Subject: Re: Least squares fit of a model to a skeleton consisting out of 3D points.
Posted by Wout De Nolf on Mon, 24 Nov 2008 16:22:53 GMT
View Forum Message <> Reply to Message

On Mon, 24 Nov 2008 05:33:18 -0800 (PST), Johan <johan@jmarais.com>
wrote:

> I have the following problem to solve and was wondering whether the
> mpfit routines of Craig Markwardt will do the job?
>
> Do have the following model:
> Let g(X,Y,Z)=1 be a quadratic function in the coordinate system
> (O,Z,Y,Z) defined by the long, horizontal and vertical axes
> (ellipsoid). Write the equation of this quadratic function in matrix
> notation as follows:
>
> g(X,Y,Z) = [X, Y, Z]*[[A1,A4,A5],[A4,A2,A6],[A5,A6,A3]]*[[X],[Y],[Z]]
> + [X, Y, Z]*[[A7],[A8],[A9]]
>
> Need to fit this model to a 3D skeleton of N points by using least
> squares by calculating the coefficients Ai .
>
> This is achieved by minimizing the total squared error between the
> exact position of the points (Xi, Yi, Zi) on the quadratic surface and
> their real position in the coordinate system (O, X, Y, Z). The
> minimizing is performed from the derivative of the equation below with
> respect to A1 ... A9:
>
> J(A1 ... A9) = for i=0,N sigma(1 ï¿½ (Xi, Yi, Zi))^2
>
> This equation yields a linear system of nine equations in which the
> values of coefficients A1 ... A9 are unknown.
>
> Anyone that can help?

Search the IDL Code Library on the ITTVIS website for Ronn Kling's
"Ellipse and Ellipsoid fitting routine" (krellipsoidfit.pro) I think
this uses orthogonal distance regression, not non-linear least squares
fitting.

If x, y and z are measured, orthogonal distance regression is the way
to go. If you insist on using NLLSQ, you could use mpfit by setting
X=[X,Y,Z] ; (you need to extract the seperate X, Y and Z in your user
routine)
Y=replicate(1,n_elements(X))
assuming g(x,y,z)=1

I'm not sure whether that works (e.g. what are the weights for Y? I'd

guess all 1...). Someone else will probably be able to tell you why to use ODR instead of NLLSQ.