Subject: Re: Fill in a logic image: possible in IDL ?
Posted by John-David T. Smith on Sun, 15 Apr 2001 02:27:01 GMT
View Forum Message <> Reply to Message

jsilva@ci.uc.pt wrote:

>
> Hello
> I have a grayscale image (512 x 512 x 1 byte from medical TAC).
> I applied a threshold to have a logic mask and tried to do a fill in that mask
> in order to remove some background. Most of languages have a fill command, like
> FILL in draw programs that FILLs the image starting in X,Y (user defined)
> coordenations.
> I search throw IDL commands and I never found any FILL command (or identical
> command).
> As most of languages (eg MATLAB and many others) have a FILL (BWFILL in MATLAB,
> as morphologic operation), did I miss the FILL in IDL or there isn?t any FILL in
> IDL?
> I tried to use POLYFILL but my mask is very irregular, not being possible to
> apply a POLYFILL in a predefined region.
> Any commend on this subject is welcome?
>
> level=100
> mask= (image LE level) ; TAC of lungs (from TAC) with many irregularities
> ; FILL to ?fill? mask starting (eg) in 1,1 pixel coordination
>
> jsilva @ ci.uc.pt

Unless I've misunderstood, it's probably easier than you think:

```
mask=where(image LE level,cnt)
if cnt gt 0 then image[mask]=background
```

or, if you know for sure there are some elements in the mask, you can
just do:

```
image[where(image LE level)]=background
```

Notice that here mask is just a list of indices. You can use multiple
masks in one of two (or more) ways, e.g.:

```
1. immask=image LE level AND image GE otherlevel AND otherim eq 1
   mask=where(immask ne 0)
2. mask=where(image LE level)
   mask=mask[where(image[mask] GE otherlevel)]
   mask=mask[where(otherim[mask] eq 1)]
```

In case 1, you compute all the comparisons on all the elements of image, and AND them all together. In case 2, you examine only those elements which survived the previous test, paring down mask at each step. Both forms have their advantages. But getting used to lists of indices as masks or collections of pixels in IDL is vital.

Good luck,

JD

P.S. Here's one other method which doesn't use subscripts, more in line with your original idea:

```
mask=image LE level
image=image*(mask)+background*(mask eq 0)
```