

---

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  
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

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

---