6.8 Least squares linear fits

It should be said that there is no good reason for the "least distance" algorithm – if x and y both contain errors, the point they delineate will not necessarily move perpendicular to the true line on which they would lie if there were no errors. This is really just a quick and dirty way of seeing how much the answer may be affected if we relax the assumption of x (say) being error-free.
Suppose the true line has the equation

$$y = mx + c$$

and we are concerned with the data point (X_i, Y_i). The line joining this point to the true line must have slope $-1/m$, and since it must pass through (X_i, Y_i) we can find its equation. It is then straightforward to find the intersection of the two lines and thus the distance d_i between them, which is

$$d_i^2 = \frac{mX_i + c - Y_i}{1 + m^2}.$$

This is non-linear in m so we have to minimize $\sum_i d_i^2$ numerically.
A fully Bayesian solution is available for the problem of fitting a straight line when there are errors in x and y - see Gull, 1989, ("Bayesian Data Analysis: Straight-line Fitting," in Maximum Entropy and Bayesian Methods, J. Skilling (ed), Kluwer Academic Publishers) - a PDF is linked from these solutions. The derivation is rather subtle and, as in the above case, there is no closed-form solution for the parameters.