Subject: ms2gt MODIS reprojection toolkit
Posted by Maarten[1] on Tue, 27 Apr 2010 09:43:38 GMT
View Forum Message <> Reply to Message

Hi Folks,

I'm trying to set up the 'modis swath to grid toolkit' reprojection software [1]. With some other background documents [2, 3], I think I have the tools set up correctly (the verification in [1] passes), but it seems I can't get the thing to run properly. Oh, in case you're wondering why I post in this group: the code is a collision of IDL, Perl and C, with more folks over here with knowledge of map projections than in the Perl and C newsgroups.

I'd like to reproject to a Cylindrical Equidistant grid (yes, I've read [3]), for later use with other satellite data (OMI/Aura most likely). The trouble with MODIS is that it is just too much data, and I always found plotting it too hard to bother. However, with the recent eruption of Eyjafjallajökull we felt the need to combine MODIS/Aqua (RGB, aerosol) with OMI/Aura (aerosol, SO2). So, here I am, trying to get ms2gt to run, to have at least one of the instruments on an easy to visualize grid.

The Cylindrical Equidistant map projection is one of the supported projections according to the documentation, however, no matter how hard I try, I always get a message that Cylindrical Equidistant is not supported (followed by a list of supported projections which, annoyingly, includes Cylindrical Equidistant).

* Can someone supply me with a working set of configuration files to start with MODIS 1KM data (so the 250 and 500 m channels are aggregated into 1 km bins) and end up with a Cylindrical Equidistand grid?

* If someone has a suggestion on how to do this with reasonable accuracy within IDL alone, then I'm all ears.

So, with that last question I even managed to get back to the main subject of the newsgroup...

Best,

Maarten

[1] http://nsidc.org/data/modis/ms2gt/
[2] http://geospatialmethods.org/documents/ppgc.html
[3] http://nsidc.org/data/psq/grids/ece_grids.html~

Subject: Re: ms2gt MODIS reprojection toolkit
Posted by lbusett@yahoo.it on Wed, 28 Apr 2010 08:25:41 GMT
View Forum Message <> Reply to Message

On 27 Apr, 11:43, Maarten <maarten.sn...@knmi.nl> wrote:
> Hi Folks,
>
> I'm trying to set up the 'modis swath to grid toolkit' reprojection
> software [1]. With some other background documents [2, 3], I think I
> have the tools set up correctly (the verification in [1] passes), but
> it seems I can't get the thing to run properly. Oh, in case you're
> wondering why I post in this group: the code is a collision of IDL,
> Perl and C, with more folks over here with knowledge of map
> projections than in the Perl and C newsgroups.
>
> I'd like to reproject to a Cylindrical Equidistant grid (yes, I've
> read [3]), for later use with other satellite data (OMI/Aura most
> likely). The trouble with MODIS is that it is just too much data, and
> I always found plotting it too hard to bother. However, with the
> recent eruption of Eyjafjallajökull we felt the need to combine MODIS/
> Aqua (RGB, aerosol) with OMI/Aura (aerosol, SO2). So, here I am,
> trying to get ms2gt to run, to have at least one of the instruments on
> an easy to visualize grid.
>
> The Cylindrical Equidistant map projection is one of the supported
> projections according to the documentation, however, no matter how
> hard I try, I always get a message that Cylindrical Equidistant is not
> supported (followed by a list of supported projections which,
> annoyingly, includes Cylindrical Equidistant).
>
> * Can someone supply me with a working set of configuration files to
> start with MODIS 1KM data (so the 250 and 500 m channels are
> aggregated into 1 km bins) and end up with a Cylindrical Equidistand
> grid?
>
> * If someone has a suggestion on how to do this with reasonable
> accuracy within IDL alone, then I'm all ears.
>
> So, with that last question I even managed to get back to the main
> subject of the newsgroup...
>
> Best,
>
> Maarten
>
> [1]http://nsidc.org/data/modis/ms2gt/
> [2]http://geospatialmethods.org/documents/ppgc.html
> [3]http://nsidc.org/data/psq/grids/ece_grids.html~

Hi Marteen,

for reprojecting SWATH MODIS data you may also consider to use the MODIS Reprojection Tool Swath software (https://lpdaac.usgs.gov/lpdaac/tools/modis_reprojection_tool_swath).
I used it to process MODIS raw radiance data and I found it very easy to use. It can easily be called from an IDL application with a simple SPAWN command.
The supported output projections do not include the cylindrical equidistant, but I think that the equirectangular projection (which is instead available) should be equivalent to it.

