
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
-