MANU2480

AUTONOMOUS SYSTEM

Mapping – Part 2

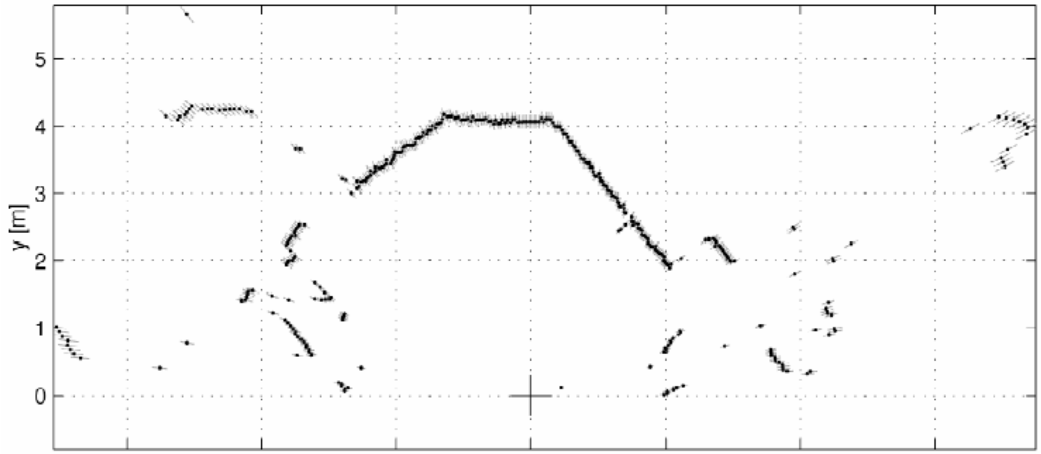
School of Science and Technology, RMIT Vietnam

Outlines

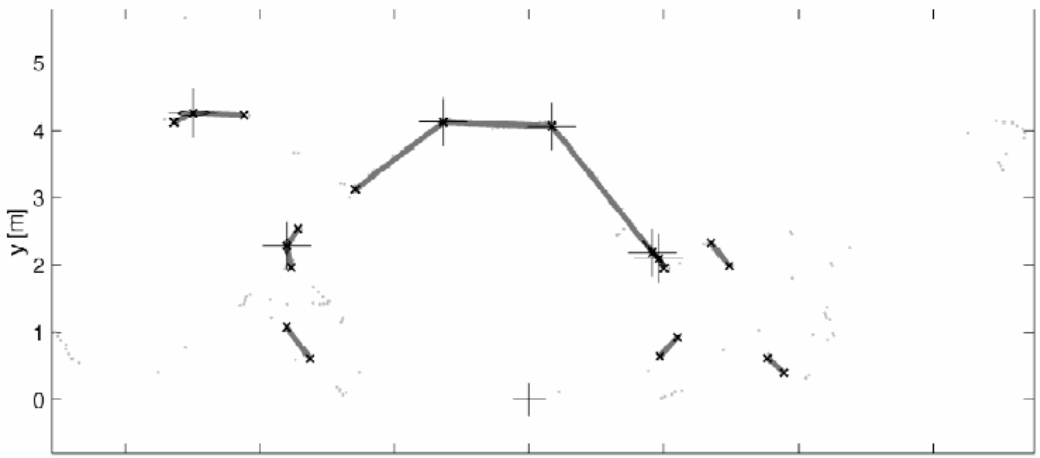
- Line Segmentation and Line Fitting
- Algorithms: Linear Regression, Split and Merge, RANSAC, Hough Transform.

Problem Statement

Raw range data



Line segments



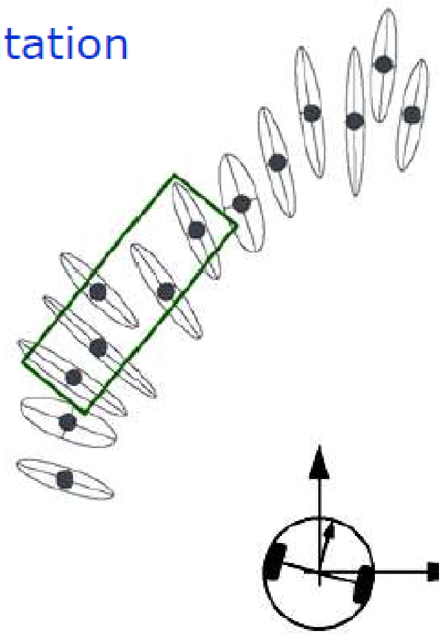
Problem Statement

Three main problems:

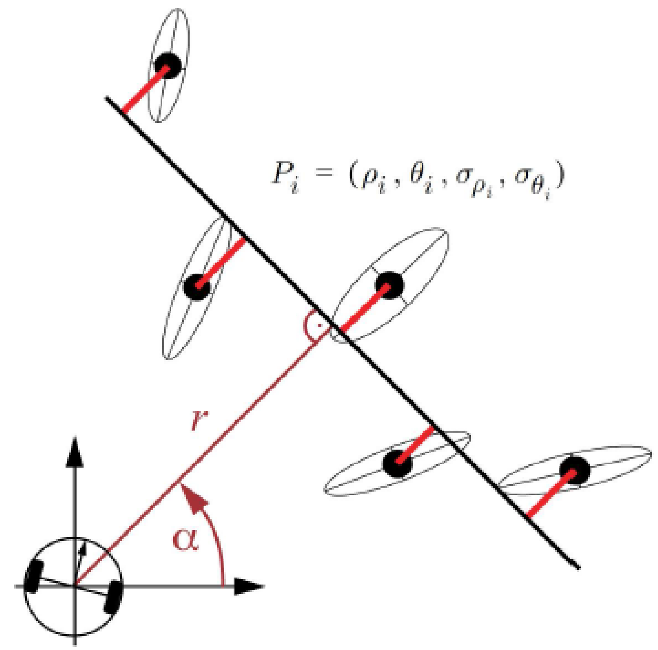
- How many lines?
- Which points belong to which line? This problem is called **segmentation**.
- Given the points that belong to a line, how to estimate the line parameters? This problem is called **line fitting**.

Problem Statement

Segmentation



Fitting



RMIT UNIVERSITY

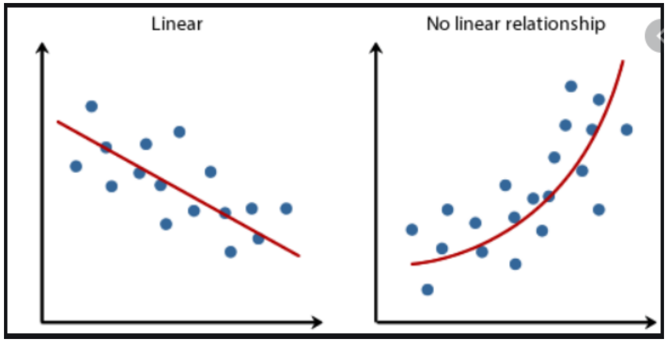
Approach

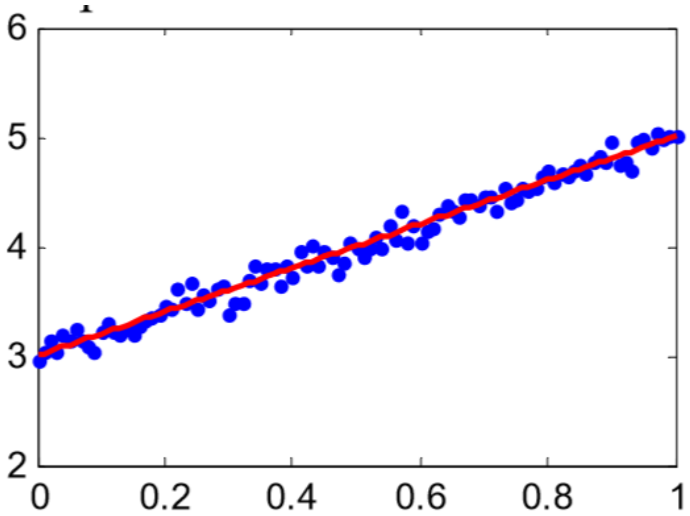
Algorithms developed to answer the above questions:

- **Linear regression**
- **Split and merge**
- **RANSAC**
- **Hough Transform**

Linear Regression

- If there is only one line to be fitted to the data points. Linear regression is applied to the points and line parameters are found.
- If line parameters are known (or estimated), the points belonging to the line can be found by thresholding their distances from the line given by those parameters.

