Subject: Re: find_boundary routine
Posted by David Fanning on Fri, 19 Mar 2004 02:47:39 GMT
View Forum Message <> Reply to Message

Brian Frizzelle writes:

> Let me apologize if this post is a bit lengthy. And in case it's
> important, I'm using IDL 6.0 on a Windows machine.
>
> I need help calculating the perimeters of regions. I have an input
> array of roughly 900 by 1200 cells which contains 15 different
> categories (coded 1 through 15). For each category, I need to find the
> total perimeter length among all of its regions.
>
> I found David Fanning's FIND_BOUNDARY routine and ran it on this
> array, but it seemed to stop at the first region that it found. So I
> wrote a script that subsetted each array first by category, then by
> region within the category (regions were created with LABEL_REGION).
> Running FIND_BOUNDARY again, I got an output for each category, but
> the numbers seemed off.
>
> So I resorted to creating a simple 5x5 array with two categories, one
> region each, to test the perimeter output of FIND_BOUNDARY.
> - Category 1 has 16 cells and a perimeter of 18.
> - Category 2 has 9 cells and a perimeter of 18.
> * FIND_BOUNDARY output a perimeter of 2 for each category.
> * FIND_BOUNDARY also only give me an output array with 2 X/Y
> coordinates.
>
> Has anyone else found this to be a problem? Maybe I'm passing in the
> wrong parameters. Here are the details of what I've done so far. Any
> help would be greatly appreciated.

Here is an example of a 5 by 5 array, with the middle 9 pixels
making up the region:

```
IDL> array=[[0,0,0,0,0],[0,1,1,1,0],[0,1,1,1,0],[0,1,1,1,0],[0,0, 0,0,0]]
IDL>indices = Where(array GT 0)
IDL>pts = Find_Boundary(indices, XSize=5, YSize=5, Perimeter=p)
IDL>Print, p
   8.0
```

I had to make a small change to the Find_Boundary to accommodate this
few a number of indices. I had to change line 316 from this:

```
   boundaryPts = IntArr(2, Long(xsize) * ysize / 4L)
```

To this:

```
boundaryPts = IntArr(2, Long(xsize) * ysize / 2L)
```

You might think the correct "perimeter" would be 9 in this case, but you have to remember that the algorithm is calculating from the center of each pixel. A quick sketch of these 9 pixels and their centers will convince you that the correct answer is 8.

Other combinations of pixels in my tests gave similar correct results. I think the algorithm is working properly.

Cheers,

David

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