
Subject: IDL Average Value Graphs

Posted by [andybohn](#) on Tue, 15 Jul 2008 20:51:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

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

Subject: Re: IDL Average Value Graphs

Posted by [Jeremy Bailin](#) on Wed, 16 Jul 2008 18:18:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Jul 15, 4:51 pm, 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

I don't know of anything built-in, but I've written similar sorts of things quite a lot using HISTOGRAM, and it's pretty straight-forward... here's a (UNTESTED!) hack at it, assuming that your image is called "image", is centered at pixel x0,y0, and you want n_azimuth pie slices and radial bins of width dr:

```
indeximage=array_indices(image, lindgen(n_elements(image)))
pixelradii = sqrt( (indeximage[0,*]-x0)^2 + (indeximage[1,*]-y0)^2 )
pixelazimuth = atan(indeximage[1,*], indeximage[0,*])
```

```
radhist = histogram(pixelradii, min=0, bin=dr, reverse_indices=radri)
nrad=n_elements(radhist)
```

```
azhist = histogram(pixelazimuth, min=0, bin=2.*!pi/n_azimuth,  
reverse_indices=azri)  
naz=n_elements(azhist)
```

```
radial_mean = fltarr(nrad)  
azimuthal_mean = fltarr(naz)  
for i=0l,nrad-1 do if radhist[i] gt 0 then radial_mean[i] =  
mean(image[radri[radri[i]:radri[i+1]-1]])  
for i=0l,naz-1 do if azhist[i] gt 0 then azimuthal_mean[i] =  
mean(image[azri[azri[i]:azri[i+1]-1]])
```

-Jeremy.

-Jeremy.

Subject: Re: IDL Average Value Graphs
Posted by [David Fanning](#) on Thu, 17 Jul 2008 15:33:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

andybohn@gmail.com writes:

> 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

Why is that?

Cheers,

David

--

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

Subject: Re: IDL Average Value Graphs
Posted by [andybohn](#) on Thu, 17 Jul 2008 17:55:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jul 17, 11:33 am, David Fanning <n...@dfanning.com> wrote:
> andyb...@gmail.com writes:

>> 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
>
> Why is that?
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:<http://www.dfanning.com/>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

When I ran POLYFILLV on a triangle, it returned just a 1d array of all the pixels in the triangle. I just wish it had an argument that returns a 2d array (the x and y coordinates of all of the pixels in the polygon). It's simple to convert back, but it's just a little inconvenient. Not a big deal.

Andy

Subject: Re: IDL Average Value Graphs
Posted by [andybohn](#) on Thu, 17 Jul 2008 18:02:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jul 17, 1:19 pm, Jean H <jghas...@DELTHIS.ucalgary.ANDTHIS.ca> wrote:

>> 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

I needed the 2d coordinates because I'm dealing with FITS images, so I
need to grab the value of each of the pixels in a given region, unless
I'm missing an easier way to do this.

Sorry, this is actually my first program in IDL, so I'm trying to get
my head out of C# and C++ syntax and methods and into IDL. I missed
the arc_x written in the middle of the array. This has been very
helpful. I have one more question about POLYFILLV. If a line goes
through a pixel, does it not include this pixel in the region? Does
it include it if it is more than half way inside the region? If so,
what happens to perfectly split pixels?

Thanks a lot for the help,
Andy

Subject: Re: IDL Average Value Graphs
Posted by [David Fanning](#) on Thu, 17 Jul 2008 19:09:02 GMT

andybohn@gmail.com writes:

> When I ran POLYFILLV on a triangle, it returned just a 1d array of all
> the pixels in the triangle. I just wish it had an argument that
> returns a 2d array (the x and y coordinates of all of the pixels in
> the polygon). It's simple to convert back, but it's just a little
> inconvenient. Not a big deal.

I am still not understanding why you have to convert anything. These are the indices of the image that are inside the triangle. Why can't you use them directly?

I *believe* (although I am not 100% certain about this) that a pixel is considered "inside" the polygon if the lower-left corner of the pixel is inside. I don't recall the details now, but I do know there is a difference between POLYFILLV and what, for example, comes back from IDLanROI, which I think uses the center of the pixel.

This goes back too many years. I may have made a note about this in my FIND_BOUNDARY program, because it came up when I was working on that project.

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 Thu, 17 Jul 2008 21:52:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

> I needed the 2d coordinates because I'm dealing with FITS images, so I
> need to grab the value of each of the pixels in a given region, unless
> I'm missing an easier way to do this.

