Subject: Re: PSF Energy inside circle
Posted by pgrigis on Wed, 23 Jul 2008 18:15:12 GMT
View Forum Message <> Reply to Message

Seems like a good situation for fitting a 2d gaussian peak to the
data.... This way you get all the parameters (intensity, FWHM)
directly...

Ciao,
Paolo

Michael Aye wrote:
> Dear all,
> as so often I am either too blind to find existing stuff or puzzled
> (if non-existing), that nobody did before what looks like a very usual
> task.
> What I want to know:
> Where in an image array (usual 2d-array with values, e.g. a CCD image)
> containing a centered 2d-gaussian light pulse lies the circle that
> contains 80 % (for example) of the "energy" of all the light on the
> image? I even only need it for the ideal situation where the center of
> the CCD aligns with the center of the 2d-gaussian light distribution.
> What I did so far:
> - Collected useful procedures like psf_gaussian, dist_circle and
> tvcircle.
> - Found the algorithm how to integrate from the center pixel towards
> outside, summing up the frame of pixels next to the previous frame. So
> my cumulative sum contains the sum of the date of 1, 9, 25 ... pixels.
>
> But I would like to go in circles, not squares! :)
> So how could I find and integrate the next "ring" of pixels? How would
> I even calculate the ever growing circumference correctly, taking into
> account that I have to sum up ever more pixels?
> Sounds like a horrible coding work and I am hoping somebody did all
> that already, because somehow that is something one would need to see
> how good an optical PSF is, or not?
>
> As usual, I am grateful for any help or hint to literature, procedures
> or calibration data of other experiments that might have done the
> same.
> Best regards,
> Michael

Subject: Re: PSF Energy inside circle
Posted by Kenneth P. Bowman on Wed, 23 Jul 2008 19:17:58 GMT
View Forum Message <> Reply to Message

In article
<8d5ea067-169e-4967-b3d9-29c2e14cf27e@f63g2000hsf.googlegroups.com>,
Michael Aye <kmichael.aye@googlemail.com> wrote:

> Dear all,
> as so often I am either too blind to find existing stuff or puzzled
> (if non-existing), that nobody did before what looks like a very usual
> task.
> What I want to know:
> Where in an image array (usual 2d-array with values, e.g. a CCD image)
> containing a centered 2d-gaussian light pulse lies the circle that
> contains 80 % (for example) of the "energy" of all the light on the
> image? I even only need it for the ideal situation where the center of
> the CCD aligns with the center of the 2d-gaussian light distribution.
> What I did so far:
> - Collected useful procedures like psf_gaussian, dist_circle and
> tvcircle.
> - Found the algorithm how to integrate from the center pixel towards
> outside, summing up the frame of pixels next to the previous frame. So
> my cumulative sum contains the sum of the date of 1, 9, 25 ... pixels.
>
> But I would like to go in circles, not squares! :)
> So how could I find and integrate the next "ring" of pixels? How would
> I even calculate the ever growing circumference correctly, taking into
> account that I have to sum up ever more pixels?
> Sounds like a horrible coding work and I am hoping somebody did all
> that already, because somehow that is something one would need to see
> how good an optical PSF is, or not?
>
> As usual, I am grateful for any help or hint to literature, procedures
> or calibration data of other experiments that might have done the
> same.
> Best regards,
> Michael

Compute the x and y coordinates of each pixel.

```
  x = REBIN(FINDGEN(nx), nx, ny)
  y = REBIN(REFORM(FINDGEN(ny), 1, ny), nx, ny)
```

You might want to add 0.5 to locate the pixel centers.

Compute the distance from each pixel to the central pixel

```
  d = SQRT((x - x0)^2 + (y - y0)^2)
```

Then find rings like this

```
   i = WHERE((d GE d1) AND (d LE d2), count)
```

Do what you want with those pixels.

You can put the WHERE statement in a loop and increment
d1 and d2 over whatever values you want.

