Least Squares Fitting of Data

David Eberly, Geometric Tools, Redmond WA 98052
https://www.geometrictools.com/

Created: July 15, 1999
Last Modified: November 20, 2016

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 the 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 the error function for the least squares minimization to be

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

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,

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

and so

$$\begin{bmatrix} \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m 1 \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m x_i y_i \\ \sum_{i=1}^m y_i \end{bmatrix} \tag{3}$$

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

If implemented directly, this formulation can lead to an ill-conditioned linear system. To avoid this, you should first compute the averages $\bar{x} = \frac{1}{m}\sum_{i=1}^m x_i$ and $\bar{y} = \frac{1}{m}\sum_{i=1}^m y_i$ and subtract them from the data, $x_i \leftarrow x_i - \bar{x}$ and $y_i \leftarrow y_i - \bar{y}$. The fitted line is $y - \bar{y} = A(x - \bar{x})$, where

$$A = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^m (x_i - \bar{x})^2} \tag{4}$$

An implementation is GteApprHeightLine2.h.

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 error 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

$$E(\mathbf{A}, \mathbf{D}) = \sum_{i=1}^{m} \left(\mathbf{Y}_i^{\mathsf{T}} \left[I - \mathbf{D}\mathbf{D}^{\mathsf{T}}\right] \mathbf{Y}_i\right) \tag{5}$$

and

$$E(\mathbf{A}, \mathbf{D}) = \mathbf{D}^{\mathsf{T}} \left(\sum_{i=1}^{m} \left[(\mathbf{Y}_i \cdot \mathbf{Y}_i)I - \mathbf{Y}_i\mathbf{Y}_i^{\mathsf{T}}\right]\right) \mathbf{D} = \mathbf{D}^{\mathsf{T}} M(\mathbf{A})\mathbf{D} \tag{6}$$

Compute the derivative of equation (5) with respect to $\mathbf{A}$ to obtain

$$\frac{\partial E}{\partial \mathbf{A}} = -2\left[I - \mathbf{D}\mathbf{D}^{\mathsf{T}}\right] \sum_{i=1}^{m} \mathbf{Y}_i \tag{7}$$

The derivative is zero when $\sum_{i=1}^{m} \mathbf{Y}_i = 0$ in which case $\mathbf{A} = (1/m)\sum_{i=1}^{m} \mathbf{X}_i$, the average of the sample points. In fact there are infinitely many solutions $\mathbf{A} + s\mathbf{D}$ for any scalar s. This is simply a statement that the average is on the best-fit line, but any other point on the line may serve as the origin for that line.

Equation (6) is a quadratic form $\mathbf{D}^{\mathsf{T}} M(\mathbf{A})\mathbf{D}$ whose minimum is the smallest eigenvalue of $M(\mathbf{A})$, computed using standard eigensystem solvers. A corresponding unit length eigenvector $\mathbf{D}$ completes our construction of the least squares line. The covariance matrix of the input points is $C = \sum_{i=1}^{m} \mathbf{Y}_i\mathbf{Y}_i^{\mathsf{T}}$. Defining $\delta = \sum_{i=1}^{m} \mathbf{Y}_i^{\mathsf{T}}\mathbf{Y}_i$, we see that $M = \delta I - C$, where I is the identity matrix. Therefore, M and C have the same eigenspaces. The eigenspace corresponding to the minimum eigenvalue of M is the same as the eigenspace corresponding to the maximum eigenvalue of C. In an implementation, it is sufficient to process C and avoid the additional cost to compute M.

An implementation for 2D is GteApprOrthogonalLine2.h and an implementation for 3D is GteApprOrthogonalLine3.h.

3 Planar Fitting of 3D Points of the 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 the error function for the least squares minimization to be

$$E(A, B, C) = \sum_{i=1}^{m} [(Ax_i + By_i + C) - z_i]^2 \tag{8}$$

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,

$$(0, 0, 0) = \nabla E = 2\sum_{i=1}^{m} [(Ax_i + By_i + C) - z_i](x_i, y_i, 1) \tag{9}$$

and so

$$\begin{bmatrix} \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m y_i & \sum_{i=1}^m 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m x_i z_i \\ \sum_{i=1}^m y_i z_i \\ \sum_{i=1}^m z_i \end{bmatrix}. \tag{10}$$

