Lect 8 Fitting

Question:

How to compose simple features to complicated model?

Parametric Model.

Use parameter to specify (uniquely) the model

e.g. for line, only need two points

for circle. radius & center

Issues:

1. Noise.
2. Extraneous data, (outliers)
3. Missing data

Method 1. (Naive). — Least Square.

for points (x_i, y_i) $i = 1, 2, \cdots n$. and model $y = mx + b$

cost function $E = \Sigma[y_i - (mx_i + b)]^2$

Turn to matrix version

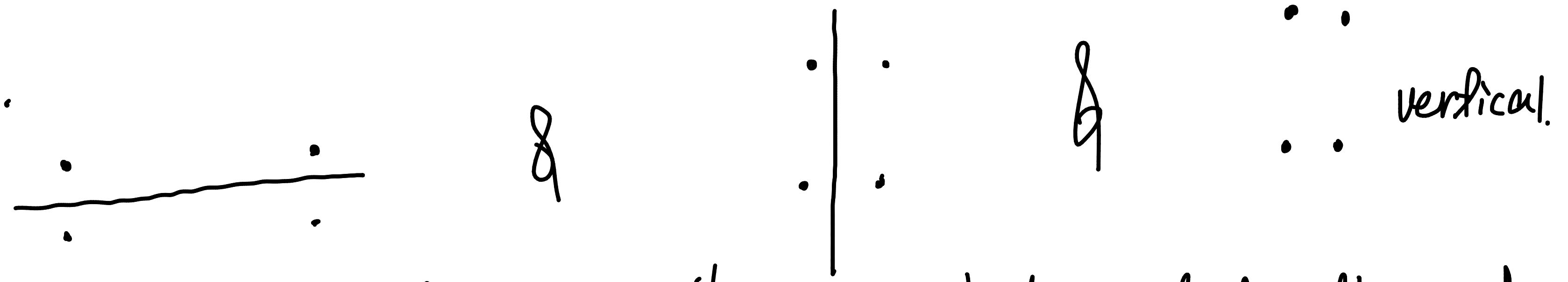
$$E = \| Y - [X, \mathbb{1}] \begin{bmatrix} m \\ b \end{bmatrix} \|_2^2 = \| Y - XB \|_2^2$$

$$B = X^+ Y$$

Problem:

This method is not rotation-invariant.

e.g.



Their relative positions are the same, but cost function value is different.

Method 2 — Total Least Square.

$$E \triangleq \sum |\text{point-to-line}|^2$$

if the line is $ax+by=d$, $a^2+b^2=1$, point is (x_i, y_i)

$$E = \sum (ax_i+by_i-d)^2$$

$$\frac{\partial}{\partial d} E = -2\sum (ax_i+by_i-d) = 0 \Rightarrow d = \frac{a}{N}\sum x_i + \frac{b}{N}\sum y_i = a\bar{x}+b\bar{y}$$

Then

$$E = \sum \left(a(x_i-\bar{x}) + b(y_i-\bar{y})\right)^2 = \| \underbrace{[X-\bar{x},\ Y-\bar{y}]}_{U} \begin{bmatrix} a \\ b \end{bmatrix} \|_2^2 = \left(U\begin{bmatrix} a \\ b \end{bmatrix}\right)^T U \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\min_{a,b} E$$

s.t. $a^2+b^2=1$

$$E = \| U \begin{bmatrix} a \\ b \end{bmatrix} \|_2^2$$

$$= \| U_\Sigma \Sigma V_\Sigma^T \begin{bmatrix} a \\ b \end{bmatrix} \|_2^2 \Rightarrow \min_{a,b} E = \Sigma^{(end)}, \quad \begin{bmatrix} a' \\ b' \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$= \| \Sigma \begin{bmatrix} a' \\ b' \end{bmatrix} \|_2^2 \qquad \text{s.t. } a^2+b^2=I$$

$$\therefore V_\Sigma^T \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ b \end{bmatrix} = V_\Sigma \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix} = V_\Sigma(:, end)$$

Note that $V_\Sigma(:, end)$ is from U.

$$U^T U = \begin{bmatrix} \Sigma(X-\bar{X})^2 & \Sigma(X-\bar{X})(Y-\bar{Y}) \\ \Sigma(X-\bar{X})(Y-\bar{Y}) & \Sigma(Y-\bar{Y})^2 \end{bmatrix} = \underbrace{\begin{bmatrix} Var(X) & Cov(X,Y) \\ Cov(X,Y) & Var(Y) \end{bmatrix}}_{\text{Second Moment}}$$

Problem: Outliers.

