CSC2503 Assignment 1

Joonil Ahn (1003311365)

1. Noise model

From the assumptions in the question, $\vec{x}_k = \widehat{\vec{x}_k} + \vec{m}$ with $m \sim \mathcal{N}(0, \sigma^2 I)$, we get

$$\mu = E\left[\hat{\vec{x}}\right] = E[\vec{x}].$$

We can rewrite the error term as follows.

$$\begin{aligned} e_k = \vec{a}^T\vec{x}_k + b + {\vec{x}_k}^T\vec{x}_k &= \vec{a}^T\vec{x}_k + b + (\vec{x}_k - \mu)^T(\vec{x}_k - \mu) + 2\mu^T\vec{x}_k - \mu^T\mu \\ &= (\vec{a}^T + 2\mu^T)\vec{x}_k - \mu^T\mu + b + (\vec{x}_k - \mu)^T(\vec{x}_k - \mu) \\ &= \underbrace{(\vec{a}^T + 2\mu^T)\vec{x}_k - \mu^T\mu + b}_{(1)} + \underbrace{\sigma^2\left[\frac{(\vec{x}_k - \mu)^T}{\sigma}\frac{(\vec{x}_k - \mu)}{\sigma}\right]}_{(2)} \end{aligned}$$

Let's look at the (1) term. Since $\vec{x}_k$ follows normal distribution with mean μ and variance σ^2, mean and variance for the (1) term is

$$Y = (\vec{a}^T + 2\mu^T)\vec{x}_k - \mu^T\mu + b$$

$$\begin{aligned} E[Y] &= (\vec{a}^T + 2\mu^T)E[\vec{x}_k] - \mu^T\mu + b \\ &= (\vec{a}^T\mu + 2\mu^T\mu - \mu^T\mu + b \\ &= \vec{a}^T\mu + b + \mu^T\mu \end{aligned}$$

$$\begin{aligned} Var[Y] &= Var[(\vec{a}^T + 2\mu^T)\vec{x}_k] \\ &= \|\vec{a} + 2\mu\|^2 * Var[\vec{x}_k] \\ &= \|\vec{a} + 2\mu\|^2\sigma^2 \end{aligned}$$

Thus, $(\vec{a}^T + 2\mu^T)\vec{x}_k - \mu^T\mu + b \sim \mathcal{N}(\vec{a}^T\mu + b + \mu^T\mu, \|\vec{a} + 2\mu\|^2\sigma^2)$

Let's see the (2) term of the equation above. We assume X to be a random variable of $\vec{x}_k$. Then

$$Z = \frac{X - E[X]}{\sigma} = \frac{X - \mu}{\sigma} \sim \mathcal{N}(0,1)$$

$$\frac{(\vec{x}_k - \mu)^T}{\sigma}\frac{(\vec{x}_k - \mu)}{\sigma} = Z^2 \sim \mathcal{X}^2(2)$$

We have the (2) term of the error above is Chi-square distribution with 2 degrees of freedom since X includes two variables. Again,

$$\begin{aligned} e_k &= (\vec{a}^T + 2\mu^T)\vec{x}_k - \mu^T\mu + b + \sigma^2\left[\frac{(\vec{x}_k - \mu)^T}{\sigma}\frac{(\vec{x}_k - \mu)}{\sigma}\right] \\ &= \mathcal{N}(\vec{a}^T\mu + b + \mu^T\mu, \|\vec{a} + 2\mu\|^2\sigma^2) + \sigma^2\mathcal{X}^2(2) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{\left(-(x - \vec{a}^T\mu - b - \mu^T\mu)^2 / 2\sigma_Q{}^2\right)} + \frac{\sigma^2}{2}e^{-x/2} \end{aligned}$$

We can also get the mean and variance for the distribution.

$$\mathrm{E}[e_k] = E[Y] + E[\sigma^2 Z] = \vec{a}^T\mu + b + \mu^T\mu + \sigma^2 E[\mathcal{X}^2(2)]$$
$$= \vec{a}^T\mu + b + \mu^T\mu + \sigma^2$$
$$Var[e_k] = Var[Y + \sigma^2 Z] = Var[Y] + \sigma^4 Var[Z] + 2 \cdot Cov[Y, \sigma^2 Z]$$
$$= \|\vec{a} + 2\mu\|^2\sigma^2 + 4\sigma^4 + 2\sigma^2 \cdot Cov[Y, Z]$$

Let's simplify $Cov[Y, Z]$.

$$Cov[Y, Z] = Cov[(\vec{a}^T + 2\mu^T)X + b, Z^2] = Cov\left[(\vec{a}^T + 2\mu^T)X, \left(\frac{X-\mu}{\sigma}\right)^2\right]$$
$$= \frac{\vec{a} + 2\mu}{\sigma^2} Cov[X, (X-\mu)^2]$$
$$= \frac{\vec{a} + 2\mu}{\sigma^2} Cov[X, X^2]$$
$$= \frac{\vec{a} + 2\mu}{\sigma^2} (E[X^3] - E[X] \cdot E[X^2])$$
$$= \frac{\vec{a} + 2\mu}{\sigma^2} (E[X^3] - \mu E[X^2])$$