Hope it helps,

Lorenzo

Subject: Re: ms2gt MODIS reprojection toolkit
Posted by Klemen on Wed, 28 Apr 2010 15:34:37 GMT
View Forum Message <> Reply to Message

Hi Maarten,

try to simply interpolate the values in IDL to a sphere. SPH_SCAT does exactly what you want but its output might be strange if you don't have the gloal coverage.

I wrote an example of using TRIANGULATE + GRIDDATA. You can rewrite it so it can fit your requirements, you probably need to calibrate the data, etc.

Cheers, Klemen

```
; modis_swath2_grid.pro
; converts swath to a regular geographical grid named
; "Equatorial Cylindrical Equidistant", "Plate-Caree", "simple
cylindrical", "lat/lon", or sometimes "unprojected"
; works for MOD021KM and MYD021KM

; in_file - input level 1B MODIS data
; dataset - "EV_1KM_Emissive", or "EV_250_Aggr1km_RefSB", or
"EV_500_Aggr1km_RefSB"
; layer_num - the corresponding number of lyer within the dataset,
e.g. 0 for band 1 at 0.6 microm, or 10 for band 31 at 11 microm
; d_resolution - spatila resolution of output (in this case decimal
degrees)
```

```
; Run as:
; a =
 modis_swath2_grid('MYD021KM.A2010107.1000.005.2010109145717. hdf',
'EV_1KM_Emissive', 10, 0.1)

; klemen.zaksek@zmaw.de, 2010


Function modis_swath2_grid, in_file, dataset, layer_num, d_resolution

 ;Tie points
 d_xL = -180. ;left
 d_xR =  180. ;right
 d_yA =   90. ;above
 d_yB =  -90. ;below

 ; Set GeoTiff geotags: http://www.remotesensing.org/geotiff/spec/contents.html
 s_geotag = {$
  MODELPIXELSCALETAG: [d_resolution, d_resolution, 0], $ ;resolution
  MODELTIEPOINTTAG: [ 0, 0, 0, d_xL, d_yA, 0], $  ;coordinates left
above
  GTMODELTYPEGEOKEY: 2, $     ;Geographic latitude-longitude System
              GTRASTERTYPEGEOKEY: 1, $    ;raster type
  GEOGRAPHICTYPEGEOKEY: 4326, $    ;geodetic datum WGS84
  GeogPrimeMeridianGeoKey: d_xL, $                    ;prime
meridian
  GEOGANGULARUNITSGEOKEY: 9102}    ;angular unit decimal degree

 ;#################################################################
##################################

 ;Open selected HDF file, read the chosen dataset, and extract the
chosen band
 i_fid = EOS_SW_OPEN(in_file, /READ)     ;open file
 if i_fid eq -1 then begin
  print, 'The input file does not exist or is not EOS HDF format!'
  GOTO, JUMP1
 endif
   i_NSwath = EOS_SW_INQSWATH(in_file, s_SwathList)
 i_swathID = EOS_SW_ATTACH(i_fid, s_SwathList)   ;attach object
 i_status_read = EOS_SW_READFIELD(i_swathID, dataset, m_modis) ;read
1000m data
 i_status_read = EOS_SW_READFIELD(i_swathID, "Latitude", m_lat) ;read
5000m latitude
 i_status_read = EOS_SW_READFIELD(i_swathID, "Longitude", m_lon) ;read
5000m longitude
 i_status_detach = EOS_SW_DETACH(i_swathID)    ;detach object
 i_status_close = EOS_SW_CLOSE(i_fid)     ;close file
```

```
 m_modis = m_modis[*,*,layer_num]     ;extract the chosen band
 in_size = size(m_modis)
 x_min = floor(min(m_lon) / d_resolution) *
d_resolution  ;interpolation borders
 x_max = ceil(max(m_lon) / d_resolution) * d_resolution
 y_min = floor(min(m_lat) / d_resolution) * d_resolution
 if y_min eq -90. then y_min = y_min + d_resolution  ;life is easier
if you do not consider data on the poles
 y_max = ceil(max(m_lat) / d_resolution) * d_resolution
 if y_max eq 90. then y_max = y_min - d_resolution
 x_pix_min = (x_min - d_xL) / d_resolution
 x_pix_max = (x_max - d_xL) / d_resolution
 y_pix_min = (d_yA - y_max) / d_resolution
 y_pix_max = (d_yA - y_min) / d_resolution

 ;################################################################
################################

 ; Average original data to "geolocation frame"
 ;first prepare indexes
 out_size = size(m_lat)     ;the output will have a reduced spatial
resolution (corresponding to the geolocation)
 ;the position 0,0 in geolocation corresponds to pixel 2,2 in original
data
 ;the geolocation is 5 times downsampled
 out_indx_col = indgen(out_size[1]) * 5L + 2L ;corresponding coloumns
of orig. data in downsampled grid
 out_indx_lin = indgen(out_size[2]) * 5L + 2L ;corresponding lines of
orig. data in downsampled grid
 out_indx_col = rebin(out_indx_col, out_size[1], out_size[2])
 out_indx_lin = rebin(reform(out_indx_lin,1,out_size[2]), out_size[1],
out_size[2])
 out_indx = out_indx_lin * in_size[1] + out_indx_col ;one dimensional
index of original data in downsampled grid

 ;compute mean value for the radiance
 m_count = make_array(out_size[1],out_size[2])  ;array containing the
number of good maeasurements
 m_mean = make_array(out_size[1],out_size[2])  ;array containing the
mean maeasurements
 for j=-2,2 do begin
  for i=-2,2 do begin
   indx = out_indx + out_size[1]*j + i
   tmp = m_modis[out_indx]
   indx_good = where(tmp le 32767)    ;do not use nodata, etc.
   m_count[indx_good] = m_count[indx_good]  + 1
   m_mean[indx_good] = m_mean[indx_good] + tmp[indx_good]
  endfor
```

```
 endfor
 m_mean = m_mean / m_count

 ;#####################################################################
################################

 ; Interpolate the data to the regular grid
 v_x = transpose(m_lon[*])     ;transform into vector
 v_y = transpose(m_lat[*])
; m_img0 = SPH_SCAT(v_x, v_y, m_mean, BOUNDS=[x_min, y_min, x_max,
y_max], GS=[d_resolution,d_resolution])
 TRIANGULATE, v_x, v_y, trg, SPHERE=sphere, /DEGREES, FVALUE=m_mean
 m_img0 = GRIDDATA(v_x, v_y, m_mean, /SPHERE, /DEGREES, /
INVERSE_DISTANCE,  TRIANGLES = trg, $
    START = [x_min, y_min], DELTA = [d_resolution,d_resolution], $
    DIMENSION = [x_pix_max-x_pix_min+1, y_pix_max-y_pix_min+1], $
    MISSING=0,  MAX_PER_SECTOR=1, SEARCH_ELLIPSE=3*d_resolution)

 ;write geotiff
 m_img0 = reverse(m_img0, 2)
 m_img = make_array((d_xR-d_xL)/d_resolution, (d_yA-d_yB)/
d_resolution)
 m_img[x_pix_min:x_pix_max,y_pix_min:y_pix_max] = m_img0
 write_tiff, in_file+'.tif', m_img, compression=1, geotiff=s_geotag, /
long

 JUMP1: print, 'END'
 return, m_img

END
```

