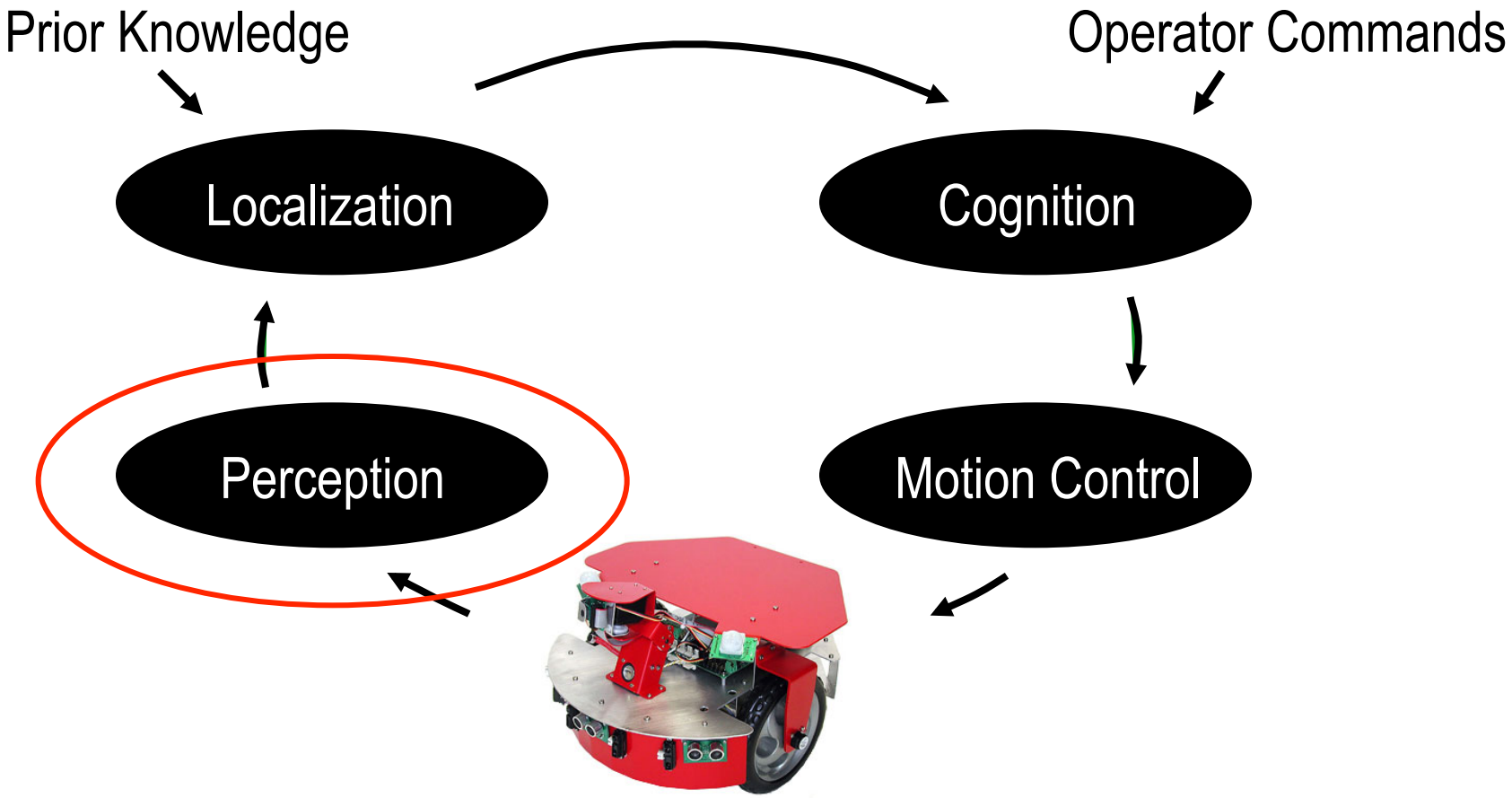
COS 495 - Lecture 11
Autonomous Robot Navigation