Then, the variance is

$$Var[e_k] = \|\vec{a} + 2\mu\|^2\sigma^2 + 2\sigma^4 + 2\sigma^2 \cdot Cov[Y, Z]$$
$$= \|\vec{a} + 2\mu\|^2\sigma^2 + 2\sigma^4 + 2\sigma^2 \frac{\vec{a} + 2\mu}{\sigma^2} (E[X^3] - \mu E[X^2])$$
$$= \|\vec{a} + 2\mu\|^2\sigma^2 + 2\sigma^4 + 2(\vec{a} + 2\mu)^T (E[X^3] - \mu E[X^2])$$

Hence, the distribution of the errors e_k can be represented as the sum of Normal distribution and Chi-square distribution. The expectation of error e_k is linearly dependent to the circle parameters $\vec{a}$ and b.

2. LS estimator

2.1. Reasonable choice for the variance of the Gaussian noise

We assume the error is a mean-zero Gaussian noise. We should find the variance to be close to the distribution in the question 1.

To estimate the difference between two distributions, KL divergence can be used.

$$KL(P||Q) = \int_{-\infty}^{\infty} P(x) \frac{P(x)}{Q(x)}$$

Let's say $P(x)$ is the pdf of Chi-square distribution and $Q(x)$ is the pdf of the normal distribution for the Gaussian error. We assume $P(x)$ is significantly affected by Chi-square distribution because the Chi-square distribution with 2-dof is actually very right skewed distribution. Also, it makes us easier to minimize the KL divergence analytically.

Since the degree of freedom for ${\sigma_P}^2 \mathcal{X}^2$ is 2,

$$P(x) = \frac{{\sigma_P}^2}{2} e^{-x/2}$$
$$Q(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{\left(-x^2/2{\sigma_Q}^2\right)}$$

The Chi-squared distribution has no value for negative inputs, so the integral will be from zero to positive infinity.

$$KL(P||Q) = \int_0^\infty \frac{{\sigma_P}^2}{2} e^{-x/2} \log\left(\frac{{\sigma_P}^2}{2} e^{-x/2} \Big/ \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{\left(-x^2/2{\sigma_Q}^2\right)}\right)$$

$$= \int_0^\infty \frac{{\sigma_P}^2}{2} e^{-x/2} \left[\log\left(\frac{{\sigma_P}^2}{2} e^{-x/2}\right) - \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{\left(-x^2/2{\sigma_Q}^2\right)}\right)\right]$$

$$= \int_0^\infty \frac{{\sigma_P}^2}{2} e^{-x/2} \left(\log\frac{{\sigma_P}^2}{2} - \frac{x}{2} + \frac{1}{2}\log 2\mu{\sigma_Q}^2 + \frac{x^2}{2{\sigma_Q}^2}\right)$$

$$= \frac{{\sigma_P}^2}{2}\log\frac{{\sigma_P}^2}{2}\int_0^\infty e^{-x/2} - \frac{{\sigma_P}^2}{4}\int_0^\infty xe^{-x/2} + \frac{{\sigma_P}^2}{4}\log 2\pi{\sigma_Q}^2\int_0^\infty e^{-x/2} + \frac{{\sigma_P}^2}{4{\sigma_Q}^2}\int_0^\infty x^2e^{-x/2}$$

$$= \frac{{\sigma_P}^2}{2}\left(\log\frac{{\sigma_P}^2}{2} + \frac{1}{2}\log 2\pi{\sigma_Q}^2\right)\int_0^\infty e^{-x/2} - \frac{{\sigma_P}^2}{4}\int_0^\infty xe^{-x/2} + \frac{{\sigma_P}^2}{4{\sigma_Q}^2}\int_0^\infty x^2e^{-x/2}$$

$$= \frac{{\sigma_P}^2}{2}\left(\log\frac{{\sigma_P}^2}{2} + \frac{1}{2}\log 2\pi{\sigma_Q}^2\right)\left[-2e^{-x/2}\right]_0^\infty - 4\frac{{\sigma_P}^2}{4}\left[-2xe^{-x/2} + 4xe^{-x/2}\right]_0^\infty$$
$$+ \frac{{\sigma_P}^2}{4{\sigma_Q}^2}\left[-2x^2e^{-x/2} - 8xe^{-x/2} - 16e^{-x/2}\right]_0^\infty$$

$$= \frac{{\sigma_P}^2}{2}\left(\log\frac{{\sigma_P}^2}{2} + \frac{1}{2}\log 2\pi{\sigma_Q}^2\right)(2) + \frac{{\sigma_P}^2}{4}(-4) + \frac{{\sigma_P}^2}{4{\sigma_Q}^2}(16)$$

$$= {\sigma_P}^2(\log(\tfrac{{\sigma_P}^2}{2}) - 1) + {\sigma_P}^2\log 2\pi{\sigma_Q}^2 + 4\tfrac{{\sigma_P}^2}{{\sigma_Q}^2}$$

