Subject: Re: image cutting,
Posted by dinhnq on Thu, 20 Dec 2001 18:30:28 GMT
View Forum Message <> Reply to Message

Hi Leon Majewski,

Thank you very much for your response.
In fact, my data is a set of landsat images (can be in both .tiff anf
.hdf format) in seprate files.
I want to do some classification (in smaller area) but in ENVI I canï¿½t
see any function can do this with seprate files. So I have to put 6
bands together in one file. The output file can be any type that can
read by ENVI.
Because I just begin learning IDL a week before so it take time to
understand your codes.
When I run "Test_HDF_Cropping_GenData", there is a message
"HDF_SD_START: Unable to start the HDF-SD interface"
When I run "Test_HDF_Cropping", the message is "Execution halted at:
HDF_CROPPING   19 E:\IDLPRO\HDF_Cropping.pro
%   TEST_HDF_CROPPING   87
E:\IDLPRO\HDF_Cropping.pro" and stop at line
"HDF_SD_GETINFO,sds_id_i,NAME=NAME"
I donï¿½t know which file(s) below have to be in working directory
before running progam:
Filename = 'C:\test2.hdf'
oFilename= 'C:\test2sm.hdf'
Filename = 'MyLargeFile.hdf'
oFilename= 'MySmallFile.hdf'

And I wrote few lines to make a multi-channel image from seprate band
like this. It works with 3 bands, but it doensï¿½t works with 6 bands.
Could you explain for me why?

```
===========
PRO Composite, Red=red, Green=green, Blue=blue

 cd, 'e:\IDLPRO\'

 band1 = READ_TIFF('tm1.tiff')
 band2 = READ_TIFF('tm2.tiff')
 band4 = READ_TIFF('tm4.tiff')
 WRITE_TIFF, 'multi3.tif', red=band1, green=band2, blue=band4,
PLANARCONFIG=2

END
===========

PRO Composite6, ba1=ba1, ba2=ba2, ba3=ba3, ba4=ba4, ba5=ba5, ba7=ba7
```

```
 cd, 'e:\IDLPRO\'
 b1 = READ_TIFF('tm1.tiff')
 b2 = READ_TIFF('tm2.tiff')
 b3 = READ_TIFF('tm3.tiff')
 b4 = READ_TIFF('tm4.tiff')
 b5 = READ_TIFF('tm5.tiff')
 b7 = READ_TIFF('tm7.tiff')
 WRITE_TIFF, 'multi6.tif', ba1=b1, ba2=b2, ba3=b3 , ba4=b4, ba5=b5,
ba7=b7 ;  PLANARCONFIG=1

END
```

