
Subject: Re: TRIANGULATE/TRIGRID problem in IDL 5.3 (SGI)
Posted by [Craig Markwardt](#) on Thu, 19 Apr 2001 20:12:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Liam E. Gumley" <Liam.Gumley@ssec.wisc.edu> writes:

> 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.
... remainder deleted ...

Hey Liam--

I haven't seen any responses so far. Unfortunately I don't have anything great to report either. I tried TRIANGULATE and TRIGRID a couple of times on some sky-mapping data (looking *up* from orbit, not down). I also ran into some very strange artifacts on the edges of the image which finally led me to a different approach [which I can discuss more upon request.]

In the past few years we have seen some queries on this pair of routines which suggest it's not entirely robust for production work. These artifacts are one thing, and the whole non-colinear points problem is another bad issue as well.

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?

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: TRIANGULATE/TRIGRID problem in IDL 5.3 (SGI)
Posted by [Kenneth P. Bowman](#) on Fri, 20 Apr 2001 00:16:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <3ADF1226.EF649510@ssec.wisc.edu>, Liam E. Gumley
<Liam.Gumley@ssec.wisc.edu> wrote:

> 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'm not certain if this is your problem, but if you have a 2-D array
with, say, missing values in it, when you interpolate you will get
artifacts along the edges of the missing regions. You can fill the
missing regions with NaN's (and lose some pixels along the edges), or
you could, for example, extrapolate instead of interpolating when one
of the interpolants is missing. That generally requires some knowledge
of the nature of the missing data (e.g., shape of the missing regions,
etc.).

In order to make contour plots of station data from Texas, we ended up
creating a mask to mask out any grid boxes outside the state boundary.
We let TRIANGULATE and TRIGRID do what they wanted, and then masked out
artifacts along the edges.

Ken

Subject: Re: TRIANGULATE/TRIGRID problem in IDL 5.3 (SGI)

```
>
>>
>> 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 probably 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 and 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 extremes 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

Subject: Re: TRIANGULATE/TRIGRID problem in IDL 5.3 (SGI)

Posted by [James Kuyper](#) on Fri, 20 Apr 2001 15:04:57 GMT

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

Ben Tupper wrote:

...

- > 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

This data set sounds familiar to me. Is it MODIS data from the Terra satellite? If so, then I'm the one responsible for the program that generates the geolocation information Liam's using. In that case, it's

not really a banana shape. Seen in a proper 3-D context, without map distortion, it's actually a bow-tie shape - the edges are actually arcs of what in spherical geometry are called "small" circles: they have radii of less than 90 degrees, bending inward. However, the radius of those "small" circles is pretty large, so they're almost great-circle arcs. The two longest arcs should have radii of about 89.6 degrees.

...

- > 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.

If I'm right about it being MODIS data, the sets of 10 represent a single frame of data. It's the complete set of 1354 frames that represents a single scan. Those two numbers, 10 and 1354, are the reason I suspect this is MODIS data. The MODIS instrument collects 3000 frames of data a second, and produces one such scan every 1.477 seconds (the missing 1.026 seconds is the time it spends not looking at the earth during each scan).

Subject: Re: TRIANGULATE/TRIGRID problem in IDL 5.3 (SGI)

Posted by [Liam E. Gumley](#) on Fri, 20 Apr 2001 19:46:47 GMT

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

Ben Tupper wrote:

- > 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.)
- >
- > 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,

I already tried the masking strategy you suggested yesterday, and the results weren't satisfactory.

However your suggestion for plotting the boundary nodes got me thinking. The online documentation for TRIGRID shows how you can plot each triangle in the connectivity list, e.g.

