Subject: MOD43B3 Col.4 Processing. unexpected error message. why?
Posted by kim20026 on Mon, 09 Jul 2007 17:46:19 GMT
View Forum Message <> Reply to Message

Good day, everyone. Even though I am still in the swamp and my previous question on MOD04 is left unsolved, I have to process MOD43B3 albedo product. I am trying to make image files from MOD43B3 hdf files now.

I did this way so far.

0. Variable (initialization/declaration)
1. Get hdf file info.
2. Input map info (sin projection)
3. Convert SIN to TM Korea

There was no problem in making images with one variable (black sky albedo), and then I added another similar variable, white sky albedo, in my new simulation. (This part is commented out and labed as part1 and part2)

However, I got unexpected error messages shown below.

Envi retrieve data: An error has occurred during processing.
Error: "Array dimensions must be greater than 0". The result may be invalid.

These messages are too broad for me and I don't know how to handle this. In addition, the ' Envi_retrieve_data' must be an internal function... something like that. I cannot find this function in ENVI help.

Any suggestions?

Harry

```
PRO MOD43B3_MakeImage_1km_03

; 0. Variable (initialization/declaration)
WorkDir = 'd:\MODIS_TSrad\MODIS_ALL\'
WorkDirOut = 'd:\MODIS_TSrad\MODIS43\Processed\'
batch_st = strcompress(WorkDir + 'batch_MOD43_TSrad.txt', /remove_all)
WorkDirSat = 'D:\MODIS_TSrad\MODIS43\MOD43\'
StrMOD = 'MOD43'
```

```
OpenR, lun, batch_st, /Get_Lun
numdates =file_lines(batch_st)
Dates = StrArr(numdates)

; Read input dates from batch file

OpenR, lun, batch_st, /Get_lun
ReadF, lun, Dates

Free_Lun, lun, /force

close, /all

FOR j = 0L, numDates-1 DO BEGIN

   ;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - -

   print, "Now processing  MOD43 data from date: ", Dates[j], '
File ', j+1, ' out of ', numDates, $
            '  overpassing time. '

; 0. Get hdf file info.

   Filename     = WorkdirSat+Dates[j]
   FileID       = HDF_OPEN(filename, /Read)
   sdFileID     = HDF_SD_Start(filename, /Read)    ;The returned
value of this function is the SD ID of the HDF file
   sdsID_albedo = HDF_SD_Select(sdFileID, 0)   ; Albedo
   sdsID_qc     = HDF_SD_Select(sdFileID, 1)   ; QC

   hdf_sd_getdata, sdsID_albedo, albedo
   hdf_sd_getdata, sdsID_qc, qc

   ;print, 'j = ', j
   ;help, albedo, qc

   albedo_black = fltarr(1200, 1200)
   albedo_white = fltarr(1200, 1200)
   qc1 = ulonarr(1200, 1200)
   qc2 = ulonarr(1200, 1200)

   albedo_black[*, *]=albedo[0, 9, *, *]
   albedo_white[*, *]=albedo[1, 9, *, *]

   qc1[*,*] = qc[0,*,*]
   qc2[*,*] = qc[1,*,*]
```

```
;help, albedo_black, albedo_white, qc1, qc2


 ;VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVV
;V  V  V                                     V  V  V
;V  V  V      1. Input map Information(SIN projection)   V  V  V
;V  V  V      2. Convert SIN to TM Korea                V  V  V
;V  V  V                                     V  V  V
 ;VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVV

;   1. Input map information (SIN)

   mc=[0.5, 0.5, 11119968.509D, 4447338.766D]
   ps=[926.6254331D, 926.6254331D]
   units   =envi_translate_projection_units('Meters')
   params1=[6371007.181D, 0, 0, 0]
   Projection_Name1 = 'SIN_MODIS'
   map_info = ENVI_MAP_INFO_CREATE(type=16, name=Projection_Name1,
params=params1, $
        UNITS = units, MC = mc, PS = ps)

   ENVI_WRITE_ENVI_FILE, albedo_black,
out_name='albedo_black_map_4.img', map_info=map_info, out_dt=4,$
       r_fid=albedo_black_map, sensor_type=32