Subject: Re: ms2gt MODIS reprojection toolkit
Posted by Maarten[1] on Wed, 28 Apr 2010 16:00:24 GMT
View Forum Message <> Reply to Message

On Apr 28, 5:34 pm, Klemen <klemen.zak...@gmail.com> wrote:
> Hi Maarten,
>
> try to simply interpolate the values in IDL to a sphere. SPH_SCAT does
> exactly what you want but its output might be strange if you don't
> have the gloal coverage.
>
> I wrote an example of using TRIANGULATE + GRIDDATA. You can rewrite it
> so it can fit your requirements, you probably need to calibrate the
> data, etc.
>
> Cheers, Klemen

Thanks for the code. Right now I'm rather busy with other stuff, but
I'll get back to this.

Best,

Maarten

Subject: Re: ms2gt MODIS reprojection toolkit
Posted by David Fanning on Wed, 28 Apr 2010 16:23:40 GMT
View Forum Message <> Reply to Message

Klemen writes:

> try to simply interpolate the values in IDL to a sphere. SPH_SCAT does
> exactly what you want but its output might be strange if you don't
> have the gloal coverage.
>
> I wrote an example of using TRIANGULATE + GRIDDATA. You can rewrite it
> so it can fit your requirements, you probably need to calibrate the
> data, etc.

