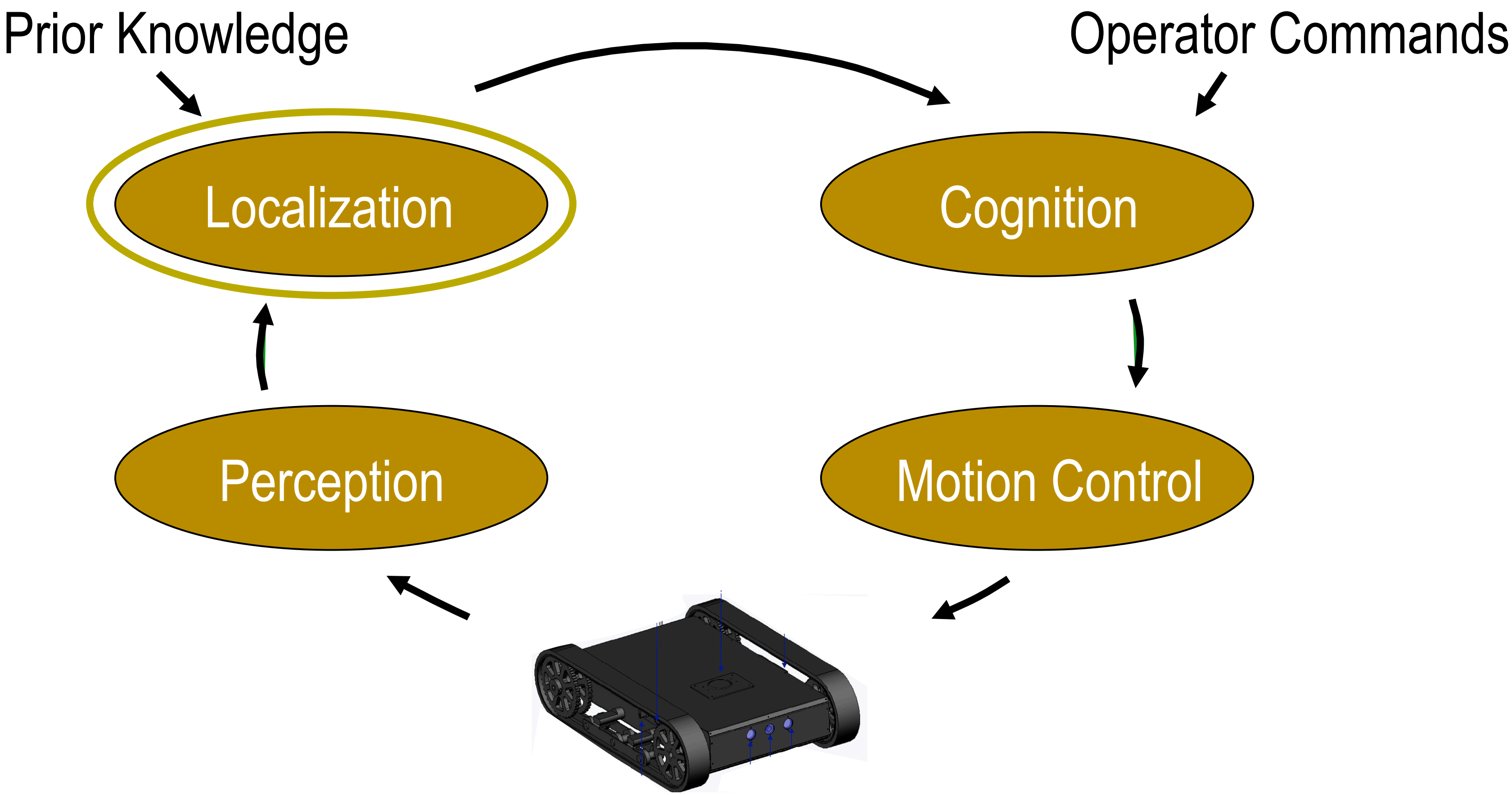
E190Q – Lecture 8
Autonomous Robot Navigation

Instructor: Chris Clark

Semester: Spring 2014

Figures courtesy of Siegwart & Nourbakhsh

Control Structures
Planning Based Control

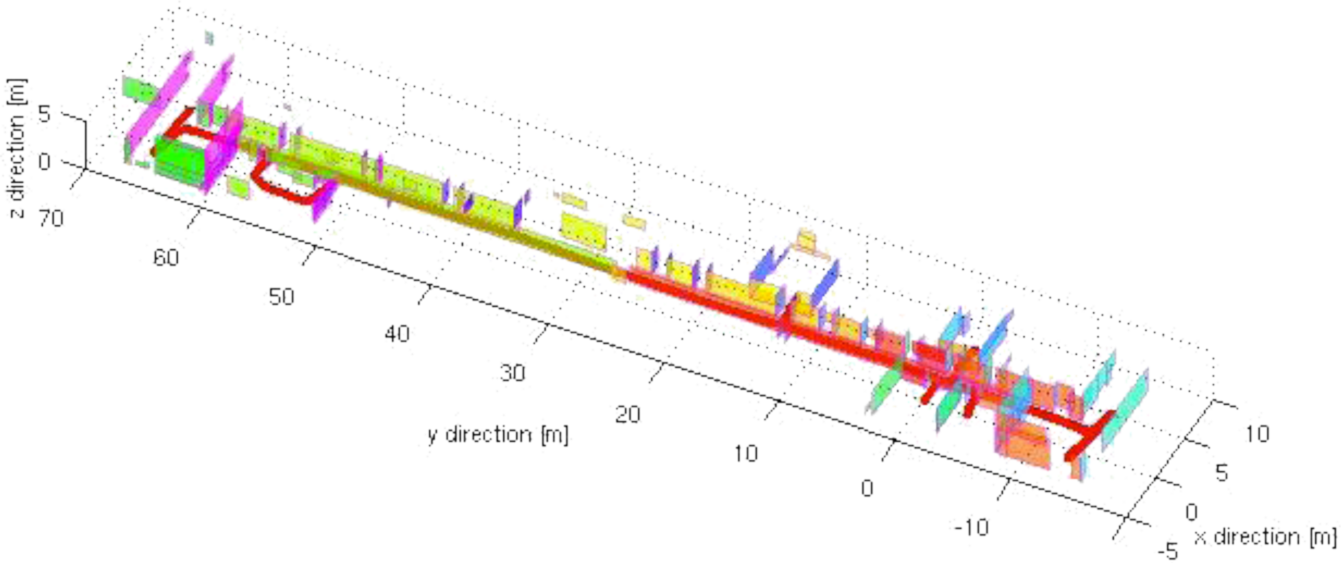


Outline – Mapping

1. Wall as Lines
   1. Segmentation
   2. Line Extraction
2. Walls as Grid Cells
   1. Evidence Grid
   2. Log Likelihood

