Subject: Re: POLYFILLV weirdness Posted by David Fanning on Fri, 17 Nov 2006 05:38:19 GMT View Forum Message <> Reply to Message

## Ted writes:

- > I've been having problems with POLYFILLV. Consider the case below: > IDL> verts = TRANSPOSE([ [ 2, 3.25, 2, 0.75, 2 ], [ 0.75, 2, 3.25, 2, > 0.7511
- > IDL > dim = [6, 6]
- > IDL> inside = POLYFILLV(verts(0,0:3), verts(1,0:3), dim(0), dim(1))

>

- > Verts defines a diamond of width 2.5, centered at [2, 2], which
- > according to the POLYFILLV documentation should be the center of pixel
- > [2, 2]. The results of the POLYFILLV call should give me 5 pixels,
- > specifically [2, 3], [1, 2], [2, 2], [3, 2], and [2, 1]. However,
- > instead I get 6 pixels, those listed plus [3, 3]. From the POLYFILLV
- > documentation I can't for the life of me figure out how this extra
- > pixel would be regarded as being inside the diamond.

- > Has anyone else encountered any funny effects at polygon edges when
- > using POLYFILLV? Any advice or commiseration would be much
- > appreciated.

Humm. Well, I've thought POLYFILLV was broken for a long while now, but here is proof. Although not for the reasons you think. In fact, I can't really tell what you think POLYFILLV should be doing, and I can't tell where you are getting your final pixel values.

POLYFILLV should return 1D subscripts into a 2D array that are enclosed by a polygon. But, in fact, extra subscripts on the bottom and left of the polygon are incorrectly returned, as the program below will demonstrate.

In your example, you should have returned four subscripts, not six. (And certainly not the five you claim should be returned.)

You will need the following Coyote programs to run the code below:

http://www.dfanning.com/programs/fsc\_color.pro http://www.dfanning.com/programs/hak.pro

If you run the TEST program you will first see the

vertices plotted on the appropriate [6,6] array. When you hit the key to continue, the six indices that are returned are plotted. Only four of these are correct. If you hit the key to continue, you will see the two incorrect indices plotted in red.

I believe I have discussed this before (although I can't find the reference) and I can't remember exactly what I concluded. I think I concluded that IDLanROI produced better results. :-)

Anyway, opinions solicited...

```
PRO TEST
verts = TRANSPOSE([ [ 2, 3.25, 2, 0.75, 2 ], $
 [0.75, 2, 3.25, 2, 0.75]])
dim = [6,6]
plot, findgen(7), /nodata, xticklen=1, yticklen=1, $
 xticks=6, yticks=6, xminor=1, yminor=1
plots, verts, color=fsc color('green')
inside = POLYFILLV(verts(0,*), verts(1,*), dim(0), dim(1))
i2d = array_indices([6,6], inside, /Dimensions)
print, i2d
hak
FOR j = 0, N_Elements(inside)-1 DO BEGIN
 PolyFill, [i2d[0,j], i2d[0,j], i2d[0,j]+1, i2d[0,j]+1, i2d[0,j]], $
        [i2d[1,j], i2d[1,j]+1, i2d[1,j]+1, i2d[1,j], i2d[1,j]]
ENDFOR
plots, verts, color=fsc_color('green')
plots, [1,3],[2,2], Linestyle=2, Color=fsc color('black')
plots, [2,2], [1, 3], Linestyle=2, Color=fsc_color('black')
hak
FOR j = 0, 1 DO BEGIN
 PolyFill, [i2d[0,j], i2d[0,j], i2d[0,j]+1, i2d[0,j]+1, i2d[0,j]], $
        [i2d[1,i], i2d[1,i]+1, i2d[1,i]+1, i2d[1,i], i2d[1,i]], $
        Color=fsc_color('red')
ENDFOR
plots, verts, color=fsc_color('green')
plots, [1,3],[2,2], Linestyle=2, Color=fsc color('black')
plots, [2,2], [1, 3], Linestyle=2, Color=fsc color('black')
END
Cheers.
David
```

David Fanning, Ph.D. Fanning Software Consulting, Inc. Coyote's Guide to IDL Programming: http://www.dfanning.com/

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: POLYFILLV weirdness

Posted by dktr.ted on Fri, 17 Nov 2006 17:19:34 GMT

View Forum Message <> Reply to Message

Thanks for the confirmation. I had some confusion as to the location of a pixel's center (whole number index versus half index), hence my problem with the expected results of POLYFILLV. But your example nicely demonstrates my suspicion that POLYFILLV is not working as it should.

Ted

Subject: Re: POLYFILLV weirdness Posted by Jean H. on Fri, 17 Nov 2006 18:58:33 GMT

View Forum Message <> Reply to Message

- > POLYFILLV should return 1D subscripts into a 2D array
- > that are enclosed by a polygon. But, in fact, extra
- > subscripts on the bottom and left of the polygon
- > are incorrectly returned, as the program below will
- > demonstrate.

PolyFillV is not using the provided polygons coordinates but a "fix()" of them.... which induce this extra line on the left and at the bottom (and a few missing pixels on the right side and on the top If I remember well). I personnaly used a round() over my polygon coordinates and it was returning much better results... though still not perfect!

Jean