Subject: Polygon filling oddities
Posted by Mark Hadfield on Thu, 04 Oct 2001 00:47:30 GMT
View Forum Message <> Reply to Message

Hello all

I have been experimenting with different methods of generating images representing filled polygons. (My motivation comes from attempts at drawing filled coastlines on various devices.) I have found some oddities when using the POLYFILLV routine and the IDLanROI object. Before I take this issue to RSI tech support I thought I'd show it to the group and ask whether what I am doing makes sense.

The procedure below (and attached) illustrates the problem. It displays a 500 x 500 image with a centred circle of radius 300 using several methods, controlled by the option argument:

0 Call POLYFILL directly to window
1 Use POLYFILLV to create an image then display image
2 Use POLYFILL to create the image in a Z buffer then display image
3 Create an IDLanROI object and use its ComputeMask method to create an image, then display image.

These all seem to work and to give identical results, except perhaps for some minor differences around the edge of the polygon.

The routine also accepts a SHIFT keyword that lets the caller shift the circle around on the image plane. With options 0 and 2 this works exactly as expected: as SHIFT is increased the circle moves to the edge of the window and eventually disappears. But with options 1 and 3 the results are unexpected: as SHIFT is made more negative the circle vanishes abruptly when it passes a threshold.

For example "mgh_test_polyfill, 1, -143" (using POLYFILLV) produces a circle with its edge not quite touching the lower left corner of the image and "mgh_test_polyfill, 1, -144" produces a blank image. Similarly "mgh_test_polyfill, 3, -101" (using IDLanROI::ComputeMask) produces a circle with its edges touching the bottom and left sides of the image and "mgh_test_polyfill, 3, -101" produces a blank image.

I guess POLYFILLV and IDLanROI are intended for dealing with regions of interest on images and it is anticipated that the vertices of the ROI will be in the positive quarter-plane. But I don't see any reason why they shouldn't be able to work with negative vertex coordinates.

So is what I've found a bug or a feature? Can others reproduce my results? (I've been using IDL 5.4.)

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

------ mgh_test_polyfill -----------

```
; Testing various methods of polygon filling

pro mgh_test_polyfill, option, SHIFT=shift

    compile_opt IDL2

    if n_elements(option) eq 0 then option = 0

    if n_elements(shift) eq 0 then shift = 0
    if n_elements(shift) eq 1 then shift = [shift,shift]

    ; Create a window dimensioned [500,500]

    window, XSIZE=500, YSIZE=500

    ; Set up coordinates defining a circle, radius 150, centred at 250

    n_vert = 50

    angle = 2.*!pi*findgen(n_vert+1)/float(n_vert)

    x = 250 + 150*sin(angle)
    y = 250 + 150*cos(angle)

    ; Shift the circle

    x = x + shift[0]
    y = y + shift[1]

    ; Generate & display and image using different methods depending on
option argument

    case option of

        0: polyfill, x, y, /DEVICE

        1: begin
            image = replicate(0B, 500, 500)
            p = polyfillv(x, y, 500, 500)
            if min(p) gt 0 then image[p] = 255B
            tv, image
```

```
      end

      2: begin
         dname = !d.name
         set_plot, 'Z'
         device, SET_RESOLUTION=[500,500]
         erase
         polyfill, x, y, /DEVICE
         image = tvrd()
         set_plot, dname
         tv, image
      end

      3: begin
         roi = obj_new('IDLanROI', x, y)
         image = roi->ComputeMask(DIMENSIONS=[500,500])
         obj_destroy, roi
         tv, image
      end

   endcase

end
```

--
Posted from clam.niwa.cri.nz [202.36.29.1]
via Mailgate.ORG Server - http://www.Mailgate.ORG

Subject: Re: Polygon filling oddities
Posted by David Fanning on Fri, 05 Oct 2001 01:14:17 GMT
View Forum Message <> Reply to Message

Mark Hadfield (m.hadfield@niwa.cri.nz) writes:

>> I can reproduce your results. But I can also
>> get the PolyFillV code to work by making this
>> change:
>>
>>   p = polyfillv(0 > x < (!D.X_Size-1), 0 > y < (!D.Y_Size-1), 500, 500)
>
> Two comments/questions:
>
> 1. Do you think you need !D.X_Size and !D.Y_Size in there? What has the
> current graphics device got to do with it? (Not arguing just curious.)

Oh, I was trying to be general. But 10 milliseconds

after I hit the Send button I realized there wasn't
much point with those damn 500s in there. :-)

In this case "499" would have worked just as
well, and would maybe have been clearer too.

> 2. Clipping x and y before calculating the filled polygon generally won't
> give the same answer as clipping the polygon, at least not when the vertex
> spacing is large. Eg think of replacing the circle in my example with a
> triangle.

Humm. I guess you're right. You might have to
do some interpolation in this case. Although,
in practice I've never worked with objects
where this mattered much.

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155