Least Squares Fitting of Data

David Eberly
Geometric Tools, LLC
http://www.geometrictools.com/

Created: July 15, 1999
Last Modified: February 9, 2008

Contents

This document describes some algorithms for fitting 2D or 3D point sets by linear or quadratic structures using least squares minimization.

1 Linear Fitting of 2D Points of Form $(x, f(x))$

This is the usual introduction to least squares fit by a line when the data represents measurements where the y-component is assumed to be functionally dependent on the x-component. Given a set of samples $\{(x_i, y_i)\}_{i=1}^m$, determine A and B so that the line $y = Ax + B$ best fits the samples in the sense that the sum of the squared errors between the y_i and the line values $Ax_i + B$ is minimized. Note that the error is measured only in the y-direction.

Define $E(A, B) = \sum_{i=1}^m [(Ax_i + B) - y_i]^2$. This function is nonnegative and its graph is a paraboloid whose vertex occurs when the gradient satistfies $\nabla E = (0, 0)$. This leads to a system of two linear equations in A and B which can be easily solved. Precisely,

$$(0, 0) = \nabla E = 2 \sum_{i=1}^{m} [(Ax_i + B) - y_i](x_i, 1)$$

and so

$$\begin{bmatrix} \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m 1 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m x_i y_i \\ \sum_{i=1}^m y_i \end{bmatrix}.$$

The solution provides the least squares solution $y = Ax + B$.

2 Linear Fitting of nD Points Using Orthogonal Regression

It is also possible to fit a line using least squares where the errors are measured *orthogonally* to the proposed line rather than measured vertically. The following argument holds for sample points and lines in n dimensions. Let the line be $\mathbf{L}(t) = t\mathbf{D} + \mathbf{A}$ where $\mathbf{D}$ is unit length. Define $\mathbf{X}_i$ to be the sample points; then

$$\mathbf{X}_i = \mathbf{A} + d_i\mathbf{D} + p_i\mathbf{D}_i^{\perp}$$

where $d_i = \mathbf{D} \cdot (\mathbf{X}_i - \mathbf{A})$ and $\mathbf{D}_i^{\perp}$ is some unit length vector perpendicular to $\mathbf{D}$ with appropriate coefficient p_i. Define $\mathbf{Y}_i = \mathbf{X}_i - \mathbf{A}$. The vector from $\mathbf{X}_i$ to its projection onto the line is

$$\mathbf{Y}_i - d_i\mathbf{D} = p_i\mathbf{D}_i^{\perp}.$$

The squared length of this vector is $p_i^2 = (\mathbf{Y}_i - d_i\mathbf{D})^2$. The energy function for the least squares minimization is $E(\mathbf{A}, \mathbf{D}) = \sum_{i=1}^m p_i^2$. Two alternate forms for this function are

$$E(\mathbf{A}, \mathbf{D}) = \sum_{i=1}^{m} \left(\mathbf{Y}_i^{\mathrm{T}} \left[I - \mathbf{D}\mathbf{D}^{\mathrm{T}} \right] \mathbf{Y}_i \right)$$

and

$$E(\mathbf{A}, \mathbf{D}) = \mathbf{D}^{\mathrm{T}} \left(\sum_{i=1}^{m} \left[(\mathbf{Y}_i \cdot \mathbf{Y}_i) I - \mathbf{Y}_i \mathbf{Y}_i^{\mathrm{T}} \right] \right) \mathbf{D} = \mathbf{D}^{\mathrm{T}} M(A) \mathbf{D}.$$

Using the first form of E in the previous equation, take the derivative with respect to A to get

$$\frac{\partial E}{\partial A} = -2\left[I - \mathbf{D}\mathbf{D}^{\mathrm{T}}\right]\sum_{i=1}^{m}\mathbf{Y}_i.$$

This partial derivative is zero whenever $\sum_{i=1}^{m}\mathbf{Y}_i = 0$ in which case $\mathbf{A} = (1/m)\sum_{i=1}^{m}\mathbf{X}_i$ (the average of the sample points).

Given $\mathbf{A}$, the matrix $M(A)$ is determined in the second form of the energy function. The quantity $\mathbf{D}^{\mathrm{T}}M(A)\mathbf{D}$ is a quadratic form whose minimum is the smallest eigenvalue of $M(A)$. This can be found by standard eigensystem solvers. A corresponding unit length eigenvector $\mathbf{D}$ completes our construction of the least squares line.