   envi_file_query, albedo_black_map, ns=ns, nl=nl, nb=nb
   dims   = [-1, 0, ns-1, 0, nl-1]
   pos_albedo_black = lindgen(nb)

    ;
 ===========================================================
==============
   ; Part 1
   ; ENVI_WRITE_ENVI_FILE, albedo_white,
out_name='albedo_white_map_4.img', map_info=map_info, out_dt=4,$
   ;    r_fid=albedo_white_map, sensor_type=32
   ;
   ; envi_file_query, albedo_white_map, ns=ns, nl=nl, nb=nb
   ; dims   = [-1, 0, ns-1, 0, nl-1]
   ; pos_albedo_white = lindgen(nb)
    ;
 ===========================================================
==============
;   2. Convert SIN to TM Korea
;    Names of MODIS products
```

```
;    MOD43B3.A2002177.h28v05.004.2003246195929.hdf

   StrDate   = STRMID(Dates[j], 9, 7)
   StrSat    = STRMID(Dates[j], 0, 7)

   out_name_albedo_black = WorkDirOut+StrSat+'\'+StrDate+'\'+StrDate
+'_albedo_black_1km_4'+'.img'
   out_name_albedo_white = WorkDirOut+StrSat+'\'+StrDate+'\'+StrDate
+'_albedo_white_1km_4'+'.img'

   DATUM          = 'Tokyo mean'
   Projection_Name2= 'Korea - TM (Middle)'
   Params2 = [6377397.2, 6356079.0, 38.000000D, 127.002890D,
200000.0, 500000.0, 1.000000]

   OUT_Proj = ENVI_PROJ_CREATE(type=3, name=Projection_Name2,
datum=Datum, params=Params2)

   envi_convert_file_map_projection, fid=albedo_black_map,
pos=pos_albedo_black, dims=dims, o_proj=OUT_Proj, $
       o_pixel_size=[1000, 1000],out_name=out_name_albedo_black,
warp_method=2, r_fid=albedo_black_TM, $
       resampling=0, background=0

   envi_file_query, albedo_black_TM, ns=ns, nl=nl, nb=nb
   dims   = [-1, 0, ns-1, 0, nl-1]
   pos_albedo_black_TM = lindgen(nb)

   ;
 ============================================================
==============
   ;  part 2
   ;    envi_convert_file_map_projection, fid=albedo_white_map,
pos=pos_albedo_white, dims=dims,       o_proj=OUT_Proj, $
   ;    o_pixel_size=[1000, 1000],out_name=out_name_albedo_white,
warp_method=2, r_fid=albedo_white_TM, $
   ;    resampling=0, background=0

   ;   envi_file_query, albedo_white_TM, ns=ns, nl=nl, nb=nb
   ;   dims   = [-1, 0, ns-1, 0, nl-1]
   ;   pos_albedo_white_TM = lindgen(nb)
    ;
 ============================================================
==============

   ;-----------------------------------------------------
   ; Cleaning memory
   ;envi_file_mng, id= albedo_black_map, /remove
```

```
    ;envi_file_mng, id= albedo_white_map, /remove
    ;---------------------------------------------------
    ; Done with SDS, close the interface
    HDF_SD_ENDACCESS, SDSID_albedo

    HDF_SD_END, sdFileID
    HDF_Close, FileID

 Close, /all, /force

ENDFOR
print, "C'est si bon! C'est fini!!!"
Free_Lun, lun, /force
close, /all

END
```

Subject: Re: MOD43B3 Col.4 Processing. unexpected error message. why?
Posted by devin.white on Wed, 11 Jul 2007 17:12:50 GMT
View Forum Message <> Reply to Message

This is just an initial stab at what might be the problem, but I noticed that you're defining your albedo arrays as floating point:

```
    albedo_black = fltarr(1200, 1200)
    albedo_white = fltarr(1200, 1200)
```

MOD43B3 albedo is stored as integer, which is half the size. Your data are most likely getting written into the file, but ENVI is expecting them to take up more room. As it attempts to read in the file you've created, it's going beyond the boundaries of your data to do so--hence the ENVI_RETRIEVE_DATA error (there was no data to retrieve). You don't get floating point output unless you take the four-dimensional albedo array and apply the necessary scale and offset factors stored in the SD. Just a thought.