Klemen,

How long does it typically take you to process a single
channel with this code? I've been waiting for about 15
minutes for the program to return. How much longer do you
think I should wait. Windows 64-bit, 6GBytes RAM. :-(

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: ms2gt MODIS reprojection toolkit
Posted by Klemen on Wed, 28 Apr 2010 16:26:53 GMT
View Forum Message <> Reply to Message

On Apr 28, 6:00 pm, Maarten <maarten.sn...@knmi.nl> wrote:
> On Apr 28, 5:34 pm, Klemen <klemen.zak...@gmail.com> wrote:

>
>> Hi Maarten,
>
>> try to simply interpolate the values in IDL to a sphere. SPH_SCAT does
>> exactly what you want but its output might be strange if you don't
>> have the gloal coverage.
>
>> I wrote an example of using TRIANGULATE + GRIDDATA. You can rewrite it
>> so it can fit your requirements, you probably need to calibrate the
>> data, etc.
>
>> Cheers, Klemen
>
> Thanks for the code. Right now I'm rather busy with other stuff, but
> I'll get back to this.
>
> Best,
>
> Maarten

Just one more thing, gridding to sphere is a few times slower than to
a grid with Cartesian coordinates. It takes me almost half a minute to
get the global geotiff with one MODIS band at 0.1 degree resolution.
Perhaps there is a faster way to do it, but that can do somebody
else. :)
Cheers, Klemen

Subject: Re: ms2gt MODIS reprojection toolkit
Posted by Klemen on Wed, 28 Apr 2010 16:35:20 GMT
View Forum Message <> Reply to Message

On Apr 28, 6:23 pm, David Fanning <n...@dfanning.com> wrote:
> Klemen writes:
>> try to simply interpolate the values in IDL to a sphere. SPH_SCAT does
>> exactly what you want but its output might be strange if you don't
>> have the gloal coverage.
>
>> I wrote an example of using TRIANGULATE + GRIDDATA. You can rewrite it
>> so it can fit your requirements, you probably need to calibrate the
>> data, etc.
>
> Klemen,
>
> How long does it typically take you to process a single
> channel with this code? I've been waiting for about 15
> minutes for the program to return. How much longer do you
> think I should wait. Windows 64-bit, 6GBytes RAM. :-(

>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

David, I just wrote Marteen that it takes less than half of minute for me (Win XP, 2Gz, 2GB RAM). What might be the problem in your case is that you have the data which have positive longitude on its left side and negative on its left side. I presume that this is your case, I had something like this when I was gridding images from Alaska using ENVI. I can add a few lines to the code to fix it, but I am about to go jogging, so I can update it in the evening. But there should be no problem if you take an image of Europe or Africa for example.
Cheers, Klemen

Subject: Re: ms2gt MODIS reprojection toolkit
Posted by Klemen on Wed, 28 Apr 2010 22:59:58 GMT
View Forum Message <> Reply to Message

