上海科技大学
ShanghaiTech University

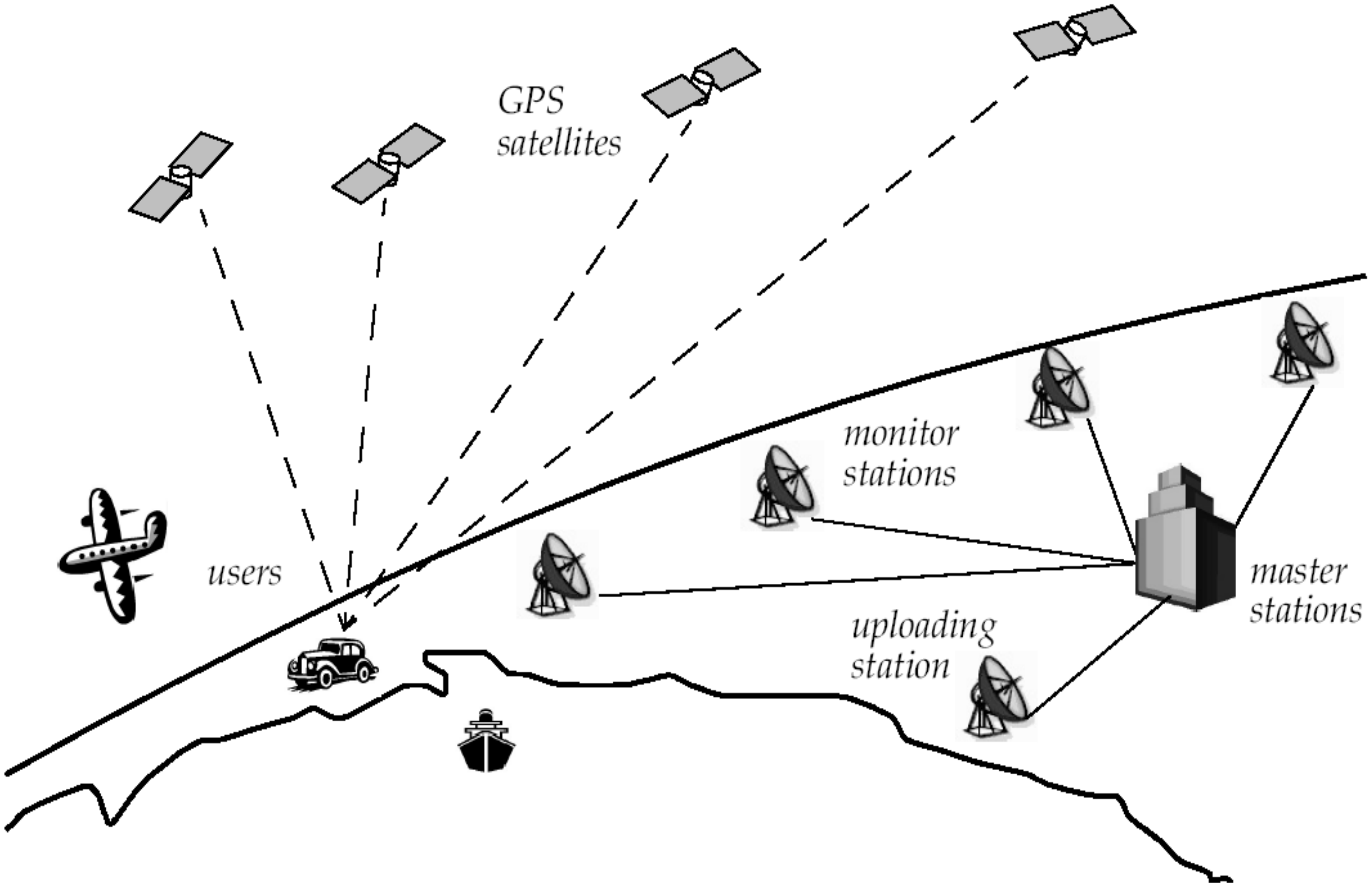
CS283: **Robotics Fall 2016: Perception I**

Sören Schwertfeger / 师泽仁

ShanghaiTech University

REVIEW SENSORS

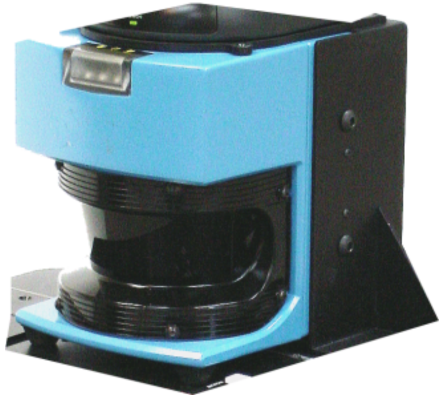
Global Positioning System (GPS) (2)



Range sensors

- Sonar --->

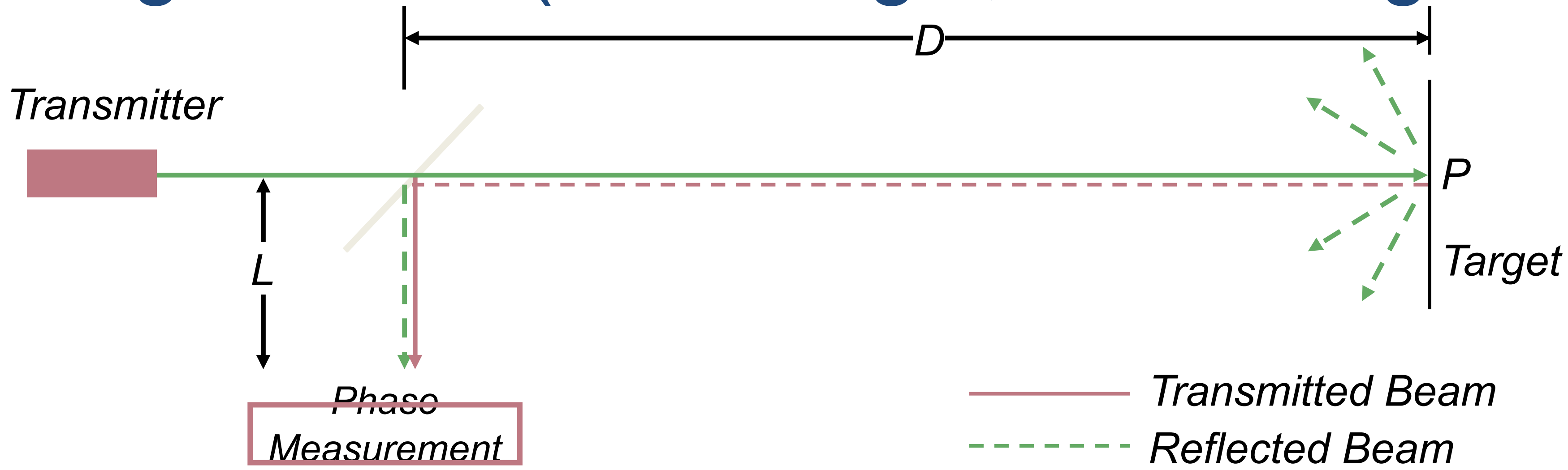
- Laser range finder --->



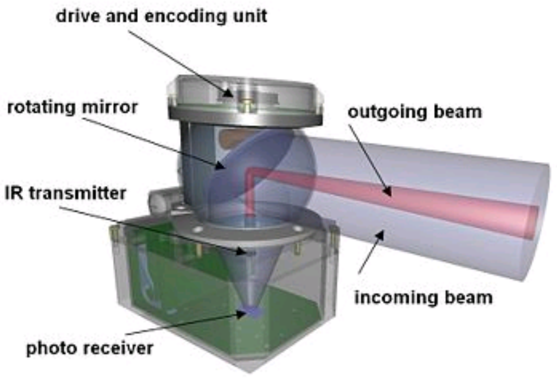
- Time of Flight Camera ------------------------------------->

- Structured light ---------->

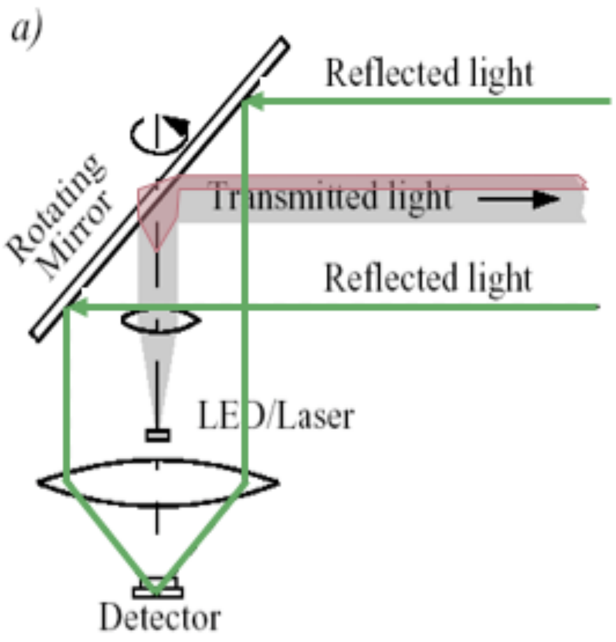
Laser Range Sensor (time of flight, electromagnetic) (1)



- Transmitted and received beams coaxial
- Transmitter illuminates a target with a collimated laser beam
- Received detects the time needed for round-trip
- A mechanical mechanism with a mirror sweeps
  - 2D or 3D measurement



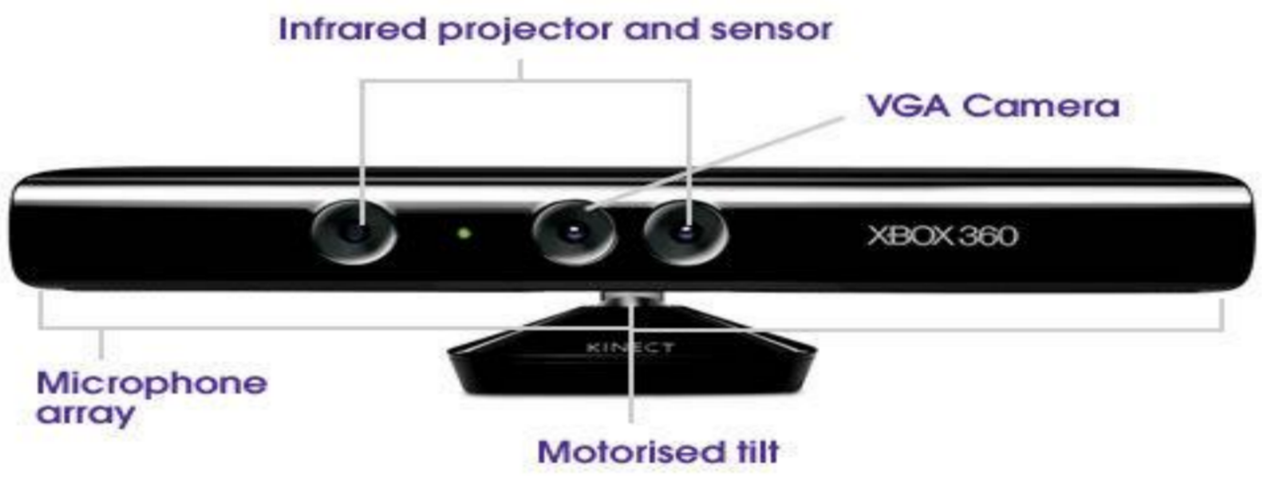

SICK


a)
Rotating Mirror
Reflected light
Transmitted light
Reflected light
LED/Laser
Detector

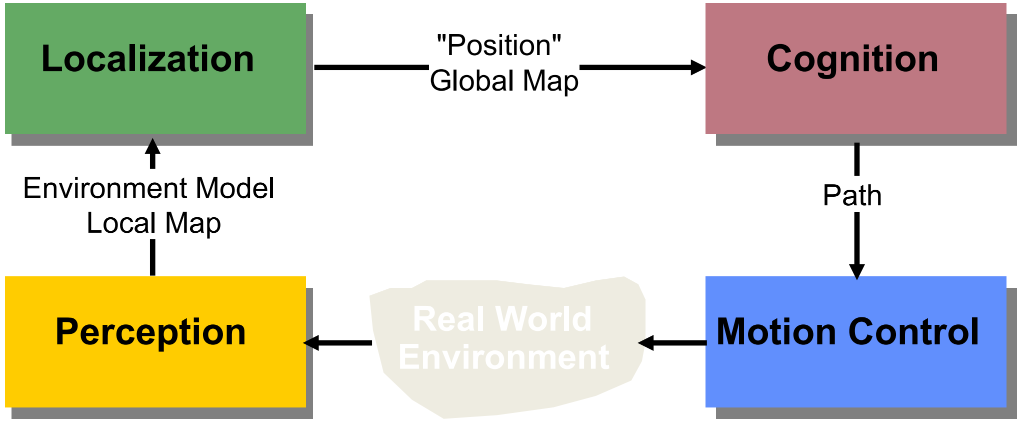

SICK

PrimeSense Cameras

- Devices: Microsoft Kinect and Asus Xtion
- Developed by Israeli company PrimeSense in 2010