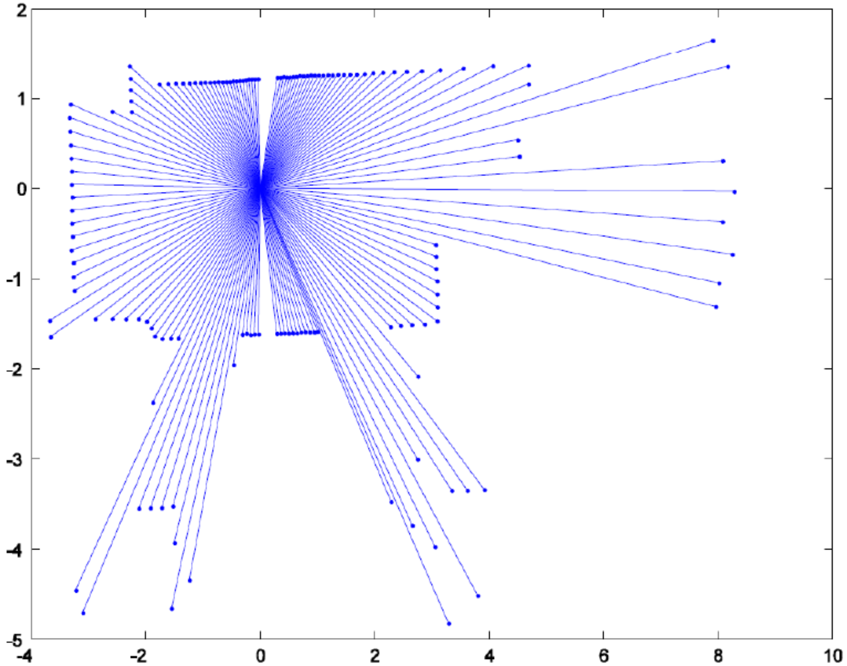




RMIT UNIVERSITY

Linear Regression

Scan point in polar form: (ρ_i, θ_i)



Linear Regression

- All measurements should satisfy the linear equation: $\rho_i \cos(\theta_i - \alpha) = r$
- But measurements are noisy, and points will be some distance from the line.

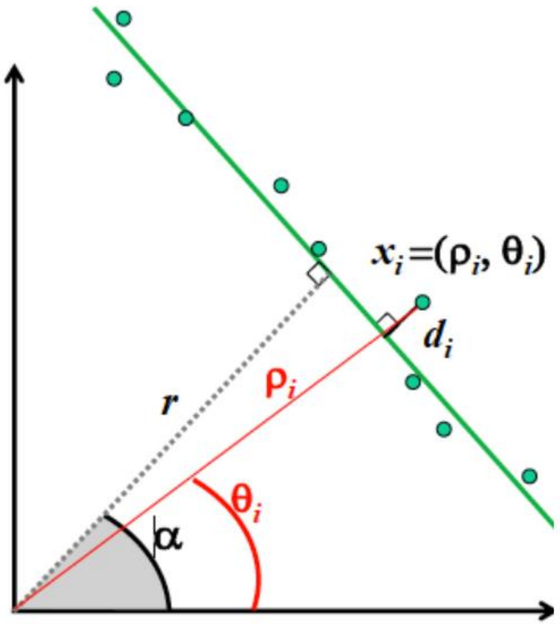
$$\rho_i \cos(\theta_i - \alpha) - r = d_i$$

- Our solution tries to minimize the error

$$S = \sum_i d_i^2 = \sum_i (\rho_i \cos(\theta_i - \alpha) - r)^2$$

- We do this by solving the system of equations

$$\frac{\partial S}{\partial \alpha} = 0 \qquad \frac{\partial S}{\partial r} = 0$$



Split and Merge

Algorithm

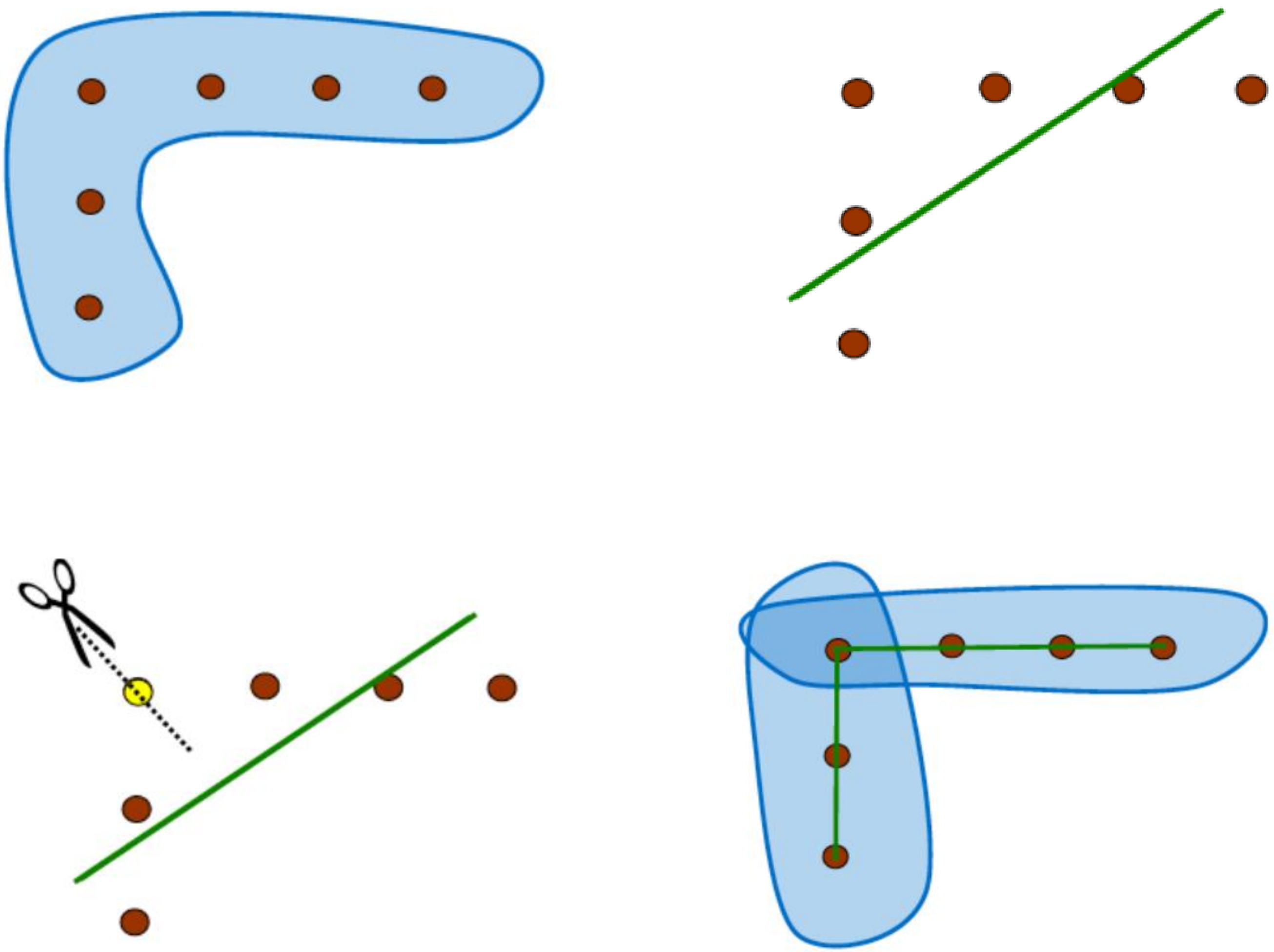
Split

- Obtain the line passing by the two extreme points
- Find the most distant point to the line
- If distance > threshold, split and repeat with the left and right point sets

