Subject: Re: How to define resolution (not dimension!) for IDLanRoi->computeMask
Posted by ben.bighair on Tue, 10 Feb 2009 13:26:00 GMT
View Forum Message <> Reply to Message

On Feb 10, 7:00 am, justspa...@yahoo.de wrote:

> Still, if I may ask again: is anyone aware of the difference between
> the 'location' and
> the 'pixel_center' keyword of IDLanROI->computeMask?
>

Hi,

My assumption has always been that Location specifies the overall position of the mask and that Pixel_Center allows you to "jiggle" around that. The picture I have in my mind is of the old "quadrat mapping" from high school biology class. We staked out an area, randomly placed squares ("quadrats" 1 m^2) and then within each quadrat determined the average number ant hills (or something) per 10 cm^2.   To me, "location" referred to the location of the quadrat relative to the study area and the "pixel_center" to the relative location of anthills within each quadrat.

But now that you have pressed the question, I am not at all sure that I have the right picture. I have a small example routine (see below) that shifts around location and pixel_center. I am pretty sure that pixel center determines the "filling" of pixels, much as described in the intro to PolyFillV. But, the more I look at the output the fuzzy my thinking gets. As I my story book hero (Freddy The Pig) says, "It's no use, thinking just confuses things."

For the benefit of all, I have pasted in the online descriptions of Location and Pixel_Center from IDLanROI::ComputeMask.

LOCATION
Set this keyword to a vector of the form [X, Y[, Z]] specifying the location of the origin of the mask. The default is [0, 0, 0]. IDL converts and maintains this value in double-precision floating-point.

PIXEL_CENTER
Set this keyword to a 2-element vector, [x, y], to indicate where the lower-left mask pixel is centered relative to a Cartesian grid. The default value is [0.0, 0.0], indicating that the lower-left pixel is centered at [0.0, 0.0].

```
**** BEGIN
pro loc_test
```

```
  loc1 = [0,0]
  loc2 = [0,0]
  loc3 = [0,0]
  loc4 = [0,0]

  pc1 = [0,0]
  pc2 = [0.5, 0]
  pc3 = [1,0]
  pc4 = [-0.5,0]

  x = [2,2,3,3]
  y = [2,3,3,2]
  roi = OBJ_NEW('IDLanROI', x,y)

  print, "******* Constant Location, Varying Center *********"
  mask1 = roi->ComputeMask(Loc = loc1, Pixel_Center = pc1, DIM =
[5,5])
  mask2 = roi->ComputeMask(Loc = loc2, Pixel_Center = pc2, DIM =
[5,5])
  mask3 = roi->ComputeMask(Loc = loc3, Pixel_Center = pc3, DIM =
[5,5])
  mask4 = roi->ComputeMask(Loc = loc4, Pixel_Center = pc4, DIM =
[5,5])

  print, "Pixel_Center = ", pc1
  print, "Location = ", loc1
  print, mask1

  print, "Pixel_Center = ", pc2
  print, "Location = ", loc2
  print, mask2

  print, "Pixel_Center = ", pc3
  print, "Location = ", loc3
  print, mask3

  print, "Pixel_Center = ", pc4
  print, "Location = ", loc4
  print, mask4


  print, "******* Constant Center, Varying Location *********"

  loc1 = [0,0]
  loc2 = [0.5,0]
  loc3 = [1,0]
  loc4 = [2,0]
  pc1 = [0,0]
  pc2 = [0,0]
  pc3 = [0,0]
  pc4 = [0,0]

  mask1 = roi->ComputeMask(Loc = loc1, Pixel_Center = pc1, DIM =
[5,5])
  mask2 = roi->ComputeMask(Loc = loc2, Pixel_Center = pc2, DIM =
[5,5])
  mask3 = roi->ComputeMask(Loc = loc3, Pixel_Center = pc3, DIM =
[5,5])
  mask4 = roi->ComputeMask(Loc = loc4, Pixel_Center = pc4, DIM =
[5,5])

  print, "Pixel_Center = ", pc1
  print, "Location = ", loc1
  print, mask1

  print, "Pixel_Center = ", pc2
  print, "Location = ", loc2
  print, mask2

  print, "Pixel_Center = ", pc3
  print, "Location = ", loc3
  print, mask3

  print, "Pixel_Center = ", pc4
  print, "Location = ", loc4
  print, mask4

end
```

***** END