To minimize the KL divergence, we take the derivative w.r.t σ_Q. Then

$$\frac{dKL}{d\sigma_Q} = 2\frac{{\sigma_P}^2}{2}\log(2\pi)\,\sigma_Q - \frac{4{\sigma_P}^2}{{\sigma_Q}^3} = 0$$

Thus,

$$\begin{aligned}{\sigma_Q}^4 &= \frac{4{\sigma_P}^2}{{\sigma_P}^2\log(2\pi)}\\ \sigma_Q &\approx 1.2146\end{aligned}$$

2.2. Beginning with the likelihood, derive a least-squares estimator

Suppose we have a set of independent measurements, $\{\vec{x}_k\}_{k=1}^{N}$, all of which arise from the same circle with centre $\vec{x}_c$ and radius r. Let's assume that the target value for the observation is t_k and the equation between t_k and the circle can be written as

$$t_k{}^2 - r^2 = \|\overrightarrow{x_k} - \overrightarrow{x_c}\|^2 - \left\|\widehat{\overrightarrow{x_k}} - \overrightarrow{x_c}\right\|^2 = e_k$$

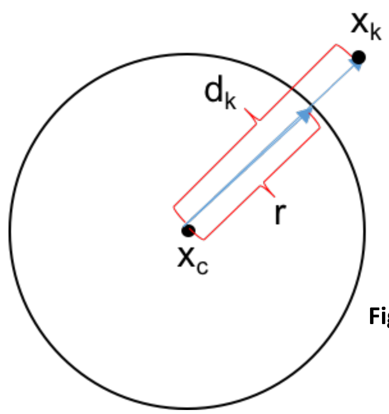

Figure 1. Circle and Target Measurement

where $\widehat{\overrightarrow{x_k}}$ is the point on the circle with the minimum distance to the $\overrightarrow{x_k}$.

Here we assume the distribution of the errors, $e = [e_1, \dots, e_N]^T$, are i.i.d. according to mean-zero Gaussian distribution. Then,

$$e \sim \mathcal{N}(0, \sigma^2 I)$$

Since the mean of e is zero and $t_k{}^2 = r^2 + e_k$, we can write

$$p(t_k{}^2 | \overrightarrow{x_k}, \overrightarrow{x_c}, r) = \mathcal{N}\left(t^2 \middle| \left\|\widehat{\overrightarrow{x_k}} - \overrightarrow{x_c}\right\|^2, \sigma^2\right) = \mathcal{N}(t^2 | r^2, \sigma^2)$$

Here we can denote the target variable as y, the square of t_k, then the likelihood function can be written in the form of

$$p(\mathbf{y} | \mathbf{x}, \overrightarrow{x_c}, r) = \prod_{k=1}^{N} \mathcal{N}(y_k | r^2, \sigma^2)$$

We want the distribution to maximize the likelihood for the observation. Hence, we can determine $\overrightarrow{x_c}$ and r such that maximizes the likelihood. For the mathematical convenience, we can take the logarithms of the likelihood function. It does not change the solution because the parameters maximizing the likelihood also maximizes the log-likelihood. The log-likelihood can be represented as

$$\begin{aligned}
\log p(\mathbf{y} | \mathbf{x}, \overrightarrow{x_c}, r) &= \log \prod_{k=1}^{N} \mathcal{N}(y_k | r^2, \sigma^2) \\
&= \sum_{k=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(t_k{}^2 - r^2)^2}{2\sigma^2}\right) \\
&= \sum_{k=1}^{N} -\frac{(\|\overrightarrow{x_k} - \overrightarrow{x_c}\|^2 - r^2)^2}{2\sigma^2} - \sum_{k=1}^{N} \log \sqrt{2\pi\sigma^2}
\end{aligned}$$

$$= -\sum_{k=1}^{N} \frac{(\|\overrightarrow{x_k} - \overrightarrow{x_c}\|^2 - r^2)^2}{2\sigma^2} - \frac{1}{2}\sum_{k=1}^{N} \log 2\pi\sigma^2$$
$$= \sum_{k=1}^{N} \frac{\left(-2\overrightarrow{x_c}^T \overrightarrow{x_k} + \overrightarrow{x_c}^T \overrightarrow{x_c} - r^2 + \overrightarrow{x_k}^T \overrightarrow{x_k}\right)^2}{2\sigma^2} - \frac{1}{2}\sum_{k=1}^{N} \log 2\pi\sigma^2$$
$$= -\sum_{k=1}^{N} \frac{(\vec{a}^T \overrightarrow{x_k} + b + \overrightarrow{x_k}^T \overrightarrow{x_k})^2}{2\sigma^2} - \frac{1}{2}\sum_{k=1}^{N} \log 2\pi\sigma^2$$

where, $\vec{a} \equiv -2\overrightarrow{x_c}$ and $b \equiv \overrightarrow{x_c}^T \overrightarrow{x_c} - r^2$.