You really should try out the IDL programming interface that comes with the MODIS Conversion Toolkit. It will save you a great deal of time and trouble. I spent several months building that plugin so members of the ENVI/IDL user community, such as yourself, could avoid many of the headaches/hassles associated with ingesting MODIS data--including this one. You can download the plugin, for free, here:

http://www.ittvis.com/codebank/search.asp?FID=485

I am always looking for feedback and suggestions for improvement. =)

On Jul 9, 1:46 pm, DirtyHarry <kim20...@gmail.com> wrote:
> Good day, everyone. Even though I am still in the swamp and my
> previous question on MOD04 is left unsolved, I have to process MOD43B3
> albedo product. I am trying to make image files from MOD43B3 hdf files
> now.
>
> I did this way so far.
>
> 0. Variable (initialization/declaration)
> 1. Get hdf file info.
> 2. Input map info (sin projection)
> 3. Convert SIN to TM Korea
>
> There was no problem in making images with one variable (black sky
> albedo), and then I added another similar variable, white sky albedo,
> in my new simulation. (This part is commented out and labed as part1
> and part2)
>
> However, I got unexpected error messages shown below.
>
> --------------
> Envi retrieve data: An error has occurred during processing.
> Error: "Array dimensions must be greater than 0". The result may be
> invalid.
> --------------
>
> These messages are too broad for me and I don't know how to handle
> this. In addition, the ' Envi_retrieve_data' must be an internal
> function... something like that. I cannot find this function in ENVI
> help.
>
> Any suggestions?
>
> Harry
>
> ------------------------------
> PRO MOD43B3_MakeImage_1km_03
>
> ; 0. Variable (initialization/declaration)
> WorkDir = 'd:\MODIS_TSrad\MODIS_ALL\'
> WorkDirOut = 'd:\MODIS_TSrad\MODIS43\Processed\'
> batch_st = strcompress(WorkDir + 'batch_MOD43_TSrad.txt', /remove_all)
> WorkDirSat = 'D:\MODIS_TSrad\MODIS43\MOD43\'
> StrMOD = 'MOD43'
>
> OpenR, lun, batch_st, /Get_Lun
> numdates =file_lines(batch_st)

```
>  Dates = StrArr(numdates)
>
>  ; Read input dates from batch file
>
>  OpenR, lun, batch_st, /Get_lun
>  ReadF, lun, Dates
>
>  Free_Lun, lun, /force
>
>  close, /all
>
>  FOR j = 0L, numDates-1 DO BEGIN
>
>     ;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
> - - - -
>
>     print, "Now processing  MOD43 data from date: ", Dates[j], '
>  File ', j+1, ' out of ', numDates, $
>               '  overpassing time. '
>
>  ; 0. Get hdf file info.
>
>     Filename     = WorkdirSat+Dates[j]
>     FileID       = HDF_OPEN(filename, /Read)
>     sdFileID     = HDF_SD_Start(filename, /Read)    ;The returned
>  value of this function is the SD ID of the HDF file
>     sdsID_albedo = HDF_SD_Select(sdFileID, 0)   ; Albedo
>     sdsID_qc     = HDF_SD_Select(sdFileID, 1)   ; QC
>
>     hdf_sd_getdata, sdsID_albedo, albedo
>     hdf_sd_getdata, sdsID_qc, qc
>
>     ;print, 'j = ', j
>     ;help, albedo, qc
>
>     albedo_black = fltarr(1200, 1200)
>     albedo_white = fltarr(1200, 1200)
>     qc1 = ulonarr(1200, 1200)
>     qc2 = ulonarr(1200, 1200)
>
>     albedo_black[*, *]=albedo[0, 9, *, *]
>     albedo_white[*, *]=albedo[1, 9, *, *]
>
>     qc1[*,*] = qc[0,*,*]
>     qc2[*,*] = qc[1,*,*]
>
>     ;help, albedo_black, albedo_white, qc1, qc2
>
```

```
>     ;VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVV
>     ;V  V  V                                        V  V  V
