
Subject: Re: contour fill

Posted by [Craig Markwardt](#) on Fri, 25 Aug 2000 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

davidf@dfanning.com (David Fanning) writes:

> Stephen O'Sullivan (so@amsta.leeds.ac.uk) writes:

>

>> Thanks for your help but the problem that I didn't really make clear is
>> that I need all the values contained within the specified closed contour,
>> not just those above or below the contour value. In this sense it's a
>> geometric thing.

>

> Oh, well then, you are in for some work. :-)

>

> You will have to use the Path_Info and Path_XY keywords
> to fish out the contour lines you are concerned with

> ...

Wow David you must be trying hard to drum up consulting business :-).
I tried it myself, having never had ROI experience before, and got
something to work in about ten minutes.

Here is an example of thresholding an image the hard way, using the
technique you just described. The first part (the wordiest) is just
to set up an image to work on. The meat of the function is CONTOUR
and POLYFILLV -- and no graphics are produced. In fact you don't even
need to set up a graphics coordinate system on the screen. The
routines GAUSS2 and PLOTIMAGE are available from my web page
(<http://cow.physics.wisc.edu/~craigm/idl/>, check the Full Listing).

Once you have the list of pixels (the value of WH below) you can do
anything with it. I did a simple threshold but of course you could do
something harder. Now of course, the *real* problem is deciding
which contour to use when there are many of them. Here I have just
taken the first one. If you know there will always be one big one
then you might be able to get away by calculating the area of each
(eg, number of pixels in each) and taking the largest.

Craig

```
;; Create image using two gaussian functions
```

```
nx = 100 & ny = nx
```

```
dx = 0.1 & dy = dx
```

```
x = findgen(nx)*dx-5. & y = x
```

```
xx = rebin(reform(x,nx,1),nx,ny)
```

```
yy = rebin(reform(y,1,ny),nx,ny)
```

```
im = gauss2(xx, yy, [2.5, -4.2, 2., 50]) + gauss2(xx, yy, [-.5, 3, 4., 30])
```

```
:: Extract contour information - in this case everything inside the 20 contour
contour, im, x, y, levels=[20], path_info=pin, path_xy=pxy, $
/path_data, /close
```

```
:: Extract the contour of interest
c0off = pin(0).offset + lindgen( pin(0).n )
c0x = (pxy(0,c0off) - x(0)) / dx ;; Convert to scan cols & lines
c0y = (pxy(1,c0off) - y(0)) / dy
```

```
:: Use POLYFILLV to create a region of interest, and hence a mask
wh = polyfillv(c0x, c0y, nx, ny)
mask = (im*0)
mask(wh) = 1
```

```
:: Plot the masked image for verification
imgx = [-5.05, 4.95] & imgy = imgx
plotimage, im*mask, range=[0,52], imgxrange=imgx, imgyrange=imgy
```

Good luck,
Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