For $n = 2$, if $\mathbf{A} = (a, b)$, then matrix $M(A)$ is given by

$$M(A) = \left(\sum_{i=1}^{m}(x_i - a)^2 + \sum_{i=1}^{n}(y_i - b)^2\right)\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \sum_{i=1}^{m}(x_i - a)^2 & \sum_{i=1}^{m}(x_i - a)(y_i - b) \\ \sum_{i=1}^{m}(x_i - a)(y_i - b) & \sum_{i=1}^{m}(y_i - b)^2 \end{bmatrix}.$$

For $n = 3$, if $\mathbf{A} = (a, b, c)$, then matrix $M(A)$ is given by

$$M(A) = \delta\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \sum_{i=1}^{m}(x_i - a)^2 & \sum_{i=1}^{m}(x_i - a)(y_i - b) & \sum_{i=1}^{m}(x_i - a)(z_i - c) \\ \sum_{i=1}^{m}(x_i - a)(y_i - b) & \sum_{i=1}^{m}(y_i - b)^2 & \sum_{i=1}^{m}(y_i - b)(z_i - c) \\ \sum_{i=1}^{m}(x_i - a)(z_i - c) & \sum_{i=1}^{m}(y_i - b)(z_i - c) & \sum_{i=1}^{m}(z_i - c)^2 \end{bmatrix}$$

where

$$\delta = \sum_{i=1}^{m}(x_i - a)^2 + \sum_{i=1}^{m}(y_i - b)^2 + \sum_{i=1}^{m}(z_i - c)^2.$$

3 Planar Fitting of 3D Points of Form $(x, y, f(x, y))$

The assumption is that the z-component of the data is functionally dependent on the x- and y-components. Given a set of samples $\{(x_i, y_i, z_i)\}_{i=1}^{m}$, determine A, B, and C so that the plane $z = Ax + By + C$ best fits the samples in the sense that the sum of the squared errors between the z_i and the plane values $Ax_i + By_i + C$ is minimized. Note that the error is measured only in the z-direction.

Define $E(A, B, C) = \sum_{i=1}^{m}[(Ax_i + By_i + C) - z_i]^2$. This function is nonnegative and its graph is a hyperparaboloid whose vertex occurs when the gradient satistfies $\nabla E = (0, 0, 0)$. This leads to a system of three linear equations in A, B, and C which can be easily solved. Precisely,

$$(0, 0, 0) = \nabla E = 2\sum_{i=1}^{m}[(Ax_i + By_i + C) - z_i](x_i, y_i, 1)$$

and so

$$\begin{bmatrix} \sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i y_i & \sum_{i=1}^{m} x_i \\ \sum_{i=1}^{m} x_i y_i & \sum_{i=1}^{m} y_i^2 & \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} y_i & \sum_{i=1}^{m} 1 \end{bmatrix}\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} x_i z_i \\ \sum_{i=1}^{m} y_i z_i \\ \sum_{i=1}^{m} z_i \end{bmatrix}.$$

The solution provides the least squares solution $z = Ax + By + C$.

4 Hyperplanar Fitting of nD Points Using Orthogonal Regression

It is also possible to fit a plane using least squares where the errors are measured *orthogonally* to the proposed plane rather than measured vertically. The following argument holds for sample points and hyperplanes in n dimensions. Let the hyperplane be $\mathbf{N} \cdot (\mathbf{X} - \mathbf{A}) = 0$ where $\mathbf{N}$ is a unit length normal to the hyperplane and $\mathbf{A}$ is a point on the hyperplane. Define $\mathbf{X}_i$ to be the sample points; then

$$\mathbf{X}_i = \mathbf{A} + \lambda_i \mathbf{N} + p_i \mathbf{N}_i^{\perp}$$

where $\lambda_i = \mathbf{N} \cdot (\mathbf{X}_i - \mathbf{A})$ and $\mathbf{N}_i^{\perp}$ is some unit length vector perpendicular to $\mathbf{N}$ with appropriate coefficient p_i. Define $\mathbf{Y}_i = \mathbf{X}_i - \mathbf{A}$. The vector from $\mathbf{X}_i$ to its projection onto the hyperplane is $\lambda_i \mathbf{N}$. The squared length of this vector is $\lambda_i^2 = (\mathbf{N} \cdot \mathbf{Y}_i)^2$. The energy function for the least squares minimization is $E(\mathbf{A}, \mathbf{N}) = \sum_{i=1}^{m} \lambda_i^2$. Two alternate forms for this function are