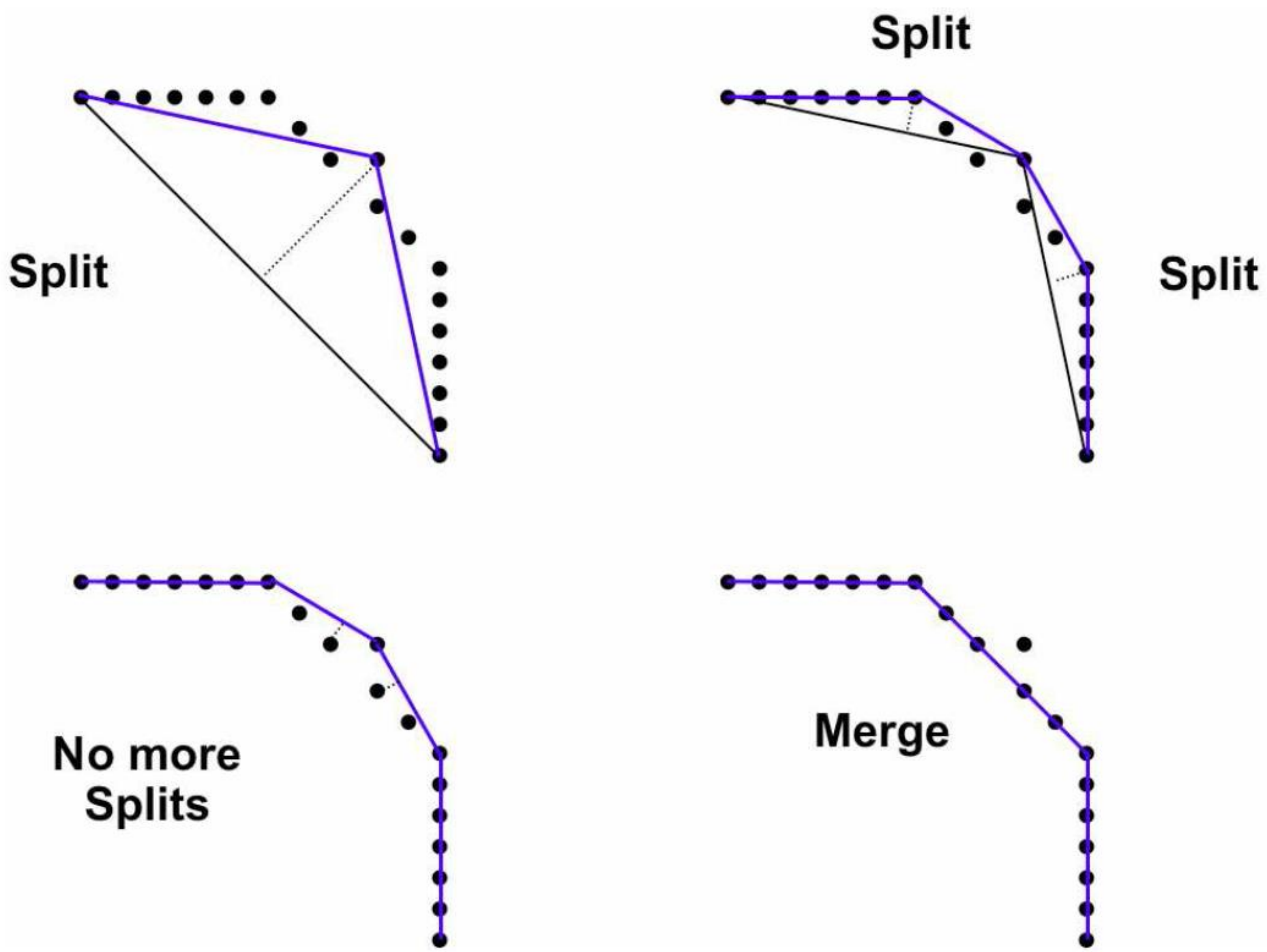
Merge

- If two consecutive segments are close/collinear enough, obtain the common line and find the most distant point
- If distance <= threshold, merge both segments

Split and Merge



Split and Merge

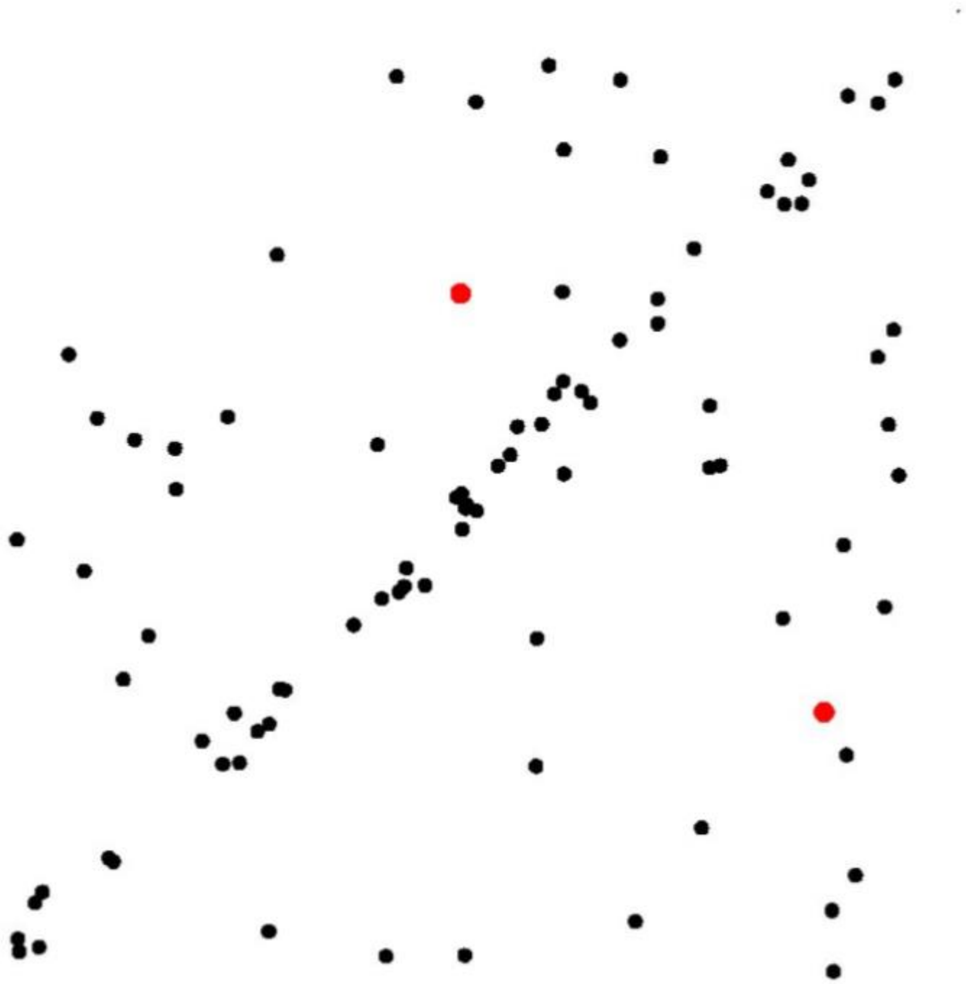


RANSAC - Random Sample Consensus

- It is a generic and robust fitting algorithm of models in the presence of outliers (points which do not satisfy a model).
- RANSAC is not restricted to line extraction and can be generally applied to any problem where the goal is to identify the inliers which satisfy a predefined mathematical model.
- Typical applications in robotics are line extraction from 2D range data (sonar or laser); plane extraction from 3D range data, and structure from motion.
- RANSAC is an iterative method and is non-deterministic in that the probability to find a line free of outliers increases as more iterations are used.
- Drawback: As a nondeterministic method, its results are different between runs.