Maximizing the log-likelihood is equivalent to minimizing negative log-likelihood, so we should minimize the negative log-likelihood to find the optimal parameters. The variance is a constant value which does not change the solution, so we can write the problem like

$$minimize \sum_{k=1}^{N} (\vec{a}^T \overrightarrow{x_k} + b + \overrightarrow{x_k}^T \overrightarrow{x_k})^2$$

The objective function above is least-square estimator since it is linear in $\vec{a}$ and b.

3. Circle proposals

We can generate initial circle proposals with as follows.

i) Randomly choose a point ${x_1}^{(1)}$ from the edgels.

ii) Compute the dot product between $\overrightarrow{n_1}^{(1)}$, the normal vector for ${x_1}^{(1)}$, and all the normal vectors for the edgels.

$$\overrightarrow{n_1}^{(1)} \cdot \vec{n}^{(i)} \ for\ all\ i$$

iii) Find the index k such that $\overrightarrow{n_1}^{(1)} \perp \vec{n}^{(k)}$ and let $\overrightarrow{n_2}^{(1)} \leftarrow \vec{n}^{(k)}$ and ${x_2}^{(1)} \leftarrow x^{(k)}$
If no vector is perpendicular to $\overrightarrow{n_1}^{(1)}$, find the vector with the minimum dot product.
If multiple vectors found, choose the point with the minimum distance to $\overrightarrow{x_1}^{(1)}$.
(See Fig 4. for the explanation)

iv) Compute the angle between the two vectors.

$$\theta = \cos^{-1}\left({\vec{n}_1}^{(1)} \cdot {\vec{n}_2}^{(1)}\right)$$

v) Compute the distance between two points.

$$d^{(1)} = \left\|{x_1}^{(1)} - {x_2}^{(1)}\right\|$$

vi) Estimate radius for a circle using basic geometry.

vii) Estimate the center from ${x_1}^{(1)}$ and radius.

viii) Go back to i). Repeat i) to vii) until generate the 'numGuesses' of circles.

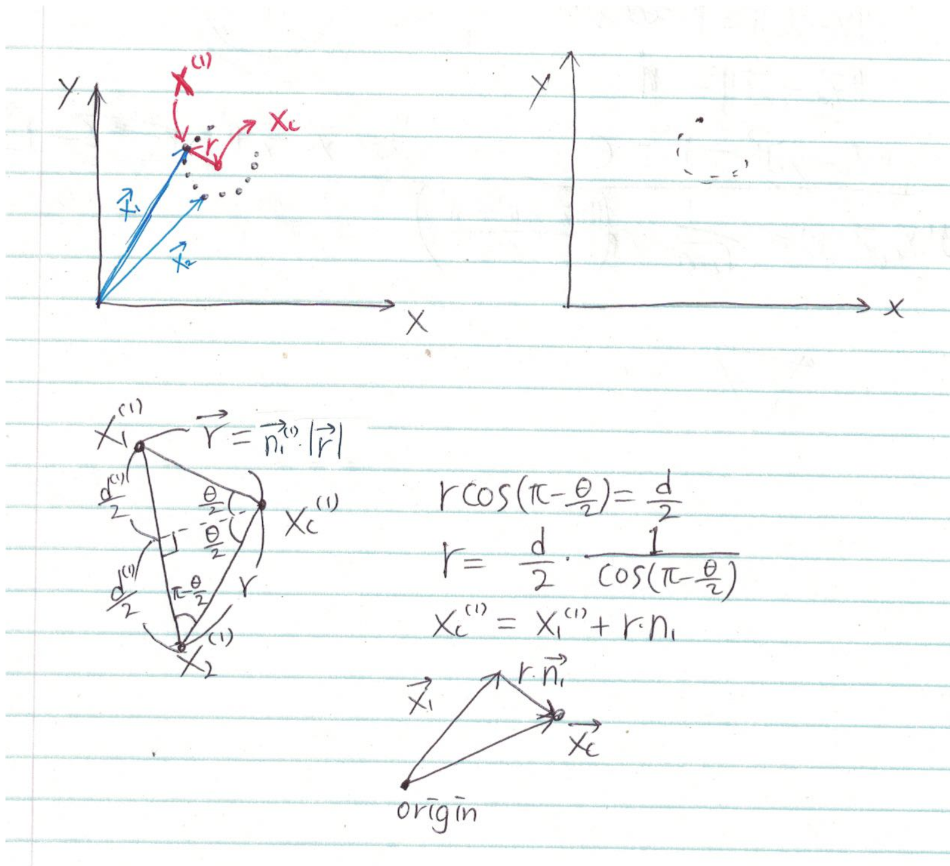

Fig 2. Estimating Radius and a Center of a Circle Using Geometry

This algorithm has two major advantages.

