
Subject: Re: Least square fitting
Posted by [Chris\[5\]](#) on Tue, 20 May 2008 11:16:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

On May 19, 11:41 pm, MichaelT <michael.theus...@gmail.com> wrote:

> Hi all,
>
> I have a problem with a rather complicated function depending on four
> parameters which I try to find using least square fitting and I don't
> know exactly how to do it.
>
> The basic problem is the following:
> I have an astronomical image of a star field and try to relate the sky
> coordinates (right ascension, declination) of the stars to the pixel
> coordinates (x, y).
>
> The function to relate the two depends on the not accurately known
> parameters focal length of the lens (f), the rotation of the field of
> view with respect of the north direction (beta) and the center sky
> coordinates of the image (bc, lc).
> I know the sky coordinates of certain stars accurately (bs, ls) as
> well as their pixel coordinates (xs, ys) and the center pixel
> coordinates of the image (xc, yc).
>
> Generally, the function is as follows (equivalent for ys', but the
> functions f1-f5 are slightly different):
>
> $xs' = f1(xc, f2(f, f3(ls, lc, bs, bc)), f4(ls, lc, bs, bc, f5(ls, lc,$
> $bs, bc)), beta)$
>
> The functions f1-f5 look rather complicated and contain many sines,
> cosines and acos etc.
>
> I then did the following as I didn't know better:
> First I minimized all this with respect of beta, then lc and bc and
> after that f starting with some initial values for the four
> parameters. Then I started all over again, beta, lc, bc, f (in a loop)
> until the difference between the known coordinates (xs, ys) and the
> calculated ones (xs', ys') reached a minimum.
>
> However, the minimum deviation reached at the end of the loop pretty
> much depends on the initially chosen values of beta, lc, bc and f.
>
> There certainly is a more optimal way to do the least square fitting
> than the one that I have chosen (partial derivatives etc.), but that
> f1 depends on f2 and f4 and these again depend on f3 and f5 is giving
> me some unsolvable problems.
>

- > How should I proceed? What would be a good method to minimize this?
- >
- > Thanks, Michael
- >
- > The main problem now is for me, that f1 depends on f2 and f4 and that
- > f2 and f4 themselves depend on f3 and f5.

How big is the field of view? If it is small enough (and the optics consistent enough) that you don't need to worry about the pixel scale (number of arcsecs/pixel) changing over the image, then the problem seems easier, since most of the parameters to solve for ought to be independent. I don't know how statistically kosher this is, but here are some thoughts:

Assuming that the pixel scale p in arcsec/pixel is a constant, then the RA,DEC (a,d) of stars are related to their pixel coords (x,y) by

$$a = a_c + ((x - x_c) \cos(\beta) - (y - y_c) \sin(\beta)) * p$$

$$d = d_c + ((x - x_c) \sin(\beta) + (y - y_c) \cos(\beta)) * p$$

where a_c, d_c, x_c, y_c are the sky and pixel coordinates of the center pixel. β is the angle of rotation of the image y axis with respect to north. I guess also here I've assumed that RA increases to the right (backwards from convention) and that the image is near the celestial equator so that 1 second of right ascension = 1 second of declination. Both of these are easy to generalize (switch some of the signs for East pointing left, and add in a $\cos(\text{dec})$ term for off-equator correction). Neither of these should affect the following analysis.

For every (a,d) (x,y) pair, compute the pixel distance and angular separation. A simple linear regression then determines p (an easier but sloppier method would be to calculate p for each pair and take a weighted average). p is independent of all of the other parameters, so (to me) it seems like you could do this separately.

Now that you know p , for a bunch of pairs of stars, compute the x displacement in pixels and the RA displacement in arcsec. Convert the x displacement in pixels to an angular distance using p . Now use the equations above to solve for β . (you can also use the y coordinates and/or decs just as easily).

Finally, determining a_c and d_c is one more linear regression.

Is this the sort of thing you are trying to do?

Chris

Subject: Re: Least square fitting
Posted by [MichaelT](#) on Tue, 20 May 2008 14:06:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks for your thoughts, Chris!

But I think the field of view is not small enough to do it this way (about $3.5^\circ \times 2.5^\circ$ and larger). The focal length is shorter than 400 mm. Otherwise, it could very well work the way you described it, I think.

These are the things I'd like to do in the end:

I have several images of the same object, but from different nights taken with an amateur instrument. So the fov is slightly different each time as well as the rotation angle (it's not a stationary scope). Knowing all the parameters would enable me to exactly overlay the images to increase the signal-to-noise ratio. I know that there are various free programs (non-IDL) that can already do that, but they all have some shortcomings. So I tried to do something myself.

Also, you could then use some neat routines from the IDL astronomy library (<http://idlastro.gsfc.nasa.gov/>) and automatically generate an overlay from the USNO catalog to determine the limiting magnitude or to label objects (e.g. `queryvizier` and `quersimbad`).

There are many other things you could do if you knew the pixel-to-coordinate conversion.

Michael
