
Subject: !x/y.window & convert_coord giving unwanted offset to my image scale
Posted by [micofelicio](#) on Thu, 23 Jul 2015 11:50:36 GMT

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

Greetings!

I'm working on turning data arrays into geoTiff files. Basically, I have 3 arrays: 1 for my actual data, and 1 for latitude and 1 longitude both corresponding to a single point in my data array. To do this, one of my procedures involves projecting my data array into, well, my desired projection. I found this code called `imagemap` by Liam. I used this as the basis for my code. Here's what I do:

```
;set/get size of size of desired output image
ns = !d.x_size
nl = !d.y_size

;create a base map projection
map_set, /mercator, limit = [ latmin, lonmin, latmax, lonmax ]

;convert my lat-lon variables from data space to normal space
p = convert_coord( lon, lat, /data, /to_normal)

;scale my data array and assign it to a new array "newimage". (newimage is the output)
newimage( p( 0, * ) * ( ns - 1 ), p( 1, * ) * ( nl - 1 ) ) = bytscl( image, min = imin, max = imax, top =
!d.table_size - 2 ) + 1B
```

This code produce a projected image as desired. I open the images (and I also have a shape file for reference) in ENVI to check if my geotiff is in the right size, scale and place. Apparently, the output image I get is too small and is way off of my desired size and scale. Soooo, I set ns and nl to my desired size since !d.x_size and !d.y_size were too small. Here's what I did:

```
;Create GLT file from my lat-lon array
ENVI_DOIT, 'ENVI_GLT_DOIT', DIMS=dim, I_PROJ=i_proj, O_PROJ=i_proj,
out_name='glt_150501', R_FID=glt_fid,rotation=0,X_FID=night_lon_id,X_POS=0,Y_FID=ni
ght_lat_id,Y_POS=0
```

```
;Extract the number of samples and lines from the GLT
ENVI_FILE_QUERY,glt_fid,ns=ns,nl=nl
```

This enlarged my image to my desired size but the scale is still wrong. There seems to be a minor offset. My data doesn't reach the edges of my image as I'd expected. The !x.window and !y.window changes when I use `map_set`. They weren't set to the lowest and highest values of the normal space.

This is where I think the problem occurs. When I call `convert_coord` and check the p-variable, my max/min lat-lon values aren't set to 0/1 (like in normal space). Instead, they are set to:

```
!x.window(0) is 0.0166717
!x.window(1) is 0.983338
```

```
!y.window(0) is 0.274023
!y.window(1) is 0.945210
```

This lead me to think that !x and !y variables had something to do with it. The size of my desired output image is 4775x3525. Using the information from !x/y.window, I get 4626x2365. When I trim the offset, I get an image of 4617x3236. This is close to my calculated window sizes (at least for the number of samples).

So this is what I did:

```
;I set the values equal to the maximum values I can get from the normal space: 0 and 1
!x.window(0) = 0
!x.window(1) = 1
!y.window(0) = 0
!y.window(1) = 1
;Apparently this does nothing.
```

That did NOTHING. It had no effect on my code so I tried to make my own simple convert_coord substitute. (This is what my present code looks like right now:

```
;;;;;BEGIN;;;;;
```

```
;Create GLT file from my lat-lon array
ENVI_DOIT, 'ENVI_GLT_DOIT', DIMS=dim, I_PROJ=i_proj, O_PROJ=i_proj,
out_name='glt_150501', R_FID=glt_fid,rotation=0,X_FID=night_lon_id,X_POS=0,Y_FID=ni
ght_lat_id,Y_POS=0
```

```
;Extract the number of samples and lines from the GLT (This turned out to have the exact size I
wanted)
```

```
ENVI_FILE_QUERY,glt_fid,ns=ns,nl=nl
```

```
!x.window(0) = 0
!x.window(1) = 1
!y.window(0) = 0
!y.window(1) = 1
```

```
;create a base map projection
```

```
map_set, /mercator, limit = [ latmin, lonmin, latmax, lonmax ]
```

```
;;;Convert_coord function attempt
```

```
;;;Just a bunch of variables I used to hasten my conversion process
diffLon = lonmax - lonmin
diffLat = latmax - latmin
sizes = size(lon)
lenX = sizes(1)
lenY = sizes(2)
print,lenX,lenY
```

```
p = fltarr(3,lenX*lenY)
```

```
;;;;;loop that converts my lat-lon array to normal space and assigns to variable p  
for i=0, lenX-1 do begin  
  for j=0, lenY-1 do begin
```

```
    ;formula for properly assigning my the correct index to p  
    index = i*lenY + j
```

```
    ; This was the scaling formula I used to convert my data space to normal space  
    p(0,index) = (lon(i,j) - lonmin) / diffLon  
    p(1,index) = (lat(i,j) - latmin) / diffLat  
    p(2,index) = 0
```

```
  endfor  
endfor
```

```
;scale my data array and assign it to a new array "newimage"  
newimage( p( 0, * ) * ( ns - 1 ), p( 1, * ) * ( nl - 1 ) ) = bytscl( image, min = imin, max = imax, top =  
!d.table_size - 2 ) + 1B
```

So what happened? I got my data to fit the extents of my image like I wanted BUT my data looks like a bunch of static. Any tips on how to work around my problem?

Thanks in advance!

-mico