Instructor: Chris Clark

Semester: Fall 2011

Figures courtesy of Siegwart & Nourbakhsh

Control Structure

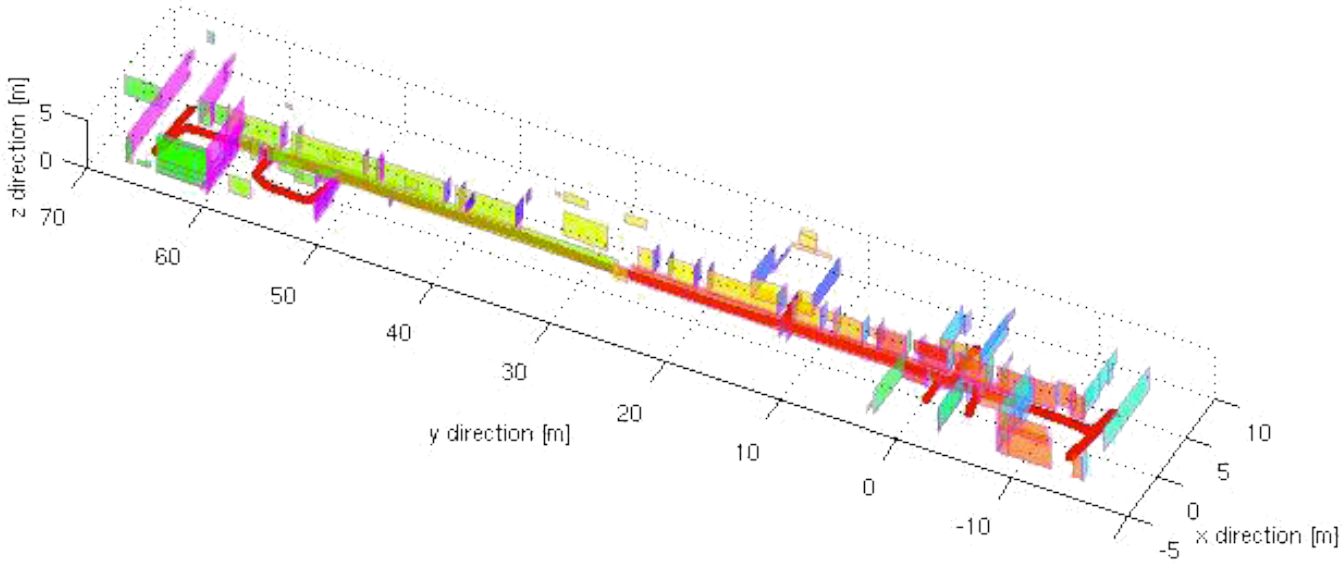


Line Extraction

- Overview of the Problem
- Line Extraction
- Segmentation

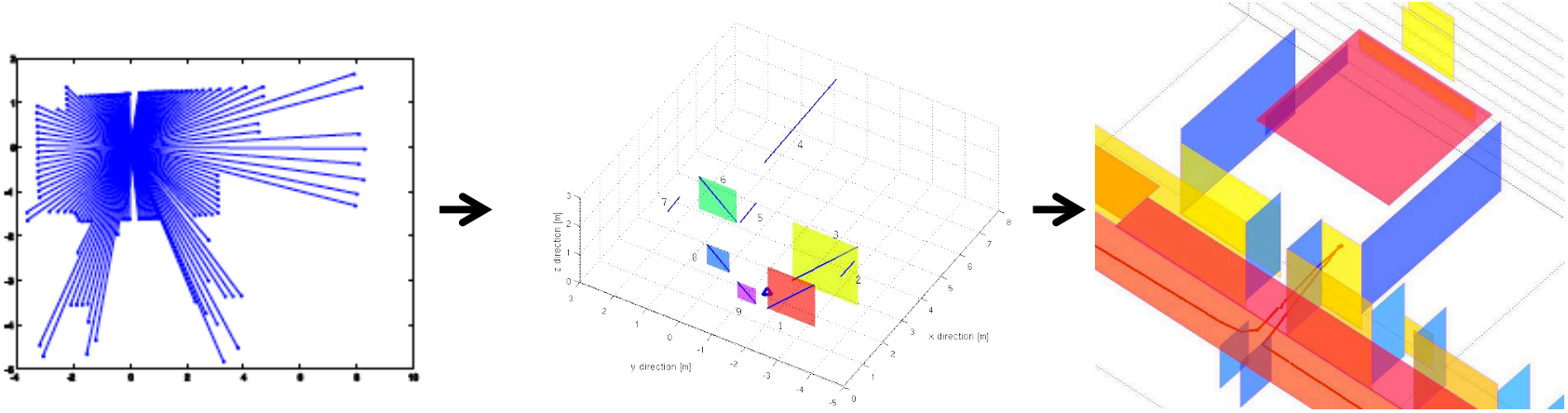
Line Extraction Problem

- Given range data, how do we extract line segments (or planes) to create?
  - These features (line segments) can be used to build maps or be compared with an existing map.



Line Extraction Problem

- From raw data, create features
  - Features are much more compact than raw data
  - Can reflect physical or abstract objects
  - Rich in information
  - Can assess accuracy of feature



Line Extraction Problem

- Three Questions
  1. How many lines are there?
  2. Which data points belong to which lines?