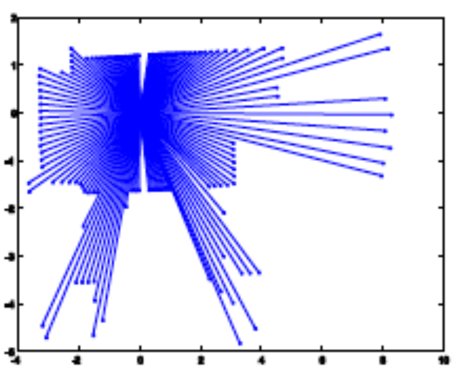
Line Extraction Problem

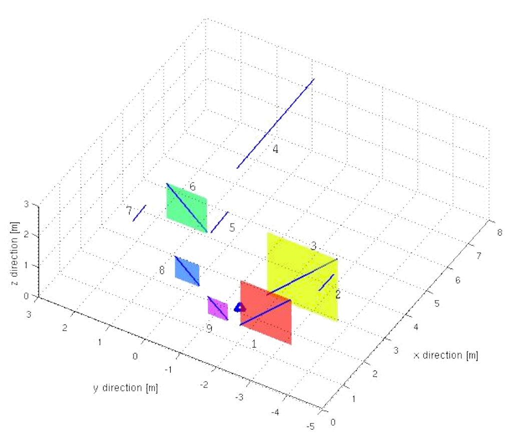
- Given range data, how do we extract line segments (or planes) to create?
  - These features (line segments) can be used to build maps or be compared with an existing map.

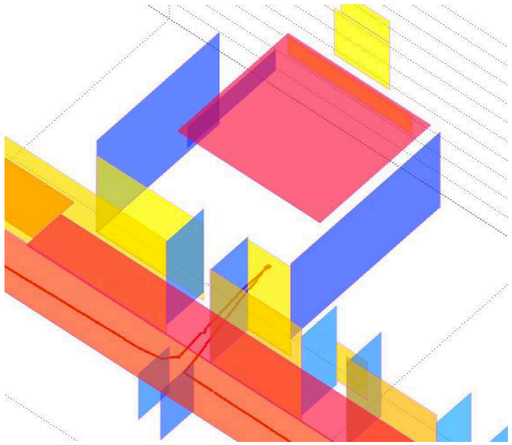


Line Extraction Problem

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

Questions 1–2: Segmentation

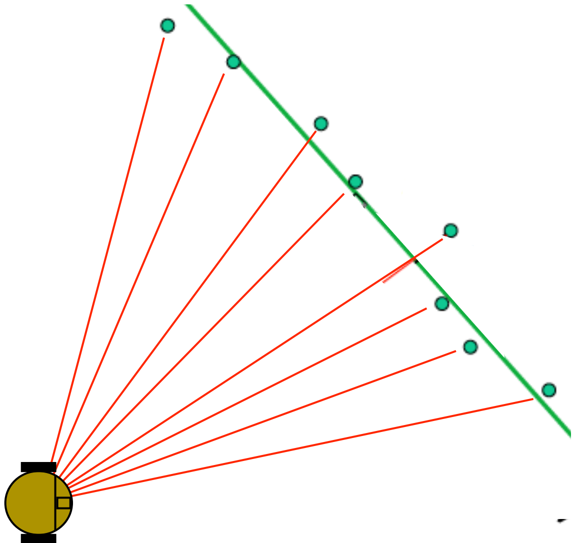
Question 3: Line Extraction

Outline – Mapping

1. Wall as Lines
    1. Line Extraction
    2. Segmentation
2. Walls as Grid Cells
    1. Evidence Grid
    2. Log Likelihood

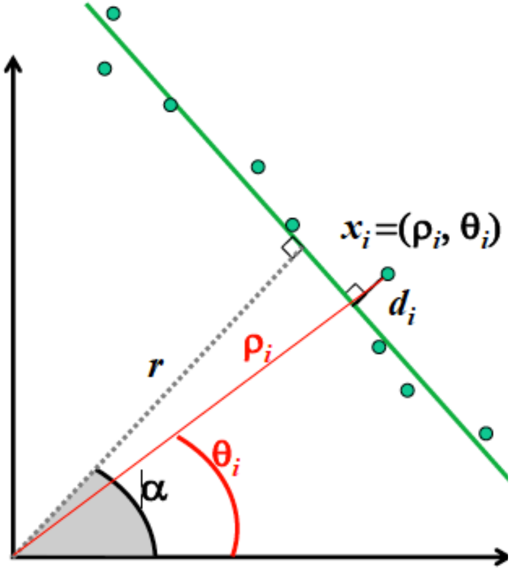
Line Extraction

- 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.



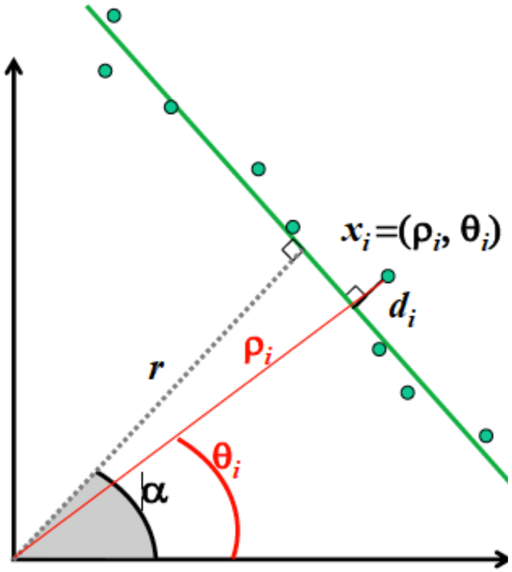
Line Extraction

- 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.