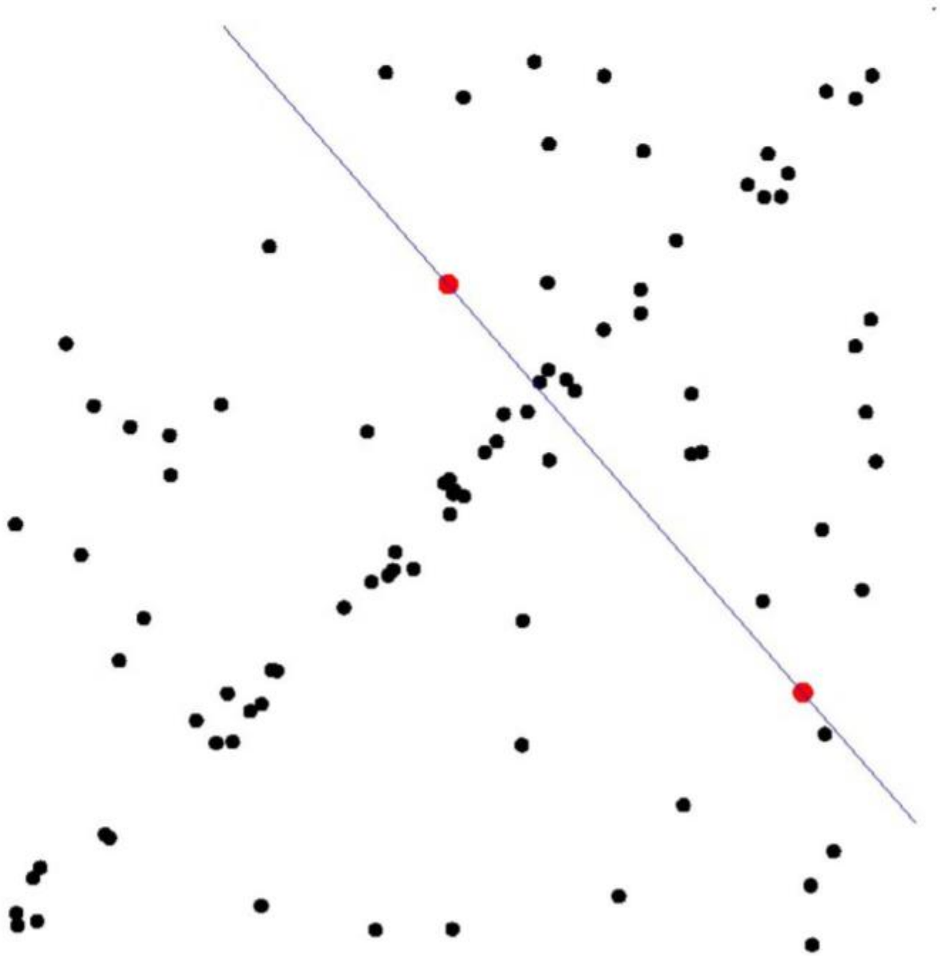
RANSAC - Random Sample Consensus

RANSAC - Random Sample Consensus



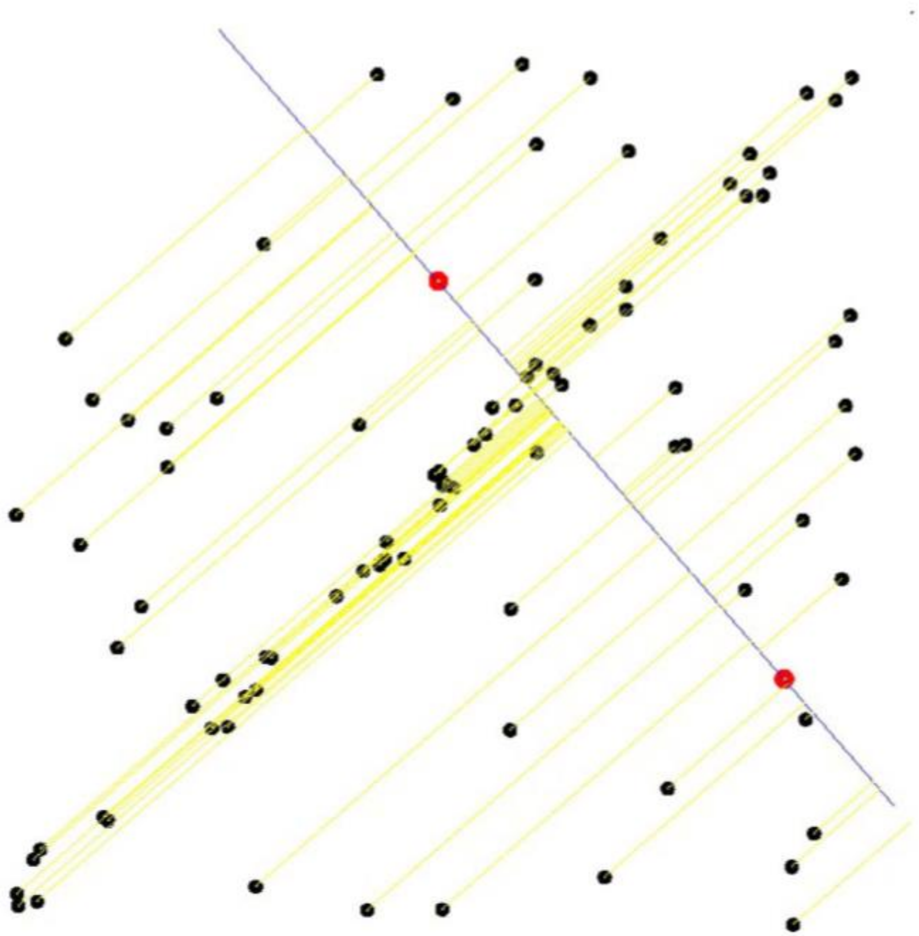
- **Select sample of 2 points at random**

RANSAC - Random Sample Consensus



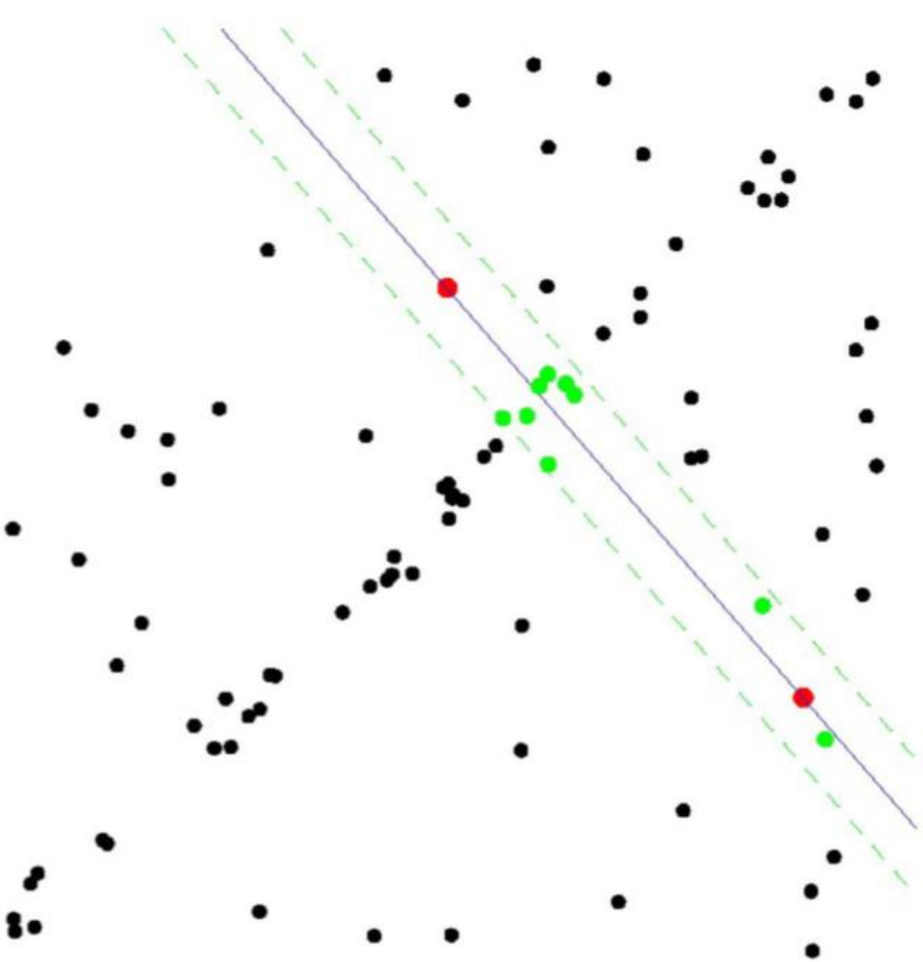
- Select sample of 2 points at random
- **Calculate model parameters that fit the data in the sample**

RANSAC - Random Sample Consensus



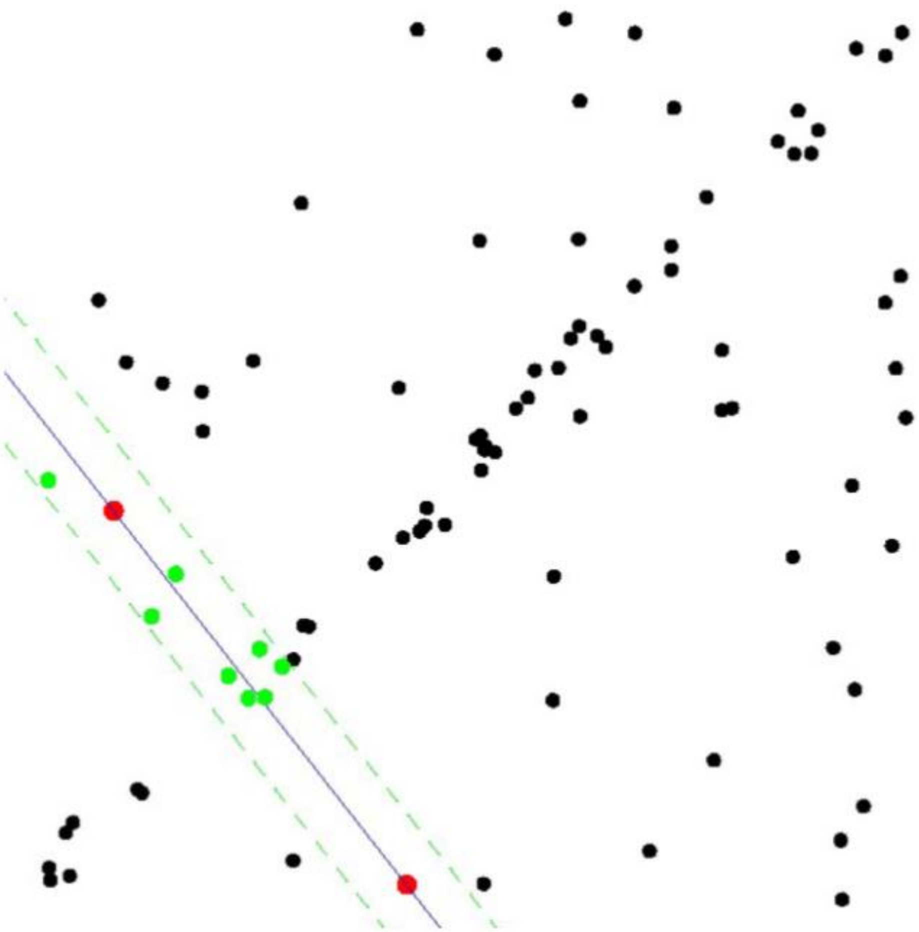
- Select sample of 2 points at random
- Calculate model parameters that fit the data in the sample
- **Calculate error function for each data point**

