Subject: Re: How to define resolution (not dimension!) for IDLanRoi->computeMask
Posted by justspam03 on Mon, 09 Feb 2009 14:09:24 GMT
View Forum Message <> Reply to Message

sorry forgot: I should add that I also don't really understand the
difference between the location and pixel_center keywords. Could
anybody please explain?
Thanks!

Subject: Re: How to define resolution (not dimension!) for IDLanRoi->computeMask
Posted by justspam03 on Mon, 09 Feb 2009 14:56:51 GMT
View Forum Message <> Reply to Message

Hi David,

eieieiei - I was afraid I'd need to do something like this :-(
Well, I guess I'll have my own overloaded computeMask... I anyway
subclass from IDLanRoi.
Thanks for the quick reply.
Oliver

Subject: Re: How to define resolution (not dimension!) for IDLanRoi->computeMask
Posted by David Fanning on Mon, 09 Feb 2009 15:11:19 GMT
View Forum Message <> Reply to Message

justspam03@yahoo.de writes:

> eieieiei - I was afraid I'd need to do something like this :-(
> Well, I guess I'll have my own overloaded computeMask... I anyway
> subclass from IDLanRoi.

Well, somehow you have to get from a data coordinate system
into a pixel coordinate system. There are probably other
ways to do it. I showed you what I do in direct graphics.
There are certainly comparable methods in object graphics,
but I don't know what they are. :-)

I suppose you could use the old Scale_Vector/Value_Locate
trick to find the device values of your ROI:

```
s = Size(image)
xvec = Scale_Vector(Findgen(s[0], 1, 5)
yvec = Scale_Vector(Findgen(s[1], 1, 5)
xdevice = Value_Locate(xvec, x)
ydevice = Value_Locate(yvec, y)
```

indices = PolyFillv(xdevice, ydevice, s[0], s[1])

I haven't tested that, but it should work. :-)

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.")

Subject: Re: How to define resolution (not dimension!) for IDLanRoi->computeMask
Posted by justspam03 on Tue, 10 Feb 2009 12:00:54 GMT
View Forum Message <> Reply to Message

Thanks. I've been hacking a bit and it seems in the end I'll extract
the data points
from the ROI, scale them properly, create a 'helper' roi and let the
latter do the
computeMask. The advantage is a.o. that this way I can also non-square
pixels (which
is a requirement).
PolyFillv might not be first choice as it implicitly maps the bounding
box of the 'Roi'
to an integer number of pixels (if I understood that correctly.)

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

Cheers
Oliver

On Feb 9, 4:11 pm, David Fanning <n...@dfanning.com> wrote:
> justspa...@yahoo.de writes:
>> eieieiei - I was afraid I'd need to do something like this :-(
>> Well, I guess I'll have my own overloaded computeMask... I anyway
>> subclass from IDLanRoi.
>
> Well, somehow you have to get from a data coordinate system
> into a pixel coordinate system. There are probably other
> ways to do it. I showed you what I do in direct graphics.

> There are certainly comparable methods in object graphics,
> but I don't know what they are. :-)
>

> I suppose you could use the old Scale_Vector/Value_Locate

> trick to find the device values of your ROI:
>
> s = Size(image)
> xvec = Scale_Vector(Findgen(s[0], 1, 5)
> yvec = Scale_Vector(Findgen(s[1], 1, 5)
> xdevice = Value_Locate(xvec, x)
> ydevice = Value_Locate(yvec, y)
> indices = PolyFillv(xdevice, ydevice, s[0], s[1])
>
> I haven't tested that, but it should work. :-)
>
> 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.")

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

Subject: Re: How to define resolution (not dimension!) for IDLanRoi->computeMask
Posted by David Fanning on Tue, 10 Feb 2009 13:52:21 GMT
View Forum Message <> Reply to Message

Ben writes:

> 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."

BIG mistake here! Follow Aristotle's advice and *never*
look into the horse's mouth. :-)

Cheers,

David

P.S. I think you are right about the pixel center
"jiggling" the location, although the image this brings
to mind is, uh, usually different from how it is used
in this context. I expect the center of the pixel will
be one of the criteria that is evaluated when the algorithm
has to decide if the pixel falls into the filled or not
filled bin. (Pixel centers to the right of the imaginary
dividing line are filled, etc.) This could be one of the
reasons POLYFILLV always seems to overfill to the bottom
and left compared to some other masking-making algorithms.

--
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.")

Subject: Re: How to define resolution (not dimension!) for IDLanRoi->computeMask
Posted by ben.bighair on Tue, 10 Feb 2009 14:13:19 GMT
View Forum Message <> Reply to Message

On Feb 10, 8:52 am, David Fanning <n...@dfanning.com> wrote:

>
> P.S. I think you are right about the pixel center
> "jiggling" the location, although the image this brings
> to mind is, uh, usually different from how it is used
> in this context.

Hi,

I laughed at this until my mind wander back to ants - jiggling ants!
Stuff of daymares. I think we actually counted blades of grass. Very
exciting - but they didn't jiggle too much.

Cheers,
Ben