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

If implemented directly, this formulation can lead to an ill-conditioned linear system. To avoid this, you should first compute the averages $\bar{x} = \frac{1}{m}\sum_{i=1}^m x_i$, $\bar{y} = \frac{1}{m}\sum_{i=1}^m y_i$, and $\bar{z} = \frac{1}{m}\sum_{i=1}^m z_i$, and then subtract them from the data, $x_i \leftarrow x_i - \bar{x}$, $y_i \leftarrow y_i - \bar{y}$, and $z_i \leftarrow z_i - \bar{z}$. The fitted plane is $z - \bar{z} = A(x-\bar{x}) + B(y-\bar{y})$, where

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m (x_i-\bar{x})^2 & \sum_{i=1}^m (x_i-\bar{x})(y_i-\bar{y}) \\ \sum_{i=1}^m (x_i-\bar{x})(y_i-\bar{y}) & \sum_{i=1}^m (y_i-\bar{y})^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^m (x_i-\bar{x})(z_i-\bar{z}) \\ \sum_{i=1}^m (y_i-\bar{y})(z_i-\bar{z}) \end{bmatrix} \tag{11}$$

An implementation is GteApprHeightPlane3.h.

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 error 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

$$E(\mathbf{A},\mathbf{N}) = \sum_{i=1}^m \left(\mathbf{Y}_i^\mathsf{T}\left[\mathbf{N}\mathbf{N}^\mathsf{T}\right]\mathbf{Y}_i\right) \tag{12}$$

and

$$E(\mathbf{A},\mathbf{N}) = \mathbf{N}^\mathsf{T}\left(\sum_{i=1}^m \mathbf{Y}_i\mathbf{Y}_i^\mathsf{T}\right)\mathbf{N} = \mathbf{N}^\mathsf{T}C\mathbf{N} \tag{13}$$

where $C = \sum_{i=1}^m \mathbf{Y}_i\mathbf{Y}_i^\mathsf{T}$.

Compute the derivative of equation (12) with respect to $\mathbf{A}$ to obtain

$$\frac{\partial E}{\partial \mathbf{A}} = -2\left(\mathbf{N}\mathbf{N}^\mathsf{T}\right)\sum_{i=1}^m \mathbf{Y}_i \tag{14}$$

The derivative is zero when $\sum_{i=1}^m \mathbf{Y}_i = 0$ in which case $\mathbf{A} = (1/m)\sum_{i=1}^m \mathbf{X}_i$, the average of the sample points. In fact there are infinitely many solutions $\mathbf{A} + \mathbf{W}$, where $\mathbf{W}$ is any vector perpendicular to $\mathbf{N}$. This is simply a statement that the average is on the best-fit hyperplane, but any other point on the hyperplane may serve as the origin for that hyperplane.

Equation (13) is a quadratic form $\mathbf{D}^{\mathsf{T}}C\mathbf{D}$ whose minimum is the smallest eigenvalue of C, computed using standard eigensystem solvers. A corresponding unit length eigenvector $\mathbf{N}$ completes our construction of the least squares hyperplane.

An implementation for 3D is GteApprOrthogonalPlane3.h.

5 kD Flat Fitting of nD Points Using Orthogonal Regression

Orthogonal regression to fit n-dimensional points by a line or by a hyperplane may be generalized to fitting by a *flat* of *affine subspace*. To emphasize the dimension of the flat, sometimes the terminology used is *k-flat*, where the dimension is k with $0 \le k \le n$. A line is a 1-flat and a hyperplane is a $(n-1)$-flat.

For dimension $n = 3$, we fit with flats that are either lines or planes. For dimensions $n \ge 4$ and flat dimensions $2 \le k \le n-2$, the generalization of orthogonal regression is the following. The bases mentioned here are for the linear portion of the affine subspace; that is, the basis vectors are relative to an origin at a point $\mathbf{A}$. The flat has an orthonormal basis $\{\mathbf{F}_j\}_{j=1}^{k}$ and the orthogonal complement has an orthonormal basis $\{\mathbf{P}_j\}_{j=1}^{n-k}$. The union of the two bases is an orthonormal basis for $\mathbb{R}^n$. Any input point $\mathbf{X}_i$, $1 \le i \le m$, can be represented by

$$\mathbf{X}_i = \mathbf{A} + \sum_{j=1}^{k} f_{ij}\mathbf{F}_j + \sum_{j=1}^{n-k} p_{ij}\mathbf{P}_j = \left(\mathbf{A} + \sum_{j=1}^{k} f_{ij}\mathbf{F}_j\right) + \left(\sum_{j=1}^{n-k} p_{ij}\mathbf{P}_j\right) \tag{15}$$

The left-parenthesized term is the portion of $\mathbf{X}_i$ that lives in the flat and the right-parenthesized is the portion that is the deviation of $\mathbf{X}_i$ from the flat. The least squares problem is about choosing the two bases so that the sum of squared lengths of the deviations is as small as possible.