RANSAC - Random Sample Consensus



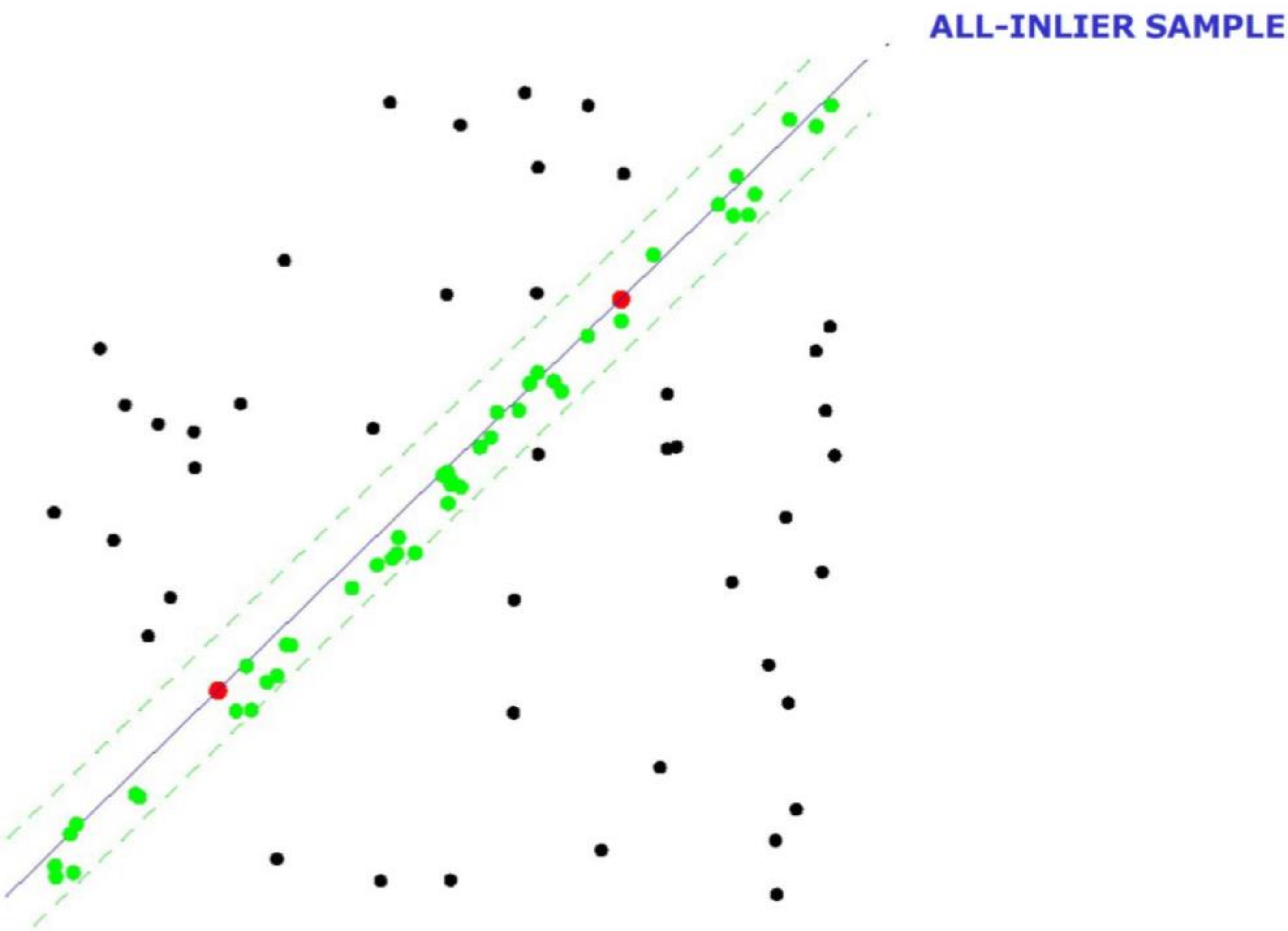
- Select sample of 2 points at random
- Calculate model parameters that fit the data in the sample
- Calculate error function for each data point
- **Select data that support current hypothesis**

RANSAC - Random Sample Consensus



- Select sample of 2 points at random
- Calculate model parameters that fit the data in the sample
- Calculate error function for each data point
- Select data that support current hypothesis
- **Repeat sampling**

RANSAC - Random Sample Consensus



RANSAC - Random Sample Consensus

- The best fit is the hypothesis supported by maximum number of points.
- Because we cannot know in advance if the observed set contains the maximum number of inliers, the ideal would be to check all possible combinations of 2 points in a dataset of N points.
- The number of combinations is given by $N(N-1)/2$, which makes it computationally unfeasible if N is too large. For example, in a laser scan of 360 points we would need to check all $360\times359/2= 64{,}620$ possibilities!
- Do we really need to check all possibilities or we can stop RANSAC after a number of iterations? The answer is that indeed we do not need to check all combinations but just a subset of them if we have a rough estimate of the percentage of inliers in our dataset.

RMIT UNIVERSITY

RANSAC - Random Sample Consensus

- What do need?
  - We need to make sure that among all randomly chosen pairs of points, at least one pair includes two inliers.
- ϵ: the fraction of inliers in the data. ϵ = number of inliers / N.
- ϵ also represents the probability of selecting an inlier.
- p: probability of selecting a pair of inliers.
- If we assume that the two points needed for estimating a line are selected independently,$p = \epsilon^2$ is the probability that both points are inliers.
- Probability that RANSAC does not select two points that are both inliers = $1 - p = 1 - \epsilon^2$.
- k: the number of pairs of points chosen (RANSAC iterations).
- Probability that RANSAC never selects two points that are both inliers $= (1-p)^k = (1 - \epsilon^2)^k$.
- p_s = Probability of success (at least one good pair) $= 1 - (1 - \epsilon^2)^k$.

RANSAC - Random Sample Consensus

$$p_s = 1 - (1 - \epsilon^2)^k \quad \Rightarrow \quad 1 - p_s = (1 - \epsilon^2)^k$$

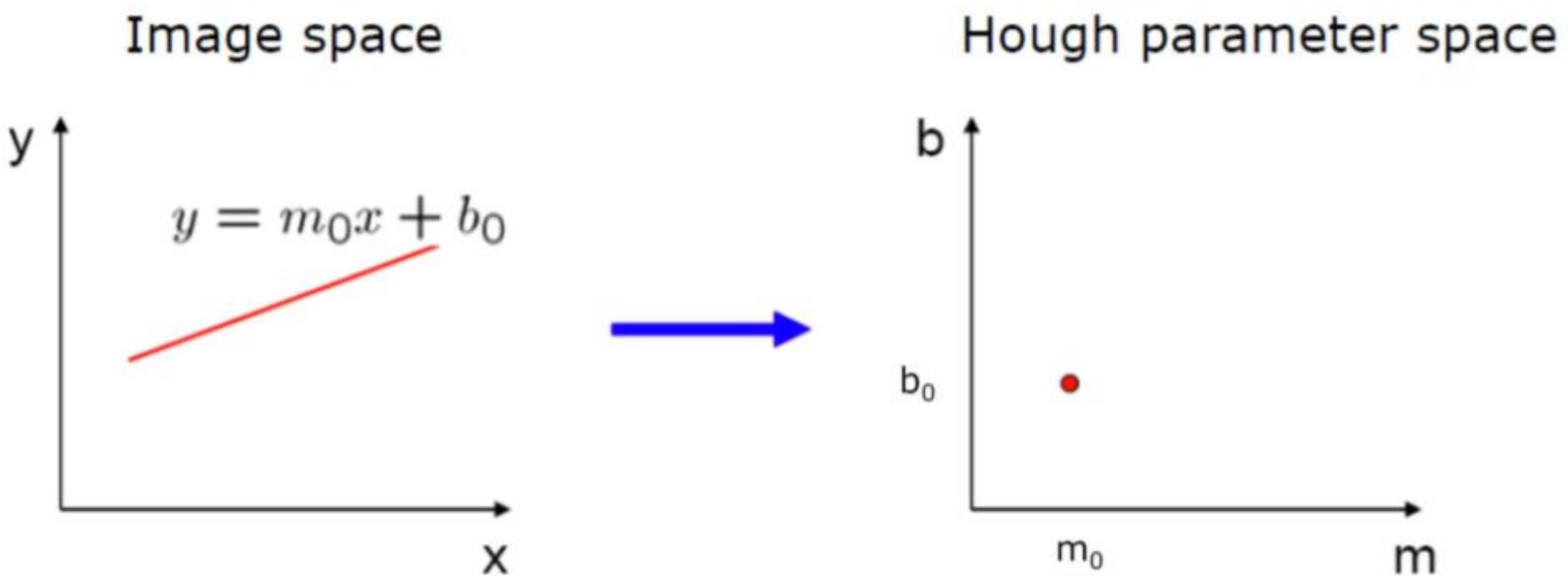
$$\Rightarrow \quad \log(1 - p_s) = k \times \log(1 - \epsilon^2)$$

$$\boxed{k = \frac{\log(1 - p_s)}{\log(1 - \epsilon^2)}}$$

- Note that k does not depend on the number of points N.
- Some values:
  - $p_s = 0.99, \epsilon = 0.5 \rightarrow k = 16$
  - $p_s = 0.99, \epsilon = 0.2 \rightarrow k = 113$
  - $p_s = 0.99, \epsilon = 0.1 \rightarrow k = 458$
- Compare the above values with 64,620 which we got for an exhaustive search.

