroi],$
           ns[iroi],$
           nl[iroi],$
           format='(i-5,a-35,i6,i9,i7,i7)'
return
end; end PRO ENVI_ROI_INFO
```