Subject: Re: Newbie needs help...
Posted by Liam E. Gumley on Thu, 11 Jan 2001 15:53:17 GMT
View Forum Message <> Reply to Message

Bruce Bowler wrote:
> As I hit the ground, I realized that displaying the data and extracting
> the data value at some lat/lon are 2 entirely different processes. I
> can use Liam's image_map to display it and came up with clever (but as
> yet untested) way to extract the data.
>
> Given a target lat/lon and BHAlat and BHAlon, how about (in pseudo-code)
>
> possiblelats = where(BHAlat eq lat{+/- some epsilon})
> possiblelons = where(BHAlon eq lon{+/- some epsilon})
> possiblevalues = intersection(possiblelats,possiblelons)
>
> if number of possiblevalues is between 1 and 10, printout the data
> otherwise adjust epsilon either up or down and try again.

I've written something similar for the MODIS Airborne Simulator (MAS),
which you could probably adapt for MODIS by tuning the epsilon values.
It works reasonably efficiently for small numbers of pixels. A couple of
other routines are required:

setintersection.pro from RSI:
http://www.dfanning.com/tips/set_operations.html

compass.pro from ESRG:
http://www.astro.washington.edu/deutsch-bin/idllibsrch?keywo rd=compass

This routine is appropriate for finding a few lat/lon locations at a
time. However it would not be very effective for overlaying coastline
lat/lon vectors on an image.

Cheers,
Liam.
http://cimss.ssec.wisc.edu/~gumley

```
;---------------------------------------
PRO MAS_LOCATE, SLAT, SLON, LAT, LON, X, Y

;+
; PURPOSE:
;    Locate a given lat/lon in a MAS image.
;
; INPUT:
;    SLAT    Latitude to locate (deg)
;    SLON    Longitude to locate (deg)
```

```
;    LAT     Array of MAS latitude values (deg)
;    LON     Array of MAS longitude values (deg)
;
; OUTPUT:
;    X       Pixel number closest to the given lat/lon (-1 if not found)
;    Y       Line number closest to the given lat/lon (-1 if not found)
;
; REVISED:
; Liam.Gumley@ssec.wisc.edu
; $Id: mas_locate.pro,v 1.2 1999/10/29 16:26:49 gumley Exp $
;-

;- Check arguments

if n_params() ne 6 then message, 'Usage: MAS_LOCATE, SLAT, SLON, LAT,
LON, X, Y'
if n_elements(slat) eq 0 then message, 'SLAT is undefined'
if n_elements(slon) eq 0 then message, 'SLON is undefined'
if n_elements(lat) eq 0 then message, 'LAT is undefined'
if n_elements(lat) eq 0 then message, 'LON is undefined'
if size(lat, /n_dim) ne 2 then message, 'LAT is not a 2D array'
if size(lon, /n_dim) ne 2 then message, 'LON is not a 2D array'
if arg_present(x) eq 0 then message, 'X cannot be modified'
if arg_present(y) eq 0 then message, 'Y cannot be modified'

;- Set default return values

x = -1L
y = -1L

;- Check that lat/lon lies within the array min/max

latmin = min(lat, max=latmax)
lonmin = min(lon, max=lonmax)
if (slat lt latmin) or (slat gt latmax) or $
   (slon lt lonmin) or (slon gt lonmax) then return

;- Find array elements close to the lat/lon

latindex = where(abs(lat - slat) lt 0.001, latcount)
lonindex = where(abs(lon - slon) lt 0.001, loncount)
if (latcount lt 1) or (loncount lt 1) then return

;- Find the intersecting elements of the arrays

result = setintersection(latindex, lonindex)
if (result[0] eq -1) then return
```

```
;- Compute the distance from the lat/lon to the array elements

compass, slat, slon, lat[result], lon[result], range, azimuth

;- Find the array element closest to the lat/lon

minrange = min(range, minindex)

;- Convert the 1D array element index to x/y

dims = size(lat, /dim)
x = result[minindex] mod dims[0]
y = result[minindex] / dims[0]

END
;----------------------------------------
```