Firstly, it is computationally fast because the algorithm uses basic calculations. It does not use linear regression or other optimization algorithms in this stage.

Secondly, the most of proposed circles will be located inside the edgels. Suppose we select any two points from the edgels in random and let the mid-point of the two points be a center. Then it may propose too large circles that are not good initial points for the robust fitting. To fit multiple circles, the initial guess should be close to at least one circle fitting among many possible circles.

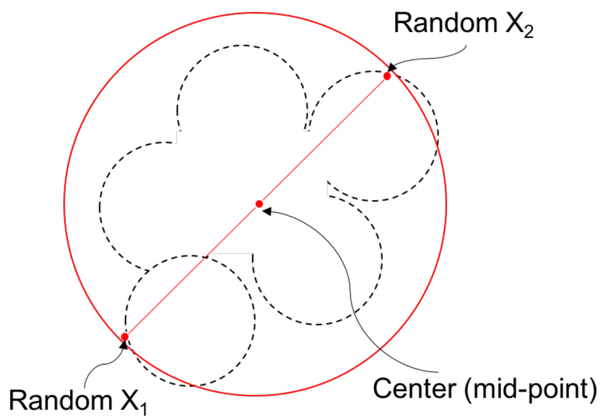

Fig 3. Drawback for Random Two-Points Method

The Fig 4 shows how the algorithm would generate circles with the normal vectors. With two perpendicular vectors, we can quickly make a circle. The second case in the Fig 4. is the most likely scenario using the algorithm. The third case can happen when the two vectors are perpendicular but too far. For the reason, the algorithm would select a point with minimum distance among many candidates.

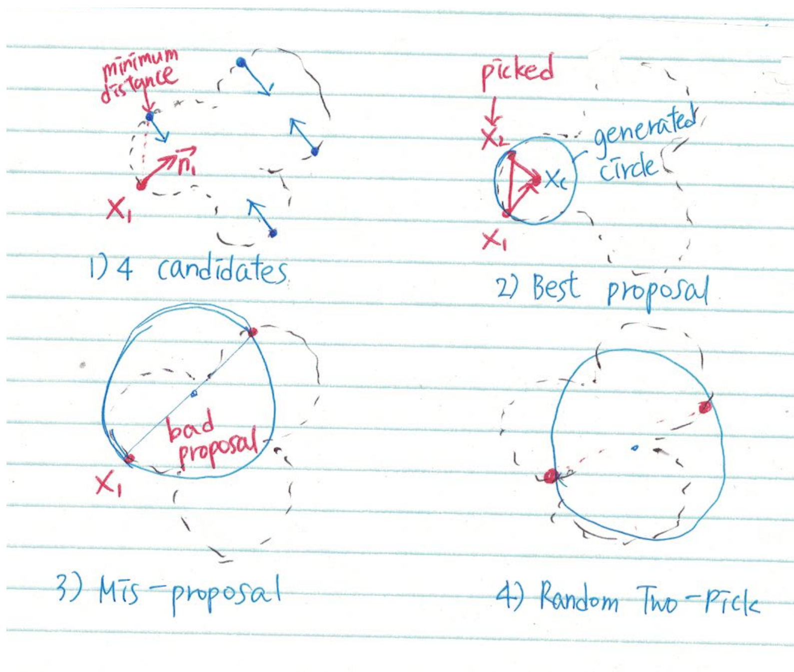

Fig 4. Circle Proposals with Two Perpendicular Vectors

4. Circle selection

We should select the best circle among the proposals generated from the first step. The best circle should be close to many edgels. For each circle, we can compute error using squared sum of distances for each circle.

$$error_{kth\ circle} = \sum_{i=1}^{P} {d_i}^2 - {r^{(k)}}^2 = \sum_{i=1}^{P} \left\| p_i - x_c^{(k)} \right\|^2 - {r^{(k)}}^2$$

However, this kind of error misleads the result because negative value exists for the error. If $\left\| p_i - x_c^{(k)} \right\|^2$ is smaller than ${r^{(k)}}^2$, the point is inside the circle. If there is only one circle to fit the edgels, positive and negative values do not have to be distinguished. For multiple circles, however, we need an initial guess to fit at least one circle. We do not prefer a circle with minimized error that may across all the possible edgels. The Fig 5. explains why negative value for the error should be penalized more than positive error.