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



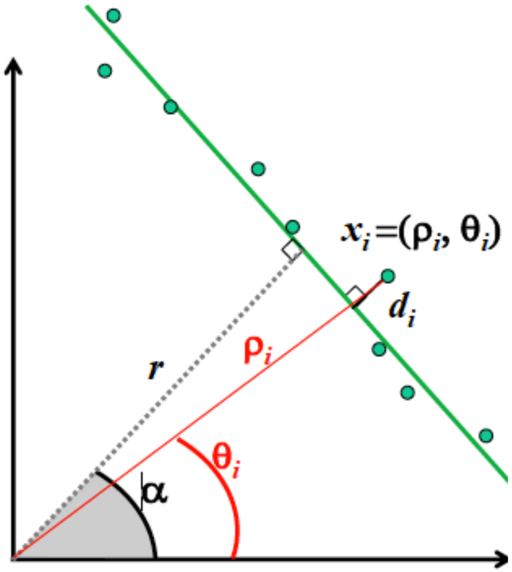
Line Extraction

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

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

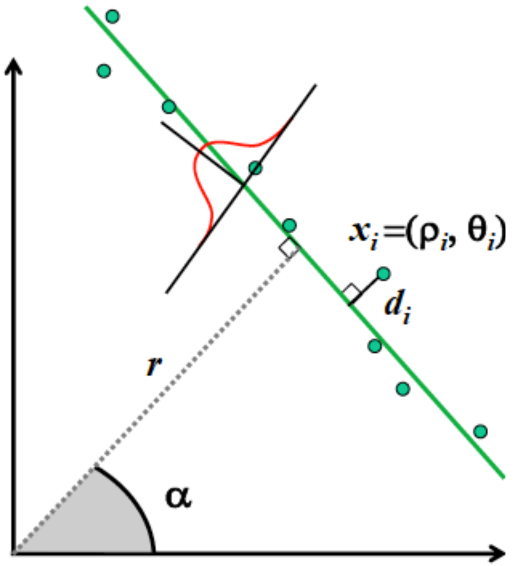
  - We do this by solving the system of equations