  Segmentation

  3. Given which points belong to which lines, how do we estimate line parameters?

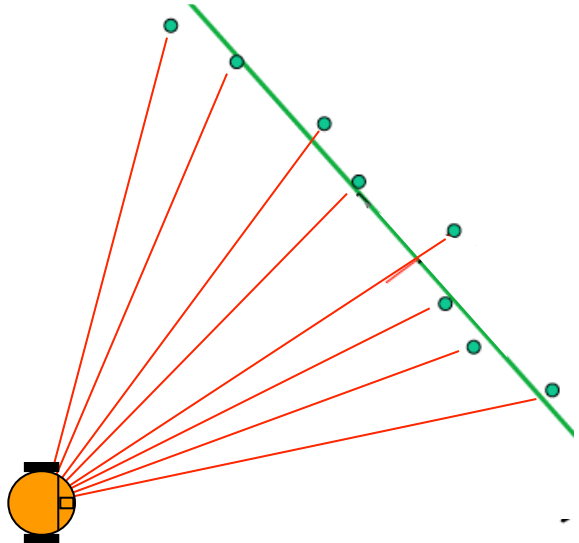
  Line Extraction

Line Extraction

- Overview of the Problem
- Line Extraction
- Segmentation

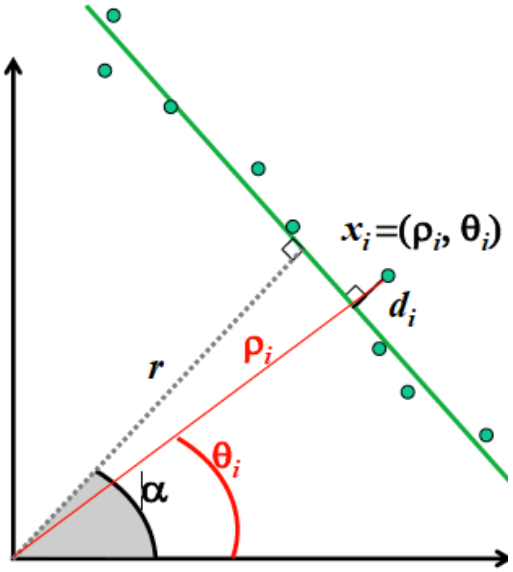
Line Extraction

- Problem:
  - Given a measurement vector of range and bearing tuples, what are the parameters that define a line feature for these measurements.



Line Extraction

- Problem (restated):
  - Given a measurement vector of N range and bearing tuples, $x_i = (\rho_i, \theta_i)$ for $i=1..N$, what are the parameters r,α that define a line feature for these measurements.



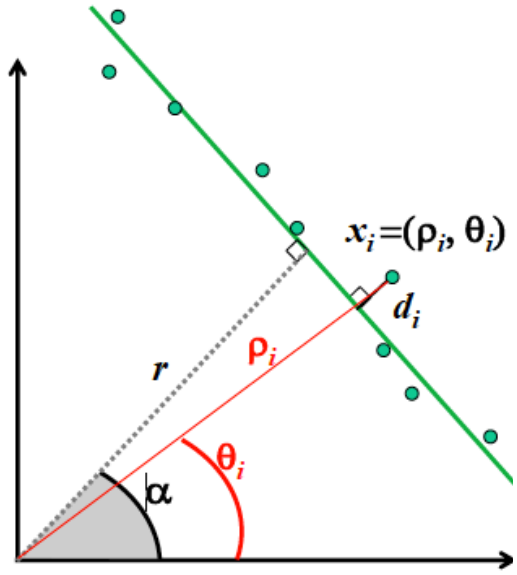
Line Extraction

- Solution: Minimize Sum of Squared Errors
  - All measurements should satisfy the linear equation:

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

  - But measurements are noisy, and points will be some distance d_i from the line.

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



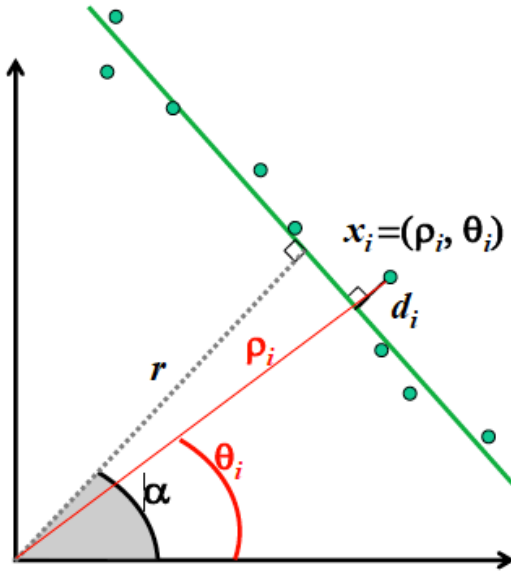
Line Extraction

- Solution: Minimize Sum of Squared Errors
  - Our solution tries to minimize the error

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

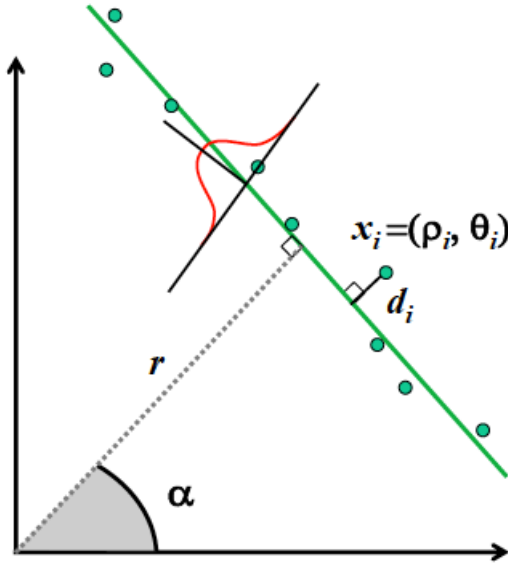
  - We do this by solving the system of equations

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



Line Extraction

- Solution: Minimize Sum of Squared Errors
  - This is known as an **Unweighted** Least Squares Solution
  - We can do better by using our confidence in each measurement
  - Recall there is a error variance associated with each measurement
  - This leads to a **Weighted** Least Square Solution

