Anchored Least Squares Arc

Dwyer
Feb 2021

= =

Problem

Given two distinct points, A and B, and a non-empty set of points $P = \{p_0, p_1, \cdots, p_{n-1}\}$, find the center point and radius for a circle which

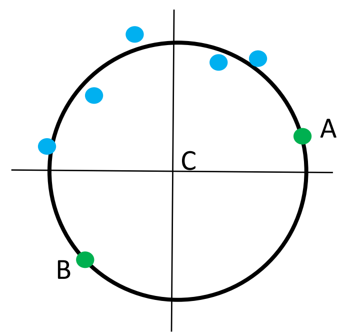
- passes through point A
- passes through point B
- is, in some sense, a 'best fit' for the points in set P

If there is only one point in P, and thus only three points total, the computed circle should be the unique circle passing through the three points.

The algorithm should recognize the case where all of the points are collinear.

= =

In many cases the points in P will lie between points A and B, as shown here, but that is not a requirement.

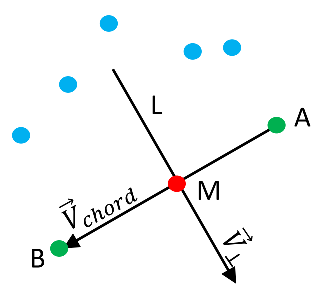


The points in P are not required to be in any particular order.

Clearly, once the center C is known the radius r is determined.

= =

Geometric Details



Let M be the midpoint of the chord AB.
Let $\vec{V}_{chord}$ be the vector from A to B ... a direction vector for the chord.
Let $\vec{V}_{\perp}$ be $\vec{V}_{chord}$ rotated 90°.
Let L be the line through M, parallel to $\vec{V}_{\perp}$.

We know that the center C must lie somewhere along line L:

= =

Normalization

The algebra is simplified if we transform the points so that :

$T(A) = (1,0)$
$T(B) = (-1,0)$

Then:

The point M is transformed to the origin.
The line L is transformed to the Y axis.
The center will lie on the Y axis, at $(0, t)$ for some choice of t.
The radius will be $r = \sqrt{1 + t^2}$

In the general case a least squares circle fit involves three parameters: the radius and the two coordinates of C. We see that the anchored arc problem involves only one parameter.

An additional advantage with the normalized problem is that it becomes trivial to see if all of the points in P lie on an arc between A and B. One of the arcs lies above the X axis, and the other below, so all that is required is to check that the transformed points have Y coordinates of the same sign. All of the points will be collinear if, and only if, the transformed points all lie on the X axis.
= =
Objective Function

In most circle fits, the objective function used is

$$J(C_x, C_y, r) = \sum (d_i - r)^2$$

where d_i is the distance from point p_i to the center C.

Experience shows that it is often preferable (whenever possible) to use distance-squared rather than distance when making comparisons. We will use the objective function

$$J(C_x, C_y, r) = \sum \left(d_i^2 - r^2\right)^2$$

This function retains the desirable property that the objective will be 0 if, and only if, all of the points lie exactly on the circle. It will, however, be more sensitive to outliers than the usual function.
= =
Solution

We assume that the problem has already been normalized, as described above. We also assume that it has been verified that the points are not all collinear.

After normalization, we have:

$$C = (0, t) \text{ for some choice of } t.$$

$$r = \sqrt{1 + t^2}$$

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

For the normalized problem, the objective is

$$\begin{aligned} J(t) &= \sum \left(d_i^2 - r^2\right)^2 \\ &= \sum \left(\left(x_i^2 + (y_i - t\,)^2\right) - (1 + t^2)\right)^2 \\ &= \sum \left(x_i^2 + y_i^2 - 2y_i t + t^2 - 1 - t^2\right)^2 \\ &= \sum \left(x_i^2 + y_i^2 - 1 - 2y_i t\right)^2 \end{aligned}$$

As usual, we optimize by taking the derivative and setting it to 0:

$$J' = \frac{d}{dt}\sum\left(x_i^2 + y_i^2 - 1 - 2y_i t\right)^2$$
$$= 2\sum\left(x_i^2 + y_i^2 - 1 - 2y_i t\right) * (-2y_i)$$
$$= -4\sum\left(x_i^2 y_i + y_i^3 - y_i - 2y_i^2 t\right)$$
$$= -4\sum\left(x_i^2 y_i + y_i^3 - y_i\right) + 8t\sum\left(y_i^2\right)$$

$$J' = 0 \quad \Rightarrow \quad -4\sum\left(x_i^2 y_i + y_i^3 - y_i\right) + 8t\sum\left(y_i^2\right) = 0$$
$$\sum\left(x_i^2 y_i + y_i^3 - y_i\right) = 2t\sum\left(y_i^2\right)$$

Observe that we have already verified that not all of the points in P lie on the X axis, so $\sum\left(y_i^2\right) > 0$.

$$t = \frac{\sum\left(x_i^2 y_i + y_i^3 - y_i\right)}{2\sum\left(y_i^2\right)}$$

= =

Normalization Details

The normalization transformation can be decomposed into three simple steps:

1. Translate M to the origin
2. Scale so that the distance from $T_1(A) = A'$ to the origin is 1.0
3. Rotate so that $T_2(A') = A''$ maps to (1,0)

We will use augmented coordinates and matrices to compute the transformations.

Translation

$$T_1 = \begin{pmatrix} 1 & 0 & -M_x \\ 0 & 1 & -M_y \\ 0 & 0 & 1 \end{pmatrix}$$

Scaling

$$s = \frac{1}{\sqrt{{A'_x}^2 + {A'_y}^2}}$$

$$T_2 = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Rotation

$$T_3 = \begin{pmatrix} A''_x & A''_y & 0 \\ -A''_y & A''_x & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Normalization

$$T = T_3 T_2 T_1$$

Inverses

$$T_1^{-1} = \begin{pmatrix} 1 & 0 & M_x \\ 0 & 1 & M_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$T_2^{-1} = \begin{pmatrix} 1/s & 0 & 0 \\ 0 & 1/s & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T_3^{-1} = \begin{pmatrix} A''_x & -A''_y & 0 \\ A''_y & A''_x & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T^{-1} = T_1^{-1} T_2^{-1} T_3^{-1}$$

= =