
Subject: TRIANGULATE/TRIGRID problem in IDL 5.3 (SGI)
Posted by [Liam E. Gumley](#) on Thu, 19 Apr 2001 16:28:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear IDL community,

I'm hoping someone can provide me with some inspiration on a problem which has bugged me for a while now.

I have 1354 x 10 float arrays of latitude, longitude, and image data from a satellite. I create a map projection, and convert the lat/lon coordinates to irregularly spaced device coordinates (x, y) in the graphics window. I then use TRIANGULATE and TRIGRID to create regularly spaced interpolated grids of the column and row coordinates in the original image array. If these interpolated column and row arrays are computed correctly, they can be used with INTERPOLATE to compute an image which is interpolated to the pixels in the graphics window. This method (in theory) should be better than the subjective technique used by my IMAGEMAP procedure.

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.

I have tried every combination of TRIANGULATE and TRIGRID keywords I can think of, to no avail. Spherical gridding does not seem to be the answer because it requires the output grid to be regularly spaced in latitude and longitude (my output grid must be evenly spaced in x/y device coordinates).

I have attached my minimal example program below, and the data file mentioned can be obtained from

<ftp://origin.ssec.wisc.edu/pub/gumley/tmp/latlon.xdr>

To run the program:

```
.compile latlon  
window, /free, xsize=1100, ysize=850  
device, decomposed=0  
latlon
```

An annotated output image is available at

<ftp://origin.ssec.wisc.edu/pub/gumley/tmp/latlon.jpg>

If anyone has suggestions, I'd be glad to hear them.

Cheers,

Liam.

<http://cimss.ssec.wisc.edu/~gumley/>

;---

PRO LATLON

;- Read lat and lon arrays

restore, 'latlon.xdr'

help, lat, lon

dims = size(lat, /dimensions)

ncol = dims[0]

nrow = dims[1]

;- Create map projection

map_set, 30, -112.5, scale=5e6, /noborder

;- Create input row/column coordinate arrays

col = rebin(findgen(ncol), ncol, nrow, /sample)

row = rebin(reform(findgen(nrow), 1, nrow), ncol, nrow, /sample)

;- Convert the lat/lon data to device coordinates

result = convert_coord(lon, lat, /data, /to_device)

x = reform(result[0, *], ncol, nrow)

y = reform(result[1, *], ncol, nrow)

;- Get edges of image in device coordinates

loc = where(x ge 0.0 and x le (!d.x_vsize - 1.0) and \$
 y ge 0.0 and y le (!d.y_vsize - 1.0))

xmin = floor(min(x[loc]))

ymin = floor(min(y[loc]))

xmax = ceil(max(x[loc]))

ymax = ceil(max(y[loc]))

;- Interpolate

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

intcol = trigrd(x[loc], y[loc], col[loc], tri, \$

 [1.0, 1.0], float([xmin, ymin, xmax, ymax]))

;- Display the interpolated column array

loadct, 13

index = where(intcol gt 0)

min_value = min(intcol[index])

tv, bytscl(intcol, min=min_value, top=(!d.table_size - 1)), \$
 xmin, ymin

END

;---