$$E(\mathbf{A}, \mathbf{N}) = \sum_{i=1}^{m} \left(\mathbf{Y}_i^{\mathrm{T}} \left[\mathbf{N}\mathbf{N}^{\mathrm{T}} \right] \mathbf{Y}_i \right)$$

and

$$E(\mathbf{A}, \mathbf{N}) = \mathbf{N}^{\mathrm{T}} \left(\sum_{i=1}^{m} \mathbf{Y}_i \mathbf{Y}_i^{\mathrm{T}} \right) \mathbf{N} = \mathbf{N}^{\mathrm{T}} M(A) \mathbf{N}.$$

Using the first form of E in the previous equation, take the derivative with respect to A to get

$$\frac{\partial E}{\partial A} = -2 \left[\mathbf{N}\mathbf{N}^{\mathrm{T}} \right] \sum_{i=1}^{m} \mathbf{Y}_i.$$

This partial derivative is zero whenever $\sum_{i=1}^{m} \mathbf{Y}_i = 0$ in which case $\mathbf{A} = (1/m) \sum_{i=1}^{m} \mathbf{X}_i$ (the average of the sample points).

Given $\mathbf{A}$, the matrix $M(A)$ is determined in the second form of the energy function. The quantity $\mathbf{N}^{\mathrm{T}} M(A) \mathbf{N}$ is a quadratic form whose minimum is the smallest eigenvalue of $M(A)$. This can be found by standard eigensystem solvers. A corresponding unit length eigenvector $\mathbf{N}$ completes our construction of the least squares hyperplane.

For $n = 3$, if $\mathbf{A} = (a, b, c)$, then matrix $M(A)$ is given by

$$M(A) = \begin{bmatrix} \sum_{i=1}^{m} (x_i - a)^2 & \sum_{i=1}^{m} (x_i - a)(y_i - b) & \sum_{i=1}^{m} (x_i - a)(z_i - c) \\ \sum_{i=1}^{m} (x_i - a)(y_i - b) & \sum_{i=1}^{m} (y_i - b)^2 & \sum_{i=1}^{m} (y_i - b)(z_i - c) \\ \sum_{i=1}^{m} (x_i - a)(z_i - c) & \sum_{i=1}^{m} (y_i - b)(z_i - c) & \sum_{i=1}^{m} (z_i - c)^2 \end{bmatrix}.$$

5 Fitting a Circle to 2D Points

Given a set of points $\{(x_i, y_i)\}_{i=1}^{m}$, $m \geq 3$, fit them with a circle $(x - a)^2 + (y - b)^2 = r^2$ where (a, b) is the circle center and r is the circle radius. An assumption of this algorithm is that not all the points are collinear. The energy function to be minimized is

$$E(a, b, r) = \sum_{i=1}^{m} (L_i - r)^2$$

where $L_i = \sqrt{(x_i - a)^2 + (y_i - b)^2}$. Take the partial derivative with respect to r to obtain

$$\frac{\partial E}{\partial r} = -2 \sum_{i=1}^{m} (L_i - r).$$

Setting equal to zero yields

$$r = \frac{1}{m} \sum_{i=1} L_i.$$

Take the partial derivative with respect to a to obtain

$$\frac{\partial E}{\partial a} = -2 \sum_{i=1}^{m} (L_i - r) \frac{\partial L_i}{\partial a} = 2 \sum_{i=1}^{m} \left((x_i - a) + r \frac{\partial L_i}{\partial a} \right)$$

and take the partial derivative with respect to b to obtain

$$\frac{\partial E}{\partial b} = -2 \sum_{i=1}^{m} (L_i - r) \frac{\partial L_i}{\partial b} = 2 \sum_{i=1}^{m} \left((y_i - b) + r \frac{\partial L_i}{\partial b} \right).$$