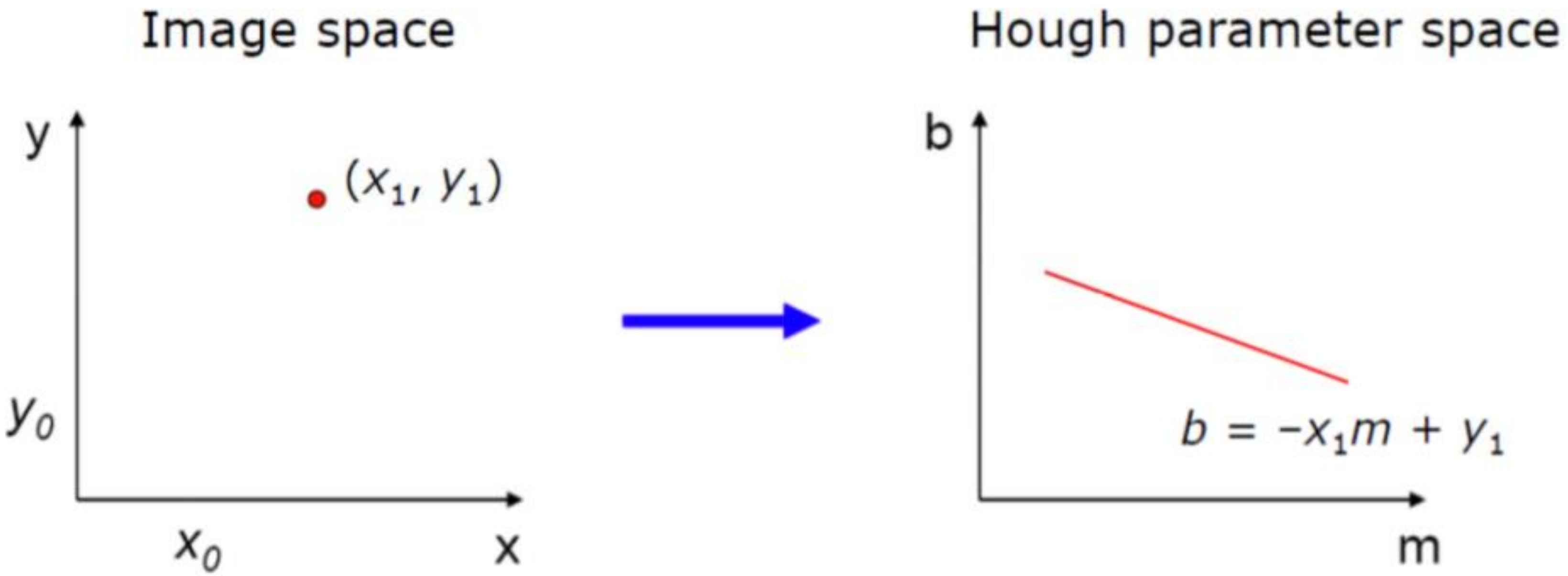
Hough Transform - Transformation between two domains

- A line in the image corresponds to a point in the Hough space



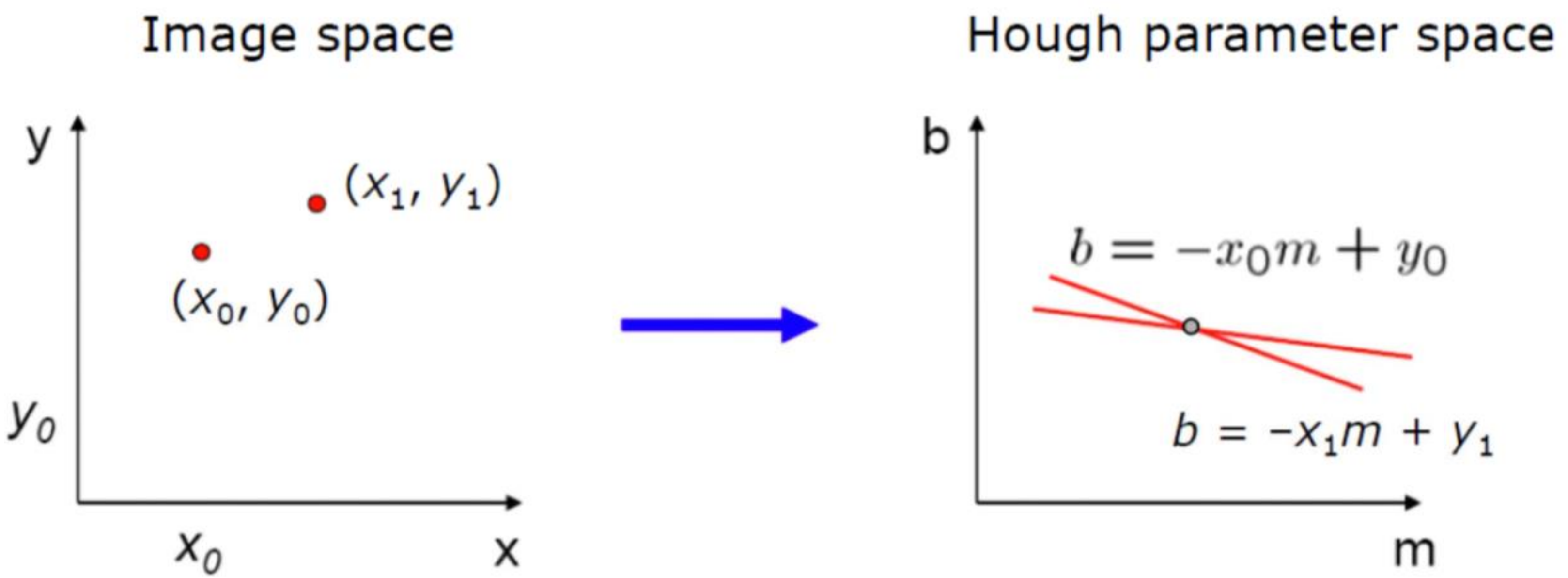
Hough Transform - Transformation between two domains

- A point in the image corresponds to a line in the Hough space



Idea in Hough Transform

- Where is the line that contains both (x0, y0) and (x1, y1)?



- It is the intersection of the lines:
  - $b = -x_0 m + y_0$
  - $b = -x_1 m + y_1$

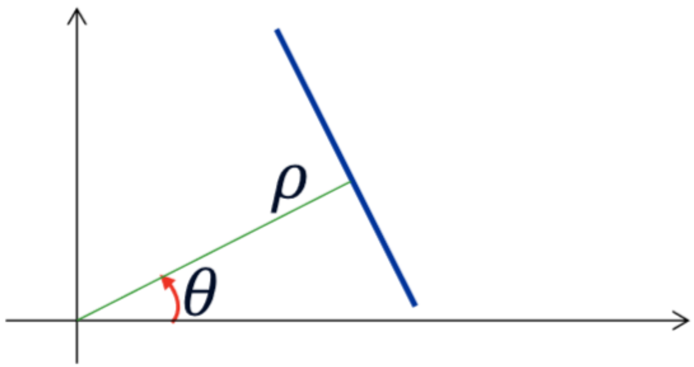
Problem with conventional presentation

Problems with the (m,b) space:

- Unbounded parameter domain.
- Vertical lines requires infinite m.

Using Polar Representation

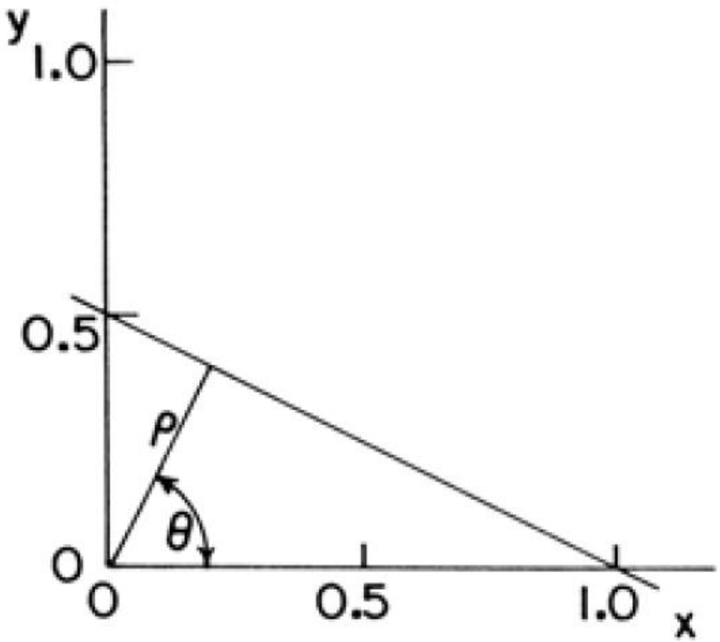
- Alternative: polar representation:
  - $x \cos\theta + y \sin\theta = \rho$
  - $-\pi < \theta \leq \pi$
  - $\rho \geq 0$



