
Subject: Need help with map projection conversion in ENVI

Posted by [BLesht](#) on Sun, 23 Jan 2011 21:21:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

I've been fighting with this problem for weeks and have finally given up on solving it myself. I hope that someone with more experience manipulating maps in ENVI can offer me some advice and guidance.

I have an image file (actually many such files) that is written as an hdf. The data in the file have been geo-referenced and projected. Some of the projection parameters (Mercator, limit, central azimuth) are included in the file as attributes of the scientific data set, but there are no explicit latitude and longitude values corresponding to the pixels in the image in the file. I know the size of the image, the datum that was used in the projection, and I think I have the affine transform matrix for the map. My goal is to convert this image from its native Mercator projection to a data file that is in geographic coordinates with a different datum.

I can use ENVI_PROJ_CREATE to create a map projection structure for the input file - for Mercator projection the parameter array includes the semi-major and semi-minor axes (obtained from knowledge of the original datum/ellipsoid), the latitude at the origin (assumed to be the equator), the longitude of the central meridian (included in the sds attribute information), and the false easting and false northing. I've used the e and f elements of the affine transform matrix for these. I also can use ENVI_PROJ_CREATE to define the output projection as geographic with the different datum.

I thought the way to approach this was to read the hdf file and then create an ENVI file in memory using ENVI_WRITE_ENVI_FILE. I thought I could simply use ENVI_CONVERT_FILE_MAP_PROJECTION to create the new file I wanted. To do, however, I needed to use ENVI_MAP_INFO_CREATE to develop a map information structure to associate with the data that I would write to the temporary ENVI file. This is where I became very confused. In addition to the input map projection, ENVI_MAP_INFO_CREATE requires a tie point (MC) array and pixel size (PS) array. I think I have enough information to fill the MC array because I know the limits of the original image from the hdf metadata, though these values are given as latitudes and longitudes rather than as meters. I'm not sure about the PS array. Because the original image was generated as ISOTROPIC, I am assuming that pixel size should be the same in each direction. Is it to be specified in angular units or linear units? It can't be angular because scale along a meridian varies with latitude in a Mercator projection. If it is linear, how do I determine the values? Are these the same as the b and c elements of the affine matrix?

Pushing on, I made some assumptions and went to the ENVI_CONVERT_FILE_MAP_PROJECTION step. I originally thought that I could write the output file into memory, but apparently, the /IN_MEMORY keyword is not allowed in this routine (seems strange..), so I use the OUT_NAME keyword. The process fails at this step - when I try to query the file I tried to write, the ENVI_QUERY_FILE routine returns -1.

Here is part of the code and the output:

```
oname=/temp.img'
ENVI_WRITE_ENVI_FILE, data, DATA_TYPE=data_type, /NO_OPEN, $
  MAP_INFO=in_map, NB=1, NL=1024, NS=1024, /IN_MEMORY, R_FID=rfid
;
foo = ENVI_GET_PROJECTION(FID=rfid)
PRINT, foo
;
ENVI_CONVERT_FILE_MAP_PROJECTION, FID=rfid, OUT_NAME=oname, POS=[0],
R_FID=ofid, DIMS=dims, O_PROJ=OUT_proj, WARP_METHOD=3
;
ENVI_OPEN_FILE, oname, R_FID=ofid
PRINT, ofid

PROJCS["Mercator",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984
",SPHEROID["WGS_1984",
6378137.0,298.257223563]],PRIMEM["Greenwich",0.0],UNIT["Degree ",
0.0174532925199433]],PROJECTION["Mercator"],PARAMETER["False_Easting
",-10187121.96],PARAMETER["False_Northing",
6518274.16],PARAMETER["Central_Meridian",-84.14],PARAMETER[ "Standard_Parallel_1",
0.0],UNIT["Degree",0.0174532925199433]]
    0}
-1
```

I've been going back and forth on this with no success. Any suggestions would be most welcome.
