Subject: Re: satellite field of view pole projection
Posted by penteado on Thu, 02 Jun 2016 18:42:58 GMT
View Forum Message <> Reply to Message

Hello,

If I understand correctly your problem, since you have the corner points, a solution might be to use pp_drawsphericalpoly from my library (http://www.ppenteado.net/idl/pp_lib/doc/).

Se the 5th example plot in http://www.ppenteado.net/idl/pp_lib/doc/pp_drawsphericalpoly .html, it is the one with polygons spanning the pole.

Paulo

On Thursday, June 2, 2016 at 7:22:27 AM UTC-7, audrey.sch...@gmail.com wrote:
> Hi everyone!
>
> I thought I had a rather simple (and common) problem, but I just can't seem to find the hang of how to solve this:
>
> I have a satellite that is flying around the Moon in a circular orbit with fixed altitude. The instrument on board the satellite is nadir pointing, has a fixed field-of-view, the surface projection of which is 150 x 25 km^2 (you can imagine the instrument being fixed in space and the Moon turning beneath the instrument). I now have a dataset of measured counts and associated longitude x latitude pair (either as boresight point, or as an array of four corner points).
>
> What I would like to do is map my counts onto the footprint of my rectangular field-of-view (around the boresight, or spanning the four corner points).
>
> This works well for the equatorial region: I just create a 360 x 180 float array, and for each measurement distribute the counts from the min(corners_longitudes), min(corners_latitudes) to the max(corners_longitudes), max(corners_latitudes).
>
> At the poles this obviously does not work, because at the poles a rectangular field-of-view projection has non-consecutive longitude, latitude values (they kind of jump around in value if part of the field-of-view is beyond the pole). I assume I have to do some triangulation, and fill all triangles within the triangles associated with the corner points. But I just can't seem to figure out how to do this.
>
> Does anyone have a solution to this problem?
>
> best,
> Audrey

Subject: Re: satellite field of view pole projection
Posted by audrey.schaufelberger on Fri, 03 Jun 2016 06:27:17 GMT

View Forum Message <> Reply to Message

thank you, that looks exactly like what I want to plot, but unfortunately I seem to already fail a step earlier in the process if I use your approach.

In your example you have a certain number of field-of-views you want to plot, which are not overlapping. In my question I was only talking about one orbit, and in that case your solution would work well.

But I actually have hundreds of orbits, where the fields-of-view overlap. I would like to average these measurements before I plot them. I am not sure how this actually influences the mapping process...

So far (for the equatorial region) I have simply been adding up the counts in one array, and have been creating a second array where I sum up the 'exposure' (how many times a surface element was seen). I then divided the first array by the second, the result of which I plotted.

If I wanted to use your approach, I would like to make use of the part where I determine what region a rectangle covers, but would then need to somehow be able to further process the result, and not directly plot it.

Any ideas?

best,
Audrey

Subject: Re: satellite field of view pole projection
Posted by audrey.schaufelberger on Fri, 03 Jun 2016 06:29:56 GMT
View Forum Message <> Reply to Message

what I forgot to mention: I have the complete SPICE kernels available... maybe that helps?

Subject: Re: satellite field of view pole projection
Posted by penteado on Fri, 24 Jun 2016 01:21:02 GMT
View Forum Message <> Reply to Message

The functionality to handle overlapping polygons, keeping track of everything that falls on each point in the map, so that one can decide on how to combine these values (take a mean, remove outliers, choose the stacking order, etc.) was lurking in pp_drawsphericalpoly, but because I had not decided on the algorithm and API, I had not documented it. I made some changes when I used it last week, and now I added some documentation, including some examples, of how to handle that (most relevant keywords are do_stack, stackmap and stackcount). I will probably add some alternative algorithms in the future, which would be chosen using other values for the keyword do_stack (at this time the only usable one is do_stack=1).

On Thursday, June 2, 2016 at 11:27:20 PM UTC-7, audrey.sch...@gmail.com wrote:

> thank you, that looks exactly like what I want to plot, but unfortunately I seem to already fail a step earlier in the process if I use your approach.
>
> In your example you have a certain number of field-of-views you want to plot, which are not overlapping. In my question I was only talking about one orbit, and in that case your solution would work well.
>
> But I actually have hundreds of orbits, where the fields-of-view overlap. I would like to average these measurements before I plot them. I am not sure how this actually influences the mapping process...
>
> So far (for the equatorial region) I have simply been adding up the counts in one array, and have been creating a second array where I sum up the 'exposure' (how many times a surface element was seen). I then divided the first array by the second, the result of which I plotted.
>
> If I wanted to use your approach, I would like to make use of the part where I determine what region a rectangle covers, but would then need to somehow be able to further process the result, and not directly plot it.
>
> Any ideas?
>
> best,
> Audrey