>     ;V  V  V      1. Input map Information(SIN projection)   V  V  V
>     ;V  V  V      2. Convert SIN to TM Korea               V  V  V
>     ;V  V  V                                        V  V  V
>     ;VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVV
>
> ;   1. Input map information (SIN)
>
>     mc=[0.5, 0.5, 11119968.509D, 4447338.766D]
>     ps=[926.6254331D, 926.6254331D]
>     units   =envi_translate_projection_units('Meters')
>     params1=[6371007.181D, 0, 0, 0]
>     Projection_Name1 = 'SIN_MODIS'
>     map_info = ENVI_MAP_INFO_CREATE(type=16, name=Projection_Name1,
> params=params1, $
>         UNITS = units, MC = mc, PS = ps)
>
>     ENVI_WRITE_ENVI_FILE, albedo_black,
> out_name='albedo_black_map_4.img', map_info=map_info, out_dt=4,$
>        r_fid=albedo_black_map, sensor_type=32
>
>     envi_file_query, albedo_black_map, ns=ns, nl=nl, nb=nb
>     dims   = [-1, 0, ns-1, 0, nl-1]
>     pos_albedo_black = lindgen(nb)
>
>      ;
>   ==========================================================
==============
>     ; Part 1
>     ; ENVI_WRITE_ENVI_FILE, albedo_white,
> out_name='albedo_white_map_4.img', map_info=map_info, out_dt=4,$
>     ;    r_fid=albedo_white_map, sensor_type=32
>     ;
>     ; envi_file_query, albedo_white_map, ns=ns, nl=nl, nb=nb
>     ; dims   = [-1, 0, ns-1, 0, nl-1]
>     ; pos_albedo_white = lindgen(nb)
>      ;
>   ==========================================================
==============
> ;   2. Convert SIN to TM Korea
> ;    Names of MODIS products
> ;    MOD43B3.A2002177.h28v05.004.2003246195929.hdf
>
>     StrDate   = STRMID(Dates[j], 9, 7)
>     StrSat    = STRMID(Dates[j], 0, 7)
```

```
>
>     out_name_albedo_black = WorkDirOut+StrSat+'\'+StrDate+'\'+StrDate
>  +'_albedo_black_1km_4'+'.img'
>     out_name_albedo_white = WorkDirOut+StrSat+'\'+StrDate+'\'+StrDate
>  +'_albedo_white_1km_4'+'.img'
>
>     DATUM          = 'Tokyo mean'
>     Projection_Name2= 'Korea - TM (Middle)'
>     Params2 = [6377397.2, 6356079.0, 38.000000D, 127.002890D,
>  200000.0, 500000.0, 1.000000]
>
>     OUT_Proj = ENVI_PROJ_CREATE(type=3, name=Projection_Name2,
>  datum=Datum, params=Params2)
>
>     envi_convert_file_map_projection, fid=albedo_black_map,
>  pos=pos_albedo_black, dims=dims, o_proj=OUT_Proj, $
>         o_pixel_size=[1000, 1000],out_name=out_name_albedo_black,
>  warp_method=2, r_fid=albedo_black_TM, $
>         resampling=0, background=0
>
>     envi_file_query, albedo_black_TM, ns=ns, nl=nl, nb=nb
>     dims   = [-1, 0, ns-1, 0, nl-1]
>     pos_albedo_black_TM = lindgen(nb)
>
>     ;
>   ============================================================
==============
>     ;  part 2
>     ;    envi_convert_file_map_projection, fid=albedo_white_map,
>  pos=pos_albedo_white, dims=dims,       o_proj=OUT_Proj, $
>     ;    o_pixel_size=[1000, 1000],out_name=out_name_albedo_white,
>  warp_method=2, r_fid=albedo_white_TM, $
>     ;    resampling=0, background=0
>
>     ;   envi_file_query, albedo_white_TM, ns=ns, nl=nl, nb=nb
>     ;   dims   = [-1, 0, ns-1, 0, nl-1]
>     ;   pos_albedo_white_TM = lindgen(nb)
>      ;
>   ============================================================
==============
>
>     ;-------------------------------------------------
>     ; Cleaning memory
>     ;envi_file_mng, id= albedo_black_map, /remove
>     ;envi_file_mng, id= albedo_white_map, /remove
>     ;-------------------------------------------------
>     ; Done with SDS, close the interface
>     HDF_SD_ENDACCESS, SDSID_albedo
```