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



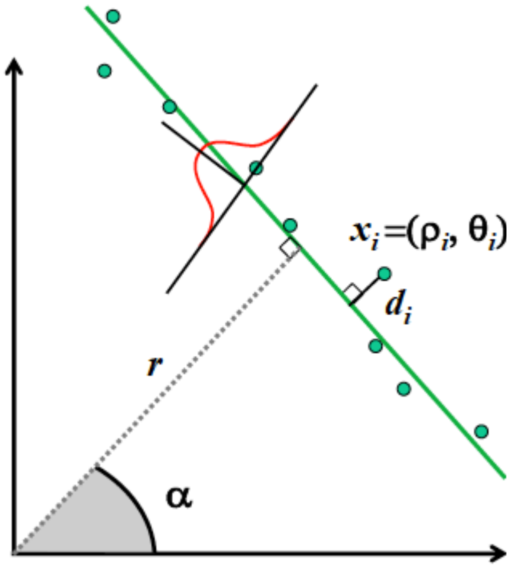
Line Extraction

- 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

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

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

Line Extraction

- Examples – Underwater Wall Mapping

Line Extraction

- Examples – Underwater Wall Mapping

Line Extraction

- Examples – Underwater Wall Mapping

Line Extraction

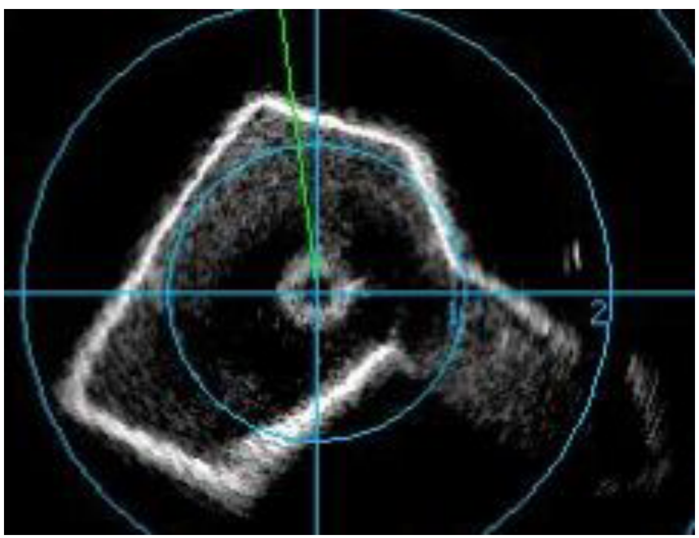
- Examples – Underwater Wall Mapping

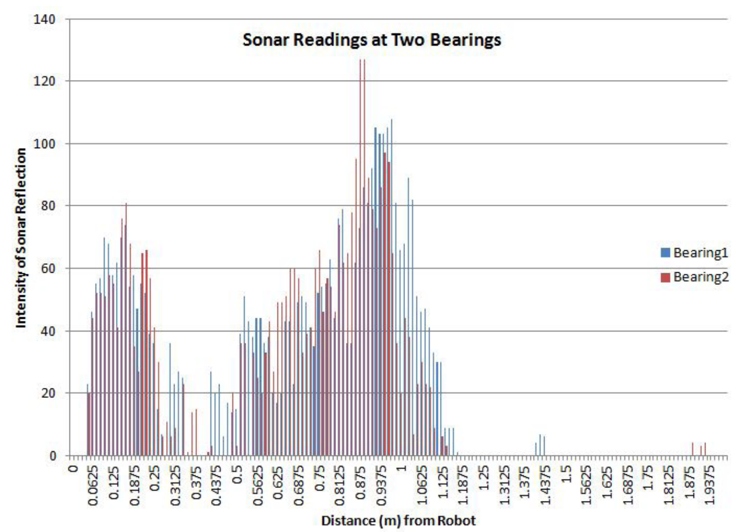
Line Extraction

- Examples – Underwater Wall Mapping

Line Extraction

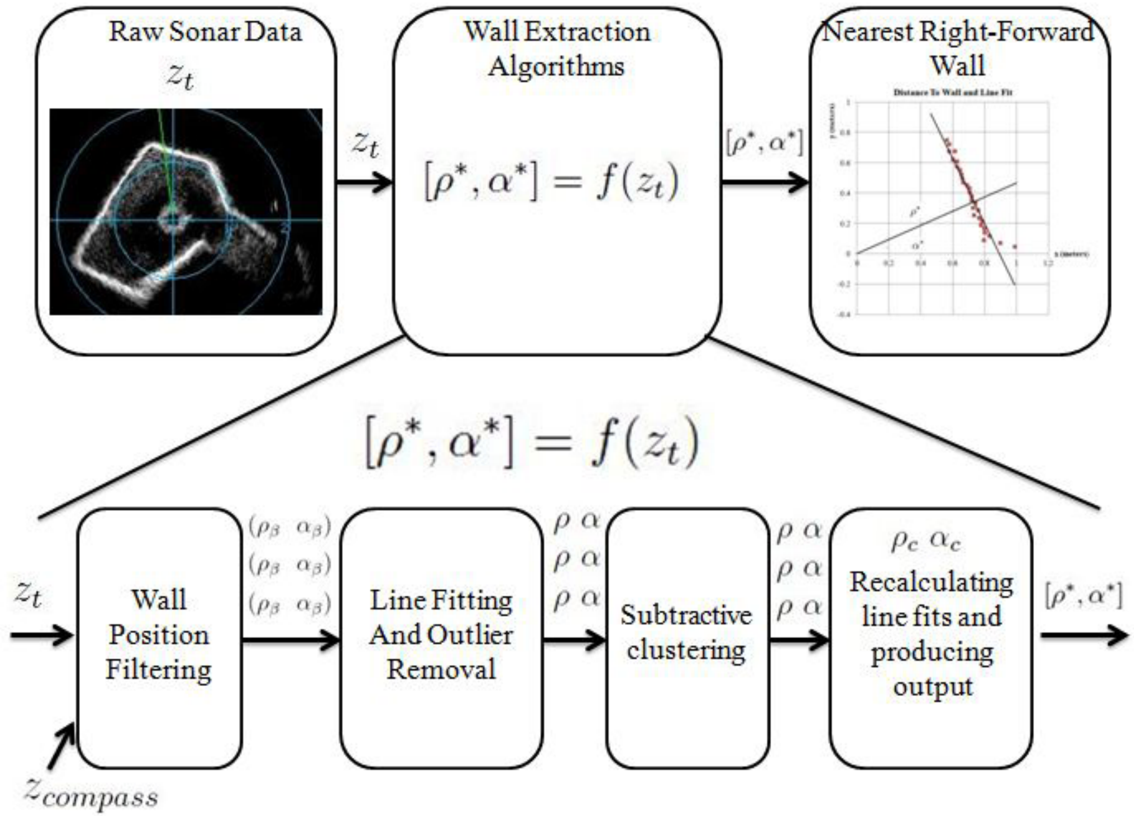
- Examples – Underwater Wall Mapping





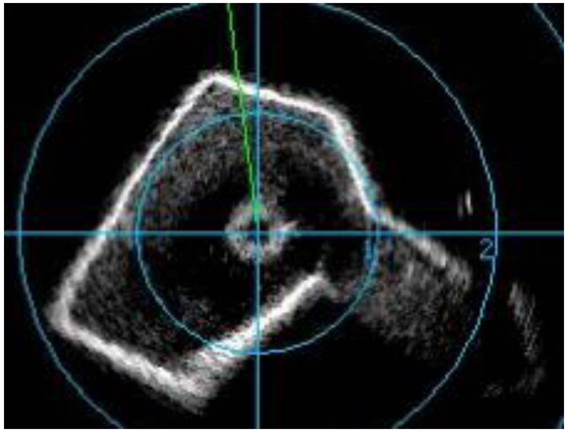
Line Extraction

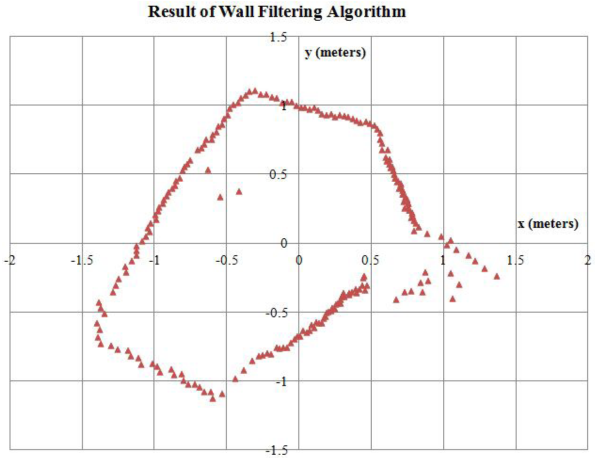
- Examples – Underwater Wall Mapping

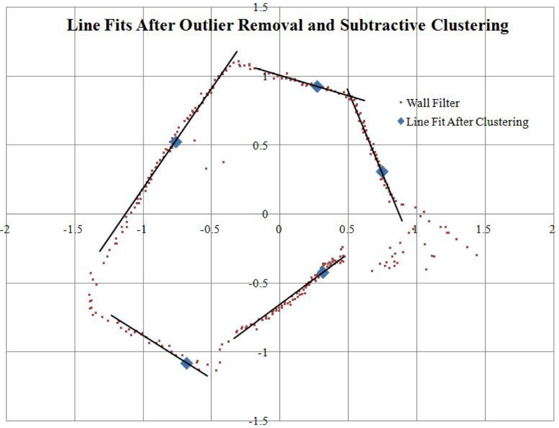


Line Extraction

- Examples – Underwater Wall Mapping







Outline – Mapping

1. Wall as Lines
    1. Line Extraction
    2. Segmentation
        - Split and Merge
        - Split and Merge – Fixed Endpoint
        - RANSAC
2. Walls as Grid Cells
    1. Evidence Grid
    2. Log Likelihood

