RANSAC Circle Fitting using 3 samples

j.w.li

May 2025

Contents

1 Introduction

This is a short description and loose proof of how exactly the library attached to this document works.

2 Circle fitting using 3 samples

All 3 points that are not on one line can be circumcircled. The following is a not so concise, but natural proof by construction. This yields us a nice formula for both the radius and center of the circle that passes through these 3 points.

We consider the general formula for a circle:

$$(x - x_c)^2 + (y - y_c)^2 - r^2 = 0 \tag{1}$$

Then we consider three points $\vec{p_i} = (x_i, y_i)^T$ where $i \in [0, 1, 2]$. Furthermore, we assume that they do not lie on one line, in other words, we know that the area of the triangle they describe is not zero.

$$\frac{1}{2}\left[x_0(y_1 - y_2) + x_1(y_2 - y_0) + x_2(y_0 - y_1)\right] \neq 0 \tag{2}$$

Continuing, we know that for these to lie on the circle they must satisfy the following system of equations:

$$\begin{cases} (x_0 - x_c)^2 + (y_0 - y_c)^2 - r^2 = 0 \\ (x_1 - x_c)^2 + (y_1 - y_c)^2 - r^2 = 0 \\ (x_2 - x_c)^2 + (y_2 - y_c)^2 - r^2 = 0 \end{cases}$$

Note that finding a arbitrary solution for these systems also proves that all 3 points as described above must lie on some circle, by construction.

We subtract the first equation from both of the other equations, this gives us the linear system:

$$\begin{cases} (x_1 - x_c)^2 + (y_1 - y_c)^2 - (x_0 - x_c)^2 - (y_0 - y_c)^2 = 0 \\ (x_2 - x_c)^2 + (y_2 - y_c)^2 - (x_0 - x_c)^2 - (y_0 - y_c)^2 = 0 \end{cases}$$

Expanding this we get

$$\begin{cases} x_1^2 + x_c^2 - 2x_1x_c + y_1^2 + y_c^2 - 2y_1y_c - x_0^2 - x_c^2 + 2x_0x_c - y_0^2 - y_c^2 + 2y_0y_c = 0 \\ x_2^2 + x_c^2 - 2x_2x_c + y_2^2 + y_c^2 - 2y_2y_c - x_0^2 - x_c^2 + 2x_0x_c - y_0^2 - y_c^2 + 2y_0y_c = 0 \end{cases}$$

Which we can simplify to

$$\begin{cases} x_c \cdot 2(x_0 - x_1) + y_c \cdot 2(y_0 - y_1) = x_0^2 + y_0^2 - x_1^2 - y_1^2 \\ x_c \cdot 2(x_0 - x_2) + y_c \cdot 2(y_0 - y_2) = x_0^2 + y_0^2 - x_2^2 - y_2^2 \end{cases}$$

Which can be described in the matrix form $A\vec{x} = \vec{b}$ where

$$A = 2\begin{bmatrix}(x_0 - x_1) & (y_0 - y_1)\\ (x_0 - x_2) & (y_0 - y_2)\end{bmatrix}, \vec{x} = \begin{pmatrix}x_c\\ y_c\end{pmatrix}, \vec{b} = \begin{pmatrix}x_0^2 + y_0^2 - x_1^2 - y_1^2\\ x_0^2 + y_0^2 - x_2^2 - y_2^2\end{pmatrix}$$

We know that the determinant of the matrix A is equal to

$$\begin{aligned}\det A &= 4[(x_0 - x_1)(y_0 - y_2) - (y_0 - y_1)(x_0 - x_2)]\\ &= 4[x_0y_0 - x_0y_2 - x_1y_0 + x_1y_2\\ &\quad - y_0x_0 + y_0x_2 + y_1x_0 - y_1x_2]\\ &= 4[-x_0y_2 - x_1y_0 + x_1y_2 + y_0x_2 + y_1x_0 - y_1x_2]\\ &= 4[x_0(y_1 - y_2) + x_1(y_2 - y_0) + x_2(y_0 - y_1)]\\ &\neq 0\end{aligned}$$

with the last step being due to the fact we know that the points do not form one line as described in Equation 2.

Because the determinant is not 0, we know that A is invertible, which means we know there must exist at least one solution, the following one:

$$\begin{aligned}\vec{x} = A^{-1}\vec{b} &= \frac{\begin{bmatrix}(y_0 - y_2) & (y_1 - y_0)\\ (x_2 - x_0) & (x_0 - x_1)\end{bmatrix}}{2[x_0(y_1 - y_2) + x_1(y_2 - y_0) + x_2(y_0 - y_1)]}\vec{b}\\ &= \frac{\begin{bmatrix}(y_0 - y_2) & (y_1 - y_0)\\ (x_2 - x_0) & (x_0 - x_1)\end{bmatrix}}{2[x_0(y_1 - y_2) + x_1(y_2 - y_0) + x_2(y_0 - y_1)]}\begin{pmatrix}x_0^2 + y_0^2 - x_1^2 - y_1^2\\ x_0^2 + y_0^2 - x_2^2 - y_2^2\end{pmatrix}\\ &= \frac{\begin{bmatrix}(y_0 - y_2)(x_0^2 + y_0^2 - x_1^2 - y_1^2) + (y_1 - y_0)(x_0^2 + y_0^2 - x_2^2 - y_2^2)\\ (x_2 - x_0)(x_0^2 + y_0^2 - x_1^2 - y_1^2) + (x_0 - x_1)(x_0^2 + y_0^2 - x_2^2 - y_2^2)\end{bmatrix}}{2[x_0(y_1 - y_2) + x_1(y_2 - y_0) + x_2(y_0 - y_1)]}\\ &= \frac{\begin{bmatrix}(y_0 - y_2)(x_0^2 + y_0^2 - x_1^2 - y_1^2) + (y_1 - y_0)(x_0^2 + y_0^2 - x_2^2 - y_2^2)\\ (x_2 - x_0)(x_0^2 + y_0^2 - x_1^2 - y_1^2) + (x_0 - x_1)(x_0^2 + y_0^2 - x_2^2 - y_2^2)\end{bmatrix}}{2[x_0(y_1 - y_2) + x_1(y_2 - y_0) + x_2(y_0 - y_1)]}\end{aligned}$$