>
>   HDF_SD_END, sdFileID
>   HDF_Close, FileID
>
>   Close, /all, /force
>
> ENDFOR
> print, "C'est si bon! C'est fini!!!"
> Free_Lun, lun, /force
> close, /all
>
> END

Subject: Re: MOD43B3 Col.4 Processing. unexpected error message. why?
Posted by James Kuyper on Wed, 11 Jul 2007 18:59:45 GMT
View Forum Message <> Reply to Message

devin.white@gmail.com wrote:
> This is just an initial stab at what might be the problem, but I
> noticed that you're defining your albedo arrays as floating point:
>
>   albedo_black = fltarr(1200, 1200)
>   albedo_white = fltarr(1200, 1200)
>
> MOD43B3 albedo is stored as integer, which is half the size.

As a result, we have:

IDL> HELP,albedo
ALBEDO   INT   = Array[2, 10, 1200, 1200]
IDL> help,albedo_black
ALBEDO_BLACK   FLOAT   = Array[1200, 1200]

However, an automatic conversion between INT and FLOAT occurs on the
following line:

albedo_black[*, *]=albedo[0, 9, *, *]

>   Your
> data are most likely getting written into the file, but ENVI is
> expecting them to take up more room.

The INT array 'albedo' is never passed to ENVI. Only the albedo_black
and albedo_white arrays are passed to ENVI_WRITE_ENVI_FILE, and those
are both float arrays. In any event, ENVI_WRITE_ENVI_FILE is supposed
to convert it's input array into whichever data type is specified by

the OUT_DT parameter. In this case, that parameter is set to 4, so no conversion is needed.

Whatever the problem is, that can't be the cause..

...
> retrieve). You don't get floating point output unless you take the
> four-dimensional albedo array and apply the necessary scale and offset
> factors stored in the SD. Just a thought.

The scale_factor and add_offset do need to be applied to calculate the correct albedo, as indicated by the filespec:

> Data conversions:
> file date =(Albedo / scale_factor) + add_offset
> Albedo =(file data - add_offset) * scale_factor

These values may be read from the file using code like the following:

```
idx_scale = HDF_SD_AttrFind(sdsID_albedo,'scale_factor')
HDF_SD_AttrInfo, sdsID_albedo, idx_scale, DATA=scale_factor
idx_offset = HDF_SD_AttrFind(sdsID_albedo, 'add_offset')
HDF_SD_AttrInfo, sdsID_albedo, idx_offset, DATA=add_offset
```

However, the problems he's been having don't seem to be related to the values stored in his variables. The message said that the problem was that something had a dimension of 0. Whether or not he applied the scale factor and offset shouldn't have any affect on that problem.

Subject: Re: MOD43B3 Col.4 Processing. unexpected error message. why?
Posted by devin.white on Thu, 12 Jul 2007 00:07:03 GMT