I am not familiar with FITS images, but I strongly believe you are missing an important IDL feature.
Read the help file under "Understanding Array Subscripts "

Elements of multidimensional arrays also can be specified using only one subscript, in which case the array is treated as a vector with the same number of points.

A0,0	A0,1
A0,1	A1,1
A0,2	A1,2

In the 2 by 3 element array, A, element A[2] is the same element as A[0, 1], and A[5] is the same element as A[1, 2].

>

I have one more question about POLYFILLV. If a line goes
> through a pixel, does it not include this pixel in the region? Does
> it include it if it is more than half way inside the region? If so,
> what happens to perfectly split pixels?
>
> Thanks a lot for the help,
> Andy

the algorithm used is a bit fuzzy.... people, including myself, often complain here that some pixels get selected while other do not (make a few tests). As David mentioned, IDLgrROI provides better results. Now it all depends on what you are doing... if you select a few 100 000 pixels, you might not care that much about having a few extra/missing pixels!

Jean

Subject: Re: IDL Average Value Graphs

Posted by [David Fanning](#) on Thu, 17 Jul 2008 23:02:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

Jean H writes:

> the algorithm used is a bit fuzzy.... people, including myself, often
> complain here that some pixels get selected while other do not (make a
> few tests). As David mentioned, IDLgrROI provides better results. Now it
> all depends on what you are doing... if you select a few 100 000 pixels,
> you might not care that much about having a few extra/missing pixels!

I found the discussion we were having way back in 2001.
And there was even a test program that shows the problem.
(It's still a problem, I guess, as I just ran it in IDL 7.0.3.)

<http://tinyurl.com/58cv6k>

Cheers,

David

P.S. Does time pass exponentially faster as you age? Weird. :-(

--

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 [andybohn](#) on Fri, 18 Jul 2008 01:11:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jul 17, 5:52 pm, Jean H <jghas...@DELTHIS.ucalgary.ANDTHIS.ca> wrote:

>> I needed the 2d coordinates because I'm dealing with FITS images, so I
>> need to grab the value of each of the pixels in a given region, unless
>> I'm missing an easier way to do this.

>

> I am not familiar with FITS images, but I strongly believe you are
> missing an important IDL feature.

> Read the help file under "Understanding Array Subscripts "

>

> *****

> Elements of multidimensional arrays also can be specified using only one
> subscript, in which case the array is treated as a vector with the same
> number of points.

> A0,0 A0,1

> A0,1 A1,1

> A0,2 A1,2

> In the 2 by 3 element array, A, element A[2] is the same element as A[0,
> 1], and A[5] is the same element as A[1, 2].

> *****

>

> I have one more question about POLYFILLV. If a line goes

>

>> through a pixel, does it not include this pixel in the region? Does
>> it include it if it is more than half way inside the region? If so,
>> what happens to perfectly split pixels?

>

>> Thanks a lot for the help,

>> Andy

>

> the algorithm used is a bit fuzzy.... people, including myself, often
> complain here that some pixels get selected while other do not (make a
> few tests). As David mentioned, IDLgrROI provides better results. Now it
> all depends on what you are doing... if you select a few 100 000 pixels,

> you might not care that much about having a few extra/missing pixels!
>
> Jean

Ah, I was not aware of this. I didn't know that you could access elements in a 2d array by calling the 1d equivalent. This isn't something I've seen in other languages. thanks for the tip

Subject: Re: IDL Average Value Graphs
Posted by [Jean H.](#) on Fri, 18 Jul 2008 17:20:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Jean H writes:
>
>> the algorithm used is a bit fuzzy.... people, including myself, often
>> complain here that some pixels get selected while other do not (make a
>> few tests). As David mentioned, IDLgrROI provides better results. Now it
>> all depends on what you are doing... if you select a few 100 000 pixels,
>> you might not care that much about having a few extra/missing pixels!
>
> I found the discussion we were having way back in 2001.
> And there was even a test program that shows the problem.
> (It's still a problem, I guess, as I just ran it in IDL 7.0.3.)
>
> <http://tinyurl.com/58cv6k>
>
> Cheers,
>
> David
>
> P.S. Does time pass exponentially faster as you age? Weird. :-(
>

hum... it was in November 2006 indeed :-) ... but I still feel as if it was yesterday... maybe because this issue was painful at this time (you know, your program is "almost" done, with "just" a few bugs/features to fix... and then you spend 3 weeks on a "detail"...)

Jean