Segmentation

- Split and Merge
  - Recursive procedure of fitting and splitting

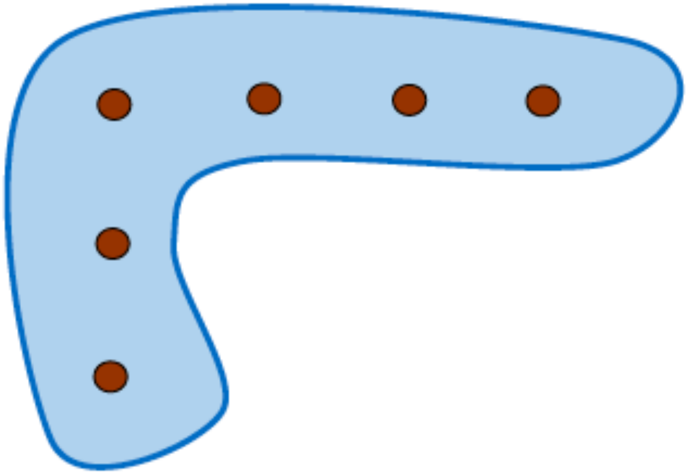
Initialise set **S** to contain all points

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

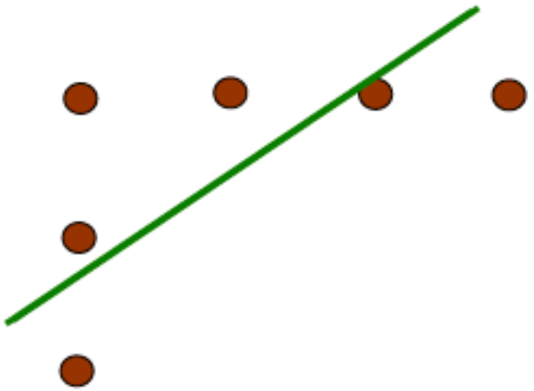
Initialise set **S** to contain all points

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

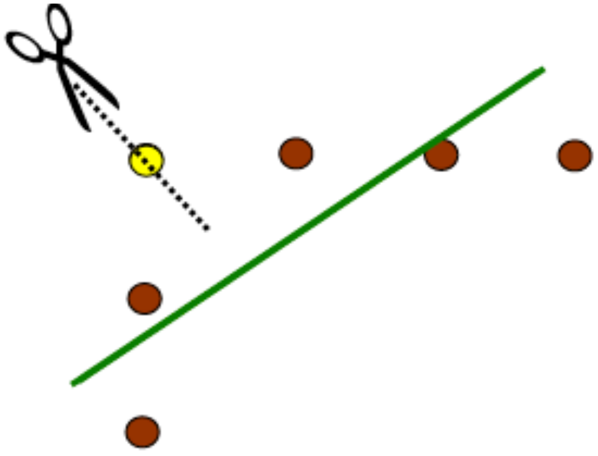
Initialise set **S** to contain all points

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

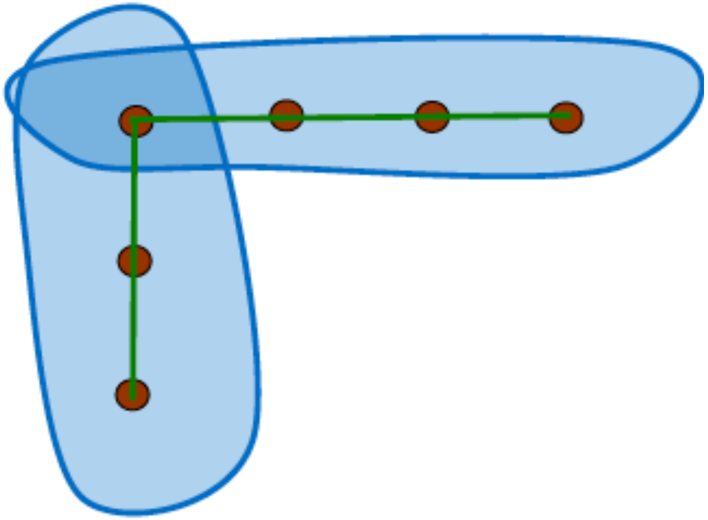
Initialise set **S** to contain all points

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

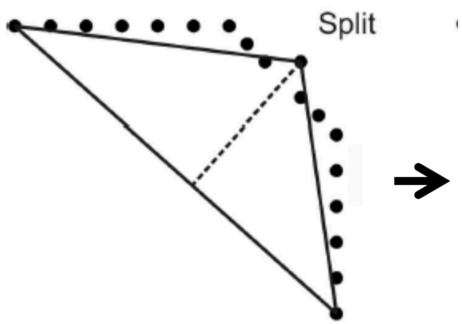


Outline – Mapping

1. Wall as Lines
   1. Line Extraction
   2. Segmentation
      - Split and Merge
      - Split and Merge – Fixed Endpoint
      - RANSAC
2. Walls as Grid Cells
   1. Evidence Grid
   2. Log Likelihood

Segmentation

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



→ → →

Outline – Mapping

1. Wall as Lines
    1. Line Extraction
    2. Segmentation
        - Split and Merge
        - Split and Merge – Fixed Endpoint
        - RANSAC
2. Walls as Grid Cells
    1. Evidence Grid
    2. Log Likelihood

