Subject: Working with 2 partially overlapping images of different array sizes
Posted by aqueous0123 on Fri, 11 Jan 2002 20:33:48 GMT
View Forum Message <> Reply to Message

Sorry, I'm new to this image processing stuff.

I have 2 images of different sizes and different, but overlapping, locations on the globe. E.g., 2 satellite images from different sensors.

```
||||||||||||||||   <- IMAGE_1 = bytarr(532,532)
|              |
| """""""""""""""
| '            |   '
| '            |   '
| '            |   '
| '            |   '
| '            |   '
||'||||||||||||||   '  <- IMAGE_2 = bytarr(250,200)
  """""""""""""""
```

Each, IMAGE_1 and IMAGE_2 has its own arrays of associated lats and lons:
LATS_1 = fltarr(532)
LONS_1 = fltarr(532)
LATS_2 = fltarr(200)
LONS_2 = fltarr(250)

Notice that IMAGE_2 is
(1) not the saxe # pixels in X and/or Y and
(2) that its land coverage is not exactly the same as IMAGE_1, its range covers an area slightly to the south and east.
In summary, IMAGE_2 has smaller size in array space but covers nearly the same geographical space, just shifted a bit.

What I wish to do is create a new array IMAGE_3 that is the same array size (532,532) as IMAGE_1 AND also has the same physical location as IMAGE_1, but with IMAGE_2's data where IMAGE_1 and IMAGE_2 overlap (the 2's in the graphic below). Where they do not overlap, I'll just fill IMAGE_3 with nearest neighbor of IMAGE_2 (the 0's in the grahic below).

```
0000000000000000   <- IMAGE_3 = bytarr(532,532)
0000000000000000
0022222222222222
```

0022222222222222
0022222222222222
0022222222222222
0022222222222222
0022222222222222
0022222222222222

So, it seems like I have to:
(1) find intersections of IMAGE_1 and IMAGE_2 using each's LAT/LON arrays
(2) extract this data from IMAGE_2, call this IMAGE_2a
(2) congrid() this IMAGE_2a to the dimensions of IMAGE_1's subarea that IMAGE_2a 'covers', creating IMAGE_2b
(3) Create final image and dummy fill for now: IMAGE_3 = IMAGE_1 & IMAGE_3[*,*] = 255b
(4) Fill IMAGE_3 with IMAGE_2b, but only where IMAGE_2b overlaps (the 2's in the graphic above)
(5) Fill rest of IMAGE_3 with nearest neighbor from subarea IMAGE_2b filled in STEP 4 (the 0's in the graphic above). Now final IMAGE_3 has same LAT/LON bounds as IMAGE_1, the same array dims as IMAGE_1, but filled with the data from IMAGE_2.

In thinking the problem, I believe my inexperience is making me over-engineer this solution. I'm not even sure how to do this all. Help is greatly appreciated. Thanks.