- Components:
  - IR camera (640 x 480 pixel)
  - IR Laser projector
  - RGB camera (640 x 480 or 1280 x 1024)
  - Field of View (FoV):
    - 57.5 degrees horizontally,
    - 43.5 degrees vertically



PERCEPTION

Sensors
Line extraction from laser scans
Uncertainties
Vision

Sildes from Roland Siegwart and Davide Scaramuzza, ETH Zurich

LINE EXTRACTION

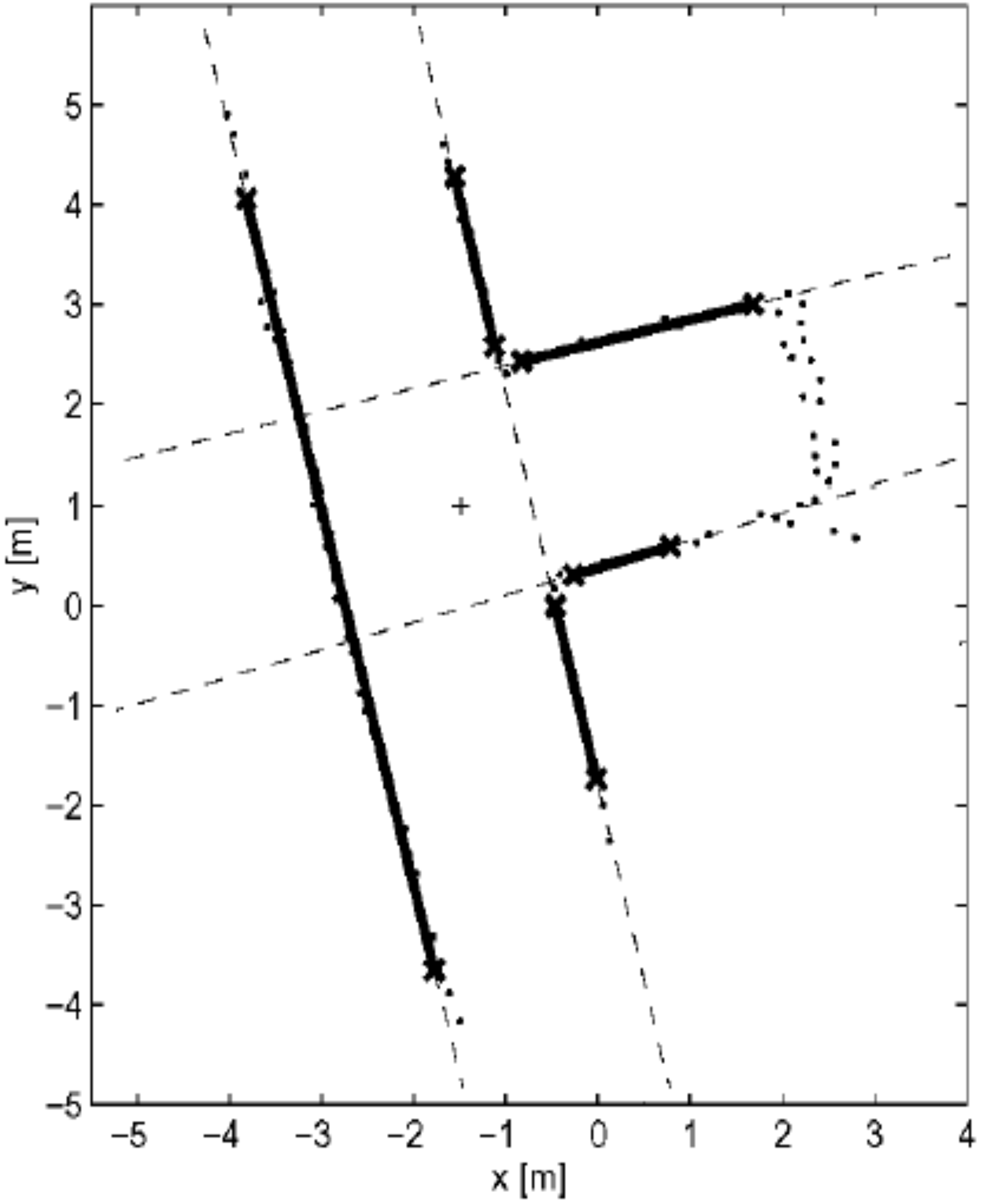
Split and merge
Linear regression
RANSAC
Hough-Transform

Line Extraction: Motivation

- Laser Range Scan
  - Example: 360 deg – black points
  - Example: dashed lines: desired line extractions

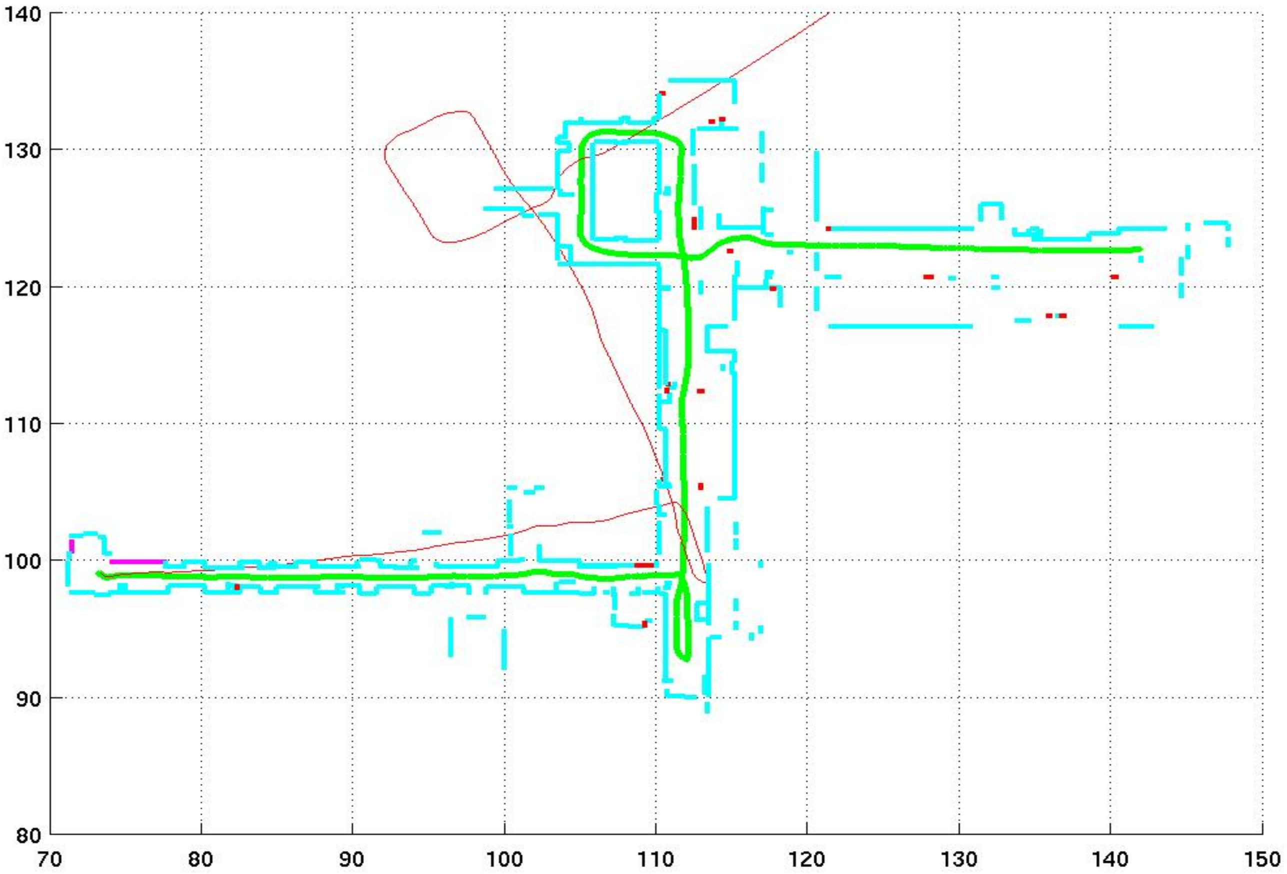
- Use detected lines for:
  - Scan registration (find out transform between frames of two consecutive LRF scans – change due to robot motion)

  OR
  - Mapping using line representation



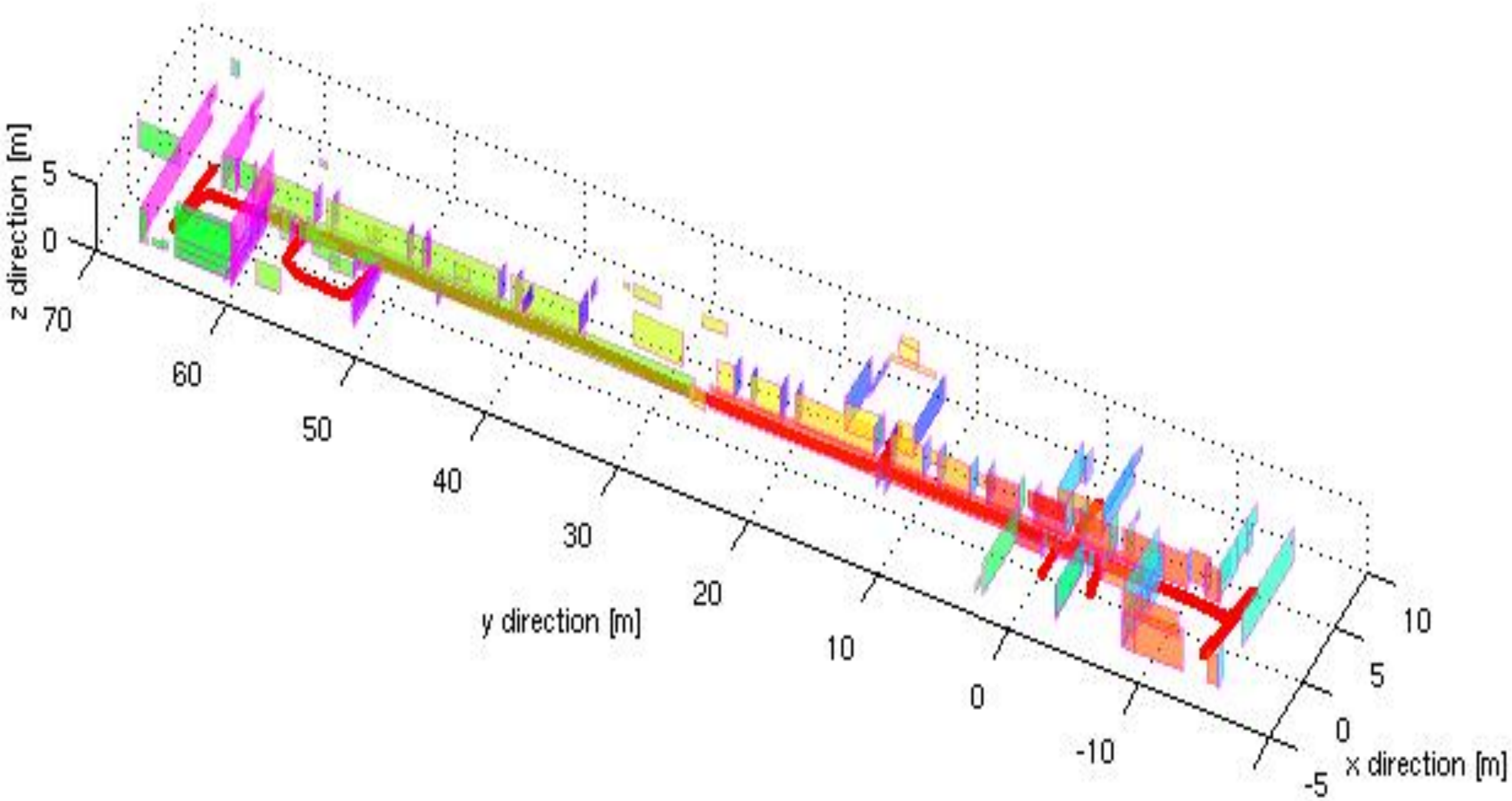
Line Extraction: Motivation

- Map of hallway built using line segments



Line Extraction: Motivation

- Map of the hallway built using orthogonal planes constructed from line segments

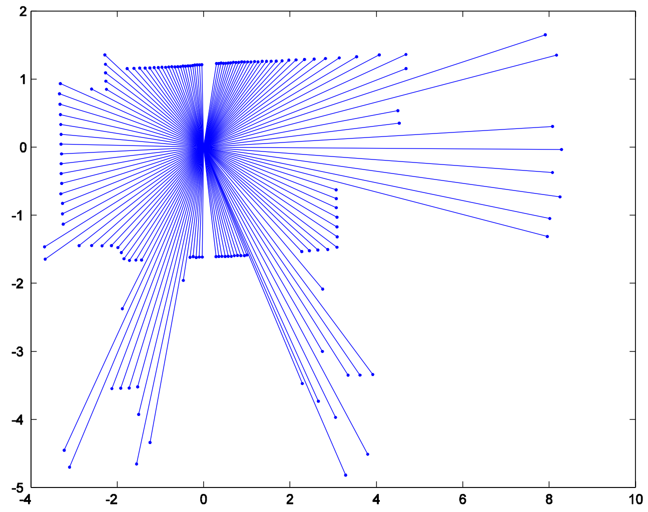


Line Extraction: Motivation

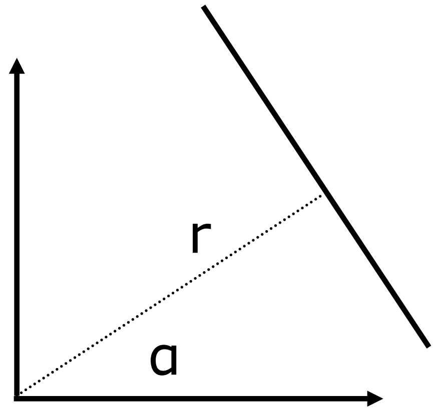
- Why laser scanner:
  - Dense and accurate range measurements
  - High sampling rate, high angular resolution
  - Good range distance and resolution.

