
Subject: Re: extract lat/lon from MOD13Q1
Posted by [Jelle](#) on Mon, 16 Jun 2008 17:57:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Mark,

I wrote this a while ago for MOD09 point retrieval for a student. I am not sure whether this is the final final version, but it should get you well on your way; It is a very long program, as it include the option to reproject .hdf files on the fly. I am sure there is a much easier, shorter solution, but I think if you look though the code, you can pick the pieces you need.

pro get_modis_roi_after_reproject

```
;-----  
; Setup procedures  
;-----
```

```
; set folder of idl files  
idlfiles = "D:\_current_projects\_danni\IDL"
```

```
; Is a reprojection required?  
reproject = 0    ; Set to 0 if files already reprojected, 1 to  
reproject
```

```
; Output folder for reprojected files  
reproject_dir = 'reprojected\'; Output subdir to save reprojected  
files
```

```
; Bands to return statistics for  
n_bands = 7  
bands_to_return = [0,1,2,3,4,5,6]
```

```
; Get the input directory  
; If reproject is set to 0, this is the folder from which files will  
be taken  
; to produce stats. Otherwise, this folder is the folder with files  
for reprojection.  
; The reprojected files, stored in reproject_dir will then be used for  
data-retrieval.
```

```
indir = envi_pickfile(title='Input directory to use', /directory)  
if (indir eq "") then return  
indir = indir + '\'
```

```

; Get the input coordinate list
; Structure: imagebasename space X-coordinate space Y-coordinate.
E.g., a point in australia:
; MOD09A1.h30v12.004_500m 32.95 -33.54

coordinate_file = DIALOG_PICKFILE(PATH=indir, title='Coordinate
listing file (imagename X-coordinate Y-coordinate)')
if (coordinate_file eq "") then return

; Get the filename for writing.
outfile = DIALOG_PICKFILE( /write, /OVERWRITE_PROMPT , PATH=indir,
$
    TITLE='Select output filename', DEFAULT_EXTENSION
='txt')

; Retrieve the pre-defined file template use for reading
restore, idlfiles+"coordinate_template.sav"

; Read the file to filelists
coordinates = read_ascii(coordinate_file,
template=coordinates_template)
basenames = coordinates.(0)[*]
xcoord = coordinates.(1)[*]
ycoord = coordinates.(2)[*]

;-----
; Assess files & basenames
;-----

;determine the unique images to be assessed
unique_tiles_index = uniq(basenames)

; How many?
array_test = size(unique_tiles_index)

if(array_test[0] eq 0) then begin ; Just one basename
    un_tiles = strarr(1)
    un_tiles[0] = basenames[unique_tiles_index]
    un_count = 1
endif else begin ; More then one basename
    un_tiles = strarr(array_test[3])
    un_tiles = basenames[unique_tiles_index]
    un_count[3] = array_test[3]
endelse

```

```
;-----  
; Reproject files, if needed  
;-----
```

if (reproject eq 1) then begin ; If the images need to be reprojected

```
    ; find all files associated with the basenames  
    for i = 0, un_count-1 do begin
```

```
        ; print filename, for debugging purposes  
        print, indir+'\' + un_tiles[i]+''.hdf'
```

```
    ; Get all files, ending on .hdf with the basename in them, case-  
insensitive comparison  
    filelist = file_search(indir+'*' + un_tiles[i]+''.hdf', /fold_case)
```

```
        ; How many files were found?  
        num_files = size(filelist)  
        num_files = num_files[3]
```

```
    ; For each file  
    for j = 0, num_files-1 do begin
```

```
        ; Open the input file  
        envi_open_file, filelist[j], r_fid=fid
```

```
    ; Remove the path from the filename  
    filename = SplitFileName(filelist[j])
```

```
    ; If no files are found  
    if (fid eq -1) then begin  
        envi_batch_exit  
        return  
    endif
```

```
    ; setup the new projection  
    envi_file_query, fid, ns=ns, nl=nl, nb=nb  
    pos = lindgen(nb)  
    dims = [-1, 0, ns-1, 0, nl-1]  
    out_name = indir + reproject_dir + filename  
    o_proj = envi_proj_create(/geographic, /south)
```

```

;-----
; Reproject the file to a new file
;-----
envi_convert_file_map_projection, fid=fid, $
    pos=pos, dims=dims, o_proj=o_proj, $
    out_name=out_name, warp_method=2, $
    resampling=0, background=0, /ZERO_EDGE

; Close the file
free_lun, fid

endfor ; Filename loop
endfor ; basefile loop

; Set the indir to the directory with reprojected images
indir = indir + reproject_dir

endif ; end of reprojecting the images

```

```

;-----
; All files have been reprojected, now get the pixels
;-----

basenames = coordinates.(0)[*]
xcoord    = coordinates.(1)[*]
ycoord    = coordinates.(2)[*]

;Create array to store values
Output = dblarr(2*n_bands+4)

; open output file
openw, out_lun, outfile, width=2000, /get_lun

; Create headerline:
headerline = 'Date  Basename  Xcoord  YCoord  XPix  YPix  '
for i=0, n_bands-1 do begin
    headerline = headerline + 'Mean_B'+StrTrim(i+1,2) + '  '
endfor
for i=0, n_bands-1 do begin
    headerline = headerline + 'StDev_B'+StrTrim(i+1,2) + '  '
endfor

```

```
endfor
```

```
printf, out_lun, headerline
```

```
num_basenames = n_elements(basenames)
```

```
for i=0, num_basenames-1 do begin
```

```
    ; Get all files, ending on .hdf with the basename in them, case-  
insensitive comparison
```

```
    filelist = file_search(indir+'*'+basenames[i]+'.hdf', /fold_case)
```

```
    ; How many files were found?
```

```
    num_files = size(filelist)
```

```
    num_files = num_files[3]
```

```
    ; For each file
```

```
    for j = 0, num_files-1 do begin
```

```
        ; open the file
```

```
        envi_open_file, filelist[j], r_fid=fid
```

```
        ; convert the lat-lon coordinates to pixelnumbers
```

```
        ENVI_CONVERT_FILE_COORDINATES, fid, XF, YF, xcoord[i], ycoord[i]
```

```
        Output[0] = xcoord[i]
```

```
        Output[1] = ycoord[i]
```

```
        Output[2] = floor(XF)
```

```
        Output[3] = floor(YF)
```

```
        ; set the extent of the image you want to extract
```

```
        dims = [-1, Output[2]-1, Output[2]+1, Output[3]-1, Output[3]+1]
```

```
        ; retrieve the pixels for each band
```

```
        for k=0, n_bands-1 do begin
```

```
            data = ENVI_GET_DATA(fid=fid, dims=dims, pos=k)
```

```
            Output[k+4] = mean(data)
```

```
            Output[k+4+n_bands] = stdev(data)
```

```
        endfor
```

```
        ; close the file & release the file
```

```
        close, fid
```

```
        free_lun, fid
```

```
        ; get the date from the filename
```

```
        justfilename = SplitFileName(filelist[j])
```

```
        filename_section = STRSPLIT(justfilename, "_", /extract)
```

```
    ; Write the data to file
    printf, out_lun, [filename_section[0], basenames[i],
String(Output[*], format="(f15.9)")]

    endfor

endfor

close, out_lun
free_lun, out_lun

print, 'All files processed'

end
```