Define $\mathbf{Y}_i = \mathbf{X}_i - \mathbf{A}$. The basis coefficients are $f_{ij} = \mathbf{F}_j \cdot \mathbf{Y}_i$ and $p_{ij} = \mathbf{P}_j \cdot \mathbf{Y}_i$. The squared length of the deviation is $\sum_{j=1}^{n-k} p_{ij}^2$. The error function for the least squares minimization is the sum of the squared lengths for all inputs, $E = \sum_{i=1}^{m}\sum_{j=1}^{n-k} p_{ij}^2$. Two alternate forms for this function are

$$E(\mathbf{A}, \mathbf{P}_1, \ldots \mathbf{P}_{n-k}) = \sum_{i=1}^{m} \mathbf{Y}_i^{\mathsf{T}} \left(\sum_{j=1}^{n-k} \mathbf{P}_j\mathbf{P}_j^{\mathsf{T}}\right) \mathbf{Y}_i \tag{16}$$

and

$$E(\mathbf{A}, \mathbf{P}_1, \ldots \mathbf{P}_{n-k}) = \sum_{j=1}^{n-k} \mathbf{P}_j^{\mathsf{T}} \left(\sum_{i=1}^{m} \mathbf{Y}_i\mathbf{Y}_i^{\mathsf{T}}\right) \mathbf{P}_j = \sum_{j=1}^{n-k} \mathbf{P}_j C \mathbf{P}_j \tag{17}$$

where $C = \sum_{i=1} \mathbf{Y}_i\mathbf{Y}_i^{\mathsf{T}}$.

Compute the derivative of equation (16) with respect to $\mathbf{A}$ to obtain

$$\frac{\partial E}{\partial \mathbf{A}} = 2\left(\sum_{j=1}^{n-k} \mathbf{P}_j\mathbf{P}_j^{\mathsf{T}}\right) \sum_{i=1}^{m} \mathbf{Y}_i \tag{18}$$

The derivative is zero when $\sum_{i=1}^{m} \mathbf{Y}_i = 0$ in which case $\mathbf{A} = (1/m)\sum_{i=1}^{m} \mathbf{X}_i$, the average of the sample points. In fact there are infinitely many solutions $\mathbf{A} + \mathbf{W}$, where $\mathbf{W}$ is any vector in the orthogonal complement

of the subspace spanned by the $\mathbf{P}_j$; this subspace is that spanned by the $\mathbf{F}_j$. This is simply a statement that the average is on the best-fit flat, but any other point on the flat may serve as the origin for that flat. Because we are choosing $\mathbf{A}$ to be the average of the input points, the matrix C is the covariance matrix for the input points.

The last term in equation (17) is a sum of quadratic forms involving the matrix C and the vectors $\mathbf{P}_j$ that are a basis (unit-length, mutually perpendicular). The minimum value of the quadratic form is the smallest eigenvalue λ_1 of C, so we may choose $\mathbf{P}_1$ to be a unit-length eigenvector of C corresponding to λ_1. We must choose $\mathbf{P}_2$ to be unit length and perpendicular to $\mathbf{P}_1$. If the eigenspace for λ_1 is 1-dimensional, the next smallest value we can attain by the quadratic form is the smallest eigenvalue λ_2 for which $\lambda_1 < \lambda_2$. $\mathbf{P}_2$ is chosen to be a corresponding unit-length eigenvector. However, if λ_1 has an eigenspace of dimension larger than 1, we can choose $\mathbf{P}_2$ in that eigenspace but which is perpendicular to $\mathbf{P}_1$.

