A brief introduction to geometric fitting algorithms pt. I

Alejandro Garnung Menéndez

27 January 2025

Contents

Geometric fitting algorithms

Suppose we have a set of 3D points denoted as:

$$\{\mathbf{x}_i\}_{i=1}^{m} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_m\}, \quad \text{where} \quad \mathbf{x}_i \in \mathbb{R}^3. \tag{1}$$

We can approach the problem of fitting geometric entities in two distinct ways:

1. **Through Direct Definition:** One approach involves solving a typically *overdetermined* linear system derived directly from the equations of the geometric entities. In this case, we "stack" as many data points as available, transforming the problem into a matrix system that can be solved using linear algebra techniques such as least squares or SVD. This method leverages the parametric forms of the geometric entities and directly applies them to the data, yielding a (hopefully linear) system that approximates the best-fitting parameters for the given points.

2. **Through Error Minimization:** Another approach is to define the equations of the geometric entities in terms of the observed data points. In this case, we minimize the *error function* that quantifies the difference between the observed data points and the predicted points defined by the entity's parameters. This optimization process typically involves (iteratively or in a single step) adjusting the parameters of the geometric entity in a parameter space until the error is minimized. By doing so, we determine the optimal parameters that best fit the data according to a chosen error criterion, such as least squares or maximum likelihood.

2D line

The equation of a 2D line can be expressed as:

$$y = mx + b,$$

where m is the slope and b is the y-intercept. The goal is to determine m and b such that the sum of squared errors between the observed y_i values and the predicted values $mx_i + b$ is minimized.

The error function to minimize is:

$$R(m, b) = \sum_{i=1}^{n} (mx_i + b - y_i)^2.$$

The solution can be found by solving the system derived from the partial derivatives of R with respect to m and b, leading to the least-squares solution for m and b. The solution (left as an exercise for the reader) is similar to:

$$\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}.$$

This approach gives us the slope m and y-intercept b of the best-fitting 2D line. A minimum of 2 points is required to fit a line (under this approach).

On the other hand, we can directly write and solve the overdetermined system for fitting a 2D line:

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \cdot \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix},$$

which, in fact, multiplying both sides by the transpose of the first matrix gives us the least-squares solution.

Remark:

To solve an overdetermined (linear) system $A\mathbf{x} = \mathbf{b}$, we can apply SVD. The decomposition of the design matrix A is:

$$A = U\Sigma V^T,$$

where U is an orthogonal matrix of left singular vectors, Σ is a diagonal matrix of singular values, and V^T is an orthogonal matrix of right singular vectors.
The solution to the system using SVD is obtained through the formula:

$$\mathbf{x} = V\Sigma^{-1}U^T\mathbf{y},$$

where $\mathbf{x}$ is the parameter vector, and $\mathbf{y}$ is the vector of observations. This approach is especially useful when the system is overdetermined and faces issues such as multicollinearity or noise in the data, as SVD robustly handles these cases.

2D circle fitting

The equation for a circle in 2D Cartesian coordinates is given by:

$$(x - h)^2 + (y - k)^2 = r^2, \tag{2}$$

where (h, k) is the center of the circle and r is the radius. To fit a circle to a set of points $\{(x_i, y_i)\}$, we aim to find h, k, and r such that the sum of squared errors between the observed points and the circle is minimized.

Define the error function as:

$$R(h, k, r) = \sum_{i=1}^{n} \left[(x_i - h)^2 + (y_i - k)^2 - r^2\right]^2.$$

The solution involves minimizing $R(h, k, r)$ by solving the corresponding system of equations. The solution (left as an exercise for the reader) is similar to:

$$\begin{bmatrix} 2\sum x_i & 2\sum y_i & 2n \\ 2\sum x_i y_i & 2\sum y_i^2 + 2\sum x_i^2 & 2\sum x_i \\ 2\sum x_i^2 + 2\sum y_i^2 & 2\sum x_i y_i & 2n \end{bmatrix} \cdot \begin{bmatrix} h \\ k \\ r^2 \end{bmatrix} = \begin{bmatrix} \sum x_i^2 + \sum y_i^2 \\ \sum x_i y_i + \sum y_i^2 \\ \sum x_i^2 + \sum y_i^2 - r^2 \end{bmatrix}.$$

Alternatively, we can define new variables for simplicity:

$$a = -2h, \quad b = -2k, \quad c = h^2 + k^2 - r^2.$$

Substituting these into (2), we get:

$$x_i^2 + y_i^2 + ax_i + by_i + c = 0.$$

This leads to the following system of linear equations for a, b, and c:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -x_1^2 - y_1^2 \\ -x_2^2 - y_2^2 \\ \vdots \\ -x_n^2 - y_n^2 \end{bmatrix}.$$

The solution provides the values of a, b, and c, from which the parameters of the 2D circle can be derived:

$$h = -\frac{a}{2}, \quad k = -\frac{b}{2}, \quad r^2 = h^2 + k^2 - c.$$

This approach gives us the center (h, k) and radius r of the best-fitting 2D circle. A minimum of 3 non-collinear points is required to fit a 2D circle (under this approach).

Sphere Fitting

In 3D Cartesian coordinates, the equation of a sphere is given by:

$$(x - h)^2 + (y - k)^2 + (z - l)^2 = r^2,$$

where (h, k, l) is the center of the sphere and r is its radius.

To fit a sphere to a set of 3D points $\{(x_i, y_i, z_i)\}$, we aim to minimize the squared error between the observed points and the sphere's equation. The error function to minimize is:

$$R(h, k, l, r) = \sum_{i=1}^{n} \left[(x_i - h)^2 + (y_i - k)^2 + (z_i - l)^2 - r^2\right]^2.$$

For simplicity, we can define new variables:

$$a = -2h, \quad b = -2k, \quad c = -2l, \quad d = h^2 + k^2 + l^2 - r^2.$$

Substituting these into the equation of the sphere, we get:

$$x_i^2 + y_i^2 + z_i^2 + ax_i + by_i + cz_i + d = 0.$$

This leads to the following system of linear equations for a, b, c, and d:

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} -(x_1^2 + y_1^2 + z_1^2) \\ -(x_2^2 + y_2^2 + z_2^2) \\ \vdots \\ -(x_n^2 + y_n^2 + z_n^2) \end{bmatrix}.$$

The solution provides the values of a, b, c, and d, from which the parameters of the sphere can be derived:

$$h = -\frac{a}{2}, \quad k = -\frac{b}{2}, \quad l = -\frac{c}{2}, \quad r^2 = h^2 + k^2 + l^2 - d.$$

This approach gives us the center (h, k, l) and radius r of the best-fitting sphere. A minimum of 4 non-colinear points is required to fit a sphere (under this approach).

Remark:

> In practice, especially when dealing with noisy data or outliers, the Random Sample Consensus (**RANSAC**) algorithm is often used for fitting. RANSAC works by randomly selecting a subset of points, fitting the geometric model to these points, and then evaluating how well the remaining points fit the model. Points that do not fit the model well (outliers) are discarded, and the model is refitted to the remaining points (inliers).