Method 3 Robust Estimation

$$E = \sum \rho(r_i(x_i, \theta), \sigma)$$

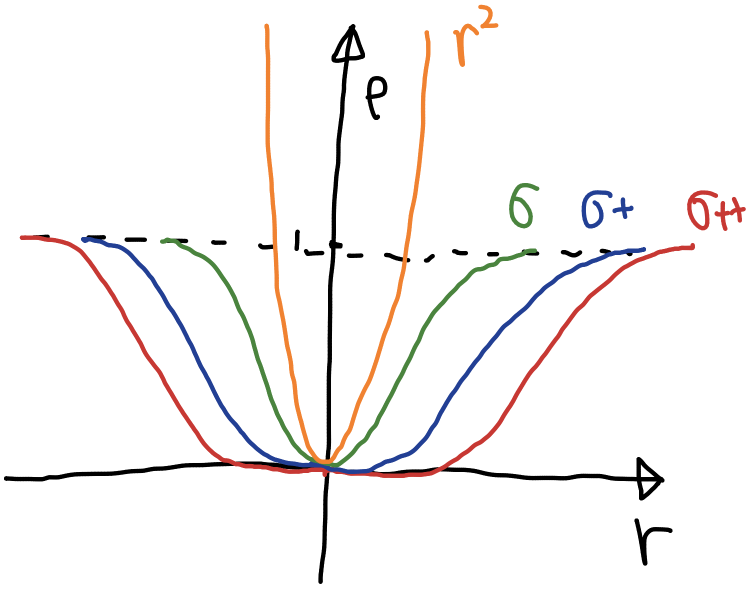
$r_i(x_i, \theta)$: residual of point x_i to θ

Can be distance, error, anything

$\rho(r, \sigma)$: penalty function.

$$\text{E.g. } \rho(r; \sigma) = \frac{r^2}{r^2+\sigma^2}$$

θ may only be solved iteratively, because $\rho(r; \sigma)$ can be non-convex.



Method 4 RANSAC (Random Sample Consensus)

1. chose the smallest set of points
2. Fit a model
3. Compute error
4. Count how many points are within tolerance
5. repeat 1-4, and save the model covering the most points.

P.S.

1. Initial points # is the minimun # required for step 1.
   Say, 2 points for a line, 3 points for a circle.
2. Tolerance t in step 2 is tricky. We can assume noise to be Gaussian, and use 3-σ rule to set it.
3. # of iterations N can be computed by:
   Target: at least 1 set is all free from outliers.
   e of points are outliers. say $e = 0.2$.
   p is probability of seeing error, say $p = 0.99$
   s is the minimum points to build model

$$\left(1-(1-e)^s\right)^N \leq 1-P \Rightarrow N \geq \log(1-P) / \log\left(1-(1-e)^s\right)$$

Adaptive Procedure

Initially, set inlier ratio to be low, sequentially update the setting.

Pros:

1. Simple and general
2. Works well in practice.

Cons:

1. lots of params
2. When inlier ratio is low, performance is bad
3. Can not get a good initialization of model based on minimum number of sample.

Mathod 5 Hough Method

Discretize parameter space to bins. each sample points vote for compatable bins. find the bin that has most votes.

E.g. Line fitting.

a point (x, y) in image space can correspond to any line passing it.

$b = -xk + y$, b & k together have 1-DoF.

So, in param space, b & k can correspond to a line.

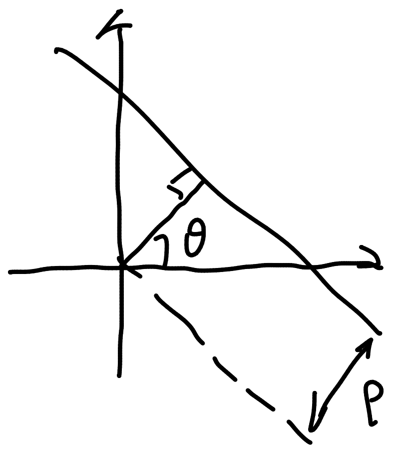
If we have line for (b_1, k_1) and line (b_2, k_2), the intersection is the $(\hat{b}, \hat{k})$ for both points.

Problem:

Some parameter can be infinite. Say verstical line.