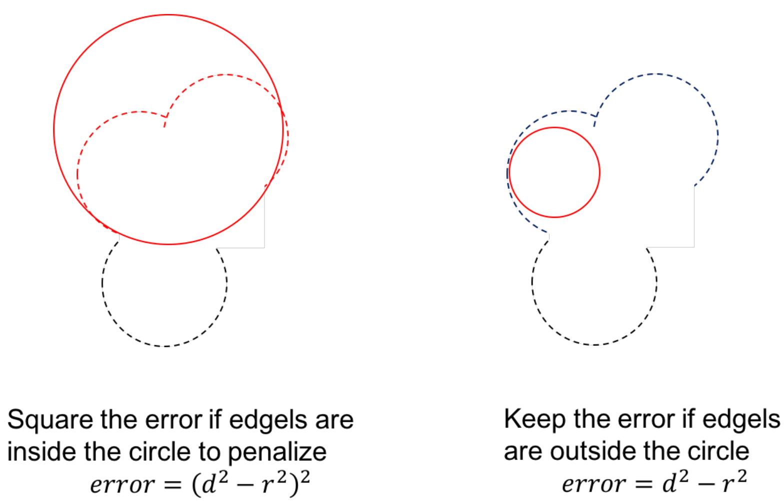

Fig 5. Difference between Two Types of Proposals

We prefer the circle on the right side than the left circle in Fig 5. because the left circle would generalize over two possible circles. It may harm the convergence of the robust fitting. Hence, we will square the error if the error is negative.

$$error1 = \sum_{i=1}^{N} \left({d_i}^2 - {r^{(k)}}^2 \right)^2 \quad , if\ {d_i}^2 < r^2$$

$$error2 = \sum_{j=1}^{M} {d_j}^2 - {r^{(k)}}^2 \ , otherwise$$

$$TotalError_{kth\ circle} = error1 + error2$$

After computing the errors for all proposals, we choose the circle with the minimum total error.

5. Robust fitting

5.1. Derivation of IRLS algorithm

Given a set of edgel positions $\{\vec{x}_k\}_{k=1}^{K}$, we should find the circle parameters to minimize the following objective function.

$$O(\vec{a}, b) = \sum_k \rho(e_k(\vec{a}, b), \sigma_g)$$

where, $e_k(\vec{a}, b) = \vec{a}^T \overrightarrow{x_k} + b + \overrightarrow{x_k}^T \overrightarrow{x_k}$

For this assignment, we used Geman-Mcclure function for $\rho(e_k(\vec{a}, b), \sigma_g)$.

$$\rho(e_k, \sigma_g) = \frac{{e_k}^2}{{\sigma_g}^2 + {e_k}^2}$$

The necessary condition for the minimum is that the derivatives w.r.t $\vec{a}$ and b are zeros.

$$\begin{aligned}\frac{\partial O}{\partial \vec{a}}(\vec{a}, b) = \sum_k \frac{\partial \rho}{\partial e}(e_k) \frac{\partial e}{\partial \vec{a}}(\vec{a}, b) &= \sum_k \frac{2{\sigma_g}^2}{\left({\sigma_g}^2 + {e_k}^2\right)^2} e_k \vec{x}_k \\ &= \sum_k w(e_k)\left(\vec{a}^T \vec{x}_k + b + {\vec{x}_k}^T \vec{x}_k\right) \vec{x}_k \\ &= 0\end{aligned}$$

$$\begin{aligned}\frac{\partial O}{\partial b}(\vec{a}, b) = \sum_k \frac{\partial \rho}{\partial e}(e_k) \frac{\partial e}{\partial b}(\vec{a}, b) &= \frac{2{\sigma_g}^2}{\left({\sigma_g}^2 + {e_k}^2\right)^2} e_k \\ &= \sum_k w(e_k)\left(\vec{a}^T \vec{x}_k + b + {\vec{x}_k}^T \vec{x}_k\right) \\ &= 0\end{aligned}$$

where $w(e_k) = \frac{1}{e_k} \frac{\partial \rho}{\partial e}(e_k) = \frac{2{\sigma_g}^2}{\left({\sigma_g}^2 + {e_k}^2\right)^2}$.

The two equations above can be rewritten in a matrix form. Let $\vec{x}_k$ be a column vector and the $\vec{a}$ and b can be grouped in a Vector $\mathbf{p} \equiv [\vec{a}, b]^T$.

$$C = \begin{bmatrix} \vec{x}_1 \dots \vec{x}_K \\ 1 \quad\;\; 1 \end{bmatrix}$$

$$d = \begin{bmatrix} {\vec{x}_1}^T \vec{x}_1 \\ \vdots \\ {\vec{x}_K}^T \vec{x}_K \end{bmatrix}$$

$$W = diag(w) = \begin{bmatrix} w_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_K \end{bmatrix}$$

C is a 3x1 matrix, **d** is a Kx1 vector, **W** is a KxK diagonal matrix and **p** is a 3x1 vector. With these matrices and vectors, we can rewrite the necessary condition.

$$(C_{3\times K}^T W_{K\times K} C_{K\times 3})\mathrm{p} + C^T{}_{3\times K} W_{K\times K} \mathrm{d}_{K\times 1} = 0_{3\times 1}$$

$A \equiv C^T WC$ and $B = C^T WD$. Then, solving the following equation gives us the solution vector p.

$$Ap + B = 0$$

$$\mathrm{p} = -A^+ B = -(A^T A)^{-1} A^T B$$

where A^+ is a pseudo inverse of A. In the solution p, the first two elements denote $\vec{a}$ and the third element denotes b. We can simply get the center and the radius for the fitted circle using

$\vec{a} = -2\vec{x}_c$ and $b = \overrightarrow{x_c}^T \overrightarrow{x_c} - r^2$.