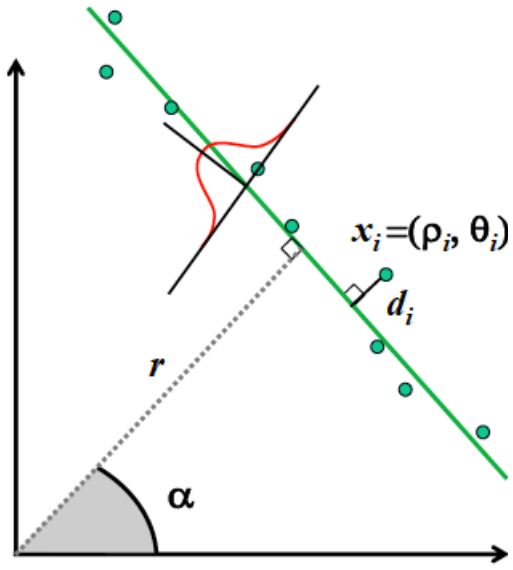


Line Extraction

- Solution: Minimize Sum of Squared Errors
  - The Weighted Least Squares Solution reformulates the error to minimize:

$$w_i = 1/\sigma_i^2$$

$$S = \sum w_i d_i^2$$



Line Extraction

- Solution: Minimize Sum of Squared Errors
  - The solution to

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

  - Results in

$$r = \frac{\sum w_i \rho_i \cos(\theta_i - \alpha)}{\sum w_i}$$

$$\alpha = \frac{1}{2}\text{atan}\left(\frac{\sum w_i \rho_i^2 \sin 2\theta_i - \frac{2}{\sum w_i}\sum\sum w_i w_j \rho_i \rho_j \cos\theta_i \sin\theta_j}{\sum w_i \rho_i^2 \cos 2\theta_i - \frac{1}{\sum w_i}\sum\sum w_i w_j \rho_i \rho_j \cos(\theta_i + \theta_j)}\right)$$

Line Extraction

- Overview of the Problem
- Line Extraction
- Segmentation
  - Split and Merge
  - Split and Merge – Fixed Endpoint
  - RANSAC

Segmentation

- Split and Merge
  - Recursive procedure of fitting and splitting

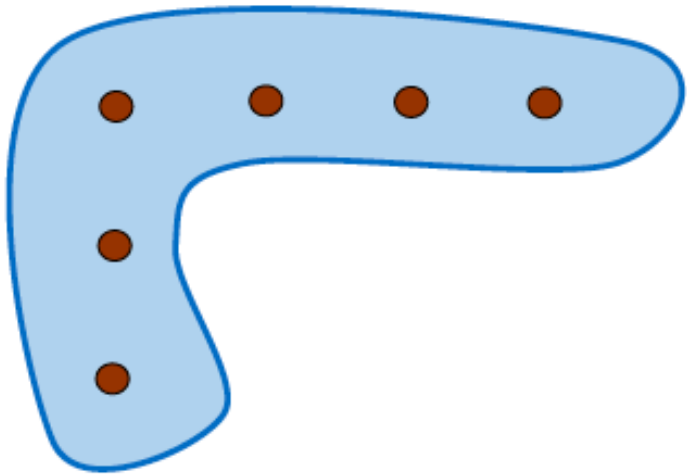
Initialise set **S** to contain all points

Split

- Fit a line to points in current set **S**
- Find the most distant point to the line
- If distance > threshold ⇨ split & repeat with 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



Segmentation

- Split and Merge
  - Recursive procedure of fitting and splitting

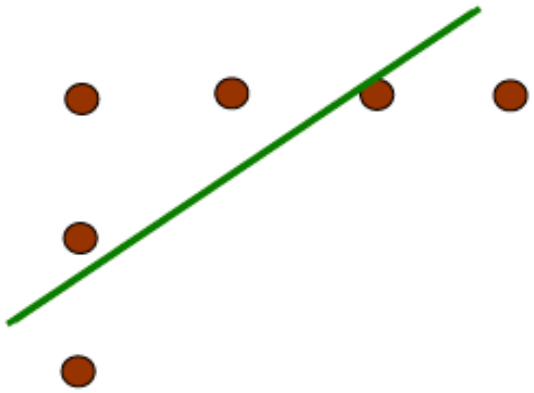
Initialise set **S** to contain all points

Split

- Fit a line to points in current set **S**
- Find the most distant point to the line
- If distance > threshold ⇨ split & repeat with 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



Segmentation

- Split and Merge
  - Recursive procedure of fitting and splitting

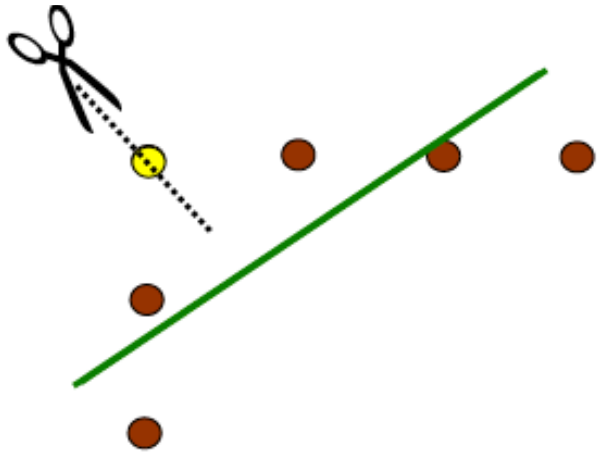
Initialise set **S** to contain all points