Solution:

Use polar interpretation of line.



$$x\cos\theta + y\sin\theta = \rho$$

then, a point $(x, y) \rightarrow (\theta, \rho)$ is a sinusoid

$\because$ $x\cos\theta + y\sin\theta$

$$= \sqrt{x^2+y^2}\left(\frac{x}{\sqrt{x^2+y^2}}\cos\theta + \frac{y}{\sqrt{x^2+y^2}}\sin\theta\right)$$

let $\gamma = \arcsin\frac{x}{\sqrt{x^2+y^2}}$

then

$x\cos\theta + y\sin\theta$

$$= \sqrt{x^2+y^2}\,(\sin\gamma\cos\theta + \cos\gamma\sin\theta)$$

$$= \sqrt{x^2+y^2}\,\sin(\theta+\gamma)$$

So, $\rho = \sqrt{x^2+y^2}\,\sin(\theta+\gamma)$, $\gamma = \arcsin\frac{x}{\sqrt{x^2+y^2}}$

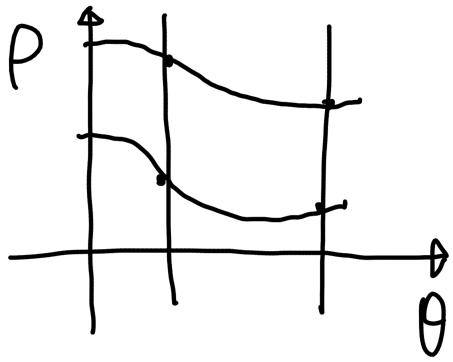
Algorithm.

Initial accumulator H

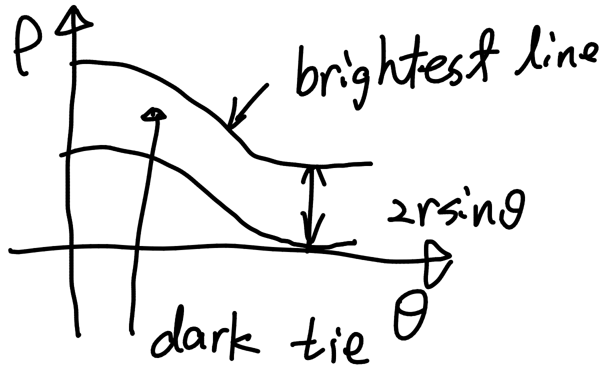
For each point (x_i, y_i), for θ from $0° \rightarrow 180°$,

Compute $\rho = x_i\cos\theta + y\sin\theta$. $H(\theta, \rho)$++

If we fit a square, we will see



For circle



Problem.

1. when noise exists, we may not see clear peaks
2. random noise will also generate peaks.

Improvement

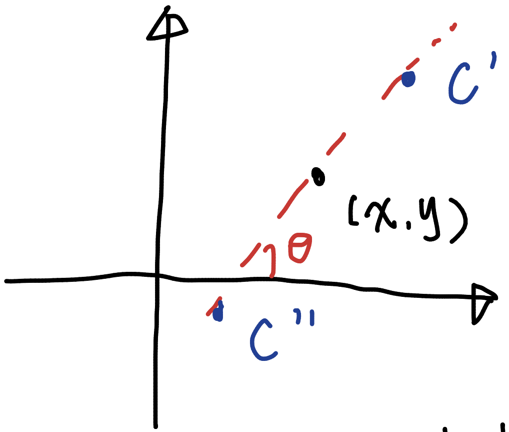
1. Choose a good grenutation, not ~~too~~ large nor too small.
2. Smooth, not only vote for certain bins, but also its neighbours.
3. Get rid of not important points.

P.S. For edge feature, let θ = gradient direction, no need to scan over θ, because we not only know its location, but also its direction

Hough transformation for circle.

for a circle, we need to determine (x, y, r)

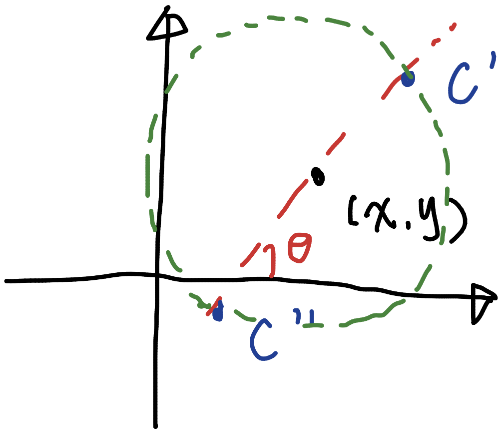
If we inspect on θ, which is the direction that circle center and (x, y) points.



