Subject: Re: Fill in a logic image: possible in IDL ?
Posted by John-David T. Smith on Sun, 15 Apr 2001 19:10:48 GMT
View Forum Message <> Reply to Message

jsilva@ci.uc.pt wrote:
>
> Hello
> To explain what I?m trying to do about a FILL in a logic image, I give an
> example (see the Computer Tomography lung image in
> http://www.ci.uc.pt/pessoal/jsilva/idl_fill.jpg)
> Thanks in advance for any suggestion.
> Jose Silva
> Physics Dep., FCTUC, Portugal

OK, I think I see now... you want to do blob coloring, aka "fill" from your favorite paint program. The IDL function label_region works well for this. It's quite simple. The JHU library contains two example programs for this, but I'll summarize:

labels=label_region(image LE level)
image[where(labels eq labels[x,y])]=fill

OK, so what does this do? label_region finds continuous blobs of non-zero values, and gives them a non-zero "ID". You find the ID of seed pixel (x,y), and where that blob exists in the array, then setting this region to whatever fill value you like.

Note that all edge pixels are considered to be in no region (label_region returns zero there). This seems dumb to me... one workaround is to fill dummy rows and columns on the exterior with 1's. You should probably also test that the region exists, or else you'll end up filling all the pixels in no region. e.g.

if labels[x,y] ne 0 then ...

Another JHU program illustrates the flexibility of this method... since you can make masks in many ways, you can label all sorts of interesting things like boundary regions... the inverse of the former problem:

labels=label_region(image ne boundary)

or even

labels=label_region(image gt bmax OR image lt bmin)

Here we label all areas bordered by boundary values in some range.

Search2D can do this too, but I think label_region gives more

flexibility. Since it just finds regions of non-zero value in a mask, you can conceive of doing all sorts of cool things. Here are a few interesting challenges:

1. Region of strictly decreasing value from a seed pixel. Or try increasing.

2. Region contiguous to seed pixel which is within 10% of the full data range to it.

3. Region contiguous to seed pixel with values in the nearest Nth percentile (think histogram).

4. Region contiguous to seed pixel which alternate even to odd.

etc.

Good luck,

JD