```
window, /free, xsize=1100, ysize=850
device, decomposed=0
restore, 'latlon.xdr'
map_set, 30, -112.5, scale=0.2e6
triangulate, lon, lat, tri, b
for i= 0L, n_elements(tri) / 3L - 1L do begin $
t = [tri[*, i], tri[0, i]] & plots, lon[t], lat[t] & endfor
plots, lon, lat, psym=2
```

This shows that many triangles are created along the bottom edge of the swath where I don't want triangles! However it also demonstrates that the triangles are created in a very consistent manner for the swath itself. The swath dimensions of 1354 x 10 are never going to change (yes, it is MODIS data), and I don't believe the connectivity list will ever change. So I can probably compute a triangle list manually just once (without using TRIANGULATE), and then use it for all cases in conjunction with TRIGRID. I'll send an update when I've implemented and tested this idea.

Cheers,
Liam.
PS Thanks to the other posters for the encouragement and suggestions.

Subject: Re: TRIANGULATE/TRIGRID problem in IDL 5.3 (SGI)
Posted by [James Kuyper](#) on Fri, 20 Apr 2001 22:26:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Liam E. Gumley" wrote:

...

> swath where I don't want triangles! However it also demonstrates that
> the triangles are created in a very consistent manner for the swath
> itself. The swath dimensions of 1354 x 10 are never going to change
> (yes, it is MODIS data), and I don't believe the connectivity list will
> ever change. So I can probably compute a triangle list manually just
> once (without using TRIANGULATE), and then use it for all cases in
> conjunction with TRIGRID. I'll send an update when I've implemented and
> tested this idea.

That will handle the normal case. However, if you need to be able to handle data even when things go a little bit wrong, then for each scan you should:

0) Check the "Geo scan quality" SDS; if the first column for a given scan has a 1 in it, we got no useful information about how fast the mirror was moving, making geolocation of that scan impossible.

1) Check the 'EV frames' value for that scan; it doesn't have to be 1354 - occasionally it's 0 due to a long data gap (> 0.5 seconds). Less frequently, it's a number between 0 and 1354; this happens if a long data gap starts or finishes during the Earth View segment of a scan.

2) Check the gflags SDS. There are things that can go wrong with geolocation on a per-pixel basis; in the pixels where gflags is >=64, either the line of sight could not be calculated, or it didn't intersect the Earth; either way, the stored values are unusable fill values.

If you follow that advice, it's going to mess up your pre-computed triangulation. On the other hand, less-than-perfect scans are rare enough that most users can afford to simply discard them.

Subject: Re: TRIANGULATE/TRIGRID problem in IDL 5.3 (SGI)
Posted by [Ben Tupper](#) on Fri, 20 Apr 2001 22:44:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

James Kuyper wrote:

> Ben Tupper wrote:
> ...
>> 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
>
> This data set sounds familiar to me. Is it MODIS data from the Terra
> satellite? If so, then I'm the one responsible for the program that
> generates the geolocation information Liam's using. In that case, it's
> not really a banana shape. Seen in a proper 3-D context, without map
> distortion, it's actually a bow-tie shape - the edges are actually arcs
> of what in spherical geometry are called "small" circles: they have

> radii of less than 90 degrees, bending inward.

Aha! A hyper-banana!

Ben

--

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

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

Subject: Re: TRIANGULATE/TRIGRID problem in IDL 5.3 (SGI)
Posted by [Ben Tupper](#) on Sun, 22 Apr 2001 00:28:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Liam E. Gumley" wrote:

>
> This shows that many triangles are created along the bottom edge of the
> swath where I don't want triangles! However it also demonstrates that
> the triangles are created in a very consistent manner for the swath
> itself. The swath dimensions of 1354 x 10 are never going to change
> (yes, it is MODIS data), and I don't believe the connectivity list will
> ever change. So I can probably compute a triangle list manually just
> once (without using TRIANGULATE), and then use it for all cases in
> conjunction with TRIGRID. I'll send an update when I've implemented and
> tested this idea.

>
>

Hi again,

Spin your own web - so to speak. You might like to check out the following web resource page for geometry:

<http://www.geom.umn.edu/software/cglist/>

Ben

--

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

Tel: (207) 563-1048

Email: PemaquidRiver@tidewater.net
