
Subject: Re: shifted .kml created with IDL
Posted by [titan](#) on Thu, 02 Aug 2012 07:00:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tuesday, July 31, 2012 6:36:37 PM UTC+2, David Fanning wrote:

> David Fanning writes:

>
>
>

>> I'm not sure if these are the *centers* of the corner

>

>> pixels or the outside edge. I would probably get this

>

>> information directly from the GeoTiff file. If you did,

>

>> you would be getting the outside edge boundaries.

>

>>

>

>> In other words, you can get this information from the

>

>> GeoTiff structure using tie points, the scale in the

>

>> X and Y direction, and the size of the image. All info

>

>> that is available for the GeoTiff file:

>

>>

>

>> http://www.idlcoyote.com/map_tips/pixel_to_ll.html

>

>

>

> All this boundary information is more easily obtained

>

> with the Coyote Library routine FindMapBoundary, of course:

>

>

>

> void = FindMapBoundary(geoTiffFile, b, X RANGE=xr, Y RANGE=yr)

>

>

>

> The boundary is in the variable b. Or you can use the ranges.

>

> Same numbers, just presented differently, depending on what

>

> you need.

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

Hi David,

I tried to collect all your useful suggestions to get the rid of this problem but I think I'm still missing something and I'm getting confused :((Do I need holiday???)

I had two approaches:

1. I'm calculating the center of the corner pixels and from this (using http://www.idlcoyote.com/map_tips/pixel_to_ll.html) I calculated the lat and lon for all pixel of the image in the Google Earth projection

```
tiff_image=path_fname+tiff_image
```

```
;mapCoord = cgGeoMap(tiff_image)
```

```
;geoTiffMapStruct = mapCoord -> GetMapStruct()
```

```
geMapStruct = Map_Proj_Init('Mercator', /GCTP, Datum=8)
```

```
image=Read_Tiff(tiff_image, GEOTIFF=geotag)
```

```
image = Reverse(image, 2) ; Necessary for TIFF files.
```

```
s=Size(image, /Dimensions)
```

```
xscale = geotag.ModelPixelScaleTag[0]
```

```
yscale = geotag.ModelPixelScaleTag[1]
```

```
tp = geotag.ModelTiePointTag[3:4]
```

```
xOrigin = tp[0]
```

```

yOrigin = tp[1] - (yscale * s[1])
xEnd = xOrigin + (xscale * s[0])
yEnd = tp[1]
xCenter = (xEnd - xOrigin) / 2.0 + xOrigin
yCenter = (yEnd - yOrigin) / 2.0 + yOrigin

```

```

uvec = Findgen(s[0]) * xscale + xOrigin
vvec = Findgen(s[1]) * yscale + yOrigin

```

```

uarray = Rebin(uvec, s[0], s[1])
varray = Rebin(Reform(vvec, 1, s[1]), s[0], s[1])

```

```

lonlat_ge= MAP_PROJ_INVERSE(uarray, varray, MAP_STRUCTURE=geMapStruct)
minLat_ge = Min(lonlat_ge[1,*], MAX=maxLat_ge)
minLon_ge = Min(lonlat_ge[0,*], MAX=maxLon_ge)
limit_ge = [minLat_ge, minLon_ge, maxLat_ge, maxLon_ge]

```

```

print, limit_ge
39.646844      6.1113937      42.130700      6.1182209

```

2. Then I tried the FindMapBoundary function to get the border of my tiff image and then using in a correct way the map_proj_image function. After reading the tiff file and creating a warped image using the Google Earth map structure I convert the uvrage in lat/lon to be passed to my kml.

```

void = FindMapBoundary(tiff_image, b, X RANGE=xr, Y RANGE=yr)
print, 'boundary ', b
print, 'X RANGE ',xr
print, 'Y RANGE ',yr

```

```

tiffCoord = cgGeoMap(tiff_image)

```

```

tiffCoord -> SetProperty, X RANGE=xr, Y RANGE=yr

```

```

MercatorCoord = Obj_New('cgMap', "MERCATOR", DATUM=8)

```

```

warped_r = Map_Proj_Image(REFORM(image[0,*,*]), b, $
    IMAGE_STRUCTURE=tiffCoord->GetMapStruct(),
MAP_STRUCTURE=MercatorCoord->GetMapStruct(), $
    UVRANGE=uvout)

```

```

warped_g = Map_Proj_Image(REFORM(image[1,*,*]), b, $
    IMAGE_STRUCTURE=tiffCoord->GetMapStruct(),
MAP_STRUCTURE=MercatorCoord->GetMapStruct(), $
    UVRANGE=uvout)

```

```

warped_b = Map_Proj_Image(REFORM(image[2,*,*]), b, $
    IMAGE_STRUCTURE=tiffCoord->GetMapStruct(),
MAP_STRUCTURE=MercatorCoord->GetMapStruct(), $
    UVRANGE=uvout)
print,'uvout'
print,uvout

s=SIZE(warped_r, /DIMENSIONS)
warped = BytArr(3, s[0], s[1])
warped[0, *, *] = ROTATE(warped_r,7)
warped[1, *, *] = ROTATE(warped_g,7)
warped[2, *, *] = ROTATE(warped_b,7)

WRITE_PNG, path_fname+'tiff_warped2mercator.png', warped

lonlat_kml= MAP_PROJ_INVERSE(uvout, MAP_STRUCTURE=geMapStruct)

boundary      680317.23      4878686.4      1045117.2      5152286.4
XRange        680317.23      1045117.2
YRange        4878686.4      5152286.4

uvout
      1252403.4      5414120.5      1789982.0      5829831.8

lonlat_kml
      11.250532      43.859578
      16.079682      46.500001

```

The problem is that I expected coordinate near to these

```

ul_lat = 47.653348
ul_lon = 8.9734384
lr_lat = 45.292876
lr_lon = 13.830713

```

which are the corners of the area in which I'm interest.

Am I still doing everything wrong?? :(

Thanks for your support!!