Ken Bowman

Subject: Re: PSF Energy inside circle
Posted by pgrigis on Wed, 23 Jul 2008 20:24:23 GMT

View Forum Message <> Reply to Message

Kenneth P. Bowman wrote:
> In article
> <8d5ea067-169e-4967-b3d9-29c2e14cf27e@f63g2000hsf.googlegroups.com>,
> Michael Aye <kmichael.aye@googlemail.com> wrote:
>
>> Dear all,
>> as so often I am either too blind to find existing stuff or puzzled
>> (if non-existing), that nobody did before what looks like a very usual
>> task.
>> What I want to know:
>> Where in an image array (usual 2d-array with values, e.g. a CCD image)
>> containing a centered 2d-gaussian light pulse lies the circle that
>> contains 80 % (for example) of the "energy" of all the light on the
>> image? I even only need it for the ideal situation where the center of
>> the CCD aligns with the center of the 2d-gaussian light distribution.
>> What I did so far:
>> - Collected useful procedures like psf_gaussian, dist_circle and
>> tvcircle.
>> - Found the algorithm how to integrate from the center pixel towards
>> outside, summing up the frame of pixels next to the previous frame. So
>> my cumulative sum contains the sum of the date of 1, 9, 25 ... pixels.
>>
>> But I would like to go in circles, not squares! :)
>> So how could I find and integrate the next "ring" of pixels? How would
>> I even calculate the ever growing circumference correctly, taking into
>> account that I have to sum up ever more pixels?
>> Sounds like a horrible coding work and I am hoping somebody did all
>> that already, because somehow that is something one would need to see
>> how good an optical PSF is, or not?
>>
>> As usual, I am grateful for any help or hint to literature, procedures
>> or calibration data of other experiments that might have done the
>> same.
>> Best regards,

>> Michael
>
> Compute the x and y coordinates of each pixel.
>
>   x = REBIN(FINDGEN(nx), nx, ny)
>   y = REBIN(REFORM(FINDGEN(ny), 1, ny), nx, ny)
>
> You might want to add 0.5 to locate the pixel centers.
>
> Compute the distance from each pixel to the central pixel
>
>   d = SQRT((x - x0)^2 + (y - y0)^2)
>
> Then find rings like this
>
>   i = WHERE((d GE d1) AND (d LE d2), count)
>
> Do what you want with those pixels.

Well, if one wants to use that strategy, I suggest
another option for finding out the radial behaviour:

-convert x,y cartesian coordinates of all pixels to r,phi in polar
coordinates
-interpolate the irregular r, phi values grid to a regular r, phi grid
-integrate (i.e. sum) the values with constant r for all r
-plot that as a function of r

That should nicely show the radial dependence of the psf...

Ciao,
Paolo

>
> You can put the WHERE statement in a loop and increment
> d1 and d2 over whatever values you want.
>
> Ken Bowman

Subject: Re: PSF Energy inside circle
Posted by wlandsman on Wed, 23 Jul 2008 20:52:41 GMT

View Forum Message <> Reply to Message

On Jul 23, 2:03 pm, Michael Aye <kmichael....@googlemail.com> wrote:
> Dear all,

> So how could I find and integrate the next "ring" of pixels? How would
> I even calculate the ever growing circumference correctly, taking into
> account that I have to sum up ever more pixels?

The options mentioned so far seem fine for large radii. For small radii, the use of integer pixel values can introduce "noise" since a pixel must be either entirely within or outside the circle. The program aper.pro
(http://idlastro.gsfc.nasa.gov/ftp/pro/idlphot/aper.pro)
sums using partial pixels, using the exact overlap area of a pixel with a circle.