Setting these two derivatives equal to zero yields

$$a = \frac{1}{m} \sum_{i=1}^{m} x_i + r \frac{1}{m} \sum_{i=1}^{m} \frac{\partial L_i}{\partial a}$$

and

$$b = \frac{1}{m} \sum_{i=1}^{m} y_i + r \frac{1}{m} \sum_{i=1}^{m} \frac{\partial L_i}{\partial b}.$$

Replacing r by its equivalent from $\partial E / \partial r = 0$ and using $\partial L_i / \partial a = (a - x_i)/L_i$ and $\partial L_i / \partial b = (b - y_i)/L_i$, we get two nonlinear equations in a and b:

$$a = \bar{x} + \bar{L} \bar{L}_a =: F(a, b)$$
$$b = \bar{y} + \bar{L} \bar{L}_b =: G(a, b)$$

where

$$\begin{aligned} \bar{x} &= \tfrac{1}{m} \textstyle\sum_{i=1}^{m} x_i \\ \bar{y} &= \tfrac{1}{m} \textstyle\sum_{i=1}^{m} y_i \\ \bar{L} &= \tfrac{1}{m} \textstyle\sum_{i=1}^{m} L_i \\ \bar{L}_a &= \tfrac{1}{m} \textstyle\sum_{i=1}^{m} \frac{a - x_i}{L_i} \\ \bar{L}_b &= \tfrac{1}{m} \textstyle\sum_{i=1}^{m} \frac{b - y_i}{L_i} \end{aligned}$$

Fixed point iteration can be applied to solving these equations: $a_0 = \bar{x}$, $b_0 = \bar{y}$, and $a_{i+1} = F(a_i, b_i)$ and $b_{i+1} = G(a_i, b_i)$ for $i \geq 0$. *Warning. I have not analyzed the convergence properties of this algorithm. In a few experiments it seems to converge just fine.*

6 Fitting a Sphere to 3D Points

Given a set of points $\{(x_i, y_i, z_i)\}_{i=1}^m$, $m \geq 4$, fit them with a sphere $(x-a)^2 + (y-b)^2 + (z-c)^2 = r^2$ where (a, b, c) is the sphere center and r is the sphere radius. An assumption of this algorithm is that not all the points are coplanar. The energy function to be minimized is

$$E(a, b, c, r) = \sum_{i=1}^{m} (L_i - r)^2$$

where $L_i = \sqrt{(x_i - a)^2 + (y_i - b)^2 + (z_i - c)}$. Take the partial derivative with respect to r to obtain

$$\frac{\partial E}{\partial r} = -2 \sum_{i=1}^{m} (L_i - r).$$

Setting equal to zero yields

$$r = \frac{1}{m} \sum_{i=1} L_i.$$

Take the partial derivative with respect to a to obtain

$$\frac{\partial E}{\partial a} = -2 \sum_{i=1}^{m} (L_i - r) \frac{\partial L_i}{\partial a} = 2 \sum_{i=1}^{m} \left((x_i - a) + r \frac{\partial L_i}{\partial a} \right),$$

take the partial derivative with respect to b to obtain

$$\frac{\partial E}{\partial b} = -2 \sum_{i=1}^{m} (L_i - r) \frac{\partial L_i}{\partial b} = 2 \sum_{i=1}^{m} \left((y_i - b) + r \frac{\partial L_i}{\partial b} \right),$$

and take the partial derivative with respect to c to obtain

$$\frac{\partial E}{\partial c} = -2 \sum_{i=1}^{m} (L_i - r) \frac{\partial L_i}{\partial c} = 2 \sum_{i=1}^{m} \left((z_i - c) + r \frac{\partial L_i}{\partial c} \right).$$

Setting these three derivatives equal to zero yields

$$a = \frac{1}{m} \sum_{i=1}^{m} x_i + r \frac{1}{m} \sum_{i=1}^{m} \frac{\partial L_i}{\partial a}$$

and

$$b = \frac{1}{m} \sum_{i=1}^{m} y_i + r \frac{1}{m} \sum_{i=1}^{m} \frac{\partial L_i}{\partial b}.$$

and

$$c = \frac{1}{m} \sum_{i=1}^{m} z_i + r \frac{1}{m} \sum_{i=1}^{m} \frac{\partial L_i}{\partial c}.$$