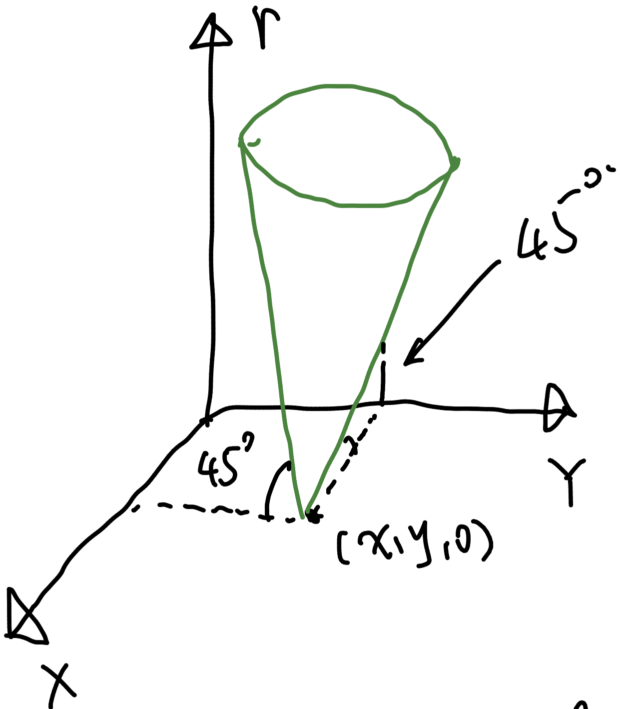
If we have specified θ, and r. potential center could only be c' or c''.

If we only fix r. loose constrain on θ.

We will have a circle.



For a specific r, we may have a circle. So, in space (x. y. r), a point will contribute to the surface of a corn



Question.

Will the inverse step functioning?

↳ Find a point in Param space, find how many image points support that circle

Bad Idea!

Because, features are sparse! The reverse method is computational intense.

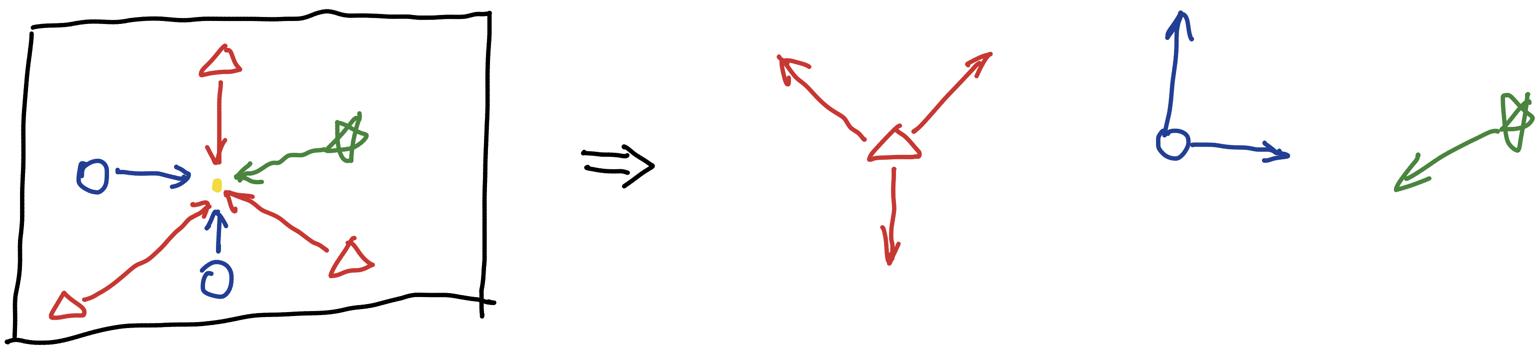
Generalized Hough Transformation

We can find some more complicated patten.

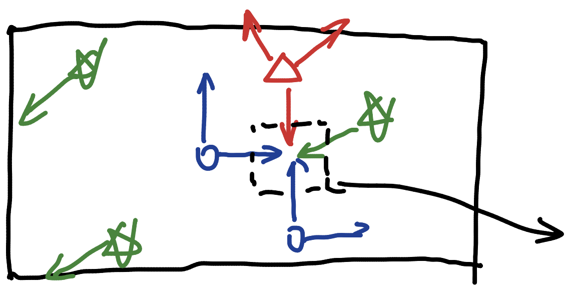
1. Template Representation.

Store the direction vector of each feature to its center

E.g.



for patten recognition.



has $1/3$ red + $2 \times 1/2$ blue + 1 green

$= 2 \cdot \frac{1}{3}$ votes

Pros:

- Can deal with non-locality & occlusion (missing some features)
- Can detect multi-instances.
- Some robust to noise

Cons:

- Search complexity increase exponentially w.r.t model param
- Non-target shape can cause spunbus peaks (false alarm)
- Grid / bin size is difficult to decide.