
Subject: Re: Map to data coordinate conversion
Posted by [PMan](#) on Thu, 18 Apr 2013 15:15:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thursday, April 18, 2013 10:40:03 AM UTC-4, David Fanning wrote:

> Paul Mallas writes:

>

>

>

>>

>

>> Hello all,

>

>>

>

>> I am looking to convert, given a geotiff image with map info, the lat/lon data to a pixel location in an image.

>

>>

>

>> What I want to do seems pretty easy, but I am having a hard time coding it in IDL for some reason. I must be missing something.

>

>>

>

>> Using the ENVI() API, this was straight forward:

>

>>

>

>> oEnvi = envi(/headless)

>

>> eMask = oEnvi->openRaster(file)

>

>> fid = ENVIRasterToFID(eMask)

>

>> envi_convert_file_coordinates, fid, xf, yf, lon, lat

>

>>

>

>> with lon, lat being my predefined input and xf, yf being my desired output.

>

>>

>

>> The analogous way to do this in IDL is (I think):

>

>>

>

>> img = image(file, /buffer)

```
>
>> xy = img->convertcoord(lon, lat, /to_data)
>
>>
>
>> however the image() function won't read the geotiff properly (but ENVI() accepts it just fine).
Perhaps my geotiff tag is not properly formed or something. Ideas on another way?
>
>
>
> I don't know how to do this in Function Graphics, but the general idea
>
> is to have an X and Y vector with the same dimensions as the image,
>
> scaled into the endpoints of the XY projected meter grid. To find a
>
> point in lat/lon space, you inverse transform these vectors to lat/lon
>
> with your map projection, then locate the specified point in the vectors
>
> with Value_Locate. The indices returned from Value_Locate are the pixel
>
> values.
>
>
>
> Cheers,
>
>
> David
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")
```

Thanks for the reply - I can calculate the positions
(with the spatial reference from the geotiff tags):

```
xf = (lon - spatialref.tie_point_map[0])/(spatialref.pixel_size[0])
```

```
yf = (spatialref.tie_point_map[1] - lat)/(spatialref.pixel_size[1])
```

which works. But I felt there was probably some IDL functionality somewhere I was missing - I hate that feeling. :)

Regards