Split

- Fit a line to points in current set **S**
- Find the most distant point to the line
- If distance > threshold ⇨ split & repeat with 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



Segmentation

- Split and Merge
  - Recursive procedure of fitting and splitting

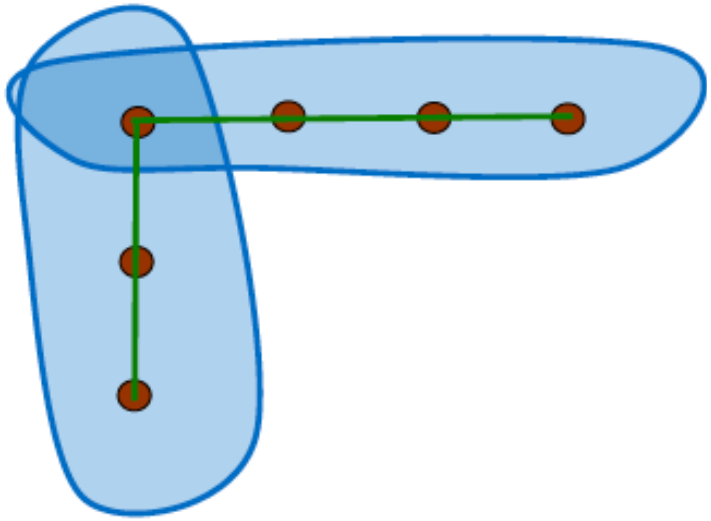
Initialise set **S** to contain all points

Split

- Fit a line to points in current set **S**
- Find the most distant point to the line
- If distance > threshold ⇨ split & repeat with 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



Line Extraction

- Overview of the Problem
- Line Extraction
- Segmentation
  - Split and Merge
  - Split and Merge – Fixed Endpoint
  - RANSAC

Segmentation

- Split and Merge - Iterative End Point
  - Recursive splitting, but simply connects end points for fitting

Line Extraction

- Overview of the Problem
- Line Extraction
- Segmentation
  - Split and Merge
  - Split and Merge – Fixed Endpoint
  - RANSAC

Segmentation

- **RANSAC** = **RAN**dom**SA**mple**C**onsensus.
  - A generic and robust fitting algorithm of models in the presence of outliers (i.e. points which do not satisfy a model)
  - Generally applicable algorithm to any problem where the goal is to **identify the inliers which satisfy a predefined model**.
  - Typical applications in robotics are: line extraction from 2D range data, plane extraction from 3D range data, feature matching…

Segmentation

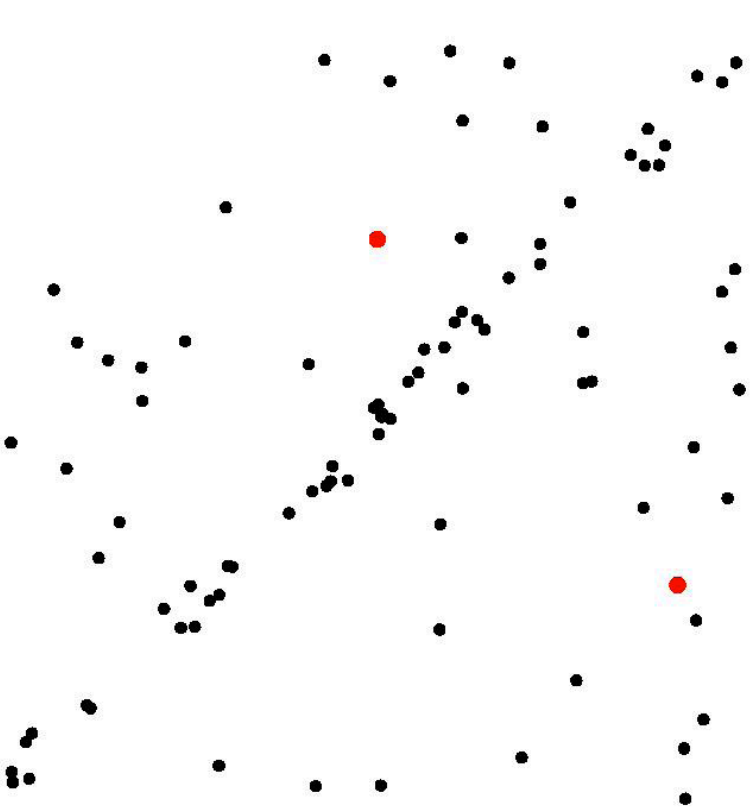
- RANSAC
  - RANSAC is an **iterative** method and is **non-deterministic** in that the probability to find a set free of outliers increases as more iterations are used
  - Drawback: A nondeterministic method, results are different between runs.

