Subject: Re: ENVI's codes
Posted by Hassan on Wed, 25 Nov 2009 11:40:09 GMT
View Forum Message <> Reply to Message

On Nov 23, 3:29 pm, jeanh
<jghasb...@DELETETHIS.environmentalmodelers.ANDTHIS.com> wrote:
> Hassan wrote:
>> On Nov 23, 12:18 am, Jiek <676215...@qq.com> wrote:
>>> On Nov 23, 5:35 am, Hassan <hkhav...@gmail.com> wrote:
>
>>>> I used the following code to display the image:
>>>> image=read_tiff(file)
>>>> imagesize=[62,488,290]
>>>> DEVICE, DECOMPOSED = 0
>>>> LOADCT, 38
>>>> WINDOW, 0, XSIZE = imageSize[1], YSIZE = imageSize[2]
>>>> TV, Image(0,*,*), 0
>>>> but there are two problems: first it's displayed upside-down and
>>>> second the way it shows the image is quite different with other
>>>> softwares like ENVI, it's more like the image is shown in 256-color
>>>> table or something like that.
>>> I found a function to read ENVI image and it work well.As follows:
>>> pro read_envi_image, infile, img, xs, ys, type, offset, mapinfo
>
>>> ;
>>> ; Copyright (c) 2003,Institute of Photogrammetry and Remote Sensing, ;
>>> (IPF),
>>> ; Technical University of Vienna. Unauthorised reproduction
>>> prohibited.
>>> ;
>>> ;+
>>> ; NAME:
>>> ; read_envi_file
>>> ;
>>> ; PURPOSE:
>>> ; IDL program, which reads standard ENVI image files (*.img).
>>> ;
>>> ;
>>> ; CATEGORY:
>>> ; Input_Output
>>> ;
>>> ; CALLING SEQUENCE:
>>> ; read_envi_file, infile, img, xs, ys, type,offset
>>> ;
>>> ; INPUTS:
>>> ; infile - input file name
>>> ;
>>> ; OPTIONAL INPUTS:

```
>>> ; None
>>> ;
>>> ; KEYWORD PARAMETERS:
>>> ; None
>>> ;
>>> ; OUTPUTS:
>>> ; img - ENVI image file, 2D array
>>> ; xs - number of samples
>>> ; ys - number of lines
>>> ; type - image data type
>>> ; offset - headeroffset
>>> ; mapinfo - information on spatial resolution (spacing) and
>>> coordinates
>>> ; of upper left corner (ulx, uly)
>>> ;
>>> ;
>>> ; OPTIONAL OUTPUTS:
>>> ; None
>>> ;
>>> ; COMMON BLOCKS:
>>> ; none
>>> ;
>>> ; SIDE EFFECTS:
>>> ;
>>> ; RESTRICTIONS:
>>> ; None
>>> ;
>>> ; PROCEDURE:
>>> ;
>>> ; EXAMPLE:
>>> ;
>>> ; REMARKS
>>> ; None
>>> ;
>>> ; MODIFICATION HISTORY:
>>> ; Written by: Carsten Pathe, c...@ipf.tuwien.ac.at
>>> ; Date: 25.08.2003
>>> ;
>>> ;-
>
>>> image = infile
>
>>> header = strsplit(infile,'.',/extract)
>>> header = header(n_elements(header)-2)+'.hdr'
>
>>> openr, unit, header, /get_lun
>
>>> header_line = ''
```

>
>>> while not eof(unit) do begin
>
>>> readf, unit, header_line
>>> tmp = strsplit(header_line(0), '=', /extract)
>>> header_keyword = strsplit(tmp(0), ' ', /extract)
>
>>> print, header_keyword
>
>>> if header_keyword(0) eq 'samples' then xs = long(tmp(1))
>>> if header_keyword(0) eq 'lines' then ys = long(tmp(1))
>>> if header_keyword(0) eq 'header' then offset = long(tmp(1))
>>> if header_keyword(0) eq 'data' then type = long(tmp(1))
>
>>> if header_keyword(0) eq 'map' then begin
>
>>> mapinfo_tmp=strsplit(tmp(1),'{',/extract)
>>> mapinfo_tmp=strsplit(mapinfo_tmp(1),',',/extract)
>
>>> mapinfo={ulx:0.,uly:0.,spacing:0.}
>>> mapinfo.ulx=mapinfo_tmp(3)
>>> mapinfo.uly=mapinfo_tmp(4)
>>> mapinfo.spacing=mapinfo_tmp(5)
>
>>> endif
>
>>> endwhile
>
>>> close,unit & free_lun, unit
>>> print, xs, ys
>
>>> if type eq 1 then img=bytarr(xs, ys)
>>> if type eq 2 then img=intarr(xs, ys)
>>> if type eq 4 then img=fltarr(xs, ys)
>>> if type eq 12 then img=uintarr(xs, ys)
>
>>> openr, unit,image, /get_lun
>>> point_lun, unit, offset
>>> readu, unit, img
>>> close, unit & free_lun, unit
>
>>> end
>
>> I run it but it seems it imports just one band, right? is the output
>> variable named 'img'?
>
> This function is incomplete. You can customize it so it also reads the
> number of bands... and the band data.

```
>  Also, if you read an Envi file, you may be interested in reading the
>  lookup table from the header (if any), so that you can display the data
>  with the exact same color in Envi and in IDL.
>
>  By the way, you say you want to do this in IDL, but you didn't specify
>  if you have to get rid of using Envi or not. If not, you can use the
>  following (note: again, the info read in the header is incomplete and
>  suits my need... feel free to read more info, such as the number of
>  bands AND the information on the band interleave..
>
>  ;Open the file in envi
>  ENVI_OPEN_FILE, fileToOpen, /NO_REALIZE , R_FID=FID
>
>  ;If it is a valid file, read the header
>  if (fid[0] eq -1) then return, -1
>  envi_file_query, fid, ns=dimX, nl=dimY, nb=nbBands, DATA_TYPE=dataType,$
>     OFFSET = HeaderOffSet, LOOKUP= lookup, class_names = classNames,$
>     num_classes = numClasses
>  map_Info = ENVI_GET_MAP_INFO(FID=FID)
>
>  ;Pack the header info in a structure (optional)
>  headerInfo = {ns:dimX, nl:dimY, lookup:lookup, map_info:map_Info,$
>        class_names:classNames, num_classes:numClasses}
>
>  ;Read the data in IDL
>  data = read_binary(fileToOpen,data_Start = HeaderOffSet,$
>        data_Type=dataType, data_Dims=[dimX,dimY],ENDIAN = "native")
>
>  Jean
```

thanks for that. but i still cant display the image as envi does,
there's no look up table in header file but i should be able to make
one.