View Forum Message <> Reply to Message

I build this sample program based on his code and it ran without throwing an error. The four output files look fine, too. Very interesting....

```
pro mod43b3_test
 compile_opt idl2

 ;Simulate a MOD43B3 albedo SD data cube returned using HDF_SD_GETDATA
 data_array = intarr(2,10,1200,1200)

 ;Get the simulated black and white sky albedo arrays
 albedo_black = fltarr(1200,1200)
 albedo_white = fltarr(1200,1200)
```

```
 albedo_black[*,*] = data_array[0,9,*,*]
 albedo_white[*,*] = data_array[1,9,*,*]

 ;Set up map information
    mc = [0.5, 0.5, 11119968.509D, 4447338.766D]
    ps = [926.6254331D, 926.6254331D]
    units = envi_translate_projection_units('Meters')
    params1 = [6371007.181D, 0, 0, 0]
    Projection_Name1 = 'SIN_MODIS'
    map_info = ENVI_MAP_INFO_CREATE(type=16, name=Projection_Name1,
params=params1, $
        UNITS = units, MC = mc, PS = ps)

 black_out_name = 'c:\albedo_black_map_4.img'
 white_out_name = 'c:\albedo_white_map_4.img'

    ENVI_WRITE_ENVI_FILE, albedo_black, out_name=black_out_name,
map_info=map_info, out_dt=4, $
       r_fid=albedo_black_map, sensor_type=32, ns=1200, nl=1200,
nb=1, interleave=0

    ENVI_WRITE_ENVI_FILE, albedo_white, out_name=white_out_name,
map_info=map_info, out_dt=4,$
       r_fid=albedo_white_map, sensor_type=32, ns=1200, nl=1200,
nb=1, interleave=0

 ;Get basic file info for both images, since they are identical
 envi_file_query, albedo_black_map, ns=ns, nl=nl, nb=nb, dims=dims
 pos = lindgen(nb)

    out_name_albedo_black = 'c:\albedo_black_1km_4.img'
    out_name_albedo_white = 'c:\albedo_white_1km_4.img'

    DATUM = 'Tokyo mean'
    Projection_Name2 = 'Korea - TM (Middle)'
    Params2 = [6377397.2, 6356079.0, 38.000000D, 127.002890D,
200000.0, 500000.0, 1.000000]

    OUT_Proj = ENVI_PROJ_CREATE(type=3, name=Projection_Name2,
datum=Datum, params=Params2)

    envi_convert_file_map_projection, fid=albedo_black_map, pos=pos,
dims=dims, o_proj=OUT_Proj, $
       o_pixel_size=[1000, 1000],out_name=out_name_albedo_black,
warp_method=2, r_fid=albedo_black_TM, $
       resampling=0, background=0

 envi_convert_file_map_projection, fid=albedo_white_map, pos=pos,
```

```
dims=dims, o_proj=OUT_Proj, $
       o_pixel_size=[1000, 1000],out_name=out_name_albedo_white,
warp_method=2, r_fid=albedo_white_TM, $
       resampling=0, background=0

 ;Get rid of intermediary files
 envi_file_mng, id=albedo_black_map, /remove, /delete
 envi_file_mng, id=albedo_white_map, /remove, /delete

end
```

On Jul 11, 2:59 pm, kuyper <kuy...@wizard.net> wrote:
> devin.wh...@gmail.com wrote:
>> This is just an initial stab at what might be the problem, but I
>> noticed that you're defining your albedo arrays as floating point:
>
>>   albedo_black = fltarr(1200, 1200)
>>   albedo_white = fltarr(1200, 1200)
>
>> MOD43B3 albedo is stored as integer, which is half the size.
>
> As a result, we have:
>
> IDL> HELP,albedo
> ALBEDO   INT   = Array[2, 10, 1200, 1200]
> IDL> help,albedo_black
> ALBEDO_BLACK   FLOAT   = Array[1200, 1200]
>
> However, an automatic conversion between INT and FLOAT occurs on the
> following line:
>
> albedo_black[*, *]=albedo[0, 9, *, *]
>
>> Your
>> data are most likely getting written into the file, but ENVI is
>> expecting them to take up more room.
>
> The INT array 'albedo' is never passed to ENVI. Only the albedo_black
> and albedo_white arrays are passed to ENVI_WRITE_ENVI_FILE, and those
> are both float arrays. In any event, ENVI_WRITE_ENVI_FILE is supposed
> to convert it's input array into whichever data type is specified by
> the OUT_DT parameter. In this case, that parameter is set to 4, so no
> conversion is needed.
>
> Whatever the problem is, that can't be the cause..
>

> ...
>
>> retrieve). You don't get floating point output unless you take the
>> four-dimensional albedo array and apply the necessary scale and offset
>> factors stored in the SD. Just a thought.
>
> The scale_factor and add_offset do need to be applied to calculate the
> correct albedo, as indicated by the filespec:
>
>> Data conversions:
>> file date =(Albedo / scale_factor) + add_offset
>> Albedo =(file data - add_offset) * scale_factor
>
> These values may be read from the file using code like the following:
>
> idx_scale = HDF_SD_AttrFind(sdsID_albedo,'scale_factor')
> HDF_SD_AttrInfo, sdsID_albedo, idx_scale, DATA=scale_factor
> idx_offset = HDF_SD_AttrFind(sdsID_albedo, 'add_offset')
> HDF_SD_AttrInfo, sdsID_albedo, idx_offset, DATA=add_offset
>
> However, the problems he's been having don't seem to be related to the
> values stored in his variables. The message said that the problem was
> that something had a dimension of 0. Whether or not he applied the
> scale factor and offset shouldn't have any affect on that problem.

