Subject: Re: IDL Average Value Graphs
Posted by David Fanning on Tue, 15 Jul 2008 21:01:56 GMT
View Forum Message <> Reply to Message

andybohn@gmail.com writes:

> Hello, I was wondering if IDL has any built-in methods for two similar
> things.
> 1. Basically cut the image like a pizza (for a variable number of
> slices) and average the values in each slice to create a plot of
> average pixel value as a function of angle
> 2. Do the same thing, but plot the average pixel value as a function
> of radius, (concentric circles).
>
> I was going to write my own method, but it would be complicated
> converting angled lines or circles into pixels.

I'll say, especially if the concentric circles have to have
equal areas! I've done this before, but I don't have the code
near at hand. I'll see if I can dig it up and if it is worth
publishing. May take me a day or so to get to it. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: IDL Average Value Graphs
Posted by Jean H. on Tue, 15 Jul 2008 21:11:55 GMT
View Forum Message <> Reply to Message

andybohn@gmail.com wrote:
> Hello, I was wondering if IDL has any built-in methods for two similar
> things.
> 1. Basically cut the image like a pizza (for a variable number of
> slices) and average the values in each slice to create a plot of
> average pixel value as a function of angle
> 2. Do the same thing, but plot the average pixel value as a function
> of radius, (concentric circles).
>
> I was going to write my own method, but it would be complicated
> converting angled lines or circles into pixels.
>

> Thanks a lot, Andy

have a look at POLYFILLV .... it will tell you which pixels are within a
polygon.

Then, download a function to create "pies" ... I have my own
http://www.ucalgary.ca/~jghasban/pie.pro ... but there is such function
in most downloadable libraries.

So, basically, you would create the pies, get the coordinates of the
pixels within and then do your maths!

Jean

Subject: Re: IDL Average Value Graphs
Posted by andybohn on Thu, 17 Jul 2008 15:21:52 GMT

View Forum Message <> Reply to Message

On Jul 15, 5:11 pm, Jean H <jghas...@DELTHIS.ucalgary.ANDTHIS.ca>
wrote:
> andyb...@gmail.com wrote:
>> Hello, I was wondering if IDL has any built-in methods for two similar
>> things.
>> 1. Basically cut the image like a pizza (for a variable number of
>> slices) and average the values in each slice to create a plot of
>> average pixel value as a function of angle
>> 2. Do the same thing, but plot the average pixel value as a function
>> of radius, (concentric circles).
>
>> I was going to write my own method, but it would be complicated
>> converting angled lines or circles into pixels.
>
>> Thanks a lot, Andy
>
> have a look at POLYFILLV .... it will tell you which pixels are within a
> polygon.
>
> Then, download a function to create "pies" ... I have my
ownhttp://www.ucalgary.ca/~jghasban/pie.pro ... but there is such function
> in most downloadable libraries.
>
> So, basically, you would create the pies, get the coordinates of the
> pixels within and then do your maths!
>
> Jean

I've been playing with POLYFILLV, and it (eventually) seems helpful.

It is a bit annoying that it returns a 1d array that has to be
converted back to a 2d array, but that's alright. I'm still playing
with the pie slices at the moment, and I didn't think of it at the
time, but if I don't inscribe a circle in the square image, then I
will get unequal areas for each slice.

So your pie program creates a 100 sided polygon that would emulate a
circle, then two more points to get back to the origin. Couldn't the
100 sided polygon be created, then for finding the pixels in between
the two circles, use POLYFILLV on the larger circle, then subtract the
smaller one? Also, I really like the way you created the polygon
array, however I'm not entirely sure how arc_x and arc_y get added to
the final points.

Thanks for the help!
Andy

Subject: Re: IDL Average Value Graphs
Posted by Jean H. on Thu, 17 Jul 2008 17:19:29 GMT
View Forum Message <> Reply to Message

> I've been playing with POLYFILLV, and it (eventually) seems helpful.
> It is a bit annoying that it returns a 1d array that has to be
> converted back to a 2d array, but that's alright.

?? ... just use the 1D array when you do your math!
pieCoord1D = [110,111,210,211]
PiePixels = data2D[pieCoord1D]

  I'm still playing
> with the pie slices at the moment, and I didn't think of it at the
> time, but if I don't inscribe a circle in the square image, then I
> will get unequal areas for each slice.

... yes... same thing will happen on the edge of the image.

> So your pie program creates a 100 sided polygon that would emulate a
> circle, then two more points to get back to the origin. Couldn't the
> 100 sided polygon be created, then for finding the pixels in between
> the two circles, use POLYFILLV on the larger circle, then subtract the
> smaller one?

Yes, this is correct. Though I would not use the pie program with an
angle of 360 degrees (what would be the consequences of the line to the
center of the circle??). Use a function to create a plain circle:
http://www.dfanning.com/tips/make_circle.html

Also, I really like the way you created the polygon
> array, however I'm not entirely sure how arc_x and arc_y get added to
> the final points.

arc_X and arc_Y are the 100 points forming the arc.
By doing x = [x_center, arc_x, x_center] , I simply add, in 1st and last position, the coord of the center of the circle. The X and Y arrays therefore represent the coord of the points starting at the center of the circle, going through the 100 points of the arc and then back to the center. It is a closed polygon.

Jean

>
> Thanks for the help!
> Andy