Replacing r by its equivalent from $\partial E/\partial r = 0$ and using $\partial L_i/\partial a = (a - x_i)/L_i$, $\partial L_i/\partial b = (b - y_i)/L_i$, and $\partial L_i/\partial c = (c - z_i)/L_i$, we get three nonlinear equations in a, b, and c:

$$a = \bar{x} + \bar{L}\bar{L}_a =: F(a, b, c)$$
$$b = \bar{y} + \bar{L}\bar{L}_b =: G(a, b, c)$$
$$c = \bar{z} + \bar{L}\bar{L}_c =: H(a, b, c)$$

where

$$\begin{aligned}
\bar{x} &= \tfrac{1}{m}\sum_{i=1}^{m} x_i \\
\bar{y} &= \tfrac{1}{m}\sum_{i=1}^{m} y_i \\
\bar{z} &= \tfrac{1}{m}\sum_{i=1}^{m} z_i \\
\bar{L} &= \tfrac{1}{m}\sum_{i=1}^{m} L_i \\
\bar{L}_a &= \tfrac{1}{m}\sum_{i=1}^{m} \tfrac{a-x_i}{L_i} \\
\bar{L}_b &= \tfrac{1}{m}\sum_{i=1}^{m} \tfrac{b-y_i}{L_i} \\
\bar{L}_c &= \tfrac{1}{m}\sum_{i=1}^{m} \tfrac{c-z_i}{L_i}
\end{aligned}$$

Fixed point iteration can be applied to solving these equations: $a_0 = \bar{x}$, $b_0 = \bar{y}$, $c_0 = \bar{z}$, and $a_{i+1} = F(a_i, b_i, c_i)$, $b_{i+1} = G(a_i, b_i, c_i)$, and $c_{i+1} = H(a_i, b_i, c_i)$ for $i \geq 0$. *Warning. I have not analyzed the convergence properties of this algorithm. In a few experiments it seems to converge just fine.*

7 Fitting an Ellipse to 2D Points

Given a set of points $\{\mathbf{X}_i\}_{i=1}^m$, $m \geq 3$, fit them with an ellipse $(\mathbf{X}-\mathbf{U})^{\mathrm{T}} R^{\mathrm{T}} D R (\mathbf{X}-\mathbf{U}) = 1$ where $\mathbf{U}$ is the ellipse center, R is an orthonormal matrix representing the ellipse orientation, and D is a diagonal matrix whose diagonal entries represent the reciprocal of the squares of the half-lengths lengths of the axes of the ellipse. An axis-aligned ellipse with center at the origin has equation $(x/a)^2 + (y/b)^2 = 1$. In this setting, $\mathbf{U} = (0,0)$, $R = I$ (the identity matrix), and $D = \mathrm{diag}(1/a^2, 1/b^2)$. The energy function to be minimized is

$$E(\mathbf{U}, R, D) = \sum_{i=1}^{m} (L_i - r)^2$$

where L_i is the distance from $\mathbf{X}_i$ to the ellipse with the given parameters.

This problem is more difficult than that of fitting circles. The distance L_i requires finding roots to a quartic polynomial. While there are closed form formulas for the roots of a quartic, these formulas are not easily manipulated algebraically or differentiated to produce an algorithm such as the one for a circle. The approach instead is to use an iterative minimizer to compute the minimum of E.

7.1 Distance From Point to Ellipse

It is sufficient to solve this problem when the ellipse is axis-aligned. For other ellipses, they can be rotated and translated to an axis-aligned ellipse centered at the origin and the distance can be measured in that system. The basic idea can be found in Graphics Gems IV (an article by John Hart on computing distance between point and ellipsoid).

Let (u, v) be the point in question. Let the ellipse be $(x/a)^2 + (y/b)^2 = 1$. The closest point (x, y) on the ellipse to (u, v) must occur so that $(x-u, y-v)$ is normal to the ellipse. Since an ellipse normal is $\nabla((x/a)^2+(y/b)^2) = (x/a^2, y/b^2)$, the orthogonality condition implies that $u-x = t*x/a^2$ and $v-y = t*y/b^2$

for some t. Solving yields $x = a^2u/(t+a^2)$ and $y = b^2v/(t+b^2)$. Replacing in the ellipse equation yields

$$\left(\frac{au}{t+a^2}\right)^2 + \left(\frac{bv}{t+b^2}\right)^2 = 1.$$

Multiplying through by the denominators yields the quartic polynomial

$$F(t) = (t+a^2)^2(t+b^2)^2 - a^2u^2(t+b^2)^2 - b^2v^2(t+a^2)^2 = 0.$$

The largest root $\bar{t}$ of the polynomial corresponds to the closest point on the ellipse.