Subject: Re: MOD43B3 Col.4 Processing. unexpected error message. why?
Posted by kim20026 on Sat, 14 Jul 2007 02:20:56 GMT
View Forum Message <> Reply to Message

Thank you James and Devin for your excellent suggestions and comments!!! I also made it without any errors. I think the problem was that I used too may envi_file_query statements.

There was no difference between using integer and float for albedo. However, as devin suggested, MOD43B3 albedo is stored as integer. I have no reason to use floating point for this... To avoid other unexpected errors, I need to use integer for albedo from now on.

Here is my final code.

```
----------------------------------
PRO MOD43B3_MakeImage_1km_final

WorkDir = 'd:\MODIS_ALL\'
WorkDirOut = 'd:\MODIS43\Processed\'
batch_st = strcompress(WorkDir + 'batch_MOD43.txt', /remove_all)
WorkDirSat = 'D:\MODIS43\MOD43\'
```

```
StrMOD = 'MOD43'

albedo = intarr(2, 10, 1200, 1200)

albedo_black = intarr(1200,1200)
albedo_white = intarr(1200,1200)

qc = ulonarr(2, 1200, 1200)
qc1 = ulonarr(1200, 1200)
qc2 = ulonarr(1200, 1200)

OpenR, lun, batch_st, /Get_Lun
numdates = file_lines(batch_st)
Dates    = StrArr(numdates)

; Read input dates from batch file

OpenR, lun, batch_st, /Get_lun
ReadF, lun, Dates

Free_Lun, lun, /force

close, /all

FOR j = 0L, numDates-1 DO BEGIN

   ;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - -

   print, "Now processing  MOD43 data from date: ", Dates[j], '
File ', j+1, ' out of ', numDates, $
          '  overpassing dates. '

   Filename     = WorkdirSat+Dates[j]
   FileID       = HDF_OPEN(filename, /read)
   sdFileID     = HDF_SD_Start(filename, /read)    ;The returned
value of this function is the SD ID of the HDF file
   sdsID_albedo = HDF_SD_Select(sdFileID, 0)   ; Albedo
   sdsID_qc     = HDF_SD_Select(sdFileID, 1)   ; QC

   hdf_sd_getdata, sdsID_albedo, albedo
   hdf_sd_getdata, sdsID_qc, qc

   albedo_black[*,*] = albedo[0, 9, *, *]
   albedo_white[*,*] = albedo[1, 9, *, *]

   qc1[*,*] = qc[0,*,*]
   qc2[*,*] = qc[1,*,*]
```

```
    ;VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVV
   ;V  V  V                                    V  V  V
   ;V  V  V      1. Input map Information(SIN projection)   V  V  V
   ;V  V  V      2. Convert SIN to TM Korea               V  V  V
   ;V  V  V                                    V  V  V
    ;VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
VVVVV

;   1. Input map information (SIN)

   mc=[0.5, 0.5, 11119968.509D, 4447338.766D]
   ps=[926.6254331D, 926.6254331D]
   units   =envi_translate_projection_units('Meters')
   params1=[6371007.181D, 0, 0, 0]
   Projection_Name1 = 'SIN_MODIS'
   map_info=ENVI_MAP_INFO_CREATE(type=16, name=Projection_Name1,
params=params1, $
       UNITS = units, MC = mc, PS = ps)

 ;------------------------------------------------------------ ---
; albedo black and white

   ENVI_WRITE_ENVI_FILE, albedo_black,
out_name='albedo_black_map.img', map_info=map_info, out_dt=4,$
      r_fid=albedo_black_map, sensor_type=32, interleave = 0

   ENVI_WRITE_ENVI_FILE, albedo_white,
out_name='albedo_white_map.img', map_info=map_info, out_dt=4,$
      r_fid=albedo_white_map, sensor_type=32, interleave = 0

   envi_file_query, albedo_black_map, ns=ns, nl=nl, nb=nb
