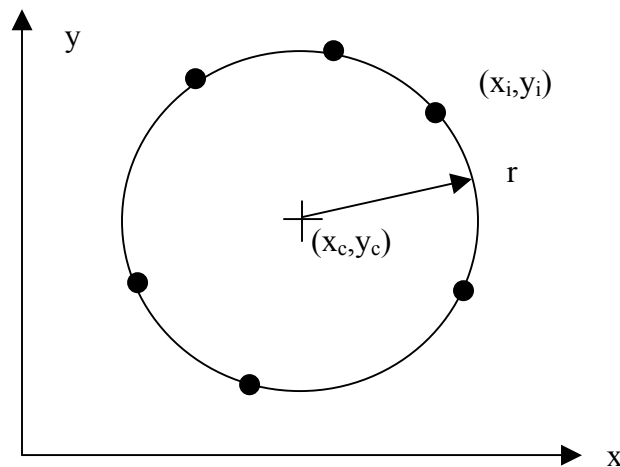


Fitting Data Points to a Circle – “Method 3” Dr. Fred DePiero, CalPoly State University

This method uses a least-squares formulation to fit N planar data points (x_i, y_i) to a circle. The error in radius is minimized (as opposed to the error in a single coordinate – as with line fitting via linear regression). No Assumptions are made with regard to the symmetry of the input data points. The center, (x_c, y_c) , and radius, r , of the circle are computed. The solution is found by minimizing a cost function. The approach defines an unknown, $\alpha = x_c^2 + y_c^2 - r^2$, that results in a linear set of equations. While α is convenient in terms of finding a solution, it does result in a least-squares formulation in which the radius is not explicitly minimized. (See below). However, Monte Carlo simulations have demonstrated the technique to work well despite this limitation.



A cost function, $G(x_c, y_c, r)$ is defined in order to minimize the error associated with each data point (x_i, y_i) and the model parameters (x_c, y_c, r) . The solution is found by finding the global minimum of the cost function, which is done by setting the gradient of $G()$ to zero

$$\nabla G(x_c, y_c, r) = \begin{bmatrix} \partial G / \partial x_c \\ \partial G / \partial y_c \\ \partial G / \partial r \end{bmatrix} = 0$$

The radial distance, d_i , from the (unknown) center (x_c, y_c) to a data point (x_i, y_i) is

$$d_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$$

Consider a preliminary version of the cost function, $G_1(x_c, y_c, r)$, used to minimize the error in the radial distances

$$G_1(x_c, y_c, r) = \sum_i |d_i - r|$$

Note that an alternate cost function, $G_2()$

$$G_2(x_c, y_c, r) = \sum_i (d_i - r)^2$$

would be minimized at the same point as $G_1()$, since minimizing the square of the radial errors is effectively the same as minimizing their absolute value. Another possibility is a cost function, $G_3()$

$$G_3(x_c, y_c, r) = \sum_i (d_i^2 - r^2)^2$$

This would also have the same global minimum. In this case the difference of the square of distances is used. This is the preferable version, as it eliminates computations involving the square root in the definition of d_i . Also, the gradient of $G()$ has partial derivatives that are continuous functions.

So $G()$ is defined to be $G_3()$.

$$G(x_c, y_c, r) = \sum_i \{ (x_i - x_c)^2 + (y_i - y_c)^2 - r^2 \}^2$$

expanding this

$$= \sum_i (x_i^2 - 2x_i x_c + y_i^2 - 2y_i y_c + \alpha)^2$$

where $\alpha = x_c^2 + y_c^2 - r^2$ is defined to be a new unknown.

Setting each partial to zero gives

$$\partial G / \partial x_c = \sum_i 2(x_i^2 - 2x_i x_c + y_i^2 - 2y_i y_c + \alpha)(-2x_i) = 0$$

$$\partial G / \partial y_c = \sum_i 2(x_i^2 - 2x_i x_c + y_i^2 - 2y_i y_c + \alpha)(-2y_i) = 0$$

$$\partial G / \partial \alpha = \sum_i 2(x_i^2 - 2x_i x_c + y_i^2 - 2y_i y_c + \alpha) = 0$$

This yields three simultaneous equations

$$\begin{cases} \sum_i x_i^3 - 2 \sum_i x_i^2 x_c + \sum_i y_i^2 x_i - 2 \sum_i y_i x_i y_c + \alpha \sum_i x_i = 0 \\ \sum_i x_i^2 y_i - 2 \sum_i x_i y_i x_c + \sum_i y_i^3 - 2 \sum_i y_i^2 y_c + \alpha \sum_i y_i = 0 \\ \sum_i x_i^2 - 2 \sum_i x_i x_c + \sum_i y_i^2 - 2 \sum_i y_i y_c + \alpha N = 0 \end{cases}$$

Rearranging to expose the unknowns

$$\begin{cases} (-2\sum_i x_i^2)x_c + (-2\sum_i y_i x_i)y_c + (\sum_i x_i)\alpha = -\sum_i x_i^3 - \sum_i y_i^2 x_i \\ (-2\sum_i x_i y_i)x_c + (-2\sum_i y_i^2)y_c + (\sum_i y_i)\alpha = -\sum_i x_i^2 y_i - \sum_i y_i^3 \\ (-2\sum_i x_i)x_c + (-2\sum_i y_i)y_c + (N)\alpha = -\sum_i x_i^2 - \sum_i y_i^2 \end{cases}$$

then permits a matrix equation in the form $Az = b$ to be setup:

$$\begin{bmatrix} -2\sum_i x_i^2 & -2\sum_i x_i y_i & \sum_i x_i \\ -2\sum_i x_i y_i & -2\sum_i y_i^2 & \sum_i y_i \\ -2\sum_i x_i & -2\sum_i y_i & N \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ \alpha \end{bmatrix} = \begin{bmatrix} -\sum_i x_i^3 - \sum_i x_i y_i^2 \\ -\sum_i x_i^2 y_i - \sum_i y_i^3 \\ -\sum_i x_i^2 - \sum_i y_i^2 \end{bmatrix}$$

with $\alpha = x_c^2 + y_c^2 - r^2$.

As described previously the radius, r , is not an explicit unknown. (It does not appear as an isolated component in the vector z , above). Hence errors in the input data (x_i, y_i) result in an optimal choice for (x_c, y_c, α) ; rather than for (x_c, y_c, r) . Monte Carlo simulations have shown that this does not appear to have a significant effect, for expected noise conditions. In fact the simulations demonstrated a reduction in the standard deviation of the radial error by a factor of 2.5, compared to the standard deviation of the noise in the (x_i, y_i) data points. This test was run for $N=8$ data points.