Generally, let $\{\lambda_\ell\}_{\ell=1}^r$ be the r distinct eigenvalues of the covariance matrix C; we know $1 \le r \le n$ and $\lambda_1 < \lambda_2 < \cdots < \lambda_r$. Let the dimension of the eigenspace for λ_ℓ be $d_\ell \ge 1$; we know that $\sum_{\ell=1}^r d_\ell = n$. List the eigenvalues and eigenvectors in order of increasing eigenvalue, including repeated values,

$$\underbrace{\begin{matrix}\lambda_1 & \cdots & \lambda_1 \\ \mathbf{V}_1^1 & \cdots & \mathbf{V}_{d_1}^1\end{matrix}}_{d_1 \text{ terms}} \quad \underbrace{\begin{matrix}\lambda_2 & \cdots & \lambda_2 \\ \mathbf{V}_1^2 & \cdots & \mathbf{V}_{d_2}^2\end{matrix}}_{d_2 \text{ terms}} \quad \cdots \quad \underbrace{\begin{matrix}\lambda_r & \cdots & \lambda_r \\ \mathbf{V}_1^r & \cdots & \mathbf{V}_{d_r}^r\end{matrix}}_{d_r \text{ terms}} \tag{19}$$

The list has n items. The eigenvalue d_ℓ has an eigenspace with orthonormal basis $\{\mathbf{V}_j^\ell\}_{j=1}^{d_\ell}$. In this list, choose the first k eigenvectors to be $\mathbf{P}_1$ through $\mathbf{P}_k$ and choose the last $n-k$ eigenvectors to be $\mathbf{F}_1$ through $\mathbf{F}_{n-k}$.

It is possible that one (or more) of the $\mathbf{P}_j$ and one (or more) of the $\mathbf{F}_j$ are in the eigenspace for the same eigenvalue. In this case, the fitted flat is not unique, and one should re-examine the choice of dimension k for the fitted flat. This is analogous to the following situations in dimension $n = 3$.

- The input points are nearly collinear but you are trying to fit those points with a plane. The covariance matrix likely has two distinct eigenvalues $\lambda_1 < \lambda_2$ with $d_1 = 2$ and $d_2 = 1$. The basis vectors are $\mathbf{P}_1 = \mathbf{V}_1^1$, $\mathbf{F}_1 = \mathbf{V}_2^1$, and $\mathbf{F}_2 = \mathbf{V}_1^2$. The first two of these are from the same eigenspace.
- The input points are spread out over a portion of a plane (and are not nearly collinear) but you are trying to fit those points with a line. The covariance matrix likely has two distinct eigenvalues $\lambda_1 < \lambda_2$ with $d_1 = 1$ and $d_2 = 2$. The basis vectors are $\mathbf{P}_1 = \mathbf{V}_1^1$, $\mathbf{P}_2 = \mathbf{V}_1^2$, and $\mathbf{F}_1 = \mathbf{V}_2^2$. The last two of these are from the same eigenspace.
- The input points are not well fit by a flat of any dimension. For example, your input points are uniformly distributed over a sphere. The covariance matrix likely has one distinct eigenvalue λ_1 of multiplicity $d_1 = 3$. Neither a line nor a plane is a good fit to the input points—in either case, a $\mathbf{P}$-vector and an $\mathbf{F}$-vector are in the same eigenspace.

The computational algorithm is to compute the average $\mathbf{A}$ and covariance matrix of the points. Use an eigensolver whose output eigenvalues are sorted in nondecreasing order. Choose the $\mathbf{F}_j$ to be the last $n-k$ eigenvectors output by the eigensolver.

6 Fitting a Circle to 2D Points

There are several ways to define a least squares error function to try to fit a circle to points. The typical approach is to define a sum of squares function and use the Levenberg–Marquardt algorithm for iterative minimization. The methods proposed here do not follow this paradigm.

The first section uses the sum of squares of the differences between the purported circle radius and the distance from an input point to the purported circle. The algorithm is effectively a fixed-point iteration.

The second section estimates the coefficients of a quadratic equation of two variables. It reduces to computing the eigenvector associated with the minimum eigenvalue of a matrix.

Regardless of the error function, in practice the fitting produces reasonable results when the input points are distributed over the circle. If the input points are restricted to a small arc of a circle, the fitting is typically not of good quality. The problem is that small perturbations in the input points can lead to large changes in the estimates for the center and radius.