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

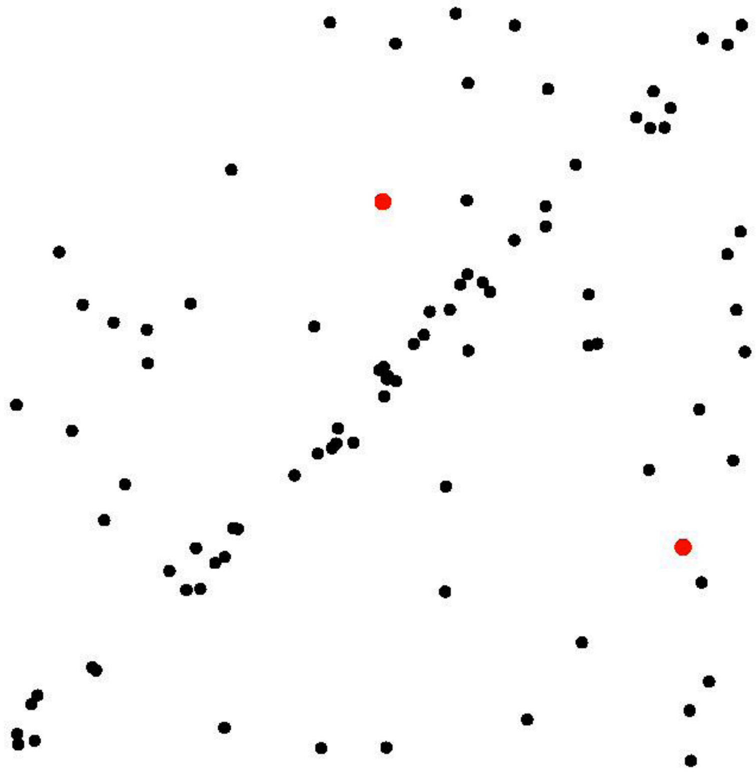
- RANSAC
  - RANSAC is an **iterative** method and is **non-deterministic** in that the probability to find a set free of outliers increases as more iterations are used
  - Drawback: A nondeterministic method, results are different between runs.