- Why line segment:
  - The simplest geometric primitive
  - Compact, requires less storage
  - Provides rich and accurate information
  - Represents most office-like environment.

Line Extraction: The Problem



- Scan point in polar form: (ρ_i, θ_i)
- Assumptions:
  - Gaussian noise
  - Negligible angular uncertainty
- Line model in polar form:
  - $x \cos \alpha + y \sin \alpha = r$
  - $-\pi < \alpha <= \pi$
  - $r >= 0$



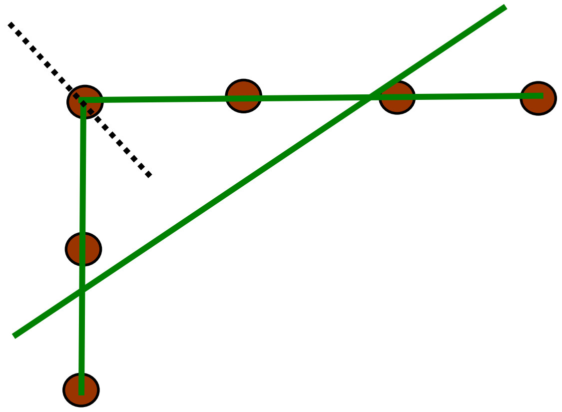
Line Extraction: The Problem (2)

- Three main problems:
  - How many lines ?
  - Which points belong to which line ?
    - This problem is called SEGMENTATION
  - Given points that belong to a line, how to estimate the line parameters ?
    - This problem is called LINE FITTING

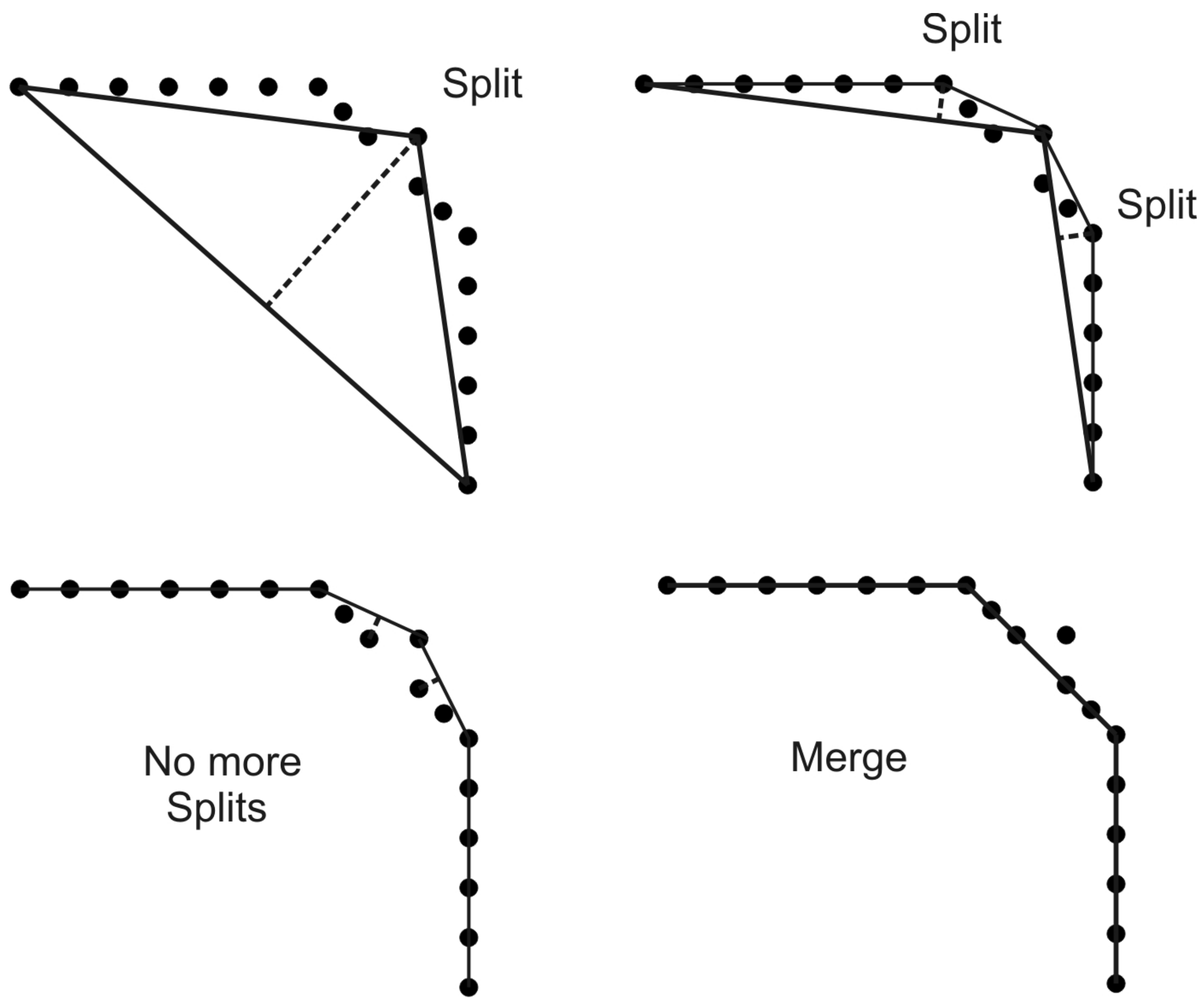
- The Algorithms we will see:
  1. Split and merge
  2. Linear regression
  3. RANSAC
  4. Hough-Transform

Algorithm 1: Split-and-Merge (standard)

- The most popular algorithm which is originated from computer vision.
- A recursive procedure of fitting and splitting.
- A slightly different version, called Iterative-End-Point-Fit, simply connects the end points for line fitting.



Algorithm 1: Split-and-Merge (Iterative-End-Point-Fit)

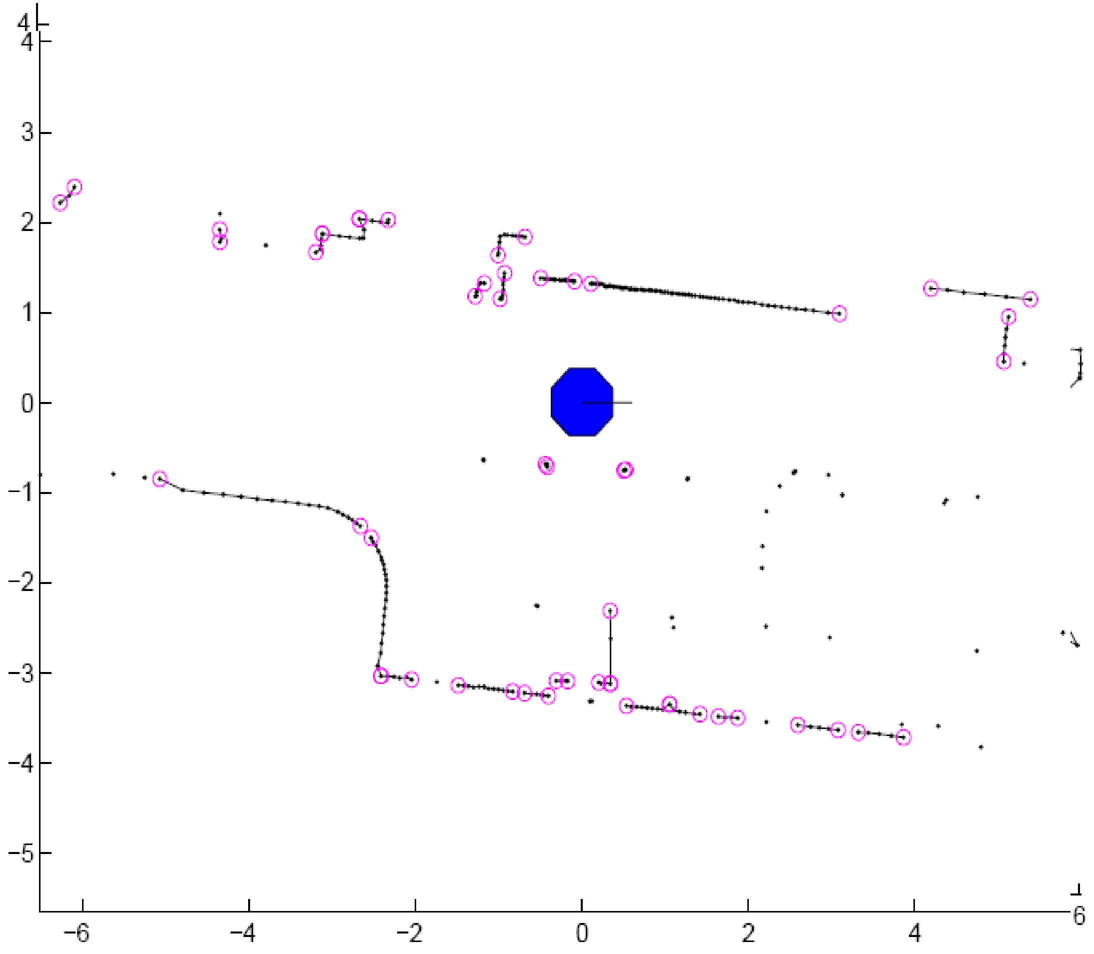


Algorithm 1: Split-and-Merge

Algorithm 1: *Split-and-Merge*

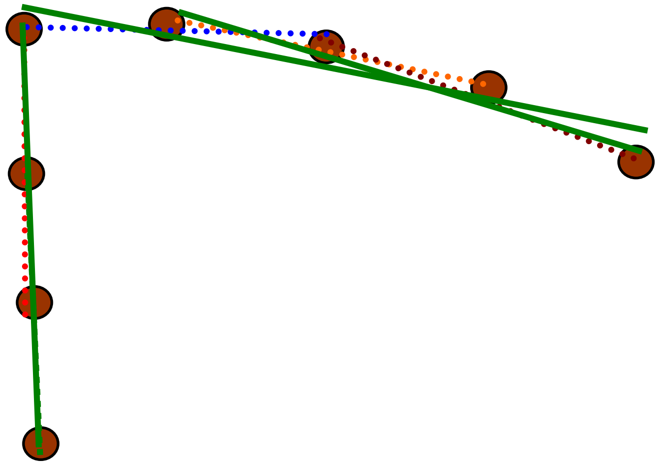
1. Initial: set s_1 consists of N points. Put s_1 in a list L
2. Fit a line to the next set s_i in L
3. Detect point P with maximum distance d_P to the line
4. If d_P is less than a threshold, continue (go to step 2)
5. Otherwise, split s_i at P into s_{i1} and s_{i2}, replace s_i in L by s_{i1} and s_{i2}, continue (go to 2)
6. When all sets (segments) in L have been checked, merge collinear segments.