Here is the brief explanation of IRLS algorithm.

(1) Initial $\vec{a}^{(k)} = -2\vec{x}_c^{\,(k)}$ and $b^{(k)} = \vec{x}_c^{\,(k)^T} \vec{x}_c^{\,(0)} - {r^{(k)}}^2$
(2) Compute the weights $w(e_k) = \frac{1}{e_k}\frac{\partial \rho}{\partial e}(e_k) = \frac{2{\sigma_g}^2}{({\sigma_g}^2 + {e_k}^2)^2}$ for all k
(3) Create the matrices C, d and W
(4) Solve the equation, $\mathrm{p} = -A^+ B$, and get $\vec{a}^{(k+1)}$ and $b^{(k+1)}$
(5) if $\left\|\vec{a}^{(k+1)} - \vec{a}^{(k)}\right\| < \varepsilon$ and $\left\|b^{(k+1)} - b^{(k)}\right\| < \varepsilon$, stop
otherwise, go back to (1)

5.2. Experiment with parameter ${\sigma_g}^2$

In practice, we need to experiment with different value of σ_g to find the most suitable value for the dataset. For example, run least square method for the simplest samples to find error range for the inliers. With the fitted circle, calculate the variance of errors of inliers and set σ_g as the variance. Starting with the variance and adjust reasonably with experiments to find good σ_g.

If σ_g is too small, however, some of the inliers may be treated as outliers. In other words, as the error increases, the influence function increases but descends too quickly so only few number of points can be inliers.

If σ_g is too large, it is hard to detect the outliers. In other words, the descending point of influence function arise at relatively large error value. So it would act like L_1 estimator.

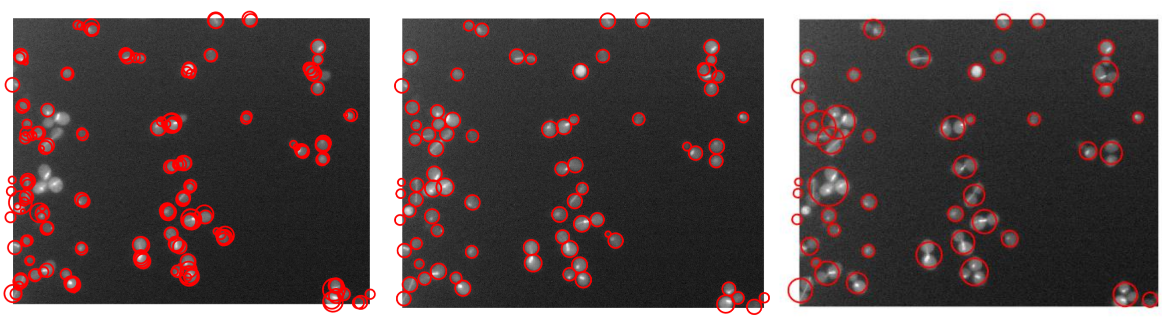

Fig 6. Fittings with Different σ, the left σ=10, the middle σ=75 and the right σ=500

It can be seen that the circles were too sensitive with close outliers (should have to be inliers) when σ_g is very small, at 10, while the circles did not distinguish the outliers when σ_g is too large, at 500.

When σ_g is 75, the algorithm showed the best result for the dataset.

6. Model update

We can determine how a model is good analyzing distributions of weights.

(1) Rescale the weights to have max possible value 1
(2) Set threshold for the weight (0.5 was used in the code)
(3) Sum(weights > threshold) which is called 'strongW' in the code
(4) Compute sum(strongW) / sum(w), we call it support ratio
(5) If the (support ratio) > 0.7, good circle

There are many possible values for the threshold so the suitable value should be determined through empirical method. The support ratio also can be found with many experiments.

