Subject: Re: object IDLanROI and polyfillv
Posted by David Fanning on Wed, 12 Sep 2001 13:56:23 GMT
View Forum Message <> Reply to Message

Karsten Rodenacker (rodena@gsf.de) writes:

> Two object graphics newbie questions:
>
> Does anyone know how to get from an IDLanROI object the initial path
> coordinate data ?

I'm not sure I completely understand this question.
Do you mean you want the initial polygon points that
you initialized the object with? If so, then you would
do something like this:

```
myRoiObj->GetProperty, Data=myPolygonPts
```

> Suprisingly using the same contour (path) coordinate data, the IDLanROI
> object returns with computemask a different one compared with a mask
> generated by polyfillv. Is there any explication for that?

The Mask_Rule keyword allows you to specify several algorithms
for which pixels will be included in the mask. You will have
to set it to whatever it is (nobody knows, I guess) that
PolyFillV uses. The documentation, of course, doesn't even
tell you which algorithm the IDLanROI object uses by default,
so the standard procedure is to spend an hour or so experimenting
until everything becomes clear. Then we would appreciate it if
you would write an article about what you learned and publish
it here. :-)

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: object IDLanROI and polyfillv
Posted by Karsten Rodenacker on Thu, 13 Sep 2001 07:11:54 GMT
View Forum Message <> Reply to Message

Thank you, David. The first answer helped. Do you think that other values for

mask_rule exist?
Attached is a piece of code illustrating at least differences of ComputeMask
and polyfillv.

David Fanning schrieb:

> Karsten Rodenacker (rodena@gsf.de) writes:
>
>> Two object graphics newbie questions:
>>
>> Does anyone know how to get from an IDLanROI object the initial path
>> coordinate data ?
>
> I'm not sure I completely understand this question.
> Do you mean you want the initial polygon points that
> you initialized the object with? If so, then you would
> do something like this:
>
> myRoiObj->GetProperty, Data=myPolygonPts
>
>> Suprisingly using the same contour (path) coordinate data, the IDLanROI
>> object returns with computemask a different one compared with a mask
>> generated by polyfillv. Is there any explication for that?
>
> The Mask_Rule keyword allows you to specify several algorithms
> for which pixels will be included in the mask. You will have
> to set it to whatever it is (nobody knows, I guess) that
> PolyFillV uses. The documentation, of course, doesn't even
> tell you which algorithm the IDLanROI object uses by default,
> so the standard procedure is to spend an hour or so experimenting
> until everything becomes clear. Then we would appreciate it if
> you would write an article about what you learned and publish
> it here. :-)
>

Here is (at least) a small piece of code showing the behaviour .

```
xroi,bytarr(256,256),regions_out=r1,/block,$
    ROI_geom=r1_geo,stat=r1_stat
smii=r1->computemask(dimensions=[256,256],mask_rule=2) ne 0
r1->getProperty,all=ttxyinfo,data=ttxy
reg=polyfillv(ttxy[0,*],ttxy[1,*],256,256)
smjj=bytarr(256,256)
smjj[reg]=1b
tvscl,smii,0
tvscl,smjj,1
tvscl,smii xor smjj,2
```

end

The third image displayed shows the difference between the output of
myRoiObj->computemask and polyfillv.
Any roi should be drawn and xroi should be left by quitting.

The behaviour has not become clear, but maybe there is some expertise for
explanation. I am not so involved with graphics. Switching between
Mask_rule=1 and Mask_rule=2 shows only deviations on the lower left side of
the mask in contrast to the upper right side. Seemingly the understanding of
interior, boundary and 'interior and boundary' differs a bit throughout idl.

Regards
--
Karsten Rodenacker (LapTop)
---------- ----------:-)
GSF - Forschungszentrum Institute of Biomathematics and Biometry
D-85758 Oberschleissheim Postfach 11 29
Tel: +49 89 31873401 | FAX: ...3369 | rodena@gsf.de | Karsten@Rodenacker.de
http://www.gsf.de/ibb/homepages/rodenacker

Subject: Re: object IDLanROI and polyfillv
Posted by David Fanning on Thu, 13 Sep 2001 14:12:59 GMT
View Forum Message <> Reply to Message

Karsten Rodenacker (Karsten@Rodenacker.de) writes:

> Thank you, David. The first answer helped. Do you think that other values for
> mask_rule exist?

It wouldn't appear so.

> Attached is a piece of code illustrating at least differences of ComputeMask
> and polyfillv...
> The behaviour has not become clear, but maybe there is some expertise for
> explanation. I am not so involved with graphics. Switching between
> Mask_rule=1 and Mask_rule=2 shows only deviations on the lower left side of
> the mask in contrast to the upper right side. Seemingly the understanding of
> interior, boundary and 'interior and boundary' differs a bit throughout idl.

The object graphics system and the direct graphics system are
two completely different ways of doing graphics, so I am not
terribly surprised to find differences between them. There
are always about a dozen different ways to accomplish
something with a computer program and no firm basis for
calling one method "right" and another "wrong".

This is especially true (as I learned in my graduate school days) of any contouring algorithm you care to write. There is (quite literally) no "right" way to draw a contour line. There are some ways that are more internally "consistent" with one another, but no right way.

Since what we are seeing here has to do with boundaries around a contour, I'm not at all surprised to see differences in the two methods. And, frankly, I wouldn't worry about which was "right". I would urge you to pick a method and stick with it, so as to get consistent results.

I ran into exactly this problem several months ago myself, when I thought an IDLanROI object would make my life easier. But I couldn't get its answers for perimeter, centroid, and area to match with answers I was getting by performing the same calculations on contour lines returned by the Contour command. In the end, I ripped the object graphics stuff out of my code so my answers could be internally consistent.

(I could probably just as well have gone with the ISOCONTOUR command for creating my contours, but frankly the time required to decipher the seemingly incomprehensible documentation appeared overwhelming at the time. It was much faster to just write my own methods for calculating the perimeter, centroid, and area for contours with which I was much more familiar.)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155