Algorithm 1: Split-and-Merge: Example application



Algorithm 2: Line-Regression

- Uses a “sliding window” of size *Nf*
- The points within each “sliding window” are fitted by a segment
- Then adjacent segments are merged if their line parameters are close



$Nf = 3$

Algorithm 2: Line-Regression

Algorithm 2: *Line-Regression*

1. Initialize sliding window size N_f
2. Fit a line to every N_f consecutive points (a window)
3. Compute a line fidelity array, each is the sum of Mahalanobis distances between every three adjacent windows
4. Construct line segments by scanning the fidelity array for consecutive elements having values less than a threshold, using an AHC algorithm
5. Merge overlapped line segments and recompute line parameters for each segment

ADMIN

PROJECTS

Projects

- Project selection due Fri, Oct 21, 22:00 in piazza. Failure to meet the due date: 10% off of project 1 score!
- Three students per project
- Projects will be split into two distinct parts!
- The group will have to write a short project proposal (LaTeX, 2 pages, English)
- Other steps:
  - Coding/ building
  - Experiments
  - Evaluation
  - Documentation!
  - Write short project report
  - Create a webpage
  - Make videos
- After the 2^{nd} part of the project: Presentation/ Demo

Project Ideas:

- 3D mapping robot
- "RoboCup Project Smartphone Robot"
- Schunk arm programming
- Omni-directional robot
- Car mapping (put sensors on my car and collect data)
- Full Speed Jackal
- Work with the FARO data (e.g. registration, extract planes, "terrain classification", path planning for ground robots)
- Some aerial project (I’m not a big fan)
- …

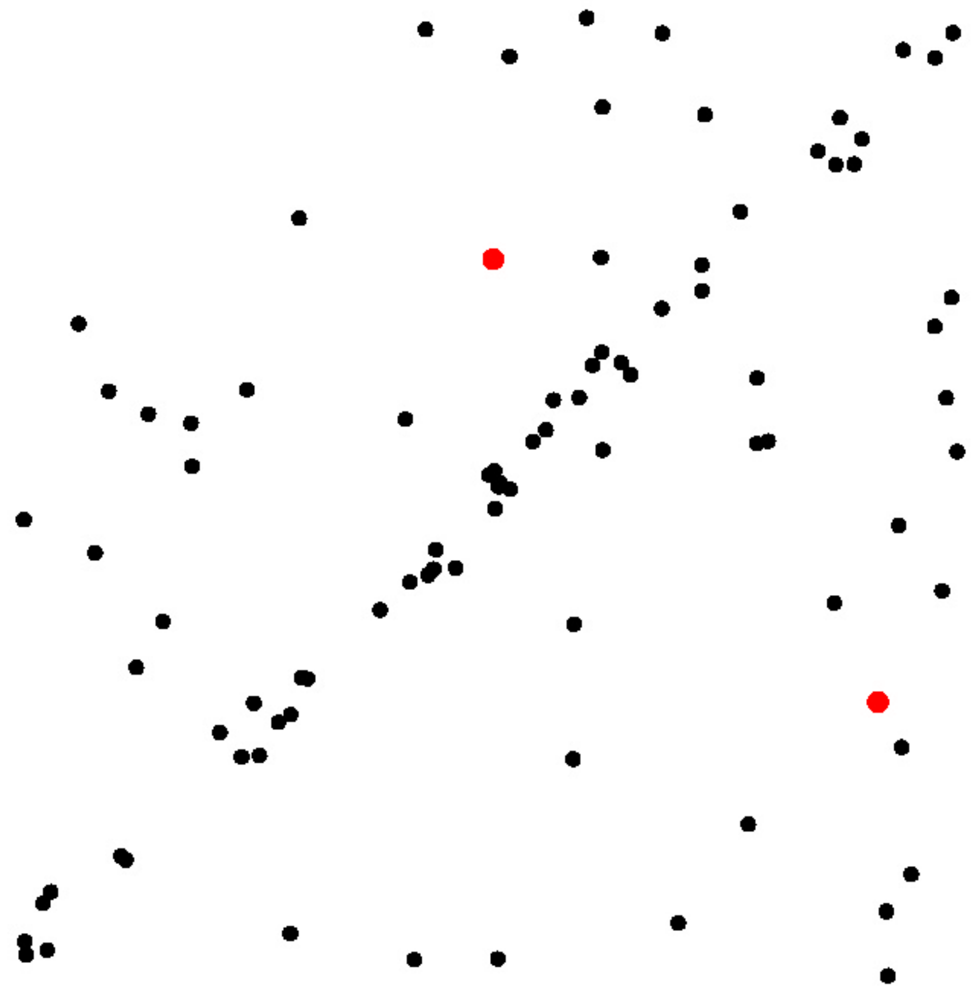
Paper presentation Friday….

Algorithm 3: RANSAC

- Acronym of 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 from laser data but it 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: A nondeterministic method, results are different between runs.

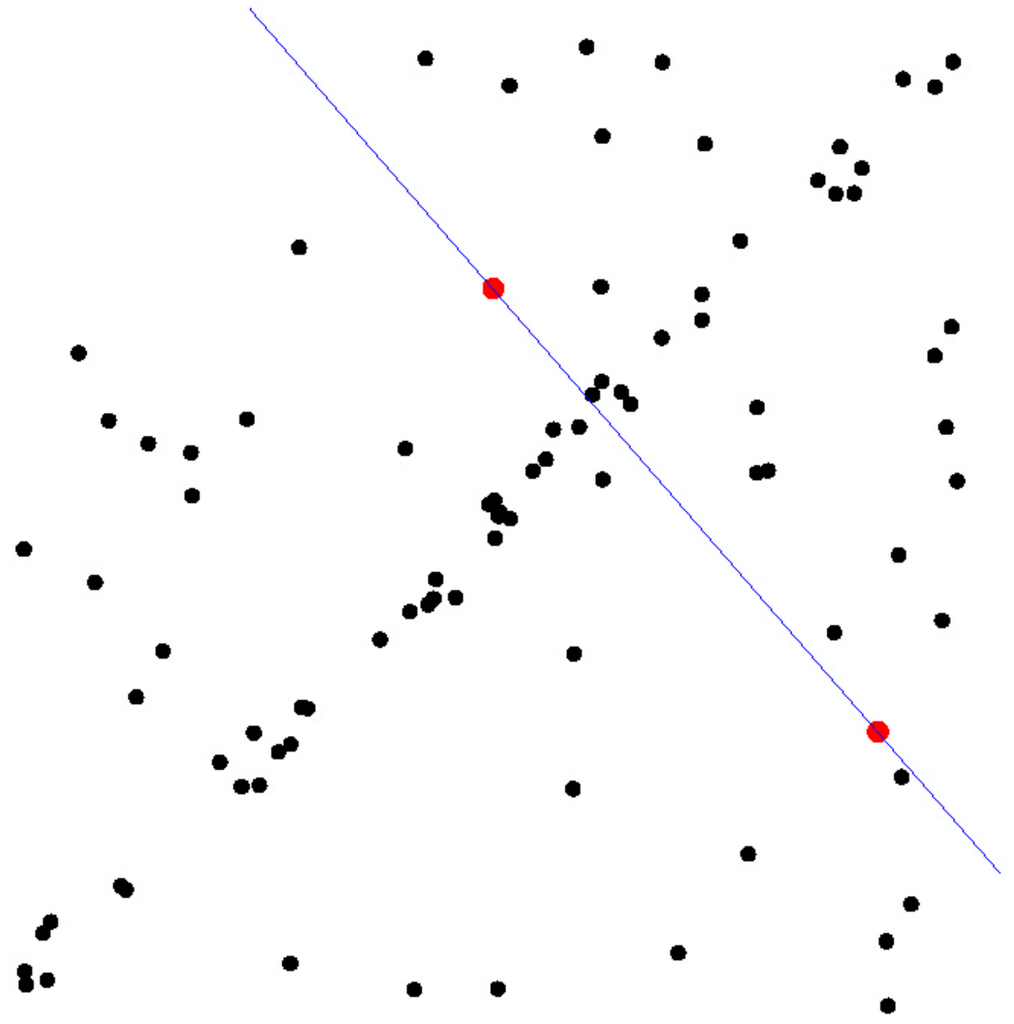
Algorithm 3: RANSAC

Algorithm 3: RANSAC



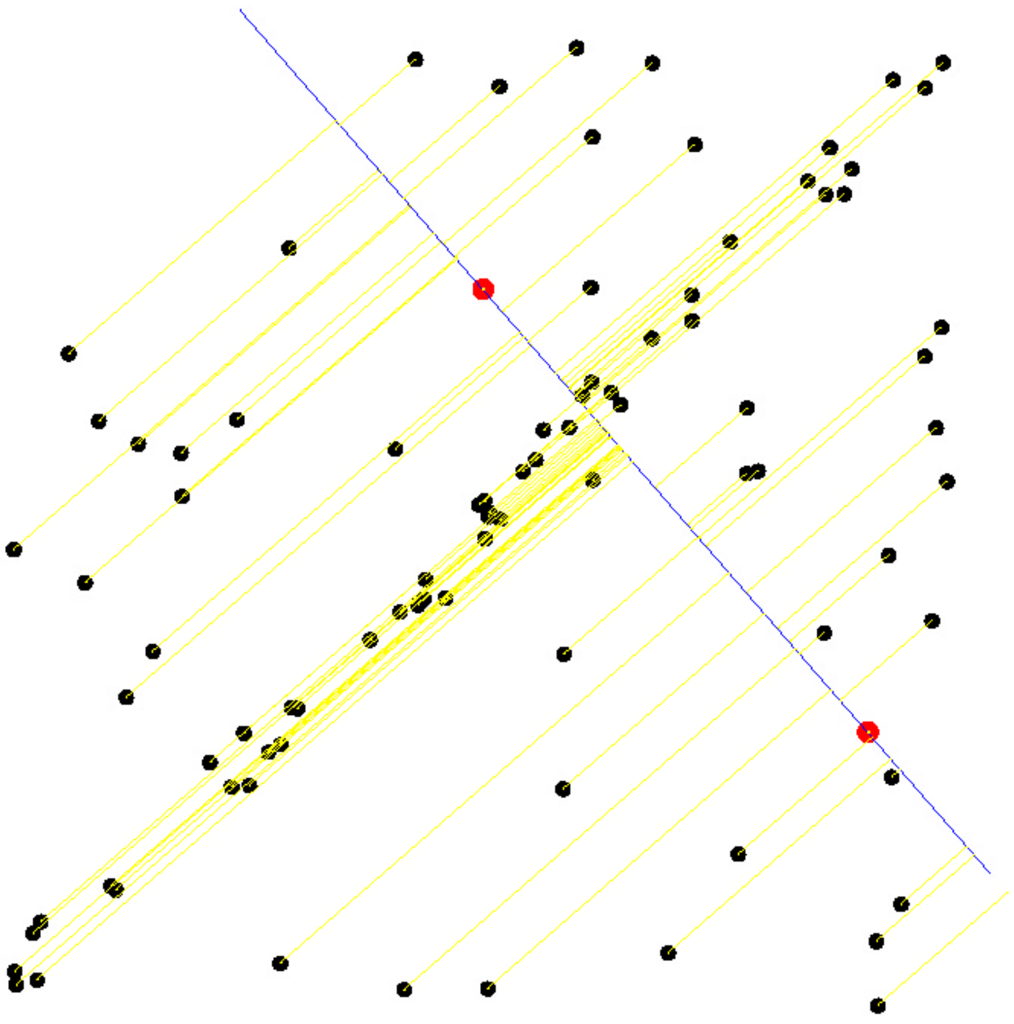
- **Select sample of 2 points at random**

Algorithm 3: RANSAC



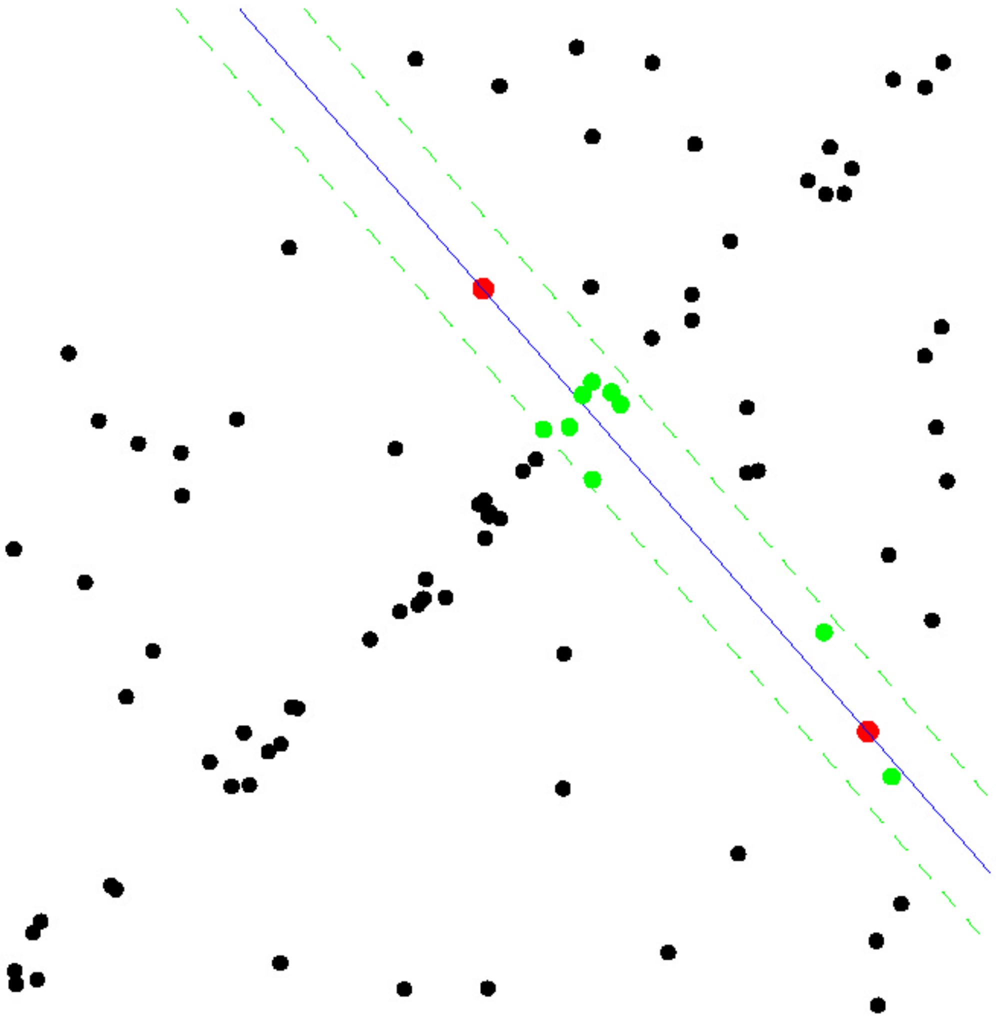
- Select sample of 2 points at random
- **Calculate model parameters that fit the data in the sample**