6.1 Direct Least Squares Algorithm

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 error function to be minimized is

$$E(a, b, r) = \sum_{i=1}^{m} (L_i - r)^2 \tag{20}$$

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

$$\frac{\partial E}{\partial r} = -2 \sum_{i=1}^{m} (L_i - r) \tag{21}$$

Setting the derivative equal to zero yields

$$r = \frac{1}{m} \sum_{i=1}^{m} L_i \tag{22}$$

Compute the partial derivatives with respect to a and b to obtain

$$\begin{aligned} \tfrac{\partial E}{\partial a} &= -2 \textstyle\sum_{i=1}^{m} (L_i - r) \tfrac{\partial L_i}{\partial a} = 2 \sum_{i=1}^{m} \left((x_i - a) + r \tfrac{\partial L_i}{\partial a} \right), \\ \tfrac{\partial E}{\partial b} &= -2 \textstyle\sum_{i=1}^{m} (L_i - r) \tfrac{\partial L_i}{\partial b} = 2 \sum_{i=1}^{m} \left((y_i - b) + r \tfrac{\partial L_i}{\partial b} \right) \end{aligned} \tag{23}$$

Setting these derivatives equal to zero yields

$$a = \frac{1}{m} \sum_{i=1}^{m} x_i + r \frac{1}{m} \sum_{i=1}^{m} \frac{\partial L_i}{\partial a}, \quad b = \frac{1}{m} \sum_{i=1}^{m} y_i + r \frac{1}{m} \sum_{i=1}^{m} \frac{\partial L_i}{\partial b} \tag{24}$$

Replace r from equation (22) and using using $\partial L_i / \partial a = (a - x_i)/L_i$ and $\partial L_i / \partial b = (b - y_i)/L_i$, we obtain two nonlinear equations in a and b,

$$a = \bar{x} + \bar{L}\bar{L}_a = F(a, b), \quad b = \bar{y} + \bar{L}\bar{L}_b = G(a, b) \tag{25}$$

where the last equality of each expression defines the functions $F(a,b)$ and $G(a,b)$ and where

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i,\ \ \bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i,\ \ \bar{L} = \frac{1}{m}\sum_{i=1}^{m} L_i,\ \ \bar{L}_a = \frac{1}{m}\sum_{i=1}^{m} \frac{a-x_i}{L_i},\ \ \bar{L}_b = \frac{1}{m}\sum_{i=1}^{m} \frac{b-y_i}{L_i} \tag{26}$$

Fixed-point iteration can be applied to solve 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 $i \geq 0$.

An implementation is GteApprCircle2.h.

6.2 Estimation of Coefficients to a Quadratic Equation

The quadratic equation that represents a circle is

$$0 = c_0 + c_1 x + c_2 y + c_3(x^2+y^2) = (c_0, c_1, c_2, c_3)\cdot(1, x, y, x^2+y^2) = \mathbf{C}\cdot\mathbf{V} \tag{27}$$

where the last equality defines $\mathbf{V}$ and a unit-length vector $\mathbf{C} = (c_0, c_1, c_2, c_3)$ with $c_3 \neq 0$ (for a nondegenerate circle). Given a set of input points $\{(x_i, y_i)\}_{i=1}^m$, the least squares error function is chosen to be

$$E(\mathbf{C}) = \sum_{i=1}^{m} (\mathbf{C}\cdot\mathbf{V}_i)^2 = \mathbf{C}^{\mathsf{T}}\left(\sum_{i=1}^{m} \mathbf{V}_i\mathbf{V}_i^{\mathsf{T}}\right)\mathbf{C} = \mathbf{C}^{\mathsf{T}}\mathbf{M}\mathbf{C} \tag{28}$$

where $\mathbf{V}_i = (1, x_i, y_i, r_i^2)$ with $r_i^2 = x_i^2 + y_i^2$. The last equality defines the 4×4 symmetric matrix M,

$$M = \begin{bmatrix} \sum_{i=1}^m 1 & \sum_{i=1}^m x_i & \sum_{i=1}^m y_i & \sum_{i=1}^m r_i^2 \\ \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m x_i r_i^2 \\ \sum_{i=1}^m y_i & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i r_i^2 \\ \sum_{i=1}^m r_i^2 & \sum_{i=1}^m x_i r_i^2 & \sum_{i=1}^m y_i r_i^2 & \sum_{i=1}^m r_i^4 \end{bmatrix} \tag{29}$$

The error function is a quadratic form that is minimized when $\mathbf{C}$ is a unit-length eigenvector corresponding to the minimum eigenvalue of M. An eigensolver can be used to compute the eigenvector.