The program is somewhat awkward to use, in part because it is very old, and in part because it propagates error sources. But if you want to compute the flux within 10 concentric circles centered at the position [500.2, 500.5] with radii of 3, 3.5, 4, 4.5, ... 7.5 then the following is a start:

```
IDL> apr = 3. + findgen(10)/2.
IDL> aper,im, 500.2, 500.5, flux, eflux, sky,skyerr, 1, apr,/
flux,setsky=0
IDL> plot, apr, flux     ;plot encircled flux vs. radius
```

In this case I force the "sky" (background) to have a value of zero. You can have aper.pro compute the background by giving it an inner and outer "sky" radii far from the central source. --Wayne

Subject: Re: PSF Energy inside circle
Posted by Bob[3] on Thu, 24 Jul 2008 12:43:02 GMT

View Forum Message <> Reply to Message

On Jul 23, 3:17 pm, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
> In article
> < 8d5ea067-169e-4967-b3d9-29c2e14cf...@f63g2000hsf.googlegroup s.com >,
> Michael Aye <kmichael....@googlemail.com> wrote:
>
>
>
>
>
>> Dear all,
>> as so often I am either too blind to find existing stuff or puzzled

>> (if non-existing), that nobody did before what looks like a very usual
>> task.
>> What I want to know:
>> Where in an image array (usual 2d-array with values, e.g. a CCD image)
>> containing a centered 2d-gaussian light pulse lies the circle that
>> contains 80 % (for example) of the "energy" of all the light on the
>> image? I even only need it for the ideal situation where the center of
>> the CCD aligns with the center of the 2d-gaussian light distribution.
>> What I did so far:
>> - Collected useful procedures like psf_gaussian, dist_circle and
>> tvcircle.
>> - Found the algorithm how to integrate from the center pixel towards
>> outside, summing up the frame of pixels next to the previous frame. So
>> my cumulative sum contains the sum of the date of 1, 9, 25 ... pixels.
>
>> But I would like to go in circles, not squares! :)
>> So how could I find and integrate the next "ring" of pixels? How would
>> I even calculate the ever growing circumference correctly, taking into
>> account that I have to sum up ever more pixels?
>> Sounds like a horrible coding work and I am hoping somebody did all
>> that already, because somehow that is something one would need to see
>> how good an optical PSF is, or not?
>
>> As usual, I am grateful for any help or hint to literature, procedures
>> or calibration data of other experiments that might have done the
>> same.
>> Best regards,
>> Michael
>
> Compute the x and y coordinates of each pixel.
>
```
>    x = REBIN(FINDGEN(nx), nx, ny)
>    y = REBIN(REFORM(FINDGEN(ny), 1, ny), nx, ny)
```
>
> You might want to add 0.5 to locate the pixel centers.
>
> Compute the distance from each pixel to the central pixel
>
```
>    d = SQRT((x - x0)^2 + (y - y0)^2)
```
>
> Then find rings like this
>
```
>    i = WHERE((d GE d1) AND (d LE d2), count)
```
>
> Do what you want with those pixels.
>
> You can put the WHERE statement in a loop and increment
> d1 and d2 over whatever values you want.

>
> Ken Bowman- Hide quoted text -
>
> - Show quoted text -

Is there an advantage of using this method to determine d over using
DIST?

Subject: Re: PSF Energy inside circle
Posted by Kenneth P. Bowman on Fri, 25 Jul 2008 17:18:41 GMT
View Forum Message <> Reply to Message

In article
<808314ce-4419-478d-8593-bc94dd7f1789@p25g2000hsf.googlegroups.com>,
Bob Crawford <Snowman42@gmail.com> wrote:

> Is there an advantage of using this method to determine d over using
> DIST?

DIST computes a different function.

```
!P.MULTI = [0, 2, 1]
x  = FINDGEN(51)
xx = REBIN(x, 51, 51)
yy = REBIN(REFORM(x, 1, 51), 51, 51)
d  = SQRT((xx - 25.0)^2 + (yy - 25.0)^2)
SURFACE, d
SURFACE, DIST(51)
```

The Euclidian distance is a circular cone. DIST computes an
array proportional to the frequency of each element.

