Subject: Re: Polygon filling oddities
Posted by Mark Hadfield on Thu, 04 Oct 2001 21:40:20 GMT
View Forum Message <> Reply to Message

From: "David Fanning" <david@dfanning.com>
> Mark Hadfield (m.hadfield@niwa.cri.nz) writes:
>
>> 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.)
>
> 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.)

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.

BTW I thought of another method using IDLanROI:ContainsPoints instead of IDLanROI:ComputeMask...

```
4: begin
   roi = obj_new('IDLanROI', x, y)
   xx = rebin(findgen(500),500,500)
   yy = rebin(findgen(1,500),500,500)
   inside = roi->ContainsPoints(xx[*],yy[*])
   obj_destroy, roi
   image = bytarr(500,500)
   image[where(inside)] = 255B
   tv, image
end
```

It gives the correct results (i.e. the circle doesn't vanish abruptly) but

the CPU time increases by a factor of ~ 500 (from 0.12 to 6 seconds on my machine).

I will raise this with RSI. I'm sure they'll call back all the IDL 5.5 CD-ROMs :-)

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

--