RANSAC



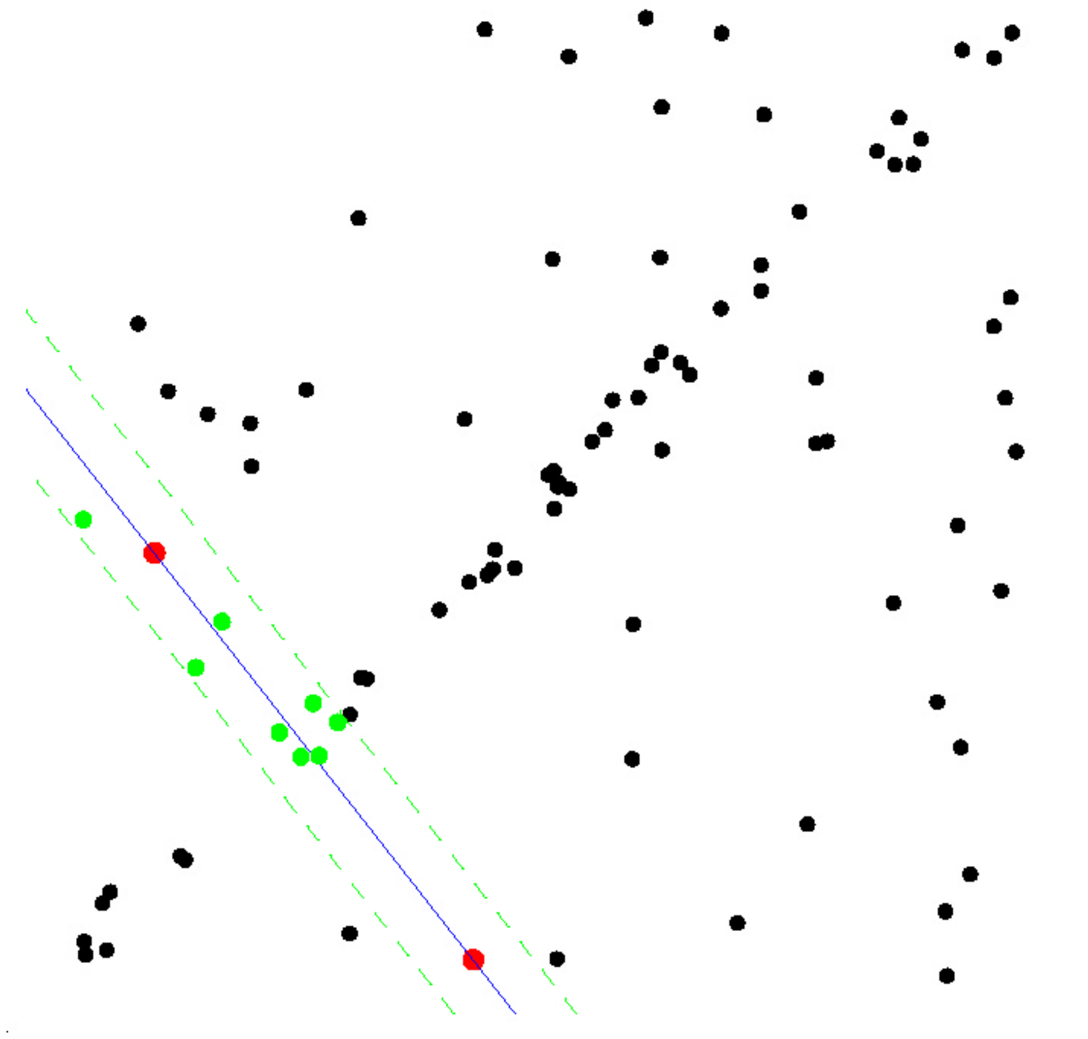
- Select sample of 2 points at random
- Calculate model parameters that fit the data in the sample
- **Calculate error function for each data point**

Algorithm 3: RANSAC



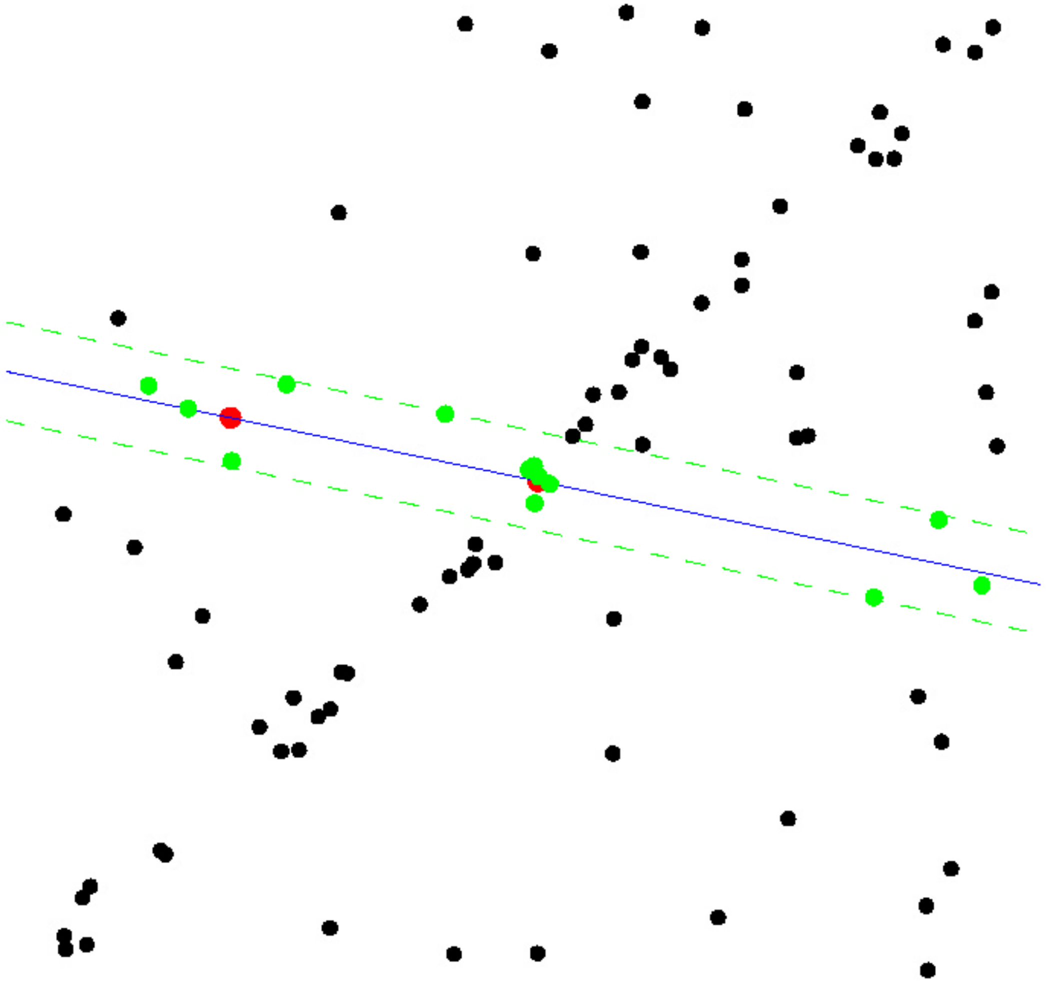
- Select sample of 2 points at random
- Calculate model parameters that fit the data in the sample
- Calculate error function for each data point
- **Select data that support current hypothesis**

Algorithm 3: RANSAC



- 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**

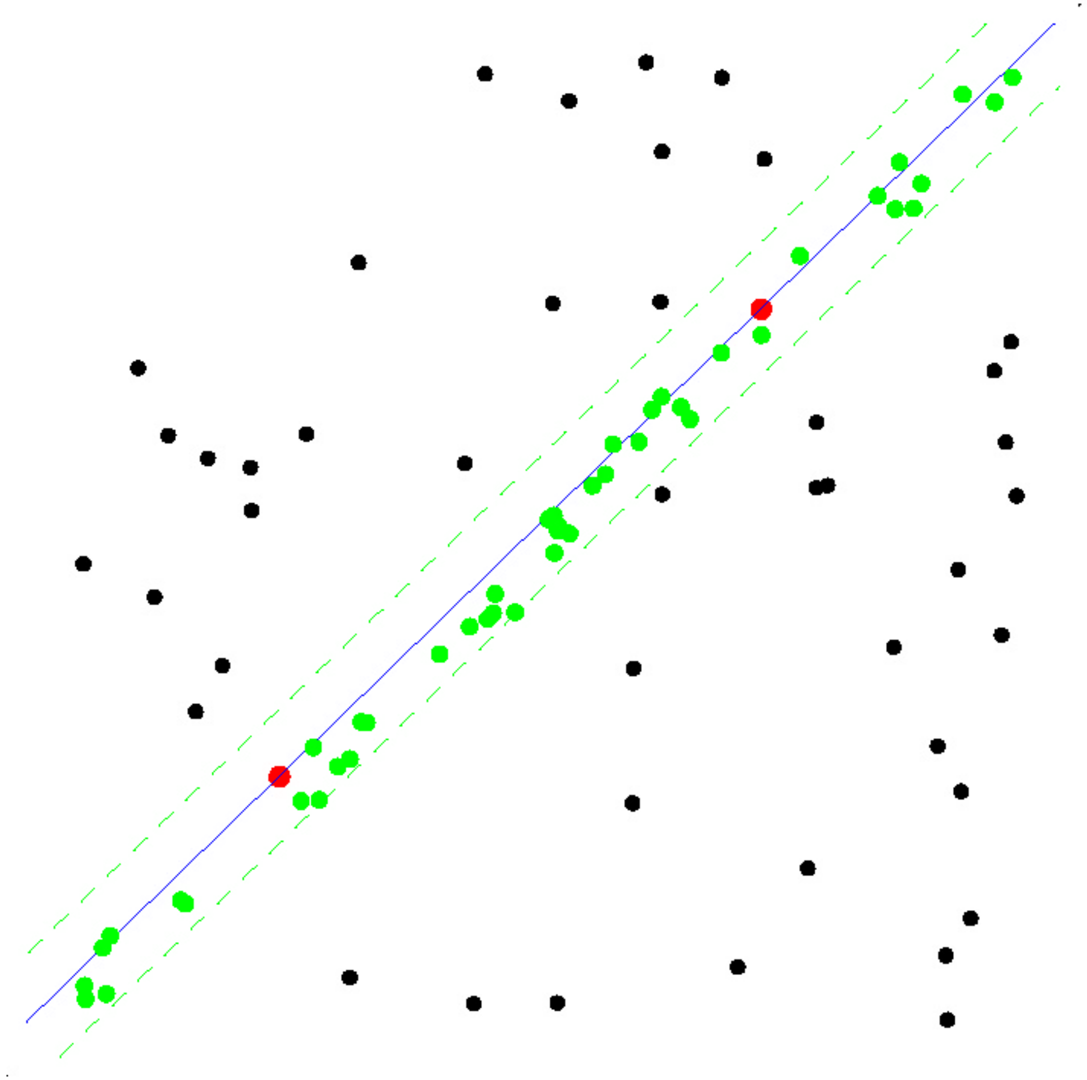
Algorithm 3: RANSAC



- 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**

Algorithm 3: RANSAC

ALL-INLIER SAMPLE



Algorithm 3: RANSAC

Algorithm 4: *RANSAC*

1. Initial: let A be a set of N points
2. **repeat**
3. Randomly select a sample of 2 points from A
4. Fit a line through the 2 points
5. Compute the distances of all other points to this line
6. Construct the inlier set (i.e. count the number of points with distance to the line $< d$)
7. Store these inliers
8. **until** Maximum number of iterations k reached
9. The set with the maximum number of inliers is chosen as a solution to the problem