Ken Bowman

Subject: Re: PSF Energy inside circle
Posted by Bob[3] on Tue, 29 Jul 2008 15:38:23 GMT
View Forum Message <> Reply to Message

On Jul 25, 1:18 pm, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
> In article
> < 808314ce-4419-478d-8593-bc94dd7f1...@p25g2000hsf.googlegroup s.com >,
> Bob Crawford <Snowma...@gmail.com> wrote:
>
>> Is there an advantage of using this method to determine d over using
>> DIST?

>
> DIST computes a different function.
>
> !P.MULTI = [0, 2, 1]
> x = FINDGEN(51)
> xx = REBIN(x, 51, 51)
> yy = REBIN(REFORM(x, 1, 51), 51, 51)
> d = SQRT((xx - 25.0)^2 + (yy - 25.0)^2)
> SURFACE, d
> SURFACE, DIST(51)
>
> The Euclidian distance is a circular cone. DIST computes an
> array proportional to the frequency of each element.
>
> Ken Bowman

Thank you - but I was referring to the simplified situation as
described in this post:
http://groups.google.ca/group/comp.lang.idl-pvwave/browse_th read/thread/69b7be558f84e5be/2c185f37b1f50503?hl=en&lnk= gst#2c185f37b1f50503

i.e. replace:
SURFACE, DIST(51)

with:
SURFACE, SHIFT(DIST(51),25,25)

Subject: Re: PSF Energy inside circle
Posted by maye on Wed, 30 Jul 2008 22:06:04 GMT

View Forum Message <> Reply to Message

On Jul 23, 9:17 pm, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
> In article
> < 8d5ea067-169e-4967-b3d9-29c2e14cf...@f63g2000hsf.googlegroup s.com >,
> Michael Aye <kmichael....@googlemail.com> wrote:
>
>
>
>> Dear all,
>> as so often I am either too blind to find existing stuff or puzzled
>> (if non-existing), that nobody did before what looks like a very usual
>> task.
>> What I want to know:
>> Where in an image array (usual 2d-array with values, e.g. a CCD image)
>> containing a centered 2d-gaussian light pulse lies the circle that
>> contains 80 % (for example) of the "energy" of all the light on the
>> image? I even only need it for the ideal situation where the center of

>> the CCD aligns with the center of the 2d-gaussian light distribution.
>> What I did so far:
>> - Collected useful procedures like psf_gaussian, dist_circle and
>> tvcircle.
>> - Found the algorithm how to integrate from the center pixel towards
>> outside, summing up the frame of pixels next to the previous frame. So
>> my cumulative sum contains the sum of the date of 1, 9, 25 ... pixels.
>
>> But I would like to go in circles, not squares! :)
>> So how could I find and integrate the next "ring" of pixels? How would
>> I even calculate the ever growing circumference correctly, taking into
>> account that I have to sum up ever more pixels?
>> Sounds like a horrible coding work and I am hoping somebody did all
>> that already, because somehow that is something one would need to see
>> how good an optical PSF is, or not?
>
>> As usual, I am grateful for any help or hint to literature, procedures
>> or calibration data of other experiments that might have done the
>> same.
>> Best regards,
>> Michael
>
> Compute the x and y coordinates of each pixel.
>
>   x = REBIN(FINDGEN(nx), nx, ny)
>   y = REBIN(REFORM(FINDGEN(ny), 1, ny), nx, ny)
>
> You might want to add 0.5 to locate the pixel centers.
>
> Compute the distance from each pixel to the central pixel
>
>   d = SQRT((x - x0)^2 + (y - y0)^2)
>
> Then find rings like this
>
>   i = WHERE((d GE d1) AND (d LE d2), count)
>
> Do what you want with those pixels.
>
> You can put the WHERE statement in a loop and increment
> d1 and d2 over whatever values you want.
>

I tried your method, it works fine, thanks!
A side question:
For a 1D Gaussian, 68 % of events/energy/.. lies insides 1 sigma, how
is this number for a 2D Gaussian? I have a hard time to find
statistical tables for 2D Gaussians? And is it possible to get that
number analytically?

Best regards,
Michael