We can rewrite this to be slightly easier to read and also be more performant on a computer

$$
\begin{aligned}
\vec{x} &= \frac{\begin{bmatrix} (y_0 - y_2)(x_0^2 + y_0^2 - x_1^2 - y_1^2) + (y_1 - y_0)(x_0^2 + y_0^2 - x_2^2 - y_2^2) \\ (x_2 - x_0)(x_0^2 + y_0^2 - x_1^2 - y_1^2) + (x_0 - x_1)(x_0^2 + y_0^2 - x_2^2 - y_2^2) \end{bmatrix}}{2[x_0(y_1 - y_2) + x_1(y_2 - y_0) + x_2(y_0 - y_1)]} \\
&= \frac{\begin{bmatrix} (y_0 - y_2)(d_0 - d_1) + (y_1 - y_0)(d_0 - d_2) \\ (x_2 - x_0)(d_0 - d_1) + (x_0 - x_1)(d_0 - d_2) \end{bmatrix}}{2[x_0(y_1 - y_2) + x_1(y_2 - y_0) + x_2(y_0 - y_1)]} \\
&= \frac{\begin{bmatrix} (y_0 - y_2)(d_{01}) + (y_1 - y_0)(d_{02}) \\ (x_2 - x_0)(d_{01}) + (x_0 - x_1)(d_{02}) \end{bmatrix}}{c}
\end{aligned}
$$

$$
x_c = \frac{d_{01}(y_0 - y_2) + d_{02}(y_1 - y_0)}{c} \wedge y_c = \frac{d_{01}(x_2 - x_0) + d_{02}(x_0 - x_1)}{c} \tag{3}
$$

where $d_i = x_i^2 + y_i^2$, $d_{ij} = d_i - d_j$ and $c = 2[x_0(y_1 - y_2) + x_1(y_2 - y_0) + x_2(y_0 - y_1)]$.

We can then use any of the equations to calculate r^2, for example using the first point:

$$
(x_0 - x_c)^2 + (y_0 - y_c)^2 = r^2
$$

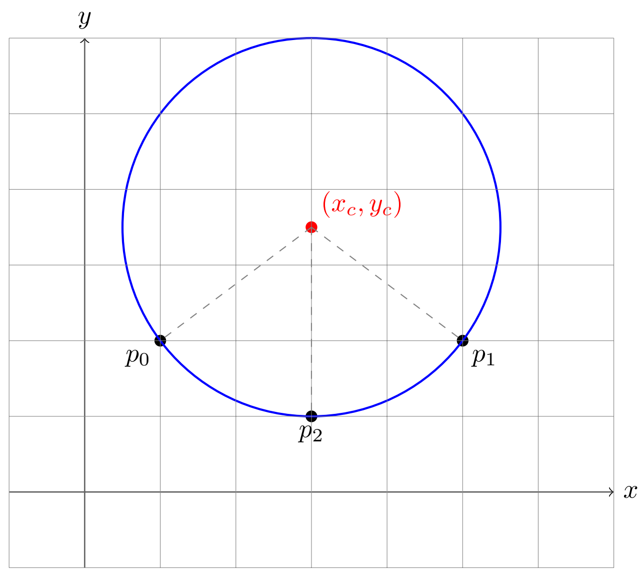
2.1 Example

As a quick example, here we can see this method being applied to the following points

$$\vec{p_0} = (1,2)^T, \vec{p_1} = (5,2)^T, \vec{p_2} = (3,1)^T$$

After doing all calculations we get the following data for the circle

$$\vec{x_c} = 3 \wedge \vec{y_c} = \frac{7}{2} \wedge r^2 = \left(\frac{5}{2}\right)^2$$



3 Circle fitting using N potentially noisy samples

We then use the aforementioned circle fitting to sample a subset of 3 from N samples. We fit a circle on these 3 points and see how many of the given samples are considered *in* the circle. This constitutes the score of the circle. You can model this as follows:

$$S_p(\vec{p}, \vec{c}, r, \epsilon) = \begin{cases} 1 & \| \|\vec{c} - \vec{p}\|_2 - r\| < \epsilon \\ 0 & else \end{cases} \tag{4}$$

$$S(\vec{c}, r, \epsilon) = \sum_{i}^{P} S_p(\vec{p}_i, \vec{c}, r, \epsilon) \tag{5}$$

where $P = \{\vec{p_i} | 0 \leq i \leq N\}$ is the set of all samples, $\vec{c}$ and r are the center and radius of the circle you're testing respectively, and ϵ is the maximum distance from the circle to be considered inside.

We simply take all circles for a max number of iterations and then see which has the higest score, taking this as our circle.

4 Circles in 3D

It is easy to see that this can easily generalise to 3-dimensional or possibly N-dimensional spaces, but personally I require this for a 2-dimensional problem, and thus will not be spending more time on this.