The resulting equation appears not to be invariant to translations of the input points. You should instead consider modifying the inputs by computing the averages $\bar{x} = \frac{1}{m}\sum_{i=1}^m x_i$ and $\bar{y} = \frac{1}{m}\sum_{i=1}^m y_i$ and then replacing $x_i \leftarrow x_i - \bar{x}$ and $y_i \leftarrow y_i - \bar{y}$. The r_i^2 are then computed from the adjusted x_i and y_i values. The matrix becomes

$$\hat{M} = \begin{bmatrix} \sum_{i=1}^m 1 & 0 & 0 & \sum_{i=1}^m r_i^2 \\ 0 & \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m x_i r_i^2 \\ 0 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i r_i^2 \\ \sum_{i=1}^m r_i^2 & \sum_{i=1}^m x_i r_i^2 & \sum_{i=1}^m y_i r_i^2 & \sum_{i=1}^m r_i^4 \end{bmatrix} \tag{30}$$

The eigenvector $\mathbf{C}$ is computed. The circle center in the original coordinate system is $(\bar{x}, \bar{y}) - (c_1, c_2)/(2c_3)$ and the circle radius is $r = \sqrt{c_1^2 + c_2^2 - 4c_0c_3}/(2c_3)$.

An implementation is template class ApprQuadraticCircle2 in GteApprQuadratic2.h.

7 Fitting a Sphere to 3D Points

There are several ways to define a least squares error function to try to fit a sphere to points. The typical approach is to define a sum of squares function and use the Levenberg–Marquardt algorithm for iterative minimization. The methods proposed here do not follow this paradigm.

The first section uses the sum of squares of the differences between the purported sphere radius and the distance from an input point to the purported sphere. The algorithm is effectively a fixed-point iteration.

The second section estimates the coefficients of a quadratic equation of three variables. It reduces to computing the eigenvector associated with the minimum eigenvalue of a matrix.

Regardless of the error function, in practice the fitting produces reasonable results when the input points are distributed over the sphere. If the input points are restricted to a small solid angle, the fitting is typically not of good quality. The problem is that small perturbations in the input points can lead to large changes in the estimates for the center and radius.

7.1 Direct Least Squares Algorithm

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 error function to be minimized is

$$E(a, b, c, r) = \sum_{i=1}^{m} (L_i - r)^2 \tag{31}$$

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

$$\frac{\partial E}{\partial r} = -2 \sum_{i=1}^{m} (L_i - r). \tag{32}$$

Setting the derivative equal to zero yields

$$r = \frac{1}{m} \sum_{i=1}^{m} L_i. \tag{33}$$

Compute the partial derivatives with respect to a, b, and c to obtain

$$\begin{aligned}
\tfrac{\partial E}{\partial a} &= -2 \textstyle\sum_{i=1}^{m} (L_i - r) \tfrac{\partial L_i}{\partial a} = 2 \textstyle\sum_{i=1}^{m} \left((x_i - a) + r \tfrac{\partial L_i}{\partial a} \right), \\
\tfrac{\partial E}{\partial b} &= -2 \textstyle\sum_{i=1}^{m} (L_i - r) \tfrac{\partial L_i}{\partial b} = 2 \textstyle\sum_{i=1}^{m} \left((y_i - b) + r \tfrac{\partial L_i}{\partial b} \right), \\
\tfrac{\partial E}{\partial c} &= -2 \textstyle\sum_{i=1}^{m} (L_i - r) \tfrac{\partial L_i}{\partial c} = 2 \textstyle\sum_{i=1}^{m} \left((z_i - c) + r \tfrac{\partial L_i}{\partial c} \right)
\end{aligned} \tag{34}$$

Setting these derivatives equal to zero yields

$$a = \frac{1}{m} \sum_{i=1}^{m} x_i + r \frac{1}{m} \sum_{i=1}^{m} \frac{\partial L_i}{\partial a}, \quad b = \frac{1}{m} \sum_{i=1}^{m} y_i + r \frac{1}{m} \sum_{i=1}^{m} \frac{\partial L_i}{\partial b}, \quad c = \frac{1}{m} \sum_{i=1}^{m} z_i + r \frac{1}{m} \sum_{i=1}^{m} \frac{\partial L_i}{\partial c} \tag{35}$$

