Subject: Area-and-time-weighted regridding of satellite data query
Posted by Jasdeep Anand on Tue, 24 Jul 2012 12:14:40 GMT
View Forum Message <> Reply to Message

Hey all.

I have several months worth of satellite data that I would like to bin
to a more regular grid, creating a time-averaged dataset (e.g a
monthly, rather than daily dataset). For each satellite swath I know
both the centre and corner lat/lon coordinates of every ground pixel,
along with the value I'm trying to grid (trace gas concentration,
let's call it "x" for the time being). The main problem I'm having is
that due to the satellite's inclination and wide-angle view, the
ground pixel area varies, from ~10 x 20 km in the middle to ~10 x 100
km at the swath edges. A typical swath can be viewed here:
 http://eoimages.gsfc.nasa.gov/images/imagerecords/38000/3823 3/fernandina_omi_2009104.jpg
- note how the orbital inclination means that the swath isn't aligned
to the geographic lat/lon grid either.

I figured that because of the alignment and varying pixel size I could
deem this dataset to be "irregular" and tried using TRIGRID and
TRIANGULATE to perform the gridding for me. However, I'm wary of any
biases the interpolation process might add to the resulting grid so I
want to create a method that accounts for the number of times a
particular grid cell in my new array overlapped a satellite pixel, and
the fraction of the grid cell that was covered by the satellite pixel.
I don't think calculating a Voronoi polygon for each data point is
appropriate here, because I already know the corner points of each
satellite pixel, but I'm not sure what else IDL can do in this case.

Are there any gridding routines that anyone has written that takes
multiple datasets and then regrids them onto a single grid that also
takes into account the corners of each pixel? Alternatively, are there
any routines that quickly works out the relative area occupied by two
overlapping pixels that I could incorporate into my own attempts at
gridding data? I've tried using OVERLAP (available here:
 http://www.oa-roma.inaf.it/facilities/IDL_Library/index.php? all_routines=1)
but it's taking far too long to check every satellite pixel with every
grid cell. Is there a better way of doing this too?

Thanks!