
Subject: Latitude longitude and Image Navigation

Posted by [AISHWARYA](#) on Wed, 22 Jun 2011 04:41:01 GMT

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

Can someone help me on Navigating the Images in IDL? I have no idea as where to start with IDL on Image Navigation. The images were acquired from telescope and target was Venus. How can I make a latitude longitude grid for venus knowing the North pole position angle and north pole position distance?

Any help would be great!

Thank you,
Aishwarya.

Subject: Re: Latitude longitude and Image Navigation

Posted by [Brian Wolven](#) on Sat, 25 Jun 2011 04:03:19 GMT

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

Well, *if* I remember correctly what I did there: Subcur is the vector of the initial sub-observer point on the planet/satellite before the rotations, given a coordinate system defined with respect to the center of that body. It's just the normalized planetocentric coordinates of the vector pointing toward the observer:

```
subcur = -1.*body.pos/length(body.pos)
```

The -1 comes in because body.pos contains the vector from the observer location (Earth in this case) to the body. subdes is the desired sub-observer location - the one specified in the image metadata or by SPICE, based on the observation time. We need to rotate our body centered coordinates such that the z-axis aligns with the specified north pole position vector, then rotate the body about the pole until our line of sight is passing through the proper sub-observer point. The various 'ang' variables were only used for diagnostic purposes to check that the desired result was being obtained.

To put it in more graphic terms, imagine that you have a globe of Venus sitting upright on your desk. In order to match your telescopic observations you need to tilt the globe until the pole axis is aligned properly, then rotate about the pole until you reach the correct observing geometry. We have a body centered lat-long grid instead of a globe, but once we know the rotation we can then project our rotated lat-long grid on the image and have it properly aligned, or infer the lat-long coordinates of a pixel in the image if we're working the other way.

There may very well be an easier way to do this, but that's what I cooked up years ago as a grad student, when presumably I was smarter than I am now. ;)

Subject: Re: Latitude longitude and Image Navigation

On Jun 25, 9:03 am, Brian Wolven <brian.wol...@gmail.com> wrote:

> Well, *if* I remember correctly what I did there: Subcur is the vector of the initial sub-observer point on the planet/satellite before the rotations, given a coordinate system defined with respect to the center of that body. It's just the normalized planetocentric coordinates of the vector pointing toward the observer:

>

> subcur = -1.*body.pos/length(body.pos)

>

> The -1 comes in because body.pos contains the vector from the observer location (Earth in this case) to the body. subdes is the desired sub-observer location - the one specified in theimagemetadata or by SPICE, based on the observation time. We need to rotate our body centered coordinates such that the z-axis aligns with the specified north pole position vector, then rotate the body about the pole until our line of sight is passing through the proper sub-observer point. The various 'ang' variables were only used for diagnostic purposes to check that the desired result was being obtained.

>

> To put it in more graphic terms, imagine that you have a globe of Venus sitting upright on your desk. In order to match your telescopic observations you need to tilt the globe until the pole axis is aligned properly, then rotate about the pole until you reach the correct observing geometry. We have a body centered lat-long grid instead of a globe, but once we know the rotation we can then project our rotated lat-long grid on theimageand have it properly aligned, or infer the lat-long coordinates of a pixel in theimageif we're working the other way.

>

> There may very well be an easier way to do this, but that's what I cooked up years ago as a grad student, when presumably I was smarter than I am now. ;)

I could partly understand your code. But, I have a few doubts. What we have in hand is a 2d image. So, how can we expect the z axis? (Sorry,if it is a silly question). My images are now oriented with North up and East right. The pixel array is 512*512 and the centre of the venus disc is found at 369,312 pixels. (all units in pixels). Having said the images are vertical, (north up so facing celestial North), I rotated the image with rot function in IDL used for 2d images, for the clockwise angle to pole : 22.84degree. Is this correct? But the pole distance is -19.09 arcseconds. Now, I have only oriented my image to face pole. The plate scale is 0.21 arcseconds. So, in pixels, pole is at 4 pixels from centre of venus disc. But, should I do something with North pole distance too?

This is how I mapped : Sub observer lat and long is the centre of the surface on venus which we are viewing (here sub obs long : 289.78, lat : -5.47 in degrees). So, I tried making a mapset with polar stereographic projection of centre (-5.47,289.78). Then i tried to overplot the rotated image (with rot 22.84 clockwise) on the grid. I faced an error : latitude out of range. I dont understand why this happens.

Any comments on my method would be helpful.

Thank you,
Aishwarya.

Subject: Re: Latitude longitude and Image Navigation
Posted by [AISHWARYA](#) on Mon, 27 Jun 2011 04:00:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jun 25, 9:03 am, Brian Wolven <brian.wol...@gmail.com> wrote:

> Well, *if* I remember correctly what I did there: Subcur is the vector of the initial sub-observer point on the planet/satellite before the rotations, given a coordinate system defined with respect to the center of that body. It's just the normalized planetocentric coordinates of the vector pointing toward the observer:

>

> subcur = -1.*body.pos/length(body.pos)

>

> The -1 comes in because body.pos contains the vector from the observer location (Earth in this case) to the body. subdes is the desired sub-observer location - the one specified in theimagemetadata or by SPICE, based on the observation time. We need to rotate our body centered coordinates such that the z-axis aligns with the specified north pole position vector, then rotate the body about the pole until our line of sight is passing through the proper sub-observer point. The various 'ang' variables were only used for diagnostic purposes to check that the desired result was being obtained.

>

> To put it in more graphic terms, imagine that you have a globe of Venus sitting upright on your desk. In order to match your telescopic observations you need to tilt the globe until the pole axis is aligned properly, then rotate about the pole until you reach the correct observing geometry. We have a body centered lat-long grid instead of a globe, but once we know the rotation we can then project our rotated lat-long grid on theimageand have it properly aligned, or infer the lat-long coordinates of a pixel in theimageif we're working the other way.

>

> There may very well be an easier way to do this, but that's what I cooked up years ago as a grad student, when presumably I was smarter than I am now. ;)

I could understand the code partly. But, what is the role of z axis ?
This is a 2 dimensional image. I tried rotating the image with centre as the pivot to 23 degree (the north pole position angle) and then used mapset with lat long as the sub observer lat long (-5.47, -109) with limints [-90,0,-180,180] since the image occupies southern hemisphere. But, I'm getting an error saying latitude is out of range.

Please help.

Thank you,
Aishwarya.