Replacing r from equation (33) 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 obtain three nonlinear equations in a, b, and c,

$$a = \bar{x} + \bar{L}\bar{L}_a = F(a,b,c), \quad b = \bar{y} + \bar{L}\bar{L}_b = G(a,b,c), \quad c = \bar{z} + \bar{L}\bar{L}_c = H(a,b,c) \tag{36}$$

where the last equality defines functions $F(a,b,c)$, $G(a,b,c)$, and $H(a,b,c)$ and where

$$\begin{gathered}\bar{x} = \tfrac{1}{m}\textstyle\sum_{i=1}^{m} x_i, \quad \bar{y} = \tfrac{1}{m}\textstyle\sum_{i=1}^{m} y_i, \quad \bar{z} = \tfrac{1}{m}\textstyle\sum_{i=1}^{m} z_i \\ \bar{L} = \tfrac{1}{m}\textstyle\sum_{i=1}^{m} L_i, \quad \bar{L}_a = \tfrac{1}{m}\textstyle\sum_{i=1}^{m} \tfrac{a-x_i}{L_i}, \quad \bar{L}_b = \tfrac{1}{m}\textstyle\sum_{i=1}^{m} \tfrac{b-y_i}{L_i}, \quad \bar{L}_c = \tfrac{1}{m}\textstyle\sum_{i=1}^{m} \tfrac{c-z_i}{L_i}\end{gathered} \tag{37}$$

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 $i \geq 0$.

An implementation is GteApprSphere3.h.

7.2 Estimation of Coefficients to a Quadratic Equation

The quadratic equation that represents a sphere is

$$0 = c_0 + c_1 x + c_2 y + c_3 z + c_4(x^2 + y^2 + z^2) = (c_0, c_1, c_2, c_3, c_4) \cdot (1, x, y, z, x^2 + y^2 + z^2) = \mathbf{C} \cdot \mathbf{V} \tag{38}$$

where the last equality defines $\mathbf{V}$ and a unit-length vector $\mathbf{C} = (c_0, c_1, c_2, c_3, c_4)$ with $c_4 \neq 0$ (for a nondegenerate sphere). Given a set of input points $\{(x_i, y_i, z_i)\}_{i=1}^m$, the least squares error function is chosen to be

$$E(\mathbf{C}) = \sum_{i=1}^{m} (\mathbf{C} \cdot \mathbf{V}_i)^2 = \mathbf{C}^{\mathsf{T}} \left(\sum_{i=1}^{m} \mathbf{V}_i \mathbf{V}_i^{\mathsf{T}} \right) \mathbf{C} = \mathbf{C}^{\mathsf{T}} \mathbf{M} \mathbf{C} \tag{39}$$

where $\mathbf{V}_i = (1, x_i, y_i, z_i, r_i^2)$ with $r_i^2 = x_i^2 + y_i^2 + z_i^2$. The last equality defines the 5×5 symmetric matrix M,

$$M = \begin{bmatrix} \sum_{i=1}^m 1 & \sum_{i=1}^m x_i & \sum_{i=1}^m y_i & \sum_{i=1}^m z_i & \sum_{i=1}^m r_i^2 \\ \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m x_i z_i & \sum_{i=1}^m x_i r_i^2 \\ \sum_{i=1}^m y_i & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i z_i & \sum_{i=1}^m y_i r_i^2 \\ \sum_{i=1}^m z_i & \sum_{i=1}^m x_i z_i & \sum_{i=1}^m y_i z_i & \sum_{i=1}^m z_i^2 & \sum_{i=1}^m z_i r_i^2 \\ \sum_{i=1}^m r_i^2 & \sum_{i=1}^m x_i r_i^2 & \sum_{i=1}^m y_i r_i^2 & \sum_{i=1}^m z_i r_i^2 & \sum_{i=1}^m r_i^4 \end{bmatrix} \tag{40}$$

The error function is a quadratic form that is minimized when $\mathbf{C}$ is a unit-length eigenvector corresponding to the minimum eigenvalue of M. An eigensolver can be used to compute the eigenvector.

The resulting equation appears not to be invariant to translations of the input points. You should instead consider modifying the inputs by computing the averages $\bar{x} = \frac{1}{m}\sum_{i=1}^m x_i$, $\bar{y} = \frac{1}{m}\sum_{i=1}^m y_i$, and $\bar{z} = \frac{1}{m}\sum_{i=1}^m z_i$ and then replacing $x_i \leftarrow x_i - \bar{x}$, $y_i \leftarrow y_i - \bar{y}$, and $z_i \leftarrow z_i - \bar{z}$. The r_i^2 are then computed from the adjusted

x_i, y_i, and z_i values. The matrix becomes

$$\hat{M} = \begin{bmatrix} \sum_{i=1}^m 1 & 0 & 0 & 0 & \sum_{i=1}^m r_i^2 \\ 0 & \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m x_i z_i & \sum_{i=1}^m x_i r_i^2 \\ 0 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i z_i & \sum_{i=1}^m y_i r_i^2 \\ 0 & \sum_{i=1}^m x_i z_i & \sum_{i=1}^m y_i z_i & \sum_{i=1}^m z_i^2 & \sum_{i=1}^m z_i r_i^2 \\ \sum_{i=1}^m r_i^2 & \sum_{i=1}^m x_i r_i^2 & \sum_{i=1}^m y_i r_i^2 & \sum_{i=1}^m z_i r_i^2 & \sum_{i=1}^m r_i^4 \end{bmatrix} \tag{41}$$

