Subject: Re: Map to data coordinate conversion
Posted by David Fanning on Thu, 18 Apr 2013 14:40:03 GMT
View Forum Message <> Reply to Message

Paul Mallas writes:

>
> Hello all,
>
> I am looking to convert, given a geotff 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.")

Subject: Re: Map to data coordinate conversion
Posted by chris_torrence@NOSPAM on Thu, 18 Apr 2013 15:12:16 GMT
View Forum Message <> Reply to Message

On Thursday, April 18, 2013 8:19:48 AM UTC-6, Paul Mallas wrote:

```
>  Hello all,
>
>
>
>  I am looking to convert, given a geotff 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?

Hi Paul,

I think you want to use the MapForward method. Something like this:

```
i = image(FILEPATH('boulder.tif',SUBDIR=['examples','data']))
print, i.mapforward(-105,40)
```

The ConvertCoord is just for converting between screen/normalized/data coordinates, and not between lat/lon and data.

Hope this helps.
-Chris
ExelisVIS

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 geotff 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

Subject: Re: Map to data coordinate conversion
Posted by Fabzi on Thu, 18 Apr 2013 15:20:40 GMT

View Forum Message <> Reply to Message

On 04/18/2013 05:12 PM, Chris Torrence wrote:
> i = image(FILEPATH('boulder.tif',SUBDIR=['examples','data']))
> print, i.mapforward(-105,40)

Wait. Does that mean that IDL now supports geotiff projections?

How do I get the Geotiff projection parameters without image() ?
Can I get a map structure?

Something like:

```
g = geotiff(FILEPATH('boulder.tif',SUBDIR=['examples','data']))

map = g->getMap()
truelat = g->getTrueLat()
```

etc.

???

Subject: Re: Map to data coordinate conversion
Posted by PMan on Thu, 18 Apr 2013 15:27:57 GMT
View Forum Message <> Reply to Message

On Thursday, April 18, 2013 11:12:16 AM UTC-4, Chris Torrence wrote:
> On Thursday, April 18, 2013 8:19:48 AM UTC-6, Paul Mallas wrote:
>
>> Hello all,
>
>>
>
>>
>
>>
>
>> I am looking to convert, given a geotff 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?

>
>
>
> Hi Paul,
>
>
>
> I think you want to use the MapForward method. Something like this:
>
>
>
> i = image(FILEPATH('boulder.tif',SUBDIR=['examples','data']))
>
> print, i.mapforward(-105,40)
>
>
>
> The ConvertCoord is just for converting between screen/normalized/data coordinates, and not between lat/lon and data.
>
>
>
> Hope this helps.
>
> -Chris
>
> ExelisVIS

Great - that's what I am thinking of.

I am encountering a problem with image() though - my geotiff won't load with correctly. I get this error:
IMAGE: Unable to invoke method on NULL object reference: <OBJREF (<NullObject>)>.

This same image works just fine with the ENVI API - perhaps there is some deficiency in my geotiff values.

Subject: Re: Map to data coordinate conversion
Posted by chris_torrence@NOSPAM on Fri, 19 Apr 2013 23:06:38 GMT
View Forum Message <> Reply to Message

Hi Fabien & Paul,

The IDL "query_tiff" routine has had a geotiff keyword for a long time. It just returns a structure of key/value pairs. The Image() routine has also supported GeoTIFF images since IDL 8.0.

Paul, is there some way I can get to your geotiff image, so I can try it out? Also, which exact version of IDL are you using?

Finally, if you can't send the image, could you try calling Image with the /DEBUG keyword, to get the stack trace where the error occurs?

-Chris
ExelisVIS

Subject: Re: Map to data coordinate conversion
Posted by David Fanning on Mon, 22 Apr 2013 12:20:16 GMT

View Forum Message <> Reply to Message

Fabien writes:

```
>
>  On 04/18/2013 05:12 PM, Chris Torrence wrote:
>>  i = image(FILEPATH('boulder.tif',SUBDIR=['examples','data']))
>>  print, i.mapforward(-105,40)
>
>  Wait. Does that mean that IDL now supports geotiff projections?
>
>  How do I get the Geotiff projection parameters without image() ?
>  Can I get a map structure?
>
>  Something like:
>  g = geotiff(FILEPATH('boulder.tif',SUBDIR=['examples','data']))
>
>  map = g->getMap()
>  truelat = g->getTrueLat()
>
>  etc.
>
>  ???
```

Something like this has been available with cgImage for quite some time,
too.

```
   IDL> file = 'AF03sep15b.n16-VIg.tif'
   IDL> cgImage, Filename=file, MapCoord=map
   IDL> Help, map
        MAP     OBJREF    = <ObjHeapVar11(CGMAP)>
```

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.")