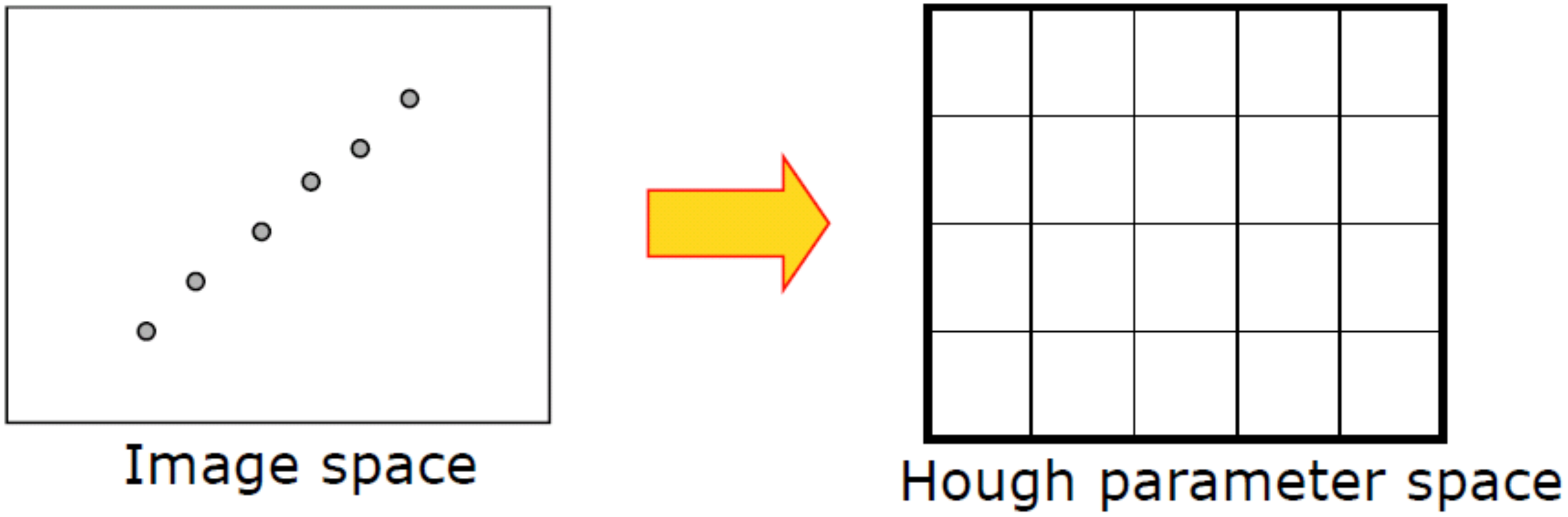
Algorithm 3: RANSAC

How many iterations does RANSAC need?

- 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*359/2= 64,620 possibilities!
- Do we really need to check all possibilities or can we stop RANSAC after 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
- This can be done in a probabilistic way

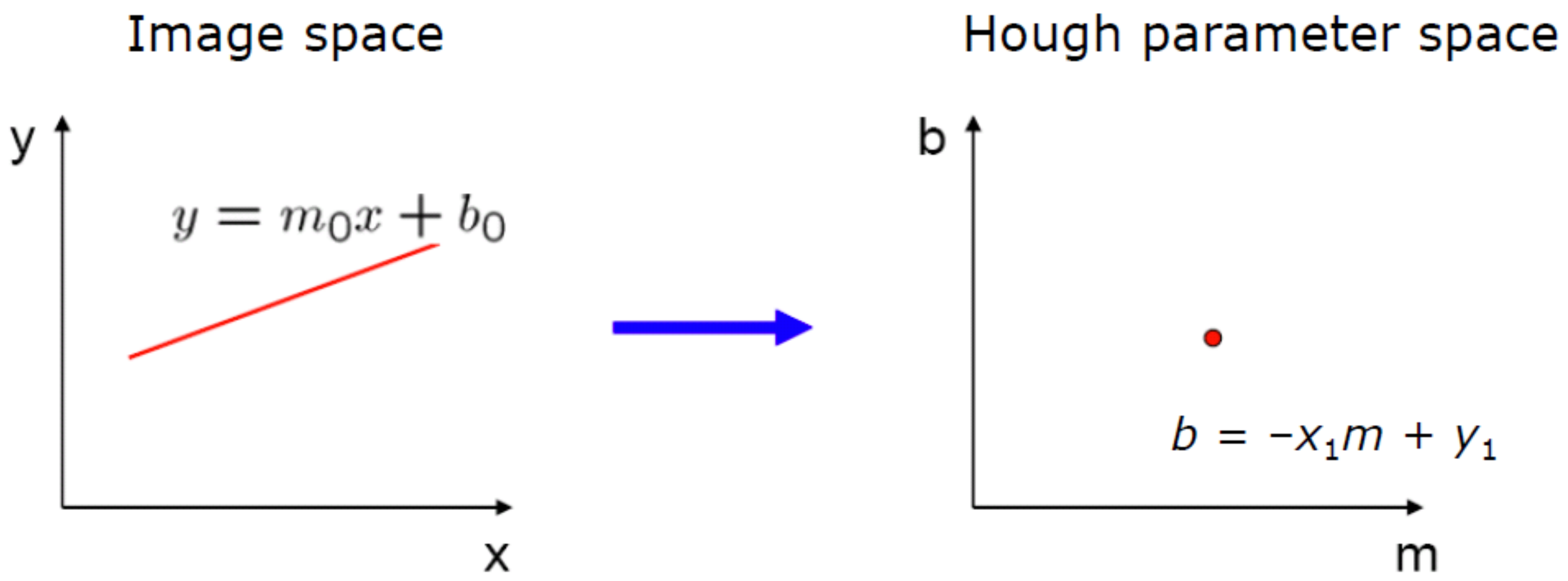
Algorithm 4: Hough-Transform

- Hough Transform uses a voting scheme



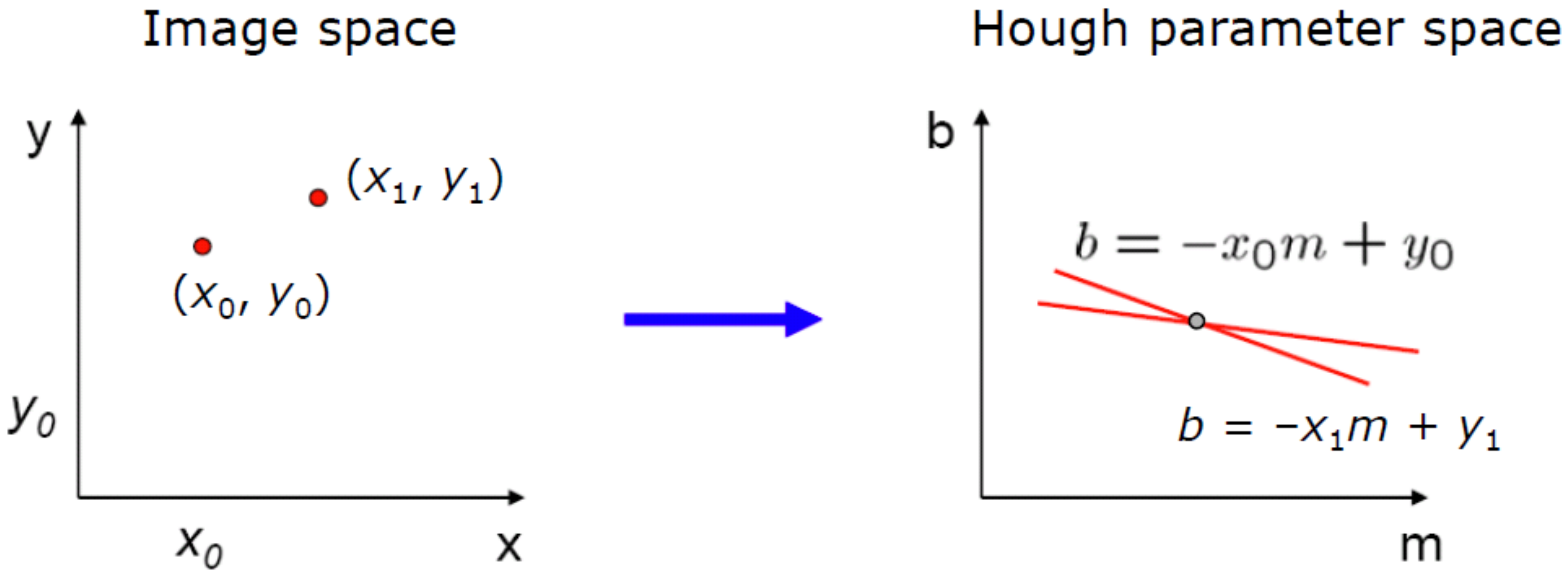
Algorithm 4: Hough-Transform

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



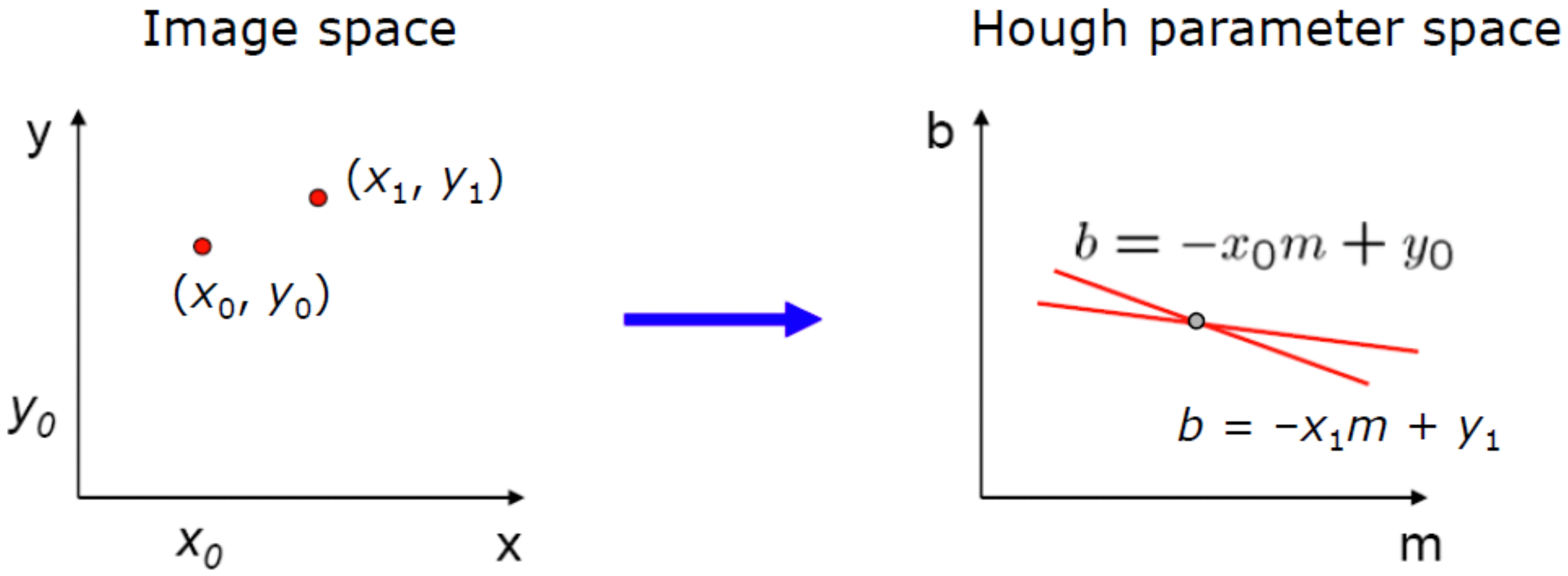
Algorithm 4: Hough-Transform

- What does a point (x_0, y_0) in the image space map to in the Hough space?