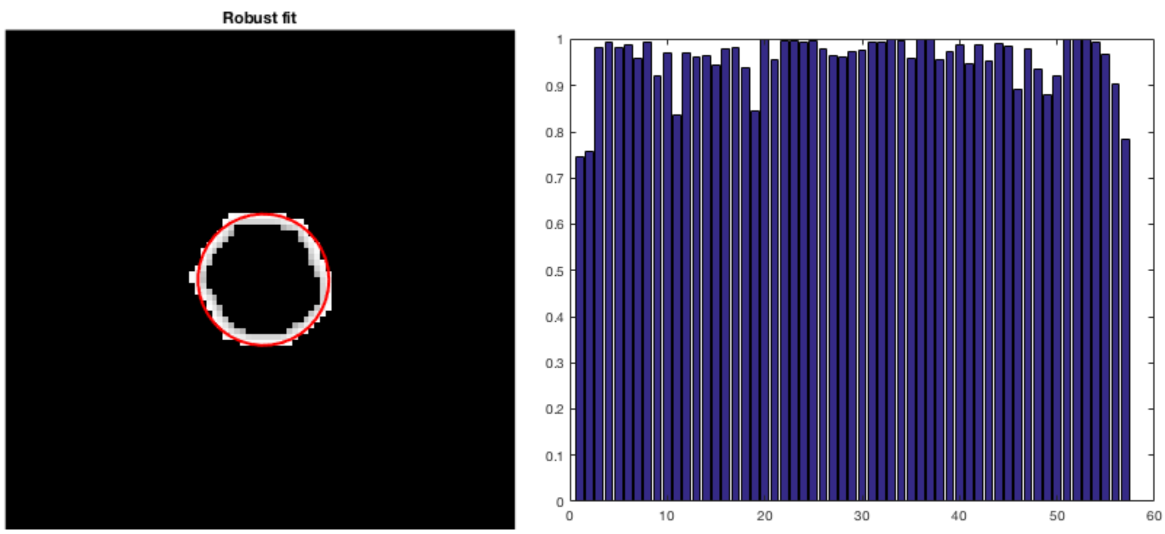

Fig 7. Robust Fit Circle and Weight Distribution for Good Example

Fig 7 shows a fitted for some edgels and a distribution of weights. It can be seen that the most of the weights are above 0.7 so sum of the weights above threshold would be close to the sum of all weight.

Fig 8 shows an bad example for the fitting. People might expect three different circles for the edgels in the figure, however, the example fitted one circle for the edgels. The distribution looks like a multi-modal distribution. Also sum of weights above threshold would not be close to the sum of all weights. Hence, we can assume if the ratio is low the model is not a good fit.

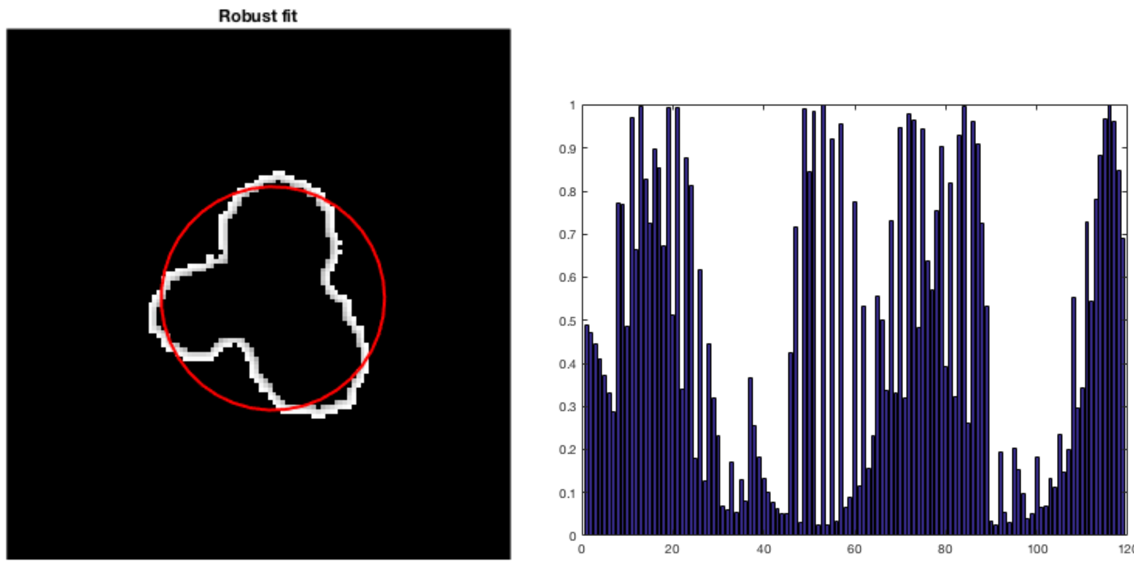

Fig 8. Robust Fit Circle and Weight Distribution for Bad Example

7. Brief evaluation

There is some noise in the measurements especially for the normal vectors. If the normal vectors do not point to the center of the possibly best circle, the initial proposals would not close to the best fit so it may fail.

If more than two cells are closely located, they may look like one cell so the cells might be fitted by one circle. To solve this problem, we can temporarily use very small σ_g then the algorithm would be strict with some outliers.

On the other hand, the performance of the algorithm can be sensitive with the threshold of weight used in isGoodCircle function. Although the sum of weights above threshold has useful information about the distribution, it may not be the most accurate way to estimate the distribution of weights. Mixed Gaussian fitting to the weight distribution might solve the problem. If a distribution is fitted with a multi modal Gaussian, then it might not be a good circle.