Subject: Re: IDL ROI Objects
Posted by Fabzi on Wed, 28 Jan 2015 17:49:07 GMT
View Forum Message <> Reply to Message

David,

the problem is, I believe, not in IDLanROI. It is in how you DRAW the mask.

Consider your example:

```
poly = [[5, 10, 10, 5, 5], [5, 5, 10, 10, 5]]

cgDisplay, 400, 400, WID=2, Title='Polygon Interior Pixel Fill - IDLanROI'
p = OBJ_NEW('IDLanROI', poly[*,0], poly[*,1])
mask = p -> ComputeMask(DIMENSIONS=[20,20])
pixels = Where(mask EQ 255)
anImage = BytArr(20,20)+1B
anImage[pixels] = 255
cgPlot, [1], XRange=[0,19], YRange=[0,19], /NoData, ASPECT=1.0
cgImage, anImage, XRange=[0,19], YRange=[0,19], /Overplot
cgPlot, [1], XRange=[0,19], YRange=[0,19], /NoData, ASPECT=1.0, $
    YTickLen=1.0, XTickLen=1.0, /NoErase
```

Note that I changed the range of the plot to [0, 19].

IDLanROI's follows a PIXEL CENTER view. so a pixel is touched from -0.5 to +0.5. IDL's graphics pixels are following a lower left pixel convention. PIXEL_CENTER does not change anything to this fact, it just shifts the whole grid.

See those three examples which are perfectly coherent with each other:

```
pro ROIbug

poly = [[5, 10, 10, 5, 5], [5, 5, 10, 10, 5]]

cgDisplay, 600, 600, WID=1, Title='Polygon rule 0 Pixel Fill - IDLanROI'
p = OBJ_NEW('IDLanROI', poly[*,0], poly[*,1])
mask = p -> ComputeMask(DIMENSIONS=[20,20], MASK_RULE=0)
pixels = Where(mask EQ 255)
anImage = BytArr(20,20)+1B
anImage[pixels] = 255
cgImage, anImage, /AXES, /SAVE
centersX = transpose(fltarr(20)+1) ## (INDGEN(20)+ 0.5)
centersY = transpose(INDGEN(20)+ 0.5) ## (fltarr(20)+1)
cgPlotS, centersX, centersY, PSYM=16, /DATA
cgPlotS, poly[*, 0]+ 0.5, poly[*, 1]+ 0.5, COLOR='blue', /DATA
```

```
cgDisplay, 600, 600, WID=2, Title='Polygon rule 1 Pixel Fill - IDLanROI'
p = OBJ_NEW('IDLanROI', poly[*,0], poly[*,1])
mask = p -> ComputeMask(DIMENSIONS=[20,20], MASK_RULE=1)
pixels = Where(mask EQ 255)
anImage = BytArr(20,20)+1B
anImage[pixels] = 255
cgImage, anImage, /AXES, /SAVE
centersX = transpose(fltarr(20)+1) ## (INDGEN(20)+ 0.5)
centersY = transpose(INDGEN(20)+ 0.5) ## (fltarr(20)+1)
cgPlotS, centersX, centersY, PSYM=16, /DATA
cgPlotS, poly[*, 0]+ 0.5, poly[*, 1]+ 0.5, COLOR='blue', /DATA

cgDisplay, 600, 600, WID=3, Title='Polygon  rule 2  Pixel Fill - IDLanROI'
p = OBJ_NEW('IDLanROI', poly[*,0], poly[*,1])
mask = p -> ComputeMask(DIMENSIONS=[20,20], MASK_RULE=2)
pixels = Where(mask EQ 255)
anImage = BytArr(20,20)+1B
anImage[pixels] = 255
cgImage, anImage, /AXES, /SAVE
centersX = transpose(fltarr(20)+1) ## (INDGEN(20)+ 0.5)
centersY = transpose(INDGEN(20)+ 0.5) ## (fltarr(20)+1)
cgPlotS, centersX, centersY, PSYM=16, /DATA
cgPlotS, poly[*, 0]+ 0.5, poly[*, 1]+ 0.5, COLOR='blue', /DATA

end
```

Cheers,

Fabien

On 28.01.2015 18:17, David Fanning wrote:
> David Fanning writes:
>
>> With IDLanROI, if you try to take interior pixels, the box is one pixel
>> too short. If you try to take *all* the pixels, the box is one pixel too
>> long. It seems impossible to get the box "just right", given a
>> particular polygon description.
>

> Here is an article that describes the problem for those who prefer not
> to run the code:
>
>   http://www.idlcoyote.com/code_tips/roipolygon.php
>
> Cheers,
>
> David
>
