Subject: Re: fitting after rebinning
Posted by biophys on Fri, 07 Dec 2007 00:17:53 GMT
View Forum Message <> Reply to Message

Yes. Using /sample with compression is "morally" wrong in most cases. It makes sense only when you have a need for speed and don't care about the details. The fitting you are doing gives you absolutely the right answer. But how to understand the results. Well I do this kinda fitting almost everyday with CCD images. If you can imagine each data point of the image array as the readout of a CCD array, you will understand this almost immediately. In other words, the data represents the readout from the center of a pixel. Now psf_gaussian is on the other hand centered on the corner of four neighboring pixels which gives you always the .5 pixel offset. Let's take a look:

```
IDL>a=psf_gaussian(npix=512,fwhm=90)
IDL>print, a[254:257,254:257]
    0.998461     0.999145     0.999145     0.998461
    0.999145     0.999829     0.999829     0.999145
    0.999145     0.999829     0.999829     0.999145
    0.998461     0.999145     0.999145     0.998461

IDL>b=rebin(a,64,64)
IDL>print, b[30:33,30:33]
    0.903193     0.943497     0.943497     0.903193
    0.943497     0.985599     0.985599     0.943497
    0.943497     0.985599     0.985599     0.943497
    0.903193     0.943497     0.943497     0.903193
IDL> c=rebin(a,64,64,/sample)
IDL> print,c[30:33,30:33]
    0.848341     0.903490     0.920975     0.898555
    0.903490     0.962224     0.980846     0.956968
    0.920975     0.980846     0.999829     0.975489
    0.898555     0.956968     0.975489     0.951742
IDL> t=gauss2dfit(a,p1,/tilt)
Compiled module: GAUSS2DFIT.
Compiled module: GAUSSFIT.
Compiled module: POLY_FIT.
Compiled module: CURVEFIT.
Program caused arithmetic error: Floating divide by 0
Program caused arithmetic error: Floating underflow
IDL> print,p1
 2.19443e-008     0.999914      38.2195      38.2195      255.500
255.500     0.000000
IDL>t=gauss2dfit(b,p2,/tilt)
CURVEFIT: Failed to converge- CHISQ increasing without bound.
Program caused arithmetic error: Floating divide by 0
Program caused arithmetic error: Floating illegal operand
```

IDL> print,p2
 1.54614e-007   0.990991   4.78601   4.78601   31.5000
31.5000   0.000000

As you can see after rebin(w/o /sample), the center is still located
at the corner of a pixel. It makes sense that if you think pixel No 0
is located at [-0.5,0.5]x[-0.5x0.5] and so on so that [255.5,255.5] or
[31.5,31.5] is exactly what you would expect where the center should
be. To proper scale the results before and after rebin you just have
to get rid of the half pixel "offset" and do the scale and add back
the half pixel back after scaling. e.g.
255.5+0.5=(31.5+0.5)*(512/64)=(7.5+.5)*(512/16)

Cheers,
BP

On Dec 6, 1:53 pm, Greg Hennessy <greg.henne...@localhost.localdomain>
wrote:
>>> The offsets go away. Hmm, I need to think on this.
>
>> I would consider your assumptions on what happens when
>> you rebin. :-)
>
> Yea. If I call A the big image, B a rebin image not using sample, and
> C the rebin,/sample image, when I plotimage the three A and B look
> the most alike, with C having an offset, but the fitted parameters
> of A and C seem more alike. Like i said, I need to think.