Segmentation

- RANSAC Example

Segmentation

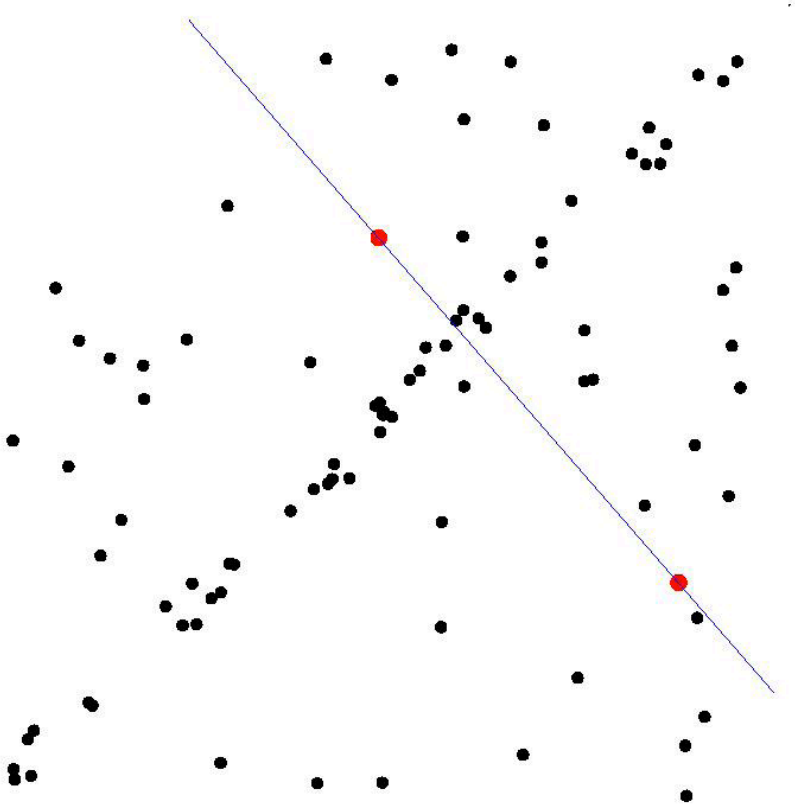
- RANSAC Example



- **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

Segmentation

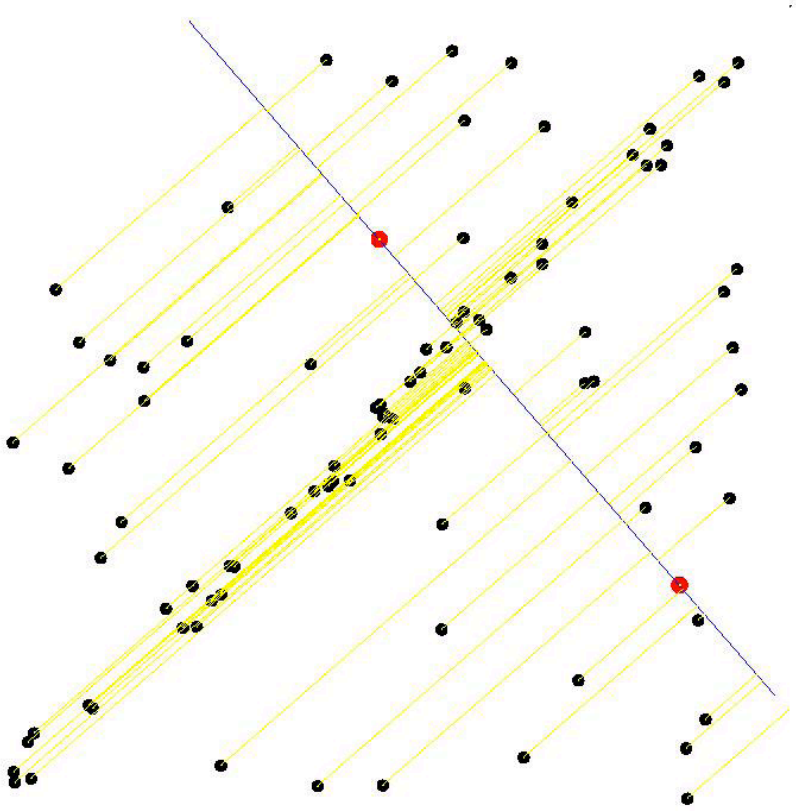
- RANSAC Example



- 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

Segmentation

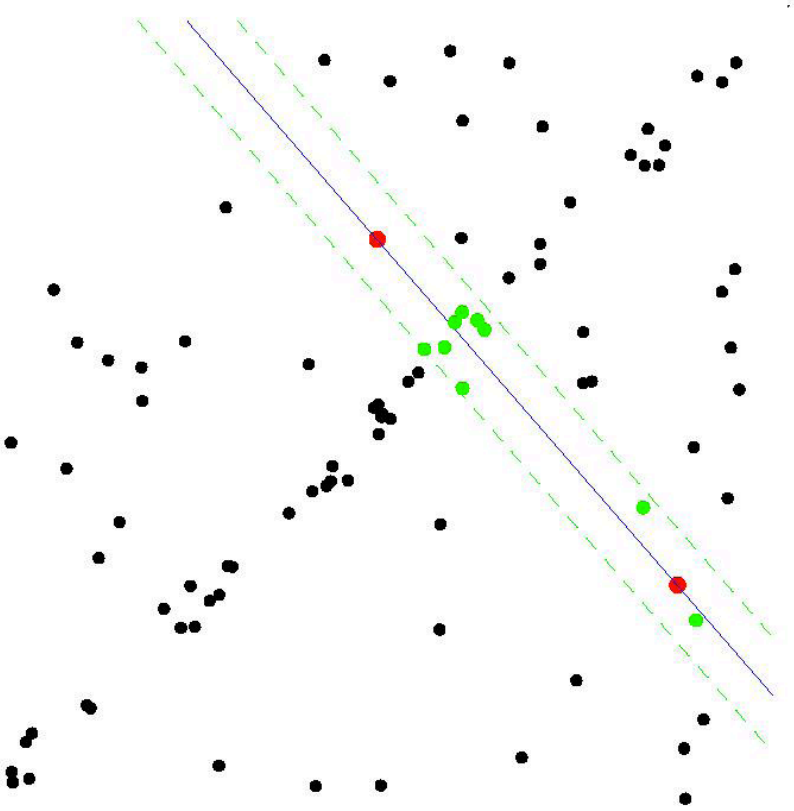
- RANSAC Example



