Subject: Re: PSF Energy inside circle

Posted by pariais 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