Ok, this works for me, less than half a min for an output resolution of 0.1 deg. I tested it over Antartica, Mediteran and Alaska...
But Marteen, just wondering why don't you rather use Labert conformal conic projection. I also visualised ash form Island eruption using it...

```
; modis_swath2grid.pro
; converts swath to a regular geographical grid named
; "Equatorial Cylindrical Equidistant", "Plate-Caree", "simple
cylindrical", "lat/lon", or sometimes "unprojected"
; works for MOD021KM and MYD021KM

; in_file - input level 1B MODIS data
; dataset - "EV_1KM_Emissive", or "EV_250_Aggr1km_RefSB", or
"EV_500_Aggr1km_RefSB"
; layer_num - the corresponding number of lyer within the dataset,
e.g. 0 for band 1 at 0.6 microm, or 10 for band 31 at 11 microm
; d_resolution - spatila resolution of output (in this case decimal
degrees)

; Run as:
```

```
; a =  modis_swath2grid('MYD021KM.A2010107.1000.005.2010109145717.h df',
'EV_1KM_Emissive', 10, 0.1)

; klemen.zaksek@zmaw.de, 2010


Function modis_swath2grid, in_file, dataset, layer_num, d_resolution

 ;Tie points
 d_xL = -180. ;left
 d_xR =  180. ;right
 d_yA =   90. ;above
 d_yB =  -90. ;below
 prime_meridian = -180.

 ;#################################################################
##################################

 ;Open selected HDF file, read the chosen dataset, and extract the
chosen band
 i_fid = EOS_SW_OPEN(in_file, /READ)           ;open file
 if i_fid eq -1 then begin
  print, 'The input file does not exist or is not EOS HDF format!'
  GOTO, JUMP1
 endif
  i_NSwath = EOS_SW_INQSWATH(in_file, s_SwathList)
 i_swathID = EOS_SW_ATTACH(i_fid, s_SwathList)     ;attach object
 i_status_read = EOS_SW_READFIELD(i_swathID, dataset, m_modis) ;read
1000m data
 i_status_read = EOS_SW_READFIELD(i_swathID, "Latitude", m_lat) ;read
5000m latitude
 i_status_read = EOS_SW_READFIELD(i_swathID, "Longitude", m_lon) ;read
5000m longitude
 i_status_detach = EOS_SW_DETACH(i_swathID)       ;detach object
 i_status_close = EOS_SW_CLOSE(i_fid)        ;close file
 m_modis = m_modis[*,*,layer_num]         ;extract the chosen band
 in_size = size(m_modis)

 tmp = where(m_lon lt -179., count_neg)
 tmp = where(m_lon gt  179., count_pos)
 if (count_neg gt 0) and  (count_pos gt 0) then begin
  indx = where(m_lon lt 0., count_lon)
  if count_lon gt 0 then begin
   m_lon[indx] = m_lon[indx] + 360.
   prime_meridian = 0.
   d_xL = 0.
   d_xR = 360.
  endif
```

```
 endif


 x_min = floor(min(m_lon) / d_resolution) *
d_resolution   ;interpolation borders
 x_max = ceil(max(m_lon) / d_resolution) * d_resolution
 y_min = floor(min(m_lat) / d_resolution) * d_resolution
 if y_min eq -90. then y_min = y_min + d_resolution    ; life is
easier if you do not consider data on the poles
 y_max = ceil(max(m_lat) / d_resolution) * d_resolution
 if y_max eq 90. then y_max = y_min - d_resolution
 x_pix_min = round((x_min - d_xL) / d_resolution)
 x_pix_max = round((x_max - d_xL) / d_resolution) -1
print, x_pix_max
 ;if x_pix_max eq round((d_xR-d_xL)/d_resolution) then x_max = x_max -
1
 y_pix_min = round((d_yA - y_max) / d_resolution)
 y_pix_max = round((d_yA - y_min) / d_resolution)

 ;#################################################################
##################################

 ; Average original data to "geolocation frame"
 ;first prepare indexes
 out_size = size(m_lat)     ;the output will have a reduced spatial
resolution (corresponding to the geolocation)
 ;the position 0,0 in geolocation corresponds to pixel 2,2 in original
data
 ;the geolocation is 5 times downsampled
 out_indx_col = indgen(out_size[1]) * 5L + 2L ;corresponding coloumns
of orig. data in downsampled grid
 out_indx_lin = indgen(out_size[2]) * 5L + 2L ;corresponding lines of
orig. data in downsampled grid
 out_indx_col = rebin(out_indx_col, out_size[1], out_size[2])
 out_indx_lin = rebin(reform(out_indx_lin,1,out_size[2]), out_size[1],
out_size[2])
 out_indx = out_indx_lin * in_size[1] + out_indx_col ;one dimensional
index of original data in downsampled grid

 ;compute mean value for the radiance
 m_count = make_array(out_size[1],out_size[2])  ;array containing the
number of good maeasurements
 m_mean = make_array(out_size[1],out_size[2])  ;array containing the
mean maeasurements
 for j=-2,2 do begin
  for i=-2,2 do begin
   indx = out_indx + out_size[1]*j + i
   tmp = m_modis[out_indx]
```

```
   indx_good = where(tmp le 32767)    ;do not use nodata, etc.
   m_count[indx_good] = m_count[indx_good]  + 1
   m_mean[indx_good] = m_mean[indx_good] + tmp[indx_good]
  endfor
 endfor
 m_mean = m_mean / m_count

 ;#################################################################
################################

 ; Interpolate the data to the regular grid
 v_x = transpose(m_lon[*])     ;transform into vector
 v_y = transpose(m_lat[*])
; m_img0 = SPH_SCAT(v_x, v_y, m_mean, BOUNDS=[x_min, y_min, x_max,
y_max], GS=[d_resolution,d_resolution])
 TRIANGULATE, v_x, v_y, trg, SPHERE=sphere, /DEGREES, FVALUE=m_mean
 m_img0 = GRIDDATA(v_x, v_y, m_mean, /SPHERE, /DEGREES, /
INVERSE_DISTANCE,  TRIANGLES = trg, $
    START = [x_min, y_min], DELTA = [d_resolution,d_resolution], $
    DIMENSION = [x_pix_max-x_pix_min+1, y_pix_max-y_pix_min+1], $
    MISSING=0,  MAX_PER_SECTOR=1, SEARCH_ELLIPSE=3*d_resolution)

 ; Set GeoTiff geotags: http://www.remotesensing.org/geotiff/spec/contents.html
 s_geotag = {$
  MODELPIXELSCALETAG: [d_resolution, d_resolution, 0], $  ;resolution
  MODELTIEPOINTTAG: [ 0, 0, 0, d_xL, d_yA, 0], $    ;coordinates left
above
  GTMODELTYPEGEOKEY: 2, $      ;Geographic latitude-longitude System
    GTRASTERTYPEGEOKEY: 1, $     ;raster type
  GEOGRAPHICTYPEGEOKEY: 4326, $    ;geodetic datum WGS84
  GeogPrimeMeridianGeoKey: prime_meridian, $ ;prime meridian
  GEOGANGULARUNITSGEOKEY: 9102}    ;angular unit decimal degree

 ;write geotiff
 m_img0 = reverse(m_img0, 2)
 print, size(m_img0), x_pix_max -x_pix_min, y_pix_max -y_pix_min
 m_img = make_array((d_xR-d_xL)/d_resolution, (d_yA-d_yB)/
d_resolution)
 m_img[x_pix_min:x_pix_max,y_pix_min:y_pix_max] = m_img0
 write_tiff, in_file+'.tif', m_img, compression=1, geotiff=s_geotag, /
long

 JUMP1: print, 'END'
 return, m_img

END
```