Algorithm 4: Hough-Transform

- Where is the line that contains both (x_0, y_0) and (x_1, y_1)?
  - It is the intersection of the lines $b = -x_0 m + y_0$ and $b = -x_1 m + y_1$



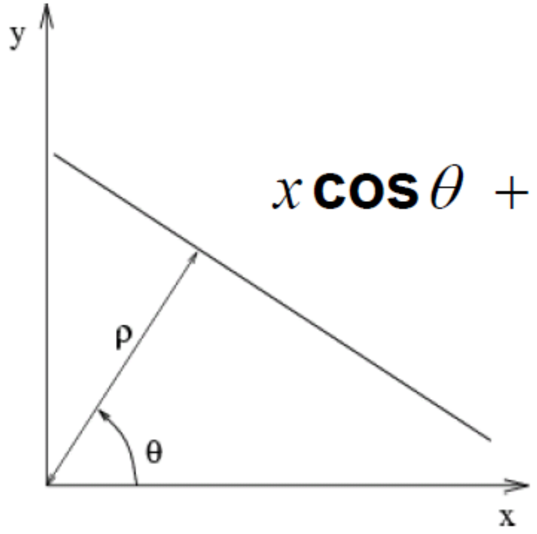
Algorithm 4: Hough-Transform

- Problems with the (m,b) space:
  - Unbounded parameter domain
  - Vertical lines require infinite m

Algorithm 4: Hough-Transform

- Problems with the (m,b) space:
  - Unbounded parameter domain
  - Vertical lines require infinite m

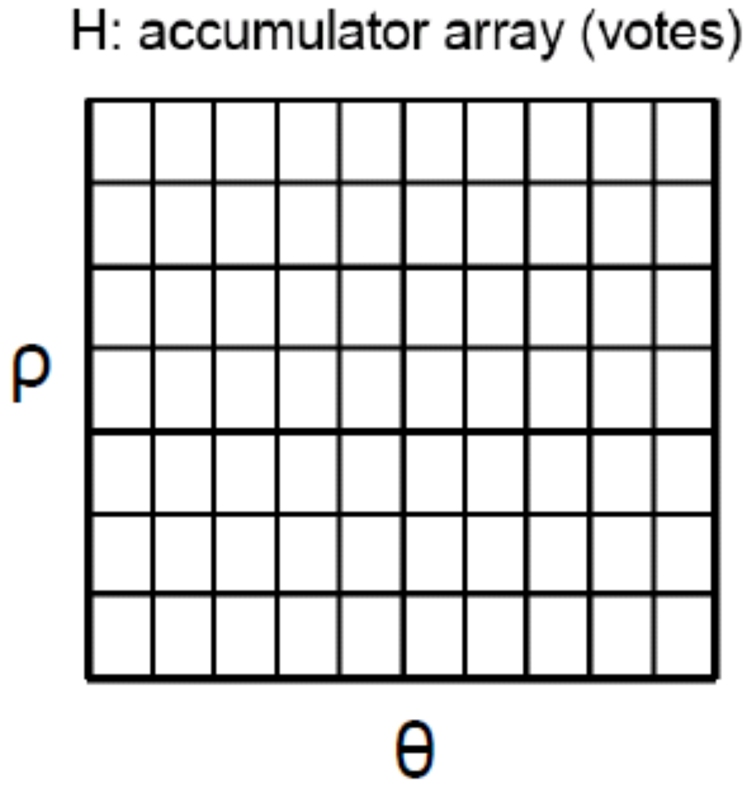
- Alternative: polar representation



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

Each point will add a sinusoid in the (θ,ρ) parameter space

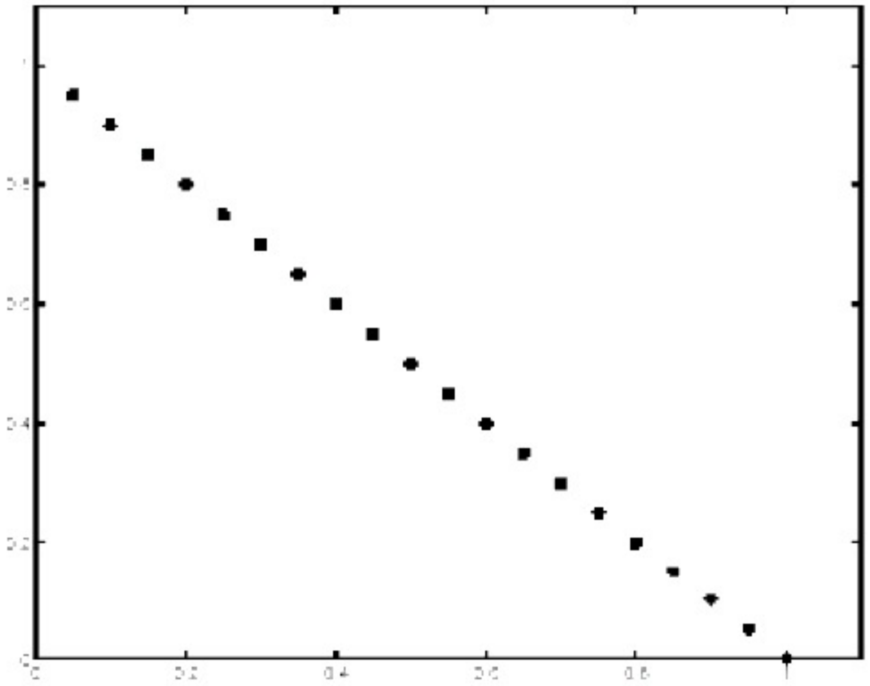
Algorithm 4: Hough-Transform



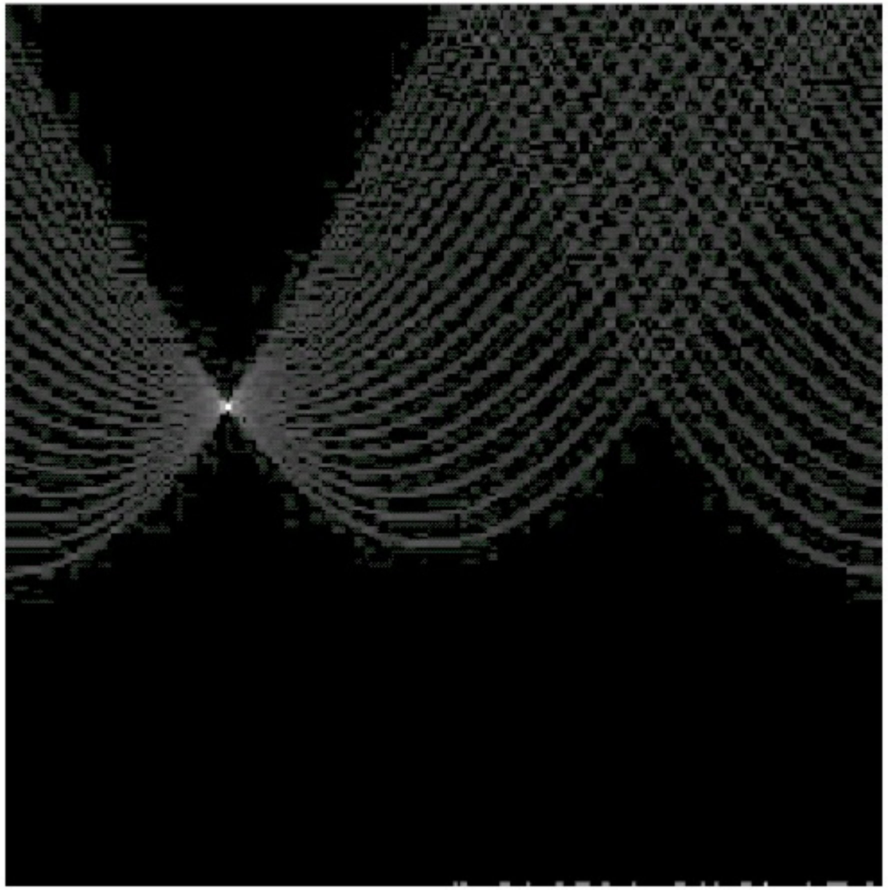
1. Initialize accumulator H to all zeros
2. For each edge point (x,y) in the image
   - For θ = 0 to 180 (with a step size of e.g. 18)
     - $\rho = x \cos \theta + y \sin \theta$
     - $H(\theta, \rho) = H(\theta, \rho) + 1$
   - end

   end
3. Find the values of (θ, ρ) where $H(\theta, \rho)$ is a local maximum
4. The detected line in the image is given by $\rho = x \cos \theta + y \sin \theta$