- 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

Segmentation

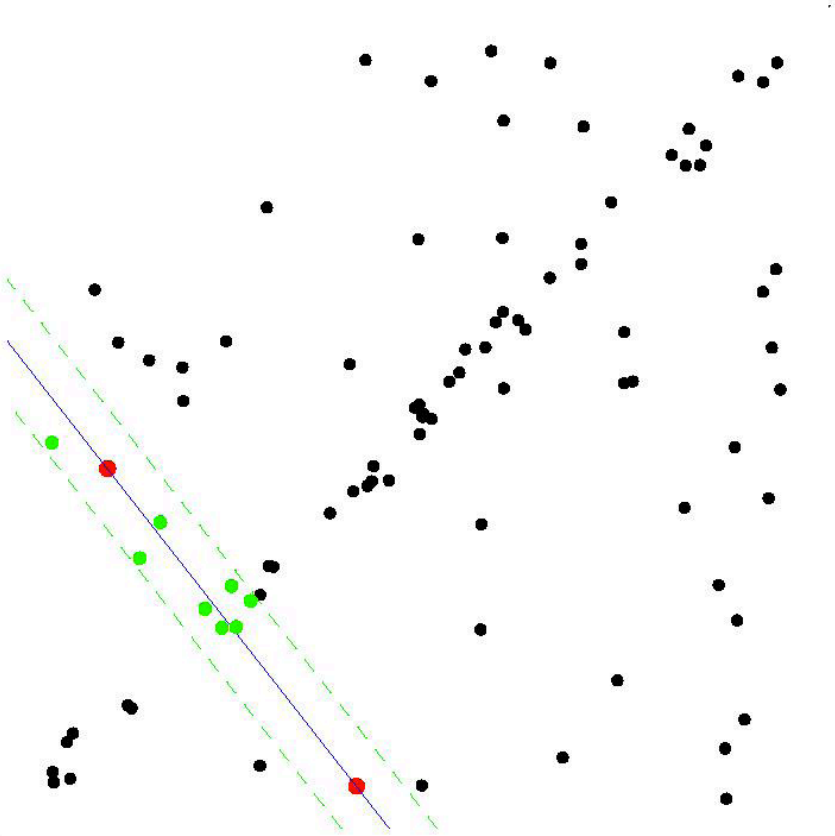
- RANSAC Example



- 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

Segmentation

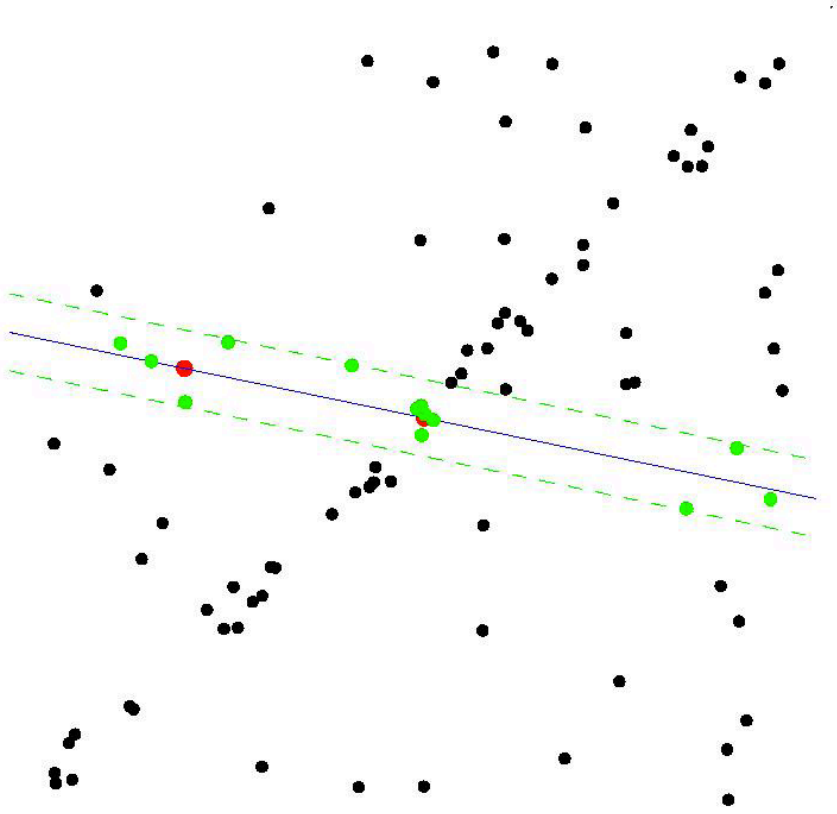
- RANSAC Example



- 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**

Segmentation

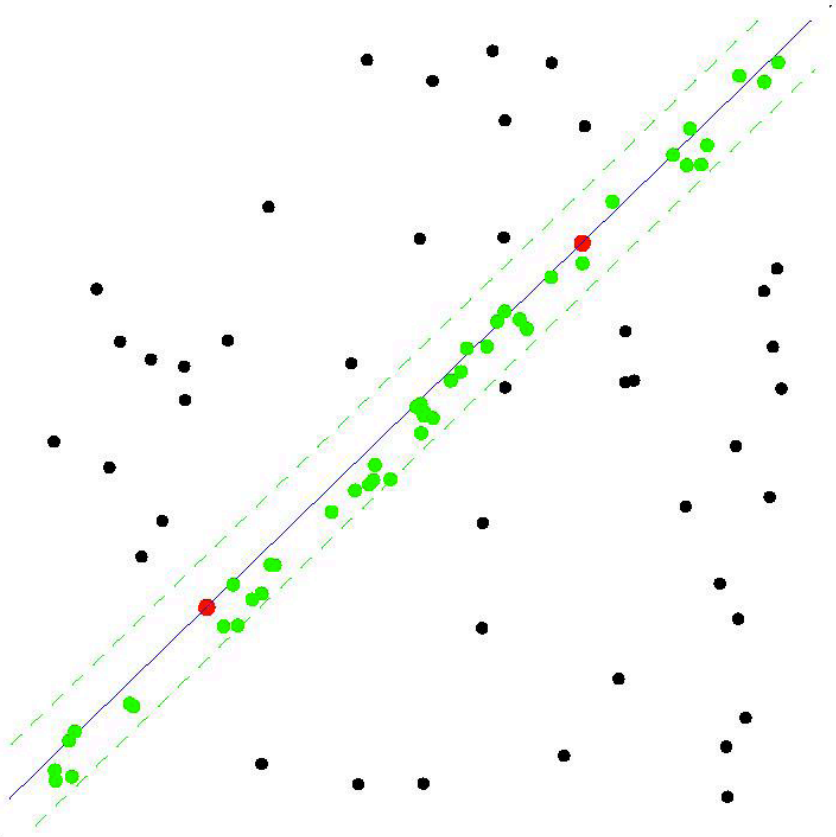