Subject: Re: ms2gt MODIS reprojection toolkit
Posted by Maarten[1] on Thu, 29 Apr 2010 08:07:14 GMT
View Forum Message <> Reply to Message

On Apr 29, 12:59 am, Klemen <klemen.zak...@gmail.com> wrote:
> Ok, this works for me, less than half a min for an output resolution
> of 0.1 deg. I tested it over Antartica, Mediteran and Alaska...
> But Marteen, just wondering why don't you rather use Lambert conformal
> conic projection. I also visualised ash form Island eruption using
> it...

I have several reasons for this:
* I don't know where the next interesting eruption will take place.
Unprojected will allow for another reprojection later on (but this
time completely within IDL, Python, ...) without special coding.
* I need to combine several instruments, and using unprojected data as
an intermediate for hte instrument with the highest spatial resolution
helps me here.
* I may want to plot the final result with a different tool
altogether.

Best,

Maarten

Subject: Re: ms2gt MODIS reprojection toolkit
Posted by Klemen on Thu, 29 Apr 2010 09:00:30 GMT
View Forum Message <> Reply to Message

I see your point. Just about the resolution. In your first post you
mentioned that you want to compare MODIS and OMi, so I just aggregated
MODIS data to 5km resolution and then do the final gridding. In this
case it makes no sense to you a better resolution than 0.05 deg.

If you need more, you should use original data and not aggregated data
for the gridding but this will be really slow. I have no idea, how
fast ms2gt is. I always prefer do do it myself in IDL. But also
because of the processing speed I prefer to use some output with
metric Cartesian coordinates (gridding is in this case much faster and
further spatial analysis as e.g. area computation are more accurate).

Klemen

Subject: Re: ms2gt MODIS reprojection toolkit
Posted by David Fanning on Thu, 29 Apr 2010 12:02:48 GMT

Klemen writes:

> If you need more, you should use original data and not aggregated data
> for the gridding but this will be really slow

I was using original data. I started it before I went
to play tennis for three hours last night. It wasn't
done by the time I got home. It *was* finished by the
time I got up for my coffee this morning. :-)

> I have no idea, how fast ms2gt is.

Well, let's just say, by comparison, it is *blazing*
fast! Never more than a minute or so in my experience.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")