Algorithm 4: Hough-Transform



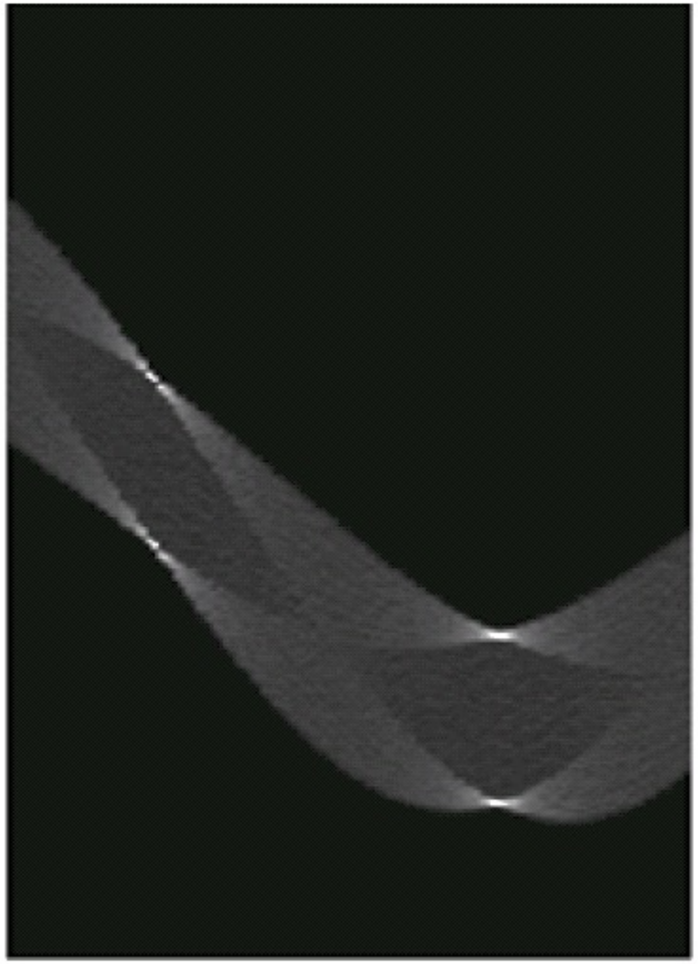
features



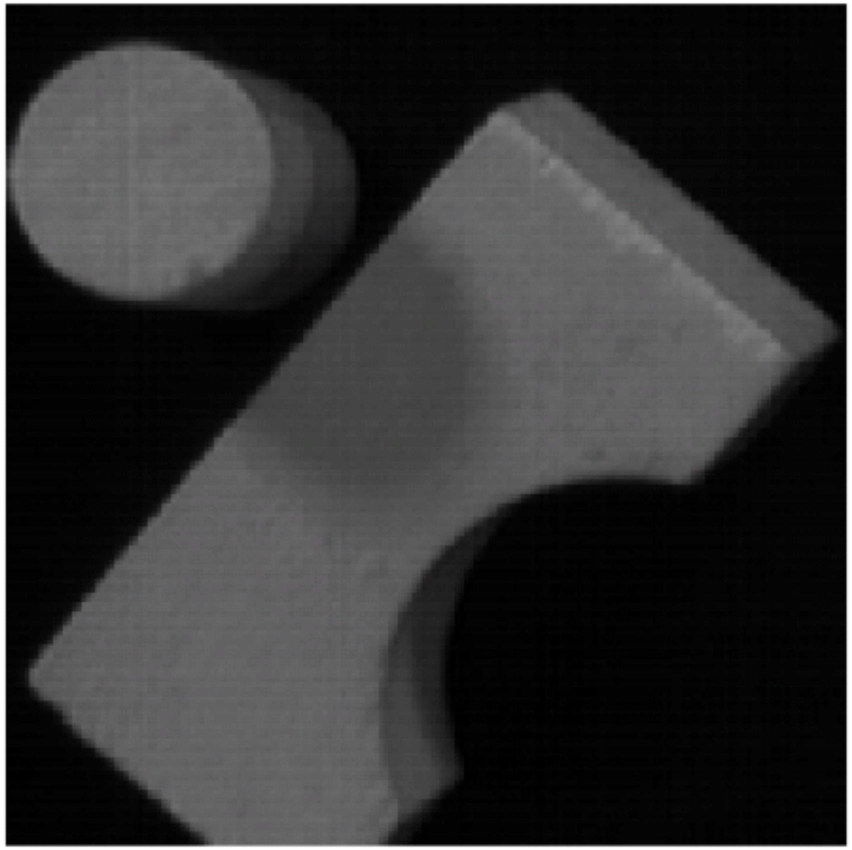
votes

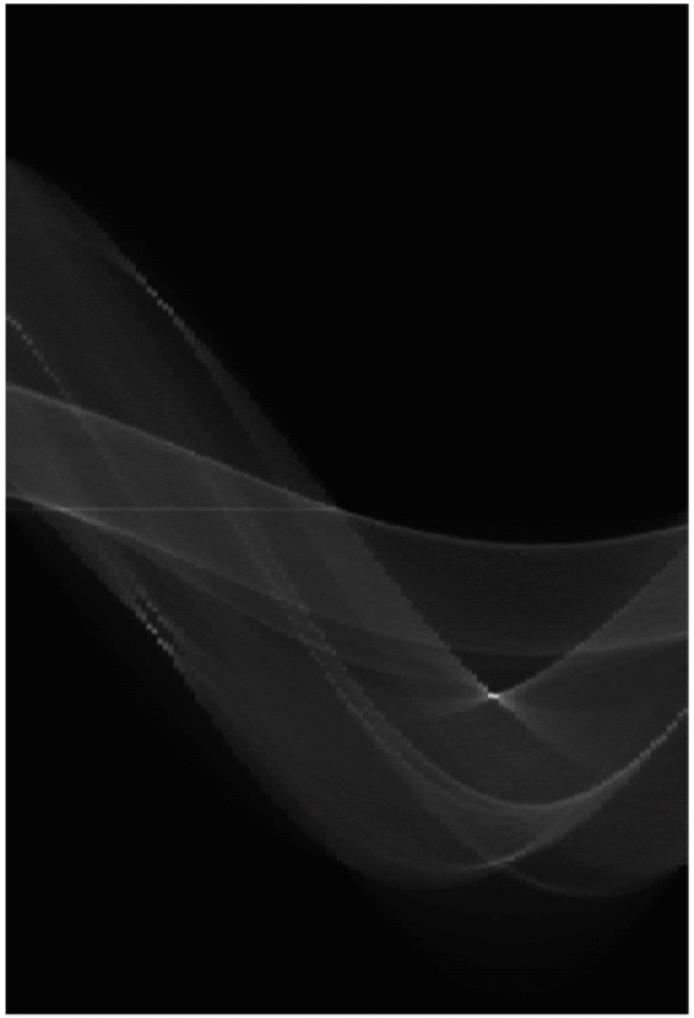
Algorithm 4: Hough-Transform

Square



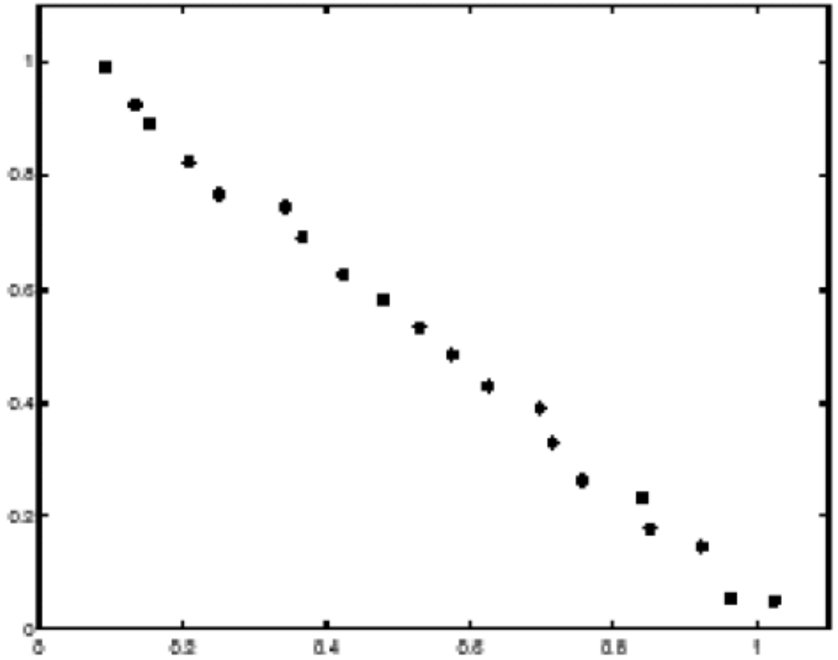
Algorithm 4: Hough-Transform



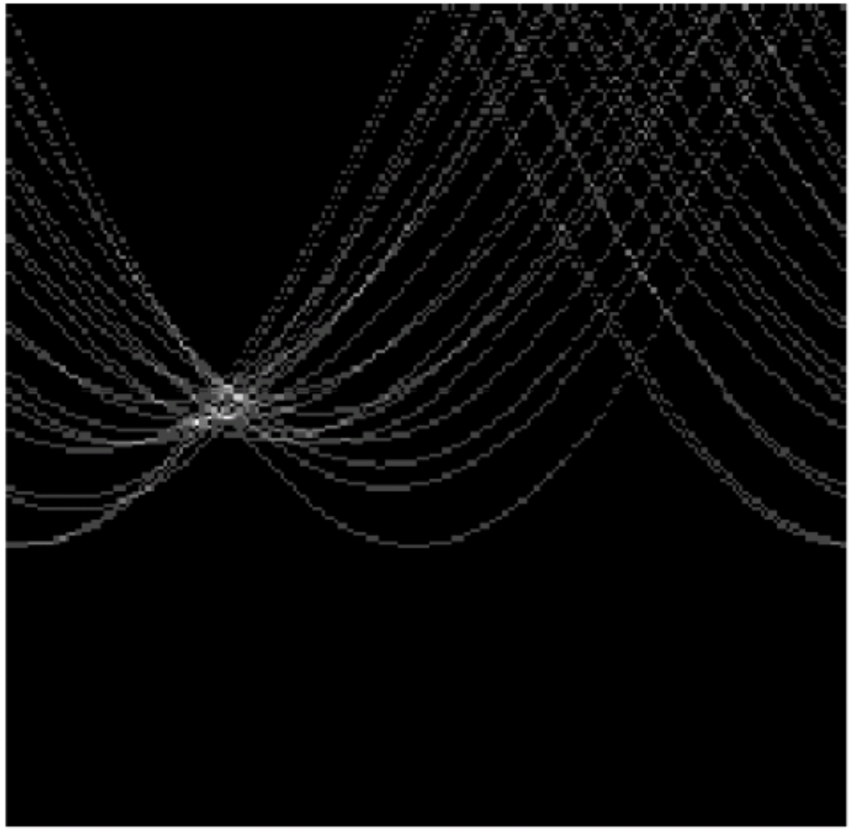


Algorithm 4: Hough-Transform

Effect of Noise



features

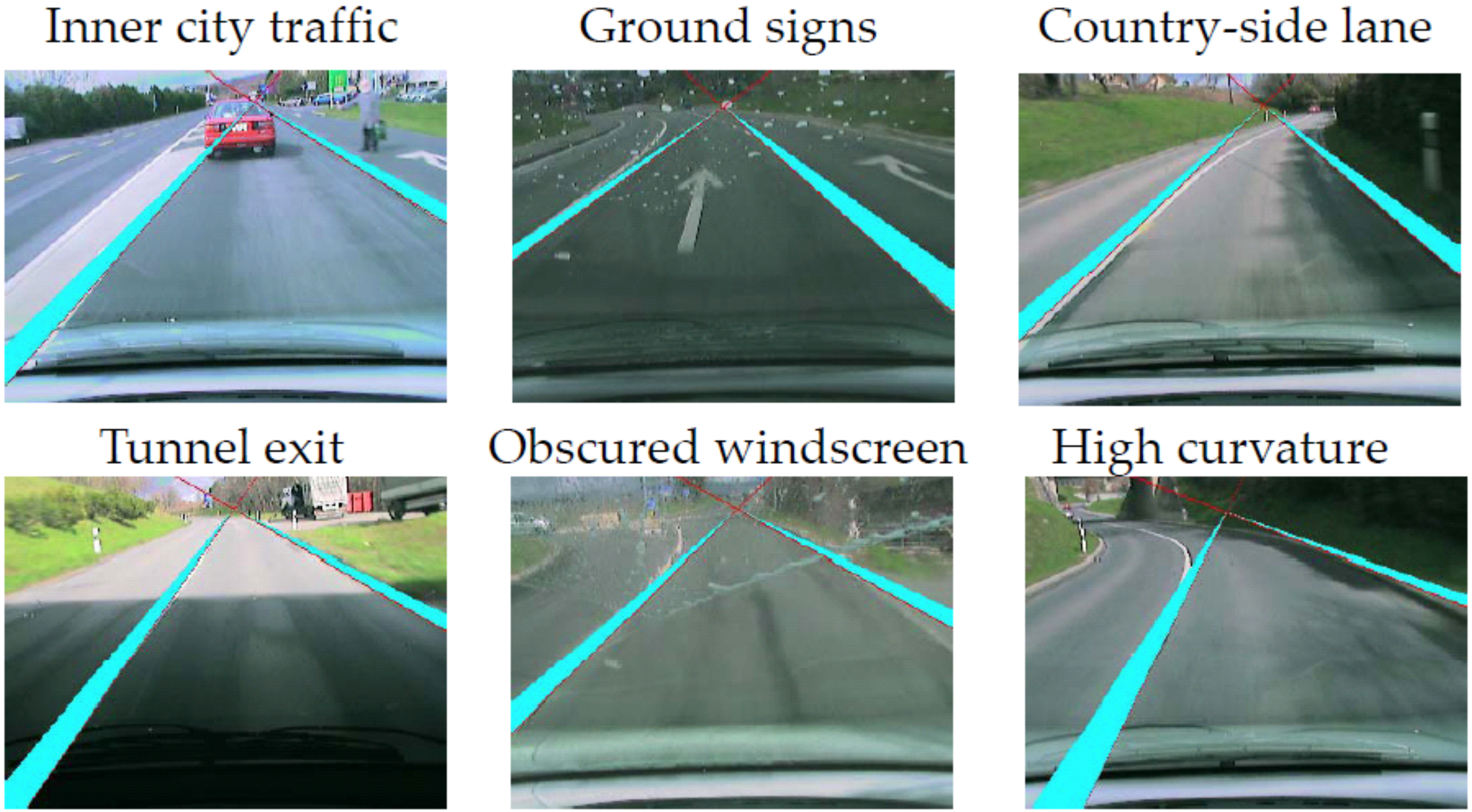


votes

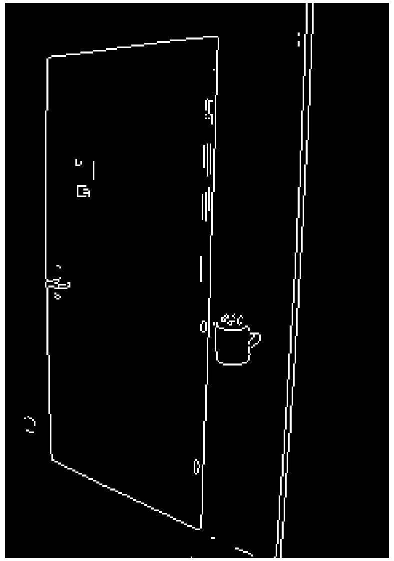
- Peak gets fuzzy and hard to locate

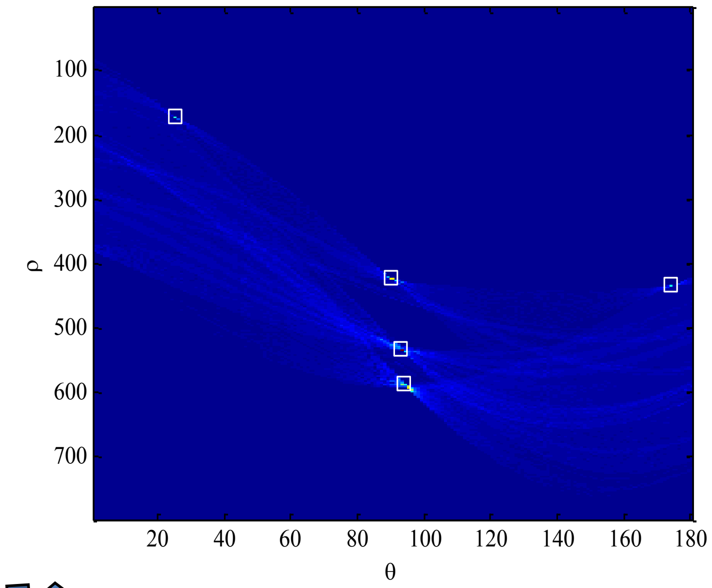
Algorithm 4: Hough-Transform

Application: Lane detection

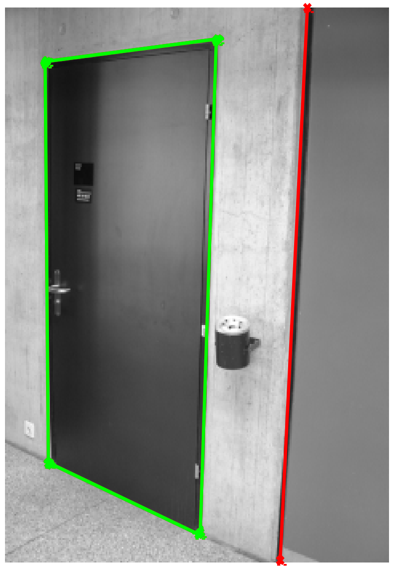


Example – Door detection using Hough Transform





Hough Transform



Hough Transform: other features

Lines:

$$p = (d, \upsilon)$$

$$g(x, y, p) := x \cdot \cos(\upsilon) + y \cdot \sin(\upsilon) - d$$

Circles:

$$p = (x_0, y_0, r)$$

$$g(x, y, p) := (x - x_0)^2 + (y - y_0)^2 - r^2$$

Ellipses:

$$p = (x_0, y_0, a, b, \psi)$$

$$g(x, y, p) := \frac{\left[(x - x_0) \cdot \cos(\psi) + (y - y_0) \cdot \sin(\psi)\right]^2}{a^2} + \frac{\left[(y - y_0) \cdot \cos(\psi) - (x - x_0) \cdot \sin(\psi)\right]^2}{b^2} - 1$$

Hough Transform

- Advantages
  - Noise and background clutter do not impair detection of local maxima
  - Partial occlusion and varying contrast are minimized
- Negatives
  - Requires time and space storage that increases exponentially with the dimensions of the parameter space

Comparison **Line Detection**

- Deterministic methods perform better with laser scans
  - Split-and-merge, Line-Regression, Hough transform
  - Make use of the sequencing property of scan points.
- Nondeterministic methods can produce high False Positives
  - RANSAC
  - Do not use the sequencing property
  - But it can cope with outliers
- Overall:
  - Split-and-merge is the fastest, best real-time applications