

justspam03@yahoo.de writes:

- > I'm stuck with the following:
- > I got a 2-dim image, say 256x256 pixels which spans (in real world
- > coordinates) the region from [1,1] to [5,5].
- > Plus I got ROIs defined on this image, say a rectangular ROI with x/y
- > coords [2,2], [2,4], [3,4], [3,2].
- >
- > Now I'd like to have a mask for this ROI with the pixel dimensions of
- > the image.
- >
- > roi = obj_new('IDLanRoi', real world ROI coordinates here)
- > mask = roi->computeMask(dimensions=[256,256], location=[1.,1.], ...
- > and now?)
- >
- > Is there a way to specify the 'real world' size of the mask pixels
- > (i.e. 1 pixel = 1/64 cm)?

I don't think this is going to work the way you want
it to work using IDLanROI. I would create the mask like
this:

```
s = Size(image, /DIMENSIONS)
x = [2, 2, 3, 3, 2]
y = [2, 4, 4, 2, 2]
mask = BytArr(s[0], s[1])
```

To find the pixels in the mask, set up a data coordinate
system, and convert ROI coordinates to device coordinates.

```
Window, XSIZE=s[0], YSIZE=s[1], /PIXMAP
thisWindow = !D.Window
xrange = [1,5]
yrange = [1,5]
!X.S = FSC_Normalize(xrange)
!Y.S = FSC_Normalize(yrange)
!X.Window = [0,1]
!Y.Window = [0,1]
!X.CRange = xrange
!Y.CRange = yrange
xy = Convert_Coord(x, y, /DATA, /TO_DEVICE)
WDelete, thisWindow
```

Now fill the pixels inside your area with 1 (or 255, whatever

is appropriate for you).

```
indices = PolyFillV(xy[0,*], xy[1,*], s[0], s[1])  
mask[indices] = 1B
```

You can find FSC_Normalize here:

http://www.dfanning.com/programs/fsc_normalize.pro

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.")