- RANSAC Example



- 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**

Segmentation

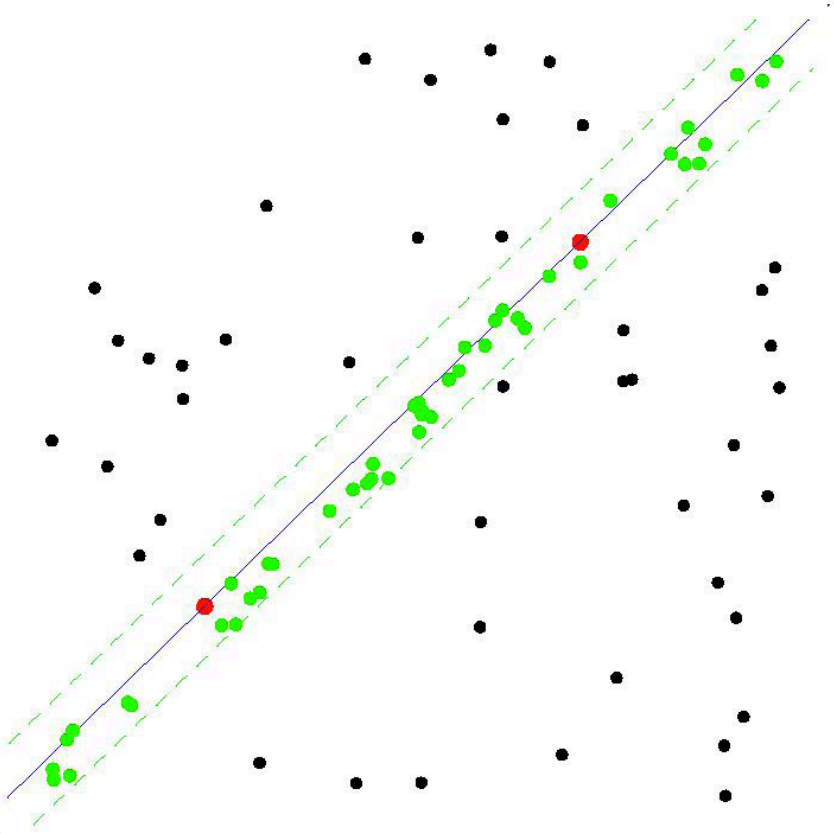
- RANSAC Example



- 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**

Segmentation

- RANSAC Example



- Stop after k iterations and select model with the max number of inliers.

Line Extracation and Segmentation

- Hopefully a new lab …