The largest root can be found by a Newton's iteration scheme. If (u,v) is inside the ellipse, then $t_0 = 0$ is a good initial guess for the iteration. If (u,v) is outside the ellipse, then $t_0 = \max\{a,b\}\sqrt{u^2+v^2}$ is a good initial guess. The iteration itself is

$$t_{i+1} = t_i - F(t_i)/F'(t_i), \quad i \geq 0.$$

Some numerical issues need to be addressed. For (u,v) near the coordinate axes, the algorithm is ill-conditioned. You need to handle those cases separately. Also, if a and b are large, then $F(t_i)$ can be quite large. In these cases you might consider uniformly scaling the data to $O(1)$ as floating point numbers first, compute distance, then rescale to get the distance in the original coordinates.

7.2 Minimization of the Energy Function

TO BE WRITTEN LATER. (The code at the web site uses a variation on Powell's direction set method.)

8 Fitting an Ellipsoid to 3D Points

Given a set of points $\{\mathbf{X}_i\}_{i=1}^m$, $m \geq 3$, fit them with an ellipsoid $(\mathbf{X}-\mathbf{U})^{\mathrm{T}}R^{\mathrm{T}}DR(\mathbf{X}-\mathbf{U}) = 1$ where $\mathbf{U}$ is the ellipsoid center and R is an orthonormal matrix representing the ellipsoid orientation. The matrix D is a diagonal matrix whose diagonal entries represent the reciprocal of the squares of the half-lengths of the axes of the ellipsoid. An axis-aligned ellipsoid with center at the origin has equation $(x/a)^2+(y/b)^2+(z/c)^2 = 1$. In this setting, $\mathbf{U} = (0,0,0)$, $R = I$ (the identity matrix), and $D = \mathrm{diag}(1/a^2, 1/b^2, 1/c^2)$. The energy function to be minimized is

$$E(\mathbf{U}, R, D) = \sum_{i=1}^{m}(L_i - r)^2$$

where L_i is the distance from $\mathbf{X}_i$ to the ellipse with the given parameters.

This problem is more difficult than that of fitting spheres. The distance L_i requires finding roots to a sixth degree polynomial. There are no closed formulas for the roots of such polynomials. The approach instead is to use an iterative minimizer to compute the minimum of E.

8.1 Distance From Point to Ellipsoid

It is sufficient to solve this problem when the ellipsoid is axis-aligned. For other ellipsoids, they can be rotated and translated to an axis-aligned ellipsoid centered at the origin and the distance can be measured

in that system. The basic idea can be found in Graphics Gems IV (an article by John Hart on computing distance between point and ellipsoid).

Let (u, v, w) be the point in question. Let the ellipse be $(x/a)^2 + (y/b)^2 + (z/c)^2 = 1$. The closest point (x, y, z) on the ellipsoid to (u, v) must occur so that $(x - u, y - v, z - w)$ is normal to the ellipsoid. Since an ellipsoid normal is $\nabla((x/a)^2 + (y/b)^2 + (z/c)^2) = (x/a^2, y/b^2, z/c^2)$, the orthogonality condition implies that $u - x = t * x/a^2$, $v - y = t * y/b^2$, and $w - z = t * z/c^2$ for some t. Solving yields $x = a^2u/(t + a^2)$, $y = b^2v/(t + b^2)$, and $z = c^2w/(t + c^2)$. Replacing in the ellipsoid equation yields

$$\left(\frac{au}{t+a^2}\right)^2 + \left(\frac{bv}{t+b^2}\right)^2 + \left(\frac{cw}{t+c^2}\right)^2 = 1.$$

Multiplying through by the denominators yields the sixth degree polynomial

$$F(t) = (t+a^2)^2(t+b^2)^2(t+c^2)^2 - a^2u^2(t+b^2)^2(t+c^2)^2 - b^2v^2(t+a^2)^2(t+c^2)^2 - c^2w^2(t+a^2)^2(t+b^2)^2 = 0.$$

The largest root $\bar{t}$ of the polynomial corresponds to the closest point on the ellipse.