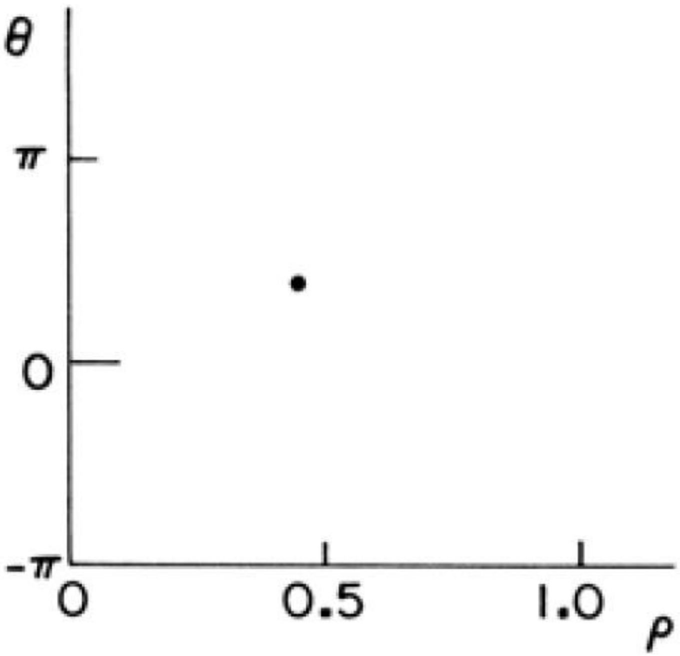
Hough Transform

- The Hough transform can be used as a means of edge linking.
- The Hough transform involves the transformation of a line in Cartesian coordinate space to a point in polar coordinate space.

Hough Transform

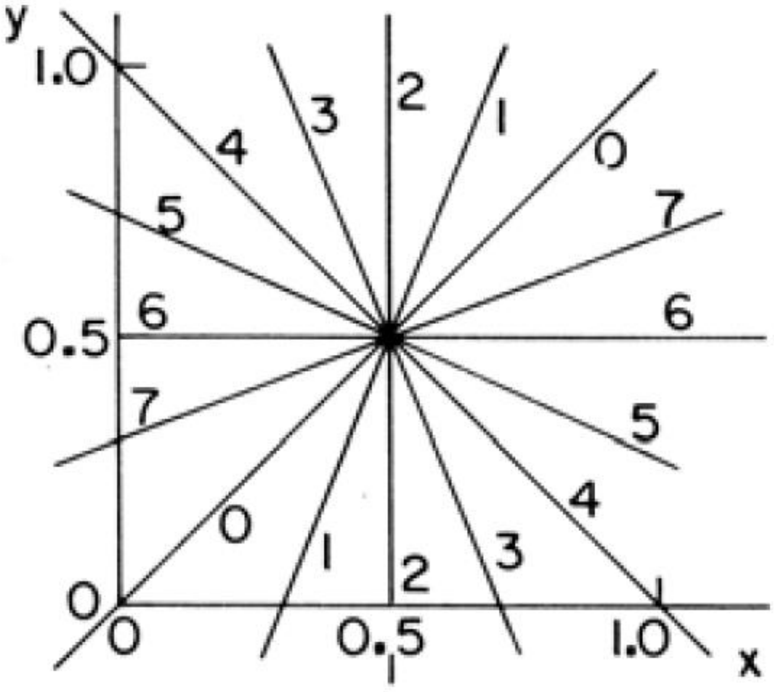


(a) Parametric line

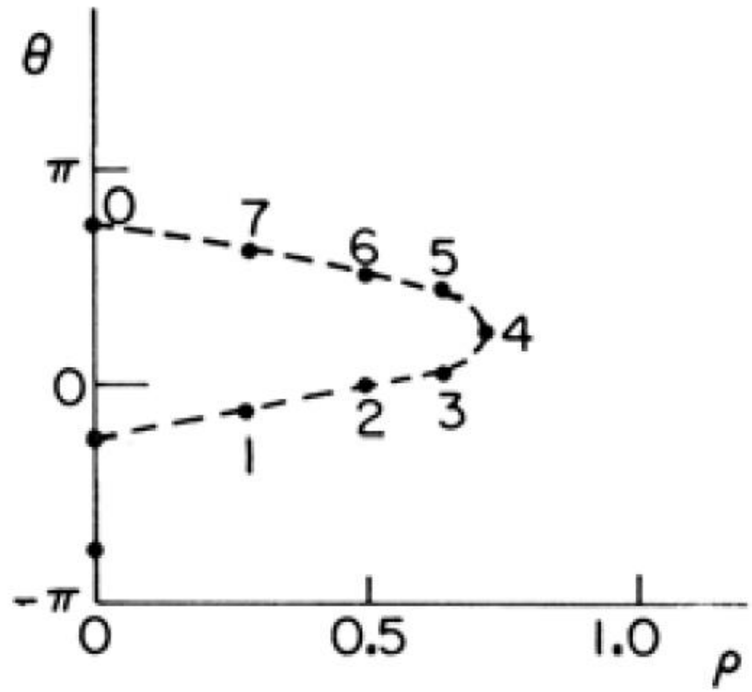


(b) Hough transform of (a)

Hough Transform

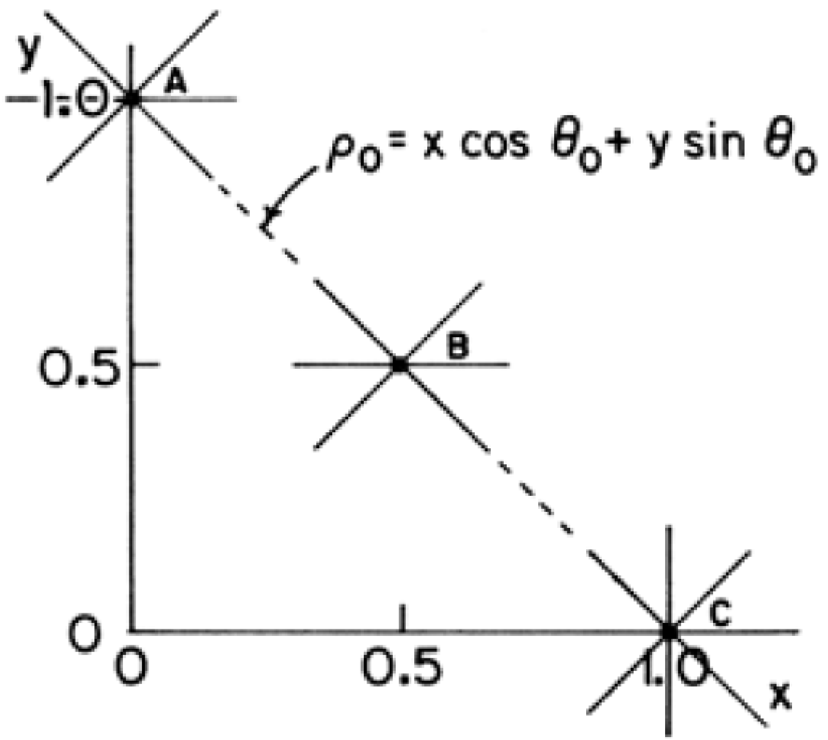


(c) Family of lines, common point

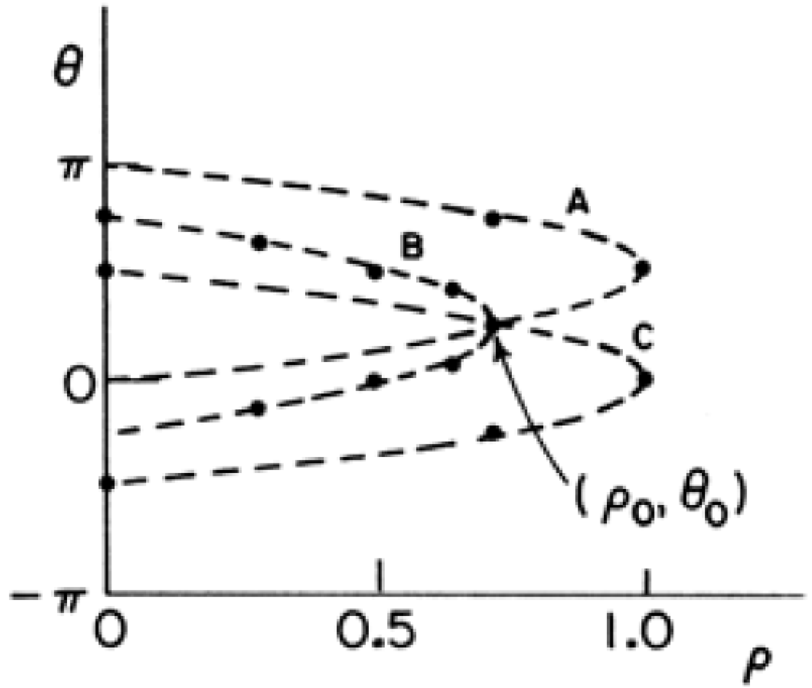


(d) Hough transform of (c)