Many thanks again,

Huong Dinh

>
> Hi Dinh
>
> You might need to provide a little more information. HDF files can store
> a wide variety of information types - each needing to be accessed in
> it's own special way.
> My guess is that you want to access a scientific data set (SDS; ie use
> the SD interface) and crop this to a 200x200 segment. Then store this in
> a new HDF file as an SDS.
>
> A couple of questions arise:
> a) is the data you are trying to access an SDS?
> b) is the original data set all in one block (12x1000x1000) or separate
>   SDSs?
> c) does the output file need to be HDF (see d)
> d) is there any meta-data that is required to be copied from the
>   original file to the output file
> e) ...
>
>
> Anyway, here is a program (included below - limited error checking /
> testing / thinking ;ï¿½)) to get you started. The main points in this
> are:
>
> a) the START and COUNT keywords to HDF_SD_GETDATA:

```
>    HDF_SD_GETDATA, sds_id_i, SmallDataSet, START=Offset, COUNT=SegSize
>
> b) the writting of data to an HDF file:
>    new_sds_id = HDF_SD_CREATE(sd_oid, $
>   string('sub '+NAME), size(SmallDataSet, /dimensions))
>    HDF_SD_ADDDATA, new_sds_id, SmallDataSet
>
> c) you need to know the index numbers of the original 12 data sets:
>    Get_SDSs = [0,1,2,3,4,5,6,7,8,9,10,11]
>
> Leon
>
> ;================================================
>
> FUNCTION HDF_Cropping, inFilename, outFilename, $
>  Offset, SegSize, Get_SDSs
>
> ;Open the HDF file in SD read mode
>  sd_id = HDF_SD_START(inFilename, /READ)
>
> ;Find out any some about the file
>  HDF_SD_FILEINFO,sd_id,NumSDS,attributes
>  IF NumSDS GE 1 THEN BEGIN
>  ;If there are SDs found in the file, create an output file
>   sd_oid = HDF_SD_START(outFilename, /CREATE)
>
>   FOR i = 0, n_elements(Get_SDSs)-1 DO BEGIN
>   ;Select the i^th data set (from the Get_SDSs above)
>    sds_id_i = HDF_SD_SELECT(sd_id, Get_SDSs[i])
>    If sds_id_i[0] ne 0 then begin
>
>    ;Retrieve the data segment
>     HDF_SD_GETINFO,sds_id_i,NAME=NAME
>     HDF_SD_GETDATA, sds_id_i, SmallDataSet, $
>      START=Offset, COUNT=SegSize
>    ;End access to the dataset
>     HDF_SD_ENDACCESS, sds_id_i
>
>    ;Now that the data segment has been obtained, store it:
>     new_sds_id = HDF_SD_CREATE(sd_oid, $
>      string('sub '+NAME), $
>      size(SmallDataSet, /dimensions))
>     HDF_SD_ADDDATA, new_sds_id, SmallDataSet
>
>     HDF_SD_ENDACCESS, new_sds_id
>    ENDIF
>   ENDFOR
>
```

```
>   ;Close the output file
>    HDF_SD_END, sd_oid
>   ;Close the input file
>    HDF_SD_END,sd_id
>   ;Return to the calling program letting it know everything went ok
>    RETURN, 1
>   ENDIF
>
>   HDF_SD_END,sd_id
>   RETURN, 0
>  END
>
>  ;=================================================
>
>  PRO Test_HDF_Cropping_GenData
>  Filename = 'C:\test2.hdf'
>  oFilename= 'C:\test2sm.hdf'
>
>  StartPos = [400,400]
>  ImageSize= [200,200]
>
>  ;generate the test data file:
>  TestData = dist(1000)
>
>  sd_oid = HDF_SD_START(Filename, /CREATE)
>  new_sds_id = HDF_SD_CREATE(sd_oid, 'My Test Data', $
>   size(TestData, /dimensions))
>  HDF_SD_ADDDATA, new_sds_id, TestData
>  HDF_SD_ENDACCESS, new_sds_id
>  HDF_SD_END, sd_oid
>
>   ;Make an array that holds the index of the data set to be subsampled
>   Get_SDSs = [0,0,0,0]
>
>   Result = HDF_Cropping(Filename, oFilename, $
>    StartPos, ImageSize, Get_SDSs)
>   a = hdf_browser(oFilename)
>  END
>
>  ;=================================================
>
>  PRO Test_HDF_Cropping
>  Filename = 'MyLargeFile.hdf'
>  oFilename= 'MySmallFile.hdf'
>
>  StartPos = [400,400]
>  ImageSize= [200,200]
>
```

```
>   ;Make an array that holds the index of the data set to be subsampled
>   Get_SDSs = [0,1,2,3,4,5,6,7,8,9,10,11]
>
>   Result = HDF_Cropping(Filename, oFilename, $
>    StartPos, ImageSize, Get_SDSs)
>   a = hdf_browser(oFilename)
>  END
>
>  ;===========================================
>
>  -------------------------
>  Leon Majewski
>
>  Remote Sensing & Satellite Research Group
>  Curtin University of Technology, Perth, Australia
>
>  email: majewski@ses.curtin.edu.au
```