Subject: Re: TRIANGULATE/TRIGRID problem in IDL 5.3 (SGI)
Posted by Ben Tupper on Fri, 20 Apr 2001 02:08:22 GMT
View Forum Message <> Reply to Message

>
>>
>> However the interpolated column and row arrays have bogus values along
>> the top and bottom edges of the satellite swath (the edges are curves in
>> the example). For example, along the bottom edge of the interpolated
>> swath there are cyan pixels where the pixels should be darker blue. This
>> causes the bilinear interpolated image (not shown here) to have very
>> noticeable artifacts.

Hi,

I have bumped into this a number of times. You pobably already know this, but IDL's TRIGRID routine interpolates out to the limits of the 'outer-hull' defined by the Delaunay triangulation. The curviness of the data swath (almost banana-shaped) has introduced a subtle concavity. There is a triangle defined to connect the ends of the banana shape. The table shown in the online help of TRIGRID indicates that data can be extrapolated beyond the triangles (which I interpret to mean that interpolation always occurs within the triangles.) The greatest difference between the data locations and the boundary occurs in the middle of the banana. Try changing the TRIANGULATE statement to the following to retrieve the boundary points.

triangulate, lon[loc], lat[loc], tri, bounds

Then after you display the image, overplot the boundary points:

plots, lon[bounds], lat[bounds], psym = -1, color = !P.color

Note that many boundary points lie along the top of the swath, but few along the bottom (within the swath concavity.)

>
>
> In this case it looks like TRIGRID is extrapolating beyond the actual
> data points. Here is an example of your data, with a close-up showing
> the pixels and where the original data points were.
>
> ftp://cow.physics.wisc.edu/pub/craigm/cmtrigrid.gif
>
> All the light-blue points are way beyond the true data points.
> Perhaps you could post-process with a mask that removes pixels beyond
> a certain distance?
>

>

Generally, I have post-masked the data just as Craig suggests here. I usually have only a couple hundred (at most) columns and rows an the concavities are not so subtle as yours - so I manually mask out the nonsense data. That doesn't seem practical for your situation. Your data comes in sets of 10 (should it be decades or decaduplets?); is each set a scan line? If so, perhaps you could assemble the extremes from each scan line to use as the masking boundary.

I tried the following and it seems to yield the appropriate boundary points to use for masking.

```
lon_b = lon[*, [0,9]]
lat_b= lat[*, [0,9]]
 ;plot each end of extemes separately
plots, lon_b[*,0], lat_b[*,0], psym = -3, color =255
plots, lon_b[*,1], lat_b[*,1], psym = -3, color = 200
```

Now all you have to do is convert those longitudes and latitudes to image coordinates and feed them to the POLYFILLV function to get the image indices to preserve, the rest gets masked.

Ben

--
Ben Tupper
248 Lower Round Pond Road
POB 106
Bristol, ME 04539

Tel: (207) 563-1048
Email: PemaquidRiver@tidewater.net