   dims   = [-1, 0, ns-1, 0, nl-1]
   pos_albedo_black = lindgen(nb)

 ;------------------------------------------------------------ ---
; qc1 and qc2

   ENVI_WRITE_ENVI_FILE, qc1 , out_name='qc1_map.img',
map_info=map_info, out_dt=2, $
   r_fid=qc1_map, sensor_type=32

   ENVI_WRITE_ENVI_FILE, qc2 , out_name='qc2_map.img',
map_info=map_info, out_dt=2, $
   r_fid=qc2_map, sensor_type=32

   envi_file_query, qc1_map, ns=ns, nl=nl, nb=nb
```

```
   dims   = [-1, 0, ns-1, 0, nl-1]
   pos_qc1 = lindgen(nb)

;  2. Convert SIN to TM Korea
;  Names of MODIS products

;  MOD11A1.A2002001.h28v05.004.2003182143054.hdf
;  MOD07_L2.A2003148.1150.004.2003173192335.hdf
;  MOD43B3.A2002177.h28v05.004.2003246195929.hdf

   StrDate   = STRMID(Dates[j], 9, 7)
   StrSat    = STRMID(Dates[j], 0, 7)

   out_name_Albedo_black = WorkDirOut+StrSat+'\'+StrDate+'\'+StrDate
+'_albedo_black_1km'+'.img'
   out_name_Albedo_white = WorkDirOut+StrSat+'\'+StrDate+'\'+StrDate
+'_albedo_white_1km'+'.img'
   out_name_QC1    = WorkDirOut+StrSat+'\'+StrDate+'\'+StrDate
+'_QC1_1km'+'.img'
   out_name_QC2    = WorkDirOut+StrSat+'\'+StrDate+'\'+StrDate
+'_QC2_1km'+'.img'

 ;---------------------------------------------------------- ---
;    Converting to Korea TM

   DATUM          = 'Tokyo mean'
   Projection_Name2= 'Korea - TM (Middle)'
   Params2 = [6377397.2, 6356079.0, 38.000000D, 127.002890D,
200000.0, 500000.0, 1.000000]

   OUT_Proj   = ENVI_PROJ_CREATE(type=3, name=Projection_Name2,
datum=Datum, params=Params2)

 ;---------------------------------------------------------- ---
; albedo black and white

   envi_convert_file_map_projection, fid=albedo_black_map,
pos=pos_albedo_black, dims=dims, o_proj=OUT_Proj, $
   o_pixel_size=[1000, 1000],out_name=out_name_albedo_black,
warp_method=2, r_fid=albedo_black_TM, $
   resampling=0, background=0

   envi_convert_file_map_projection, fid=albedo_white_map,
pos=pos_albedo_black, dims=dims, o_proj=OUT_Proj, $
   o_pixel_size=[1000, 1000],out_name=out_name_albedo_white,
warp_method=2, r_fid=albedo_white_TM, $
   resampling=0, background=0
```

```
 ;------------------------------------------------------------ ---
; qc1 and qc2

    envi_convert_file_map_projection, fid=qc1_map, pos=pos_qc1,
dims=dims, o_proj=OUT_Proj, $
    o_pixel_size=[1000, 1000],out_name=out_name_QC1, warp_method=2,
r_fid=qc1_TM, $
    resampling=0, background=0 ; NOTICE fid=qc1_map!

    envi_convert_file_map_projection, fid=qc2_map, pos=pos_qc1,
dims=dims, o_proj=OUT_Proj, $
    o_pixel_size=[1000, 1000],out_name=out_name_qc2, warp_method=2,
r_fid=qc2_TM, $
    resampling=0, background=0 ; NOTICE fid=qc2_map!

    Close, /all, /force

    ;-------------------------------------------------------
    ; Cleaning memory
    ;envi_file_mng, id= albedo_black_map, /remove
    ;envi_file_mng, id= albedo_white_map, /remove
    ;envi_file_mng, id= albedo_qc1_map, /remove
    ;envi_file_mng, id= albedo_qc2_map, /remove

    ;-------------------------------------------------------
    ; Done with SDS, close the interface
    HDF_SD_ENDACCESS, SDSID_albedo
    HDF_SD_ENDACCESS, SDSID_qc

    HDF_SD_END, sdFileID
    HDF_Close, FileID


ENDFOR

print, "Look's O.K!!!"
Free_Lun, lun, /force

END
```