Segmentation

- RANSAC Example

Segmentation

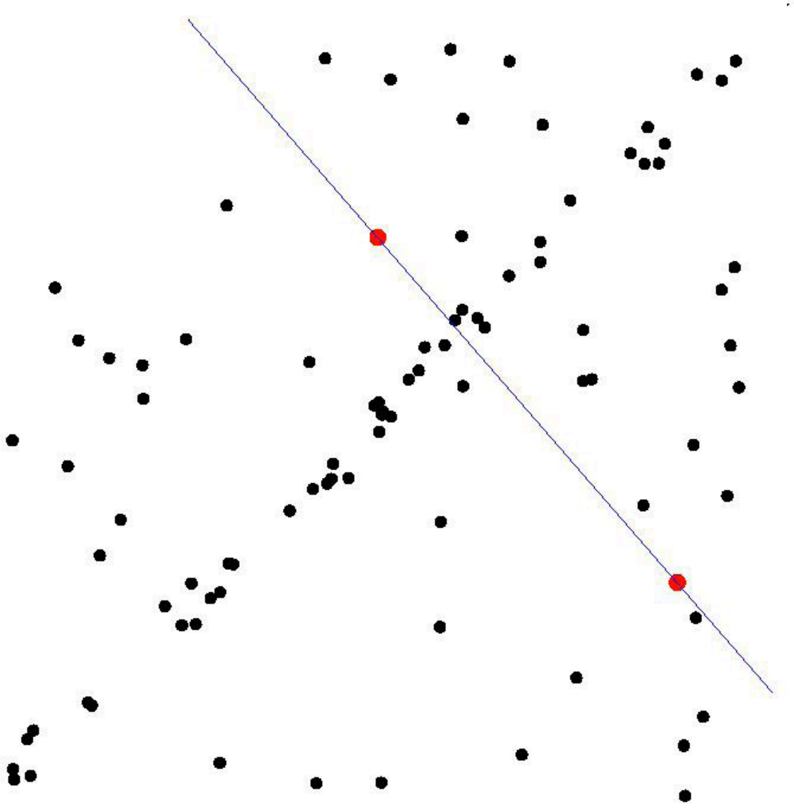
- RANSAC Example



- **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

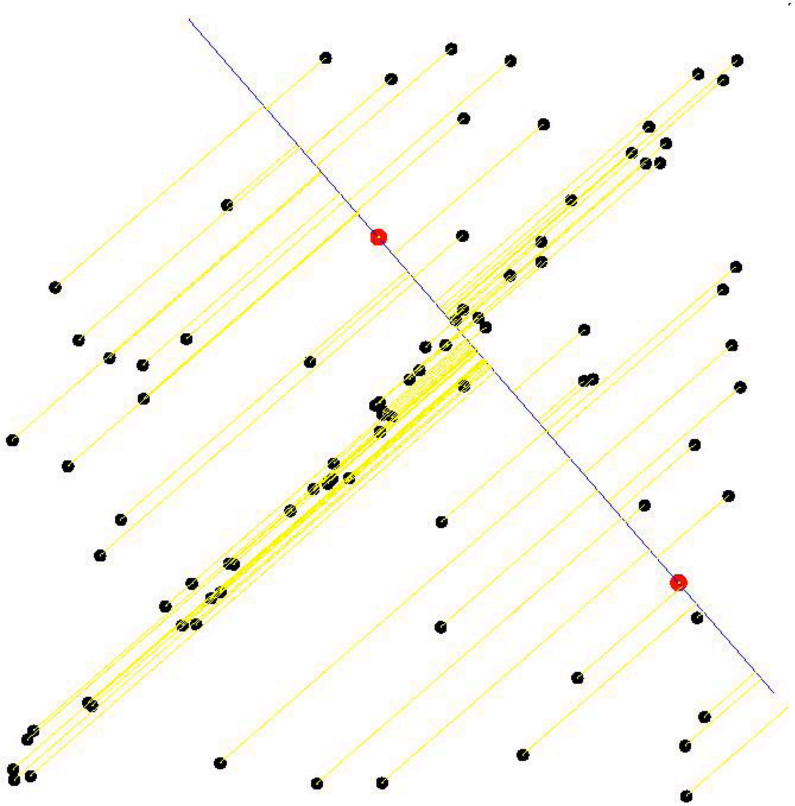
- RANSAC Example



- 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

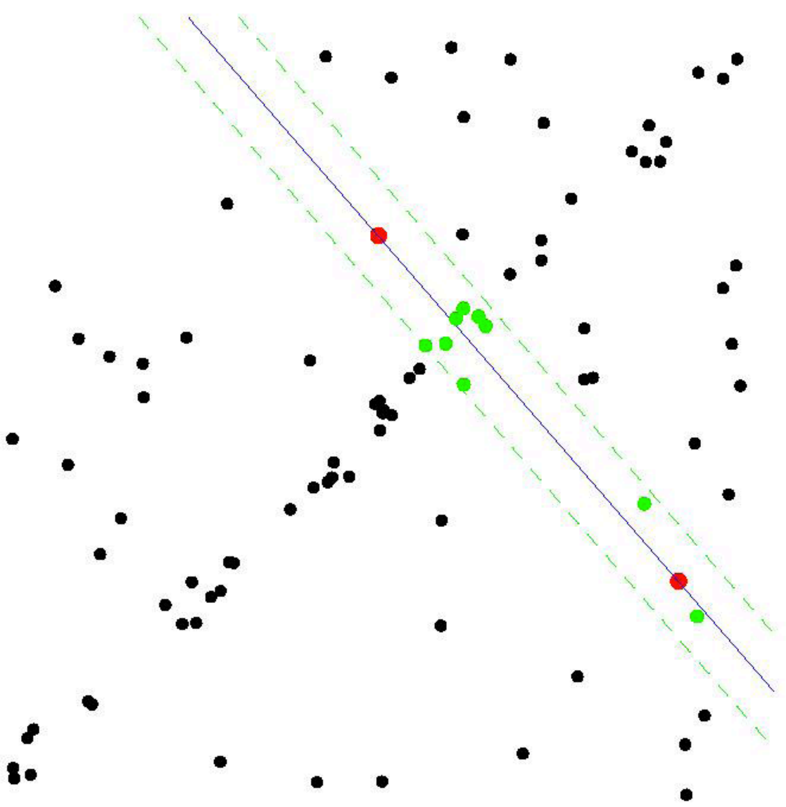
- RANSAC Example



- 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

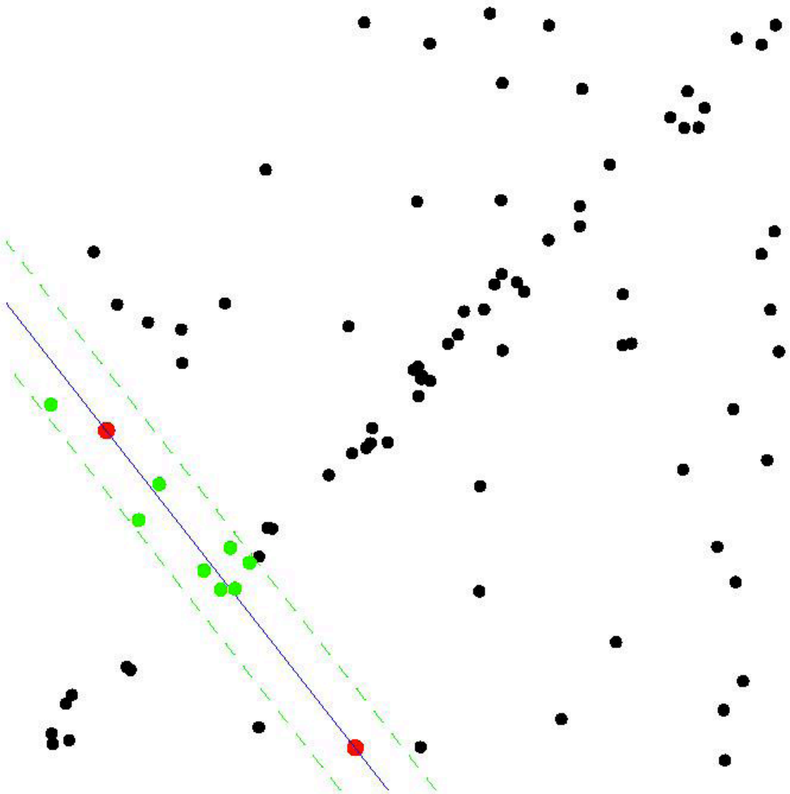
- RANSAC Example



- 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

