

---

Subject: Re: Finding pixel values of GeoTIFF image based on lat/lon (ENVI and IDL give different results).

Posted by [DavidF\[1\]](#) on Mon, 05 Nov 2012 18:15:41 GMT

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

---

Sobriquet writes:

> I have ~700 GeoTIFF images of dimensions around 4000x4000. I have the lat/long coordinates of 9 points from which I need to extract information. The images are projected on the Albers Conical Equal Area projection. I need to find the closest pixel value (sample, line image coordinates) that matches the lat/lon pairs. I previously used the approach described here: [http://www.idlcoyote.com/map\\_tips/pixel\\_to\\_ll.html](http://www.idlcoyote.com/map_tips/pixel_to_ll.html)

>

> However, I was unable to find a matching pair of lat/long in both arrays or convert the closest match between the two to pixel coordinates.

Humm. I'm not sure you are following the approach described in that article, because I see no evidence in your code that you are reversing your TIFF image in Y, etc., etc.

If I were going to do this I *would* follow the approach in the article, exactly, up to the point where I had the two vectors uvec and vvec. Then, I would convert the point (presumably in lat/lon) you want to find into projected meter values and find the image index with Value\_Locate, like this:

```
pt_lon = -148.23300
```

```
pt_lat = 64.7000
```

```
xy = Map_Proj_Forward(pt_lon, pt_lat, Map_Structure=Albermap)
```

```
pt_x = xy[0]
```

```
pt_y = xy[1]
```

```
xindex = Value_Locate(uvec, pt_x)
```

```
yindex = Value_Locate(vvec, pt_y)
```

```
Print, 'Image Value: ', image[xindex,yindex]
```

```
ll = Map_Proj_Inverse(uvec[xindex], vvec[yindex], Map_Structure=Albermap)
```

```
Print, 'Nearest Pixel Location (lon/lat): '
```

```
Print, ' Longitude: ', ll[0], ' Image X Coord: ', xindex
```

```
Print, ' Latitude: ', ll[1], ' Image Y Coord: ', yindex
```

I wouldn't use the SPHERE\_RADIUS keyword in Map\_Proj\_Init either. I can't tell if it is hurting you, but it can't be doing you any good. :-)

Cheers,

David

---

---

Subject: Re: Finding pixel values of GeoTIFF image based on lat/lon (ENVI and IDL give different results).

Posted by [Sobriquet](#) on Tue, 06 Nov 2012 01:05:02 GMT

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

---

David,

Thanks for the reply.

Your suggestion to remove the SPHERE\_RADIUS keyword was accurate: removing it makes no difference.

Here is what I got:

Image Value: 3085.77  
Nearest Pixel Location (lon/lat):  
Longitude: -148.23405 Image X Coord: 805  
Latitude: 64.699193 Image Y Coord: 1662

What's interesting is that when I input the lat/lons in ENVI it approximates it to the same values that I get when I enter my original lat/lons (64.7000, -148.23300 // 3122). However, when I input the x,y coordinates into ENVI, I get this

(65.4425, -148.09711726 // 3552.166016),

which is very far from my original location. In fact, when I enter the nearest pixel location lat/lons into Google Earth, it lands ~85km away from my input lat/long.

Now, I know ENVI is not perfect. A quick survey on Google Earth also shows that the approximation of the location I entered is about 100 m off.

I think the problem is that since the resolution of the arrays is very high (100 m), there are a lot of possible matches in the position vectors. Value\_Locate is searching through the u and v vectors individually and finding the closest match for each value without taking into account the relationship between the two vectors, i.e., it is not finding the closest u match in relation to the v match that together make the nearest point possible to the input values. As a result, you might get very close u and v individual matches for your input lat/lon that translate into a very off pair of sample/line values in your pixel coordinates.

I hate to be a stickler, but is there a way to establish a relationship between the two position vectors that would help solve the ambiguity?

Thanks again

---

---

Subject: Re: Finding pixel values of GeoTIFF image based on lat/lon (ENVI and IDL give different results).

Posted by [David Fanning](#) on Tue, 06 Nov 2012 01:48:47 GMT

Sobriquet writes:

> Here is what I got:

>

> Image Value: 3085.77

> Nearest Pixel Location (lon/lat):

> Longitude: -148.23405 Image X Coord: 805

> Latitude: 64.699193 Image Y Coord: 1662

>

> What's interesting is that when I input the lat/lons in ENVI it approximates it to the same values that I get when I enter my original lat/lons (64.7000, -148.23300 // 3122). However, when I input the x,y coordinates into ENVI, I get this

>

> (65.4425, -148.09711726 // 3552.166016),

>

> which is very far from my original location. In fact, when I enter the nearest pixel location lat/lons into Google Earth, it lands ~85km away from my input lat/long.

Well, ENVI, remember, uses a different coordinate system than IDL does. Theirs starts at 1, IDL's starts at 0.

> Now, I know ENVI is not perfect. A quick survey on Google Earth also shows that the approximation of the location I entered is about 100 m off.

Did you convert your coordinate from an Albers projection to the Equirectangular projection used by Google Earth? If not, this can account for differences of at least 100 meters, depending upon where you are on the Earth.

> I think the problem is that since the resolution of the arrays is very high (100 m), there are a lot of possible matches in the position vectors. Value\_Locate is searching through the u and v vectors individually and finding the closest match for each value without taking into account the relationship between the two vectors, i.e., it is not finding the closest u match in relation to the v match that together make the nearest point possible to the input values. As a result, you might get very close u and v individual matches for your input lat/lon that translate into a very off pair of sample/line values in your pixel coordinates.

On the contrary, there is only one possible match on this grid. It is true that the actual location of the grid crossing point can be off from the true location of your point by at something like 100 meters (if this is the size of the grid), but there are not "multiple" matches. There is one possible match to a particular point.

> I hate to be a stickler, but is there a way to establish a relationship between the two position vectors that would help solve the ambiguity?

Value\_Locate is finding the index of the value that is closest to your

chosen point that doesn't exceed your point value. You could subtract the grid value from your point, and if it is larger than half a grid size, you could say the next index is closer than the one Value\_Locate found.

```
xindex = Value_Locate(uvec, pt_x)
IF pt_x = uvec[xindex] > (uvec[1]-uvec[0])/2 THEN xindex = xindex + 1
```

I guess it depends on how anal you want to be. :-)

Cheers,

David

--

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

---

---

Subject: Re: Finding pixel values of GeoTIFF image based on lat/lon (ENVI and IDL give different results).

Posted by [David Fanning](#) on Tue, 06 Nov 2012 01:53:24 GMT

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

---

David Fanning writes:

```
> xindex = Value_Locate(uvec, pt_x)
> IF pt_x = uvec[xindex] > (uvec[1]-uvec[0])/2 THEN xindex = xindex + 1
```

Whoops!

Shouldn't write code with one hand on a burrito! :-(

```
xindex = Value_Locate(uvec, pt_x)
IF (pt_x - uvec[xindex]) GT ((uvec[1]-uvec[0])/2.0) THEN $
    xindex = xindex + 1
```

Cheers,

David

--

David Fanning, Ph.D.  
Fanning Software Consulting, Inc.  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thue. ("Perhaps thos speakest truth.")

---

---

Subject: Re: Finding pixel values of GeoTIFF image based on lat/lon (ENVI and IDL give different results).

Posted by [Sobriquet](#) on Wed, 07 Nov 2012 16:57:32 GMT

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

---

Thanks for the help!

I found that indeed the subtraction between the grid value from my point was more than 50 (~90).  
I'll go with the next pixel over.

---