The eigenvector $\mathbf{C}$ is computed. The sphere center in the original coordinate system is $(\bar{x}, \bar{y}, \bar{z}) - (c_1, c_2, c_3)/(2c_4)$ and the sphere radius is $r = \sqrt{c_1^2 + c_2^2 + c_3^2 - 4c_0c_4}/(2c_4)$.

An implementation is template class ApprQuadraticSphere3 in GteApprQuadratic3.h.

8 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 $(\mathbf{X} - \mathbf{U})^\mathsf{T} R^\mathsf{T} DR(\mathbf{X} - \mathbf{U}) = 1$ 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 $D = \text{diag}(1/a^2, 1/b^2)$. The error function to be minimized is

$$E(\mathbf{U}, R, D) = \sum_{i=1}^m (L_i - r)^2 \tag{42}$$

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 is computed according to the algorithm described in Distance from a Point to an Ellipse, an Ellipsoid, or a Hyperellipsoid. The function E is minimized iteratively using Powell's direction-set method to search for a minimum. An implementation is GteApprEllipse2.h.

9 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 $(\mathbf{X} - \mathbf{U})^\mathsf{T} R^\mathsf{T} DR(\mathbf{X} - \mathbf{U}) = 1$ 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 $D = \text{diag}(1/a^2, 1/b^2, 1/c^2)$. The error function to be minimized is

$$E(\mathbf{U}, R, D) = \sum_{i=1}^m (L_i - r)^2 \tag{43}$$

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 is computed according to the algorithm described in Distance from a Point to an Ellipse, an Ellipsoid, or a Hyperellipsoid. The function E is minimized iteratively using Powell's direction-set method to search for a minimum. An implementation is GteApprEllipsoid3.h.

10 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

$$z = f(x, y) = p_1x^2 + p_2xy + p_3y^2 + p_4x + p_5y + p_6 = \mathbf{P} \cdot \mathbf{Q}(x, y) \tag{44}$$

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

$$E(\mathbf{P}) = \sum_{i=1}^{m} (\mathbf{P} \cdot \mathbf{Q}_i - z_i)^2 \tag{45}$$

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

$$\nabla E = 2 \sum_{i=1}^{m} (\mathbf{P} \cdot \mathbf{Q}_i - z_i)\mathbf{Q}_i = \mathbf{0}. \tag{46}$$

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

$$\left(\sum_{i=1}^{m} \mathbf{Q}_i \mathbf{Q}_i^{\mathsf{T}} \right) \mathbf{P} = \sum_{i=1}^{m} z_i \mathbf{Q}_i. \tag{47}$$

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

Define the 6×6 symmetric matrix $A = \sum_{i=1}^m \mathbf{Q}_i\mathbf{Q}_i^{\mathsf{T}}$ 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, $s(x^3y) = \sum_{i=1}^m x_i^3 y_i$:

$$\begin{bmatrix} s(x^4) & s(x^3y) & s(x^2y^2) & s(x^3) & s(x^2y) & s(x^2) \\ s(x^3y) & s(x^2y^2) & s(xy^3) & s(x^2y) & s(xy^2) & s(xy) \\ s(x^2y^2) & s(xy^3) & s(y^4) & s(xy^2) & s(y^3) & s(y^2) \\ s(x^3) & s(x^2y) & s(xy^2) & s(x^2) & s(xy) & s(x) \\ s(x^2y) & s(xy^2) & s(y^3) & s(xy) & s(y^2) & s(y) \\ s(x^2) & s(xy) & s(y^2) & s(x) & s(y) & s(1) \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} s(zx^2) \\ s(zxy) \\ s(zy^2) \\ s(zx) \\ s(zy) \\ s(z) \end{bmatrix} \tag{48}$$

An implementation is GteApprParaboloid3.h.