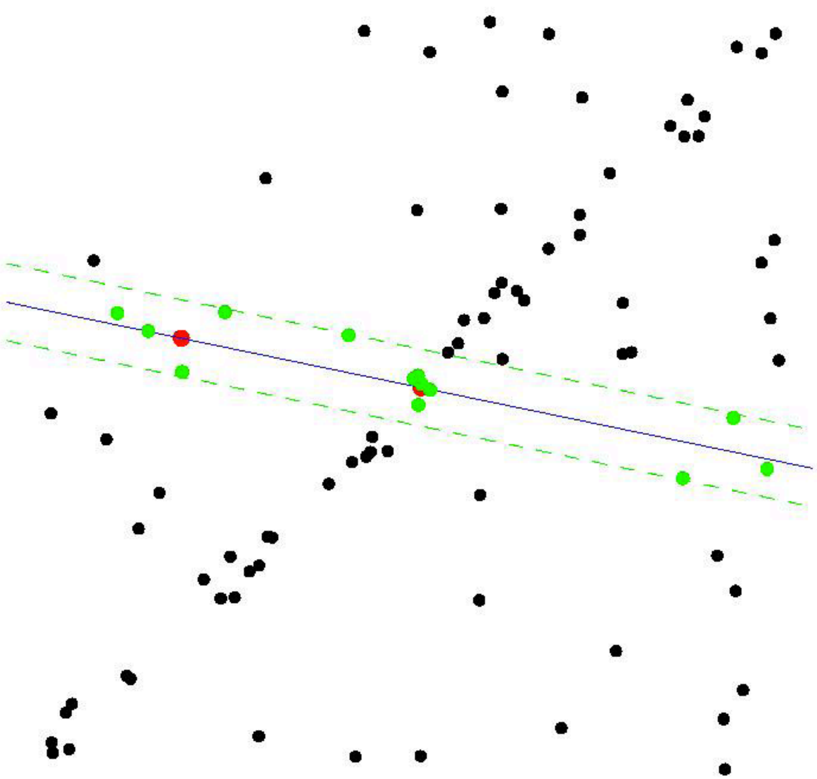
- RANSAC Example



- 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

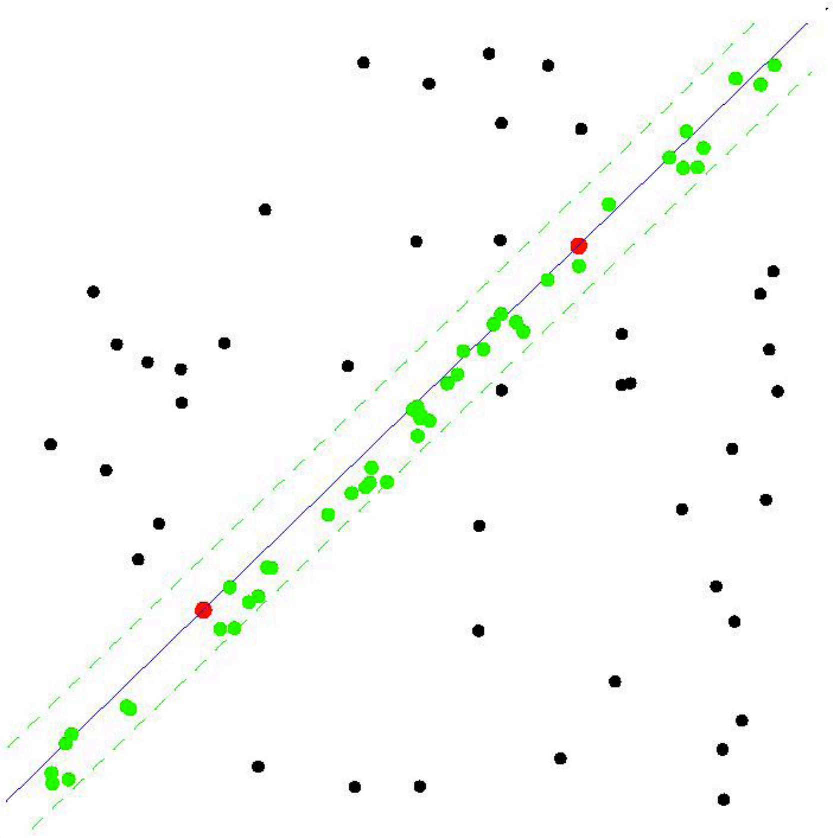
- RANSAC Example



- 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

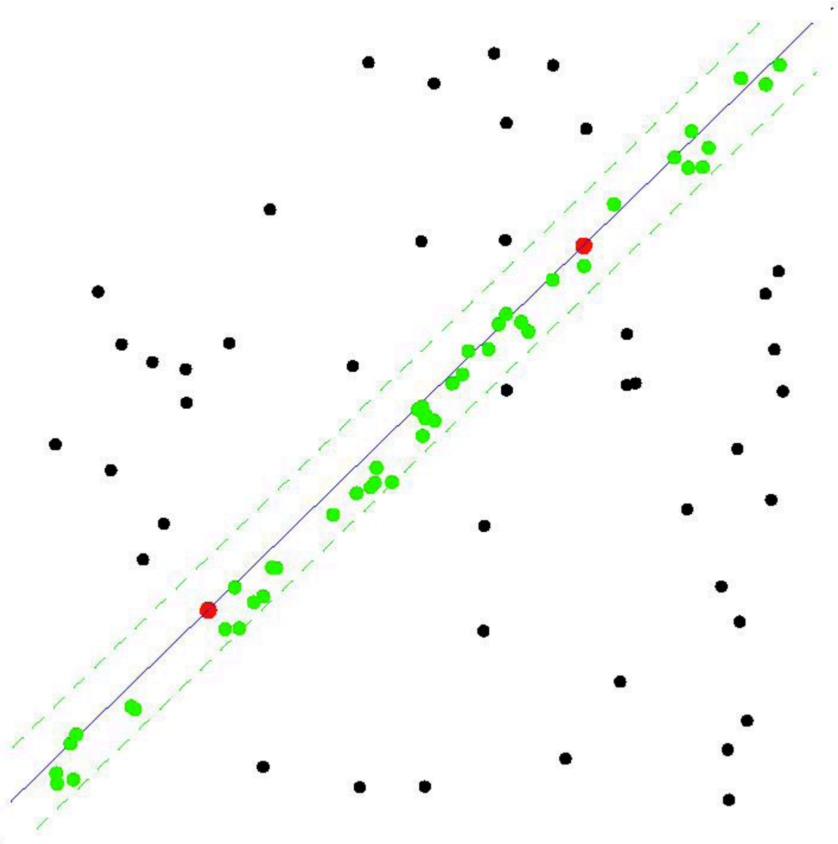
- RANSAC Example



- 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



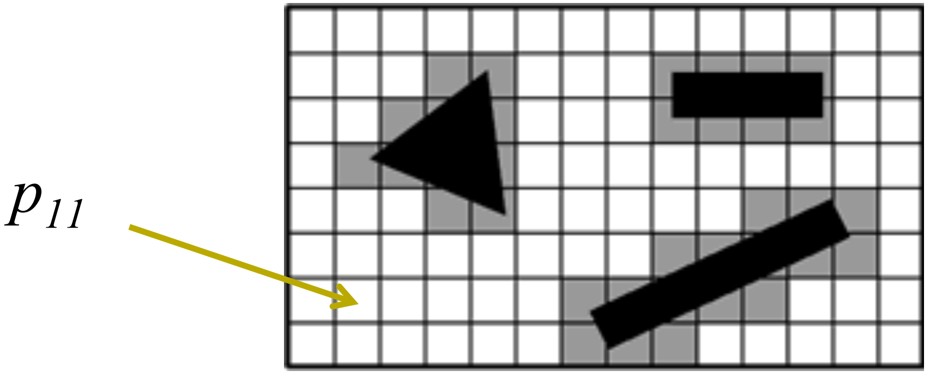
- Stop after k iterations and select model with the max number of inliers.

Outline – Mapping