The largest root can be found by a Newton's iteration scheme. If (u, v, w) is inside the ellipse, then $t_0 = 0$ is a good initial guess for the iteration. If (u, v, w) is outside the ellipse, then $t_0 = \max\{a, b, c\}\sqrt{u^2 + v^2 + w^2}$ is a good initial guess. The iteration itself is

$$t_{i+1} = t_i - F(t_i)/F'(t_i), \quad i \geq 0.$$

Some numerical issues need to be addressed. For (u, v, w) near the coordinate planes, the algorithm is ill-conditioned. You need to handle those cases separately. Also, if a, b, and c are large, then $F(t_i)$ can be quite large. In these cases you might consider uniformly scaling the data to $O(1)$ as floating point numbers first, compute distance, then rescale to get the distance in the original coordinates.

8.2 Minimization of the Energy Function

TO BE WRITTEN LATER. (The code at the web site uses a variation on Powell's direction set method.)

9 Fitting a Paraboloid to 3D Points of the Form $(x, y, f(x, y))$

Given a set of samples $\{(x_i, y_i, z_i)\}_{i=1}^m$ and assuming that the true values lie on a paraboloid

$$z = f(x, y) = p_1x^2 + p_2xy + p_3y^2 + p_4x + p_5y + p_6 = \mathbf{P} \cdot \mathbf{Q}(x, y)$$

where $\mathbf{P} = (p_1, p_2, p_3, p_4, p_5, p_6)$ and $\mathbf{Q}(x, y) = (x^2, xy, y^2, x, y, 1)$, select $\mathbf{P}$ to minimize the sum of squared errors

$$E(\mathbf{P}) = \sum_{i=1}^{m} (\mathbf{P} \cdot \mathbf{Q}_i - z_i)^2$$

where $\mathbf{Q}_i = \mathbf{Q}(x_i, y_i)$. The minimum occurs when the gradient of E is the zero vector,

$$\nabla E = 2\sum_{i=1}^{m} (\mathbf{P} \cdot \mathbf{Q}_i - z_i)\mathbf{Q}_i = \mathbf{0}.$$

Some algebra converts this to a system of 6 equations in 6 unknowns:

$$\left(\sum_{i=1}^{m} \mathbf{Q}_i \mathbf{Q}_i^{\mathrm{T}}\right) \mathbf{P} = \sum_{i=1}^{m} z_i \mathbf{Q}_i.$$

The product $\mathbf{Q}_i \mathbf{Q}_i^{\mathrm{T}}$ is a product of the 6×1 matrix $\mathbf{Q}_i$ with the 1×6 matrix $\mathbf{Q}_i^{\mathrm{T}}$, the result being a 6×6 matrix.

Define the 6×6 symmetric matrix $A = \sum_{i=1}^{m} \mathbf{Q}_i \mathbf{Q}_i^{\mathrm{T}}$ and the 6×1 vector $\mathbf{B} = \sum_{i=1}^{m} z_i \mathbf{Q}_i$. The choice for $\mathbf{P}$ is the solution to the linear system of equations $A\mathbf{P} = \mathbf{B}$. The entries of A and $\mathbf{B}$ indicate summations over the appropriate product of variables. For example, $s(x^3y) = \sum_{i=1}^{m} x_i^3 y_i$:

$$\begin{bmatrix} s(x^4) & s(x^3y) & s(x^2y^2) & s(x^3) & s(x^2y) & s(x^2) \\ s(x^3y) & s(x^2y^2) & s(xy^3) & s(x^2y) & s(xy^2) & s(xy) \\ s(x^2y^2) & s(xy^3) & s(y^4) & s(xy^2) & s(y^3) & s(y^2) \\ s(x^3) & s(x^2y) & s(xy^2) & s(x^2) & s(xy) & s(x) \\ s(x^2y) & s(xy^2) & s(y^3) & s(xy) & s(y^2) & s(y) \\ s(x^2) & s(xy) & s(y^2) & s(x) & s(y) & s(1) \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} s(zx^2) \\ s(zxy) \\ s(zy^2) \\ s(zx) \\ s(zy) \\ s(z) \end{bmatrix}$$