Hough Transform



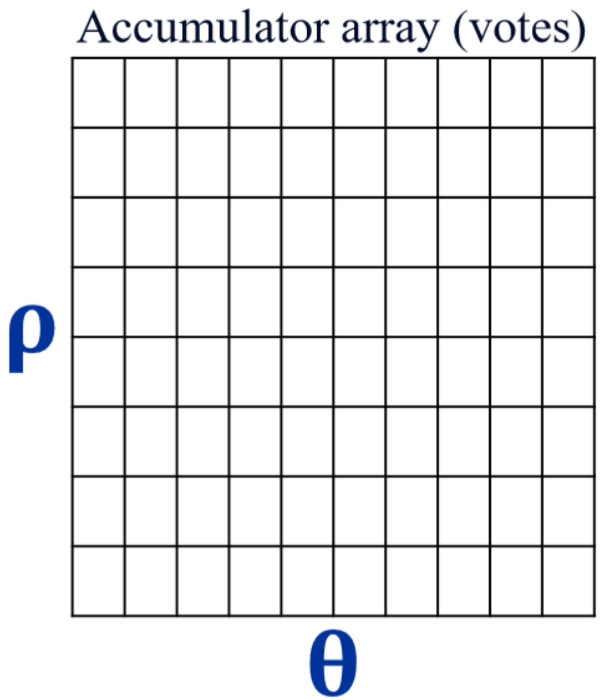
(e) Colinear points



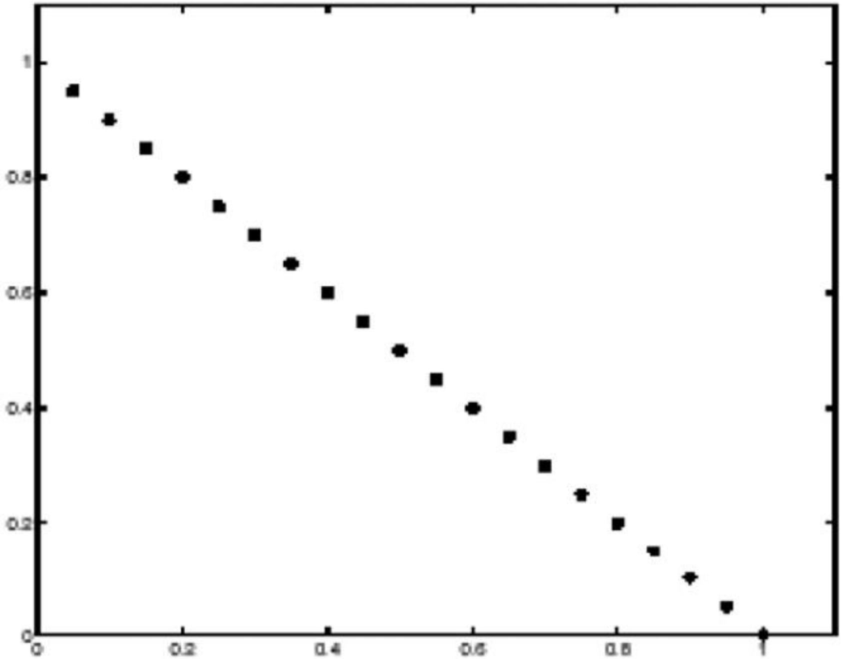
(f) Hough transform of (e)

Algorithm

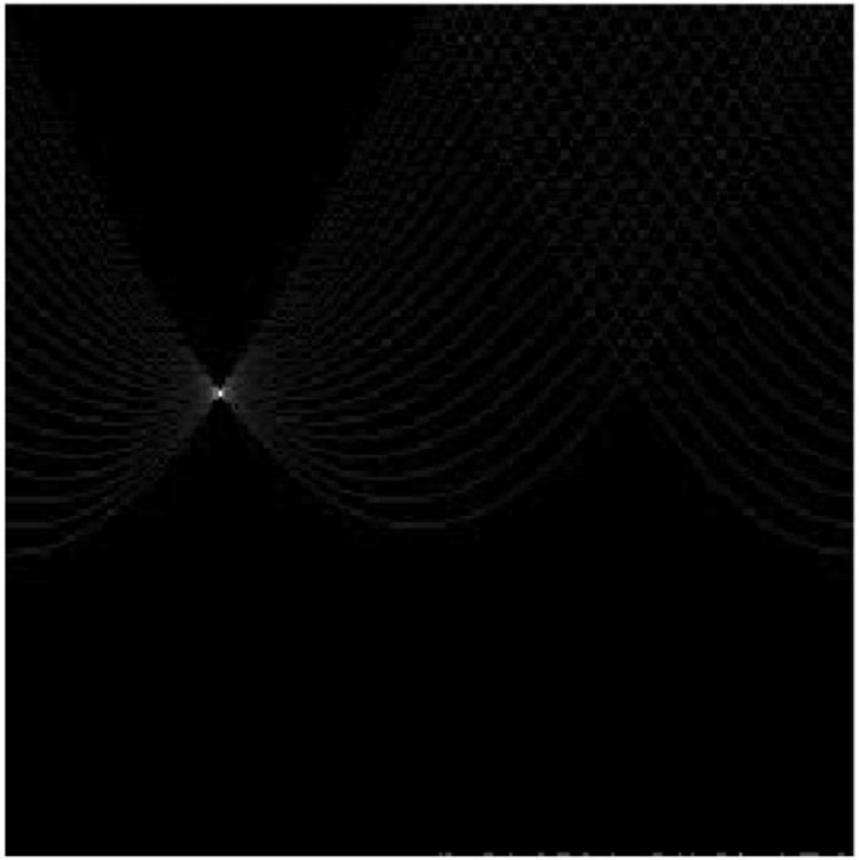
- Initialize accumulator H to all zeros.
- For each edge point (x, y) in the image
  - For $\theta = 0$ to 180
    - $\rho = x \cos\theta + y \sin\theta$
    - $H(\theta, \rho) = H(\theta, \rho) + 1$
  - end
- end
- Find the values of (θ, ρ) where $H(\theta, \rho)$ is a local maximum.
- The detected line in the image is given by $\rho = x \cos\theta + y \sin\theta$.



Example



features



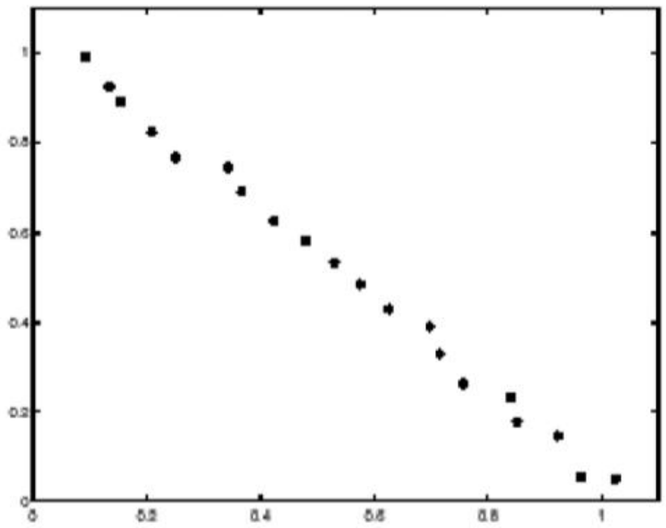
votes

Example

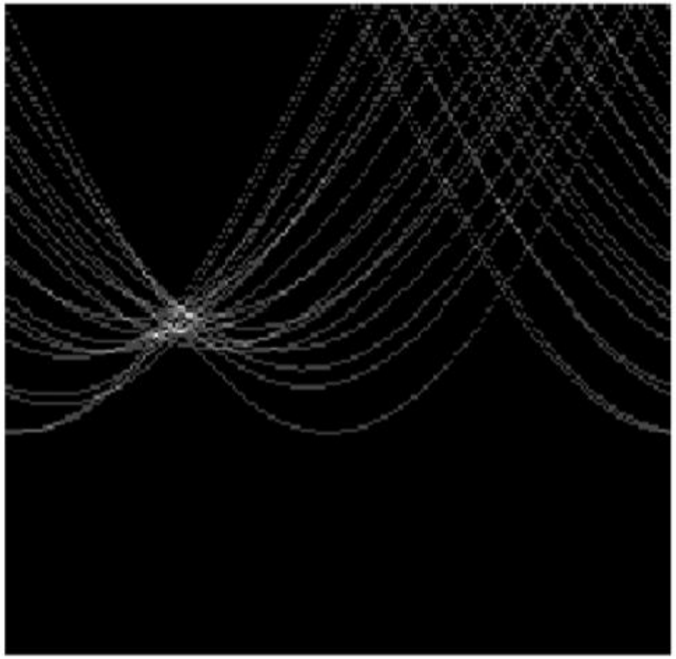
Square

Effect of Noise

- Effect of noise:



features



votes

- Peak gets fuzzy and hard to locate.

Thank you for your attendance :D

RMIT UNIVERSITY

Reference

- *MATHWORKS official tutorial.*
- *Lecture slides from RMIT Melbourne Autonomous System course, delivered by Prof Reza Hoseinnezhad.*
- *Introduction to Autonomous Mobile Robots by Roland Siegwart and Ilah R. Nourbakhsh.*
- *Chris Clark – Introduction to Robotic Navigation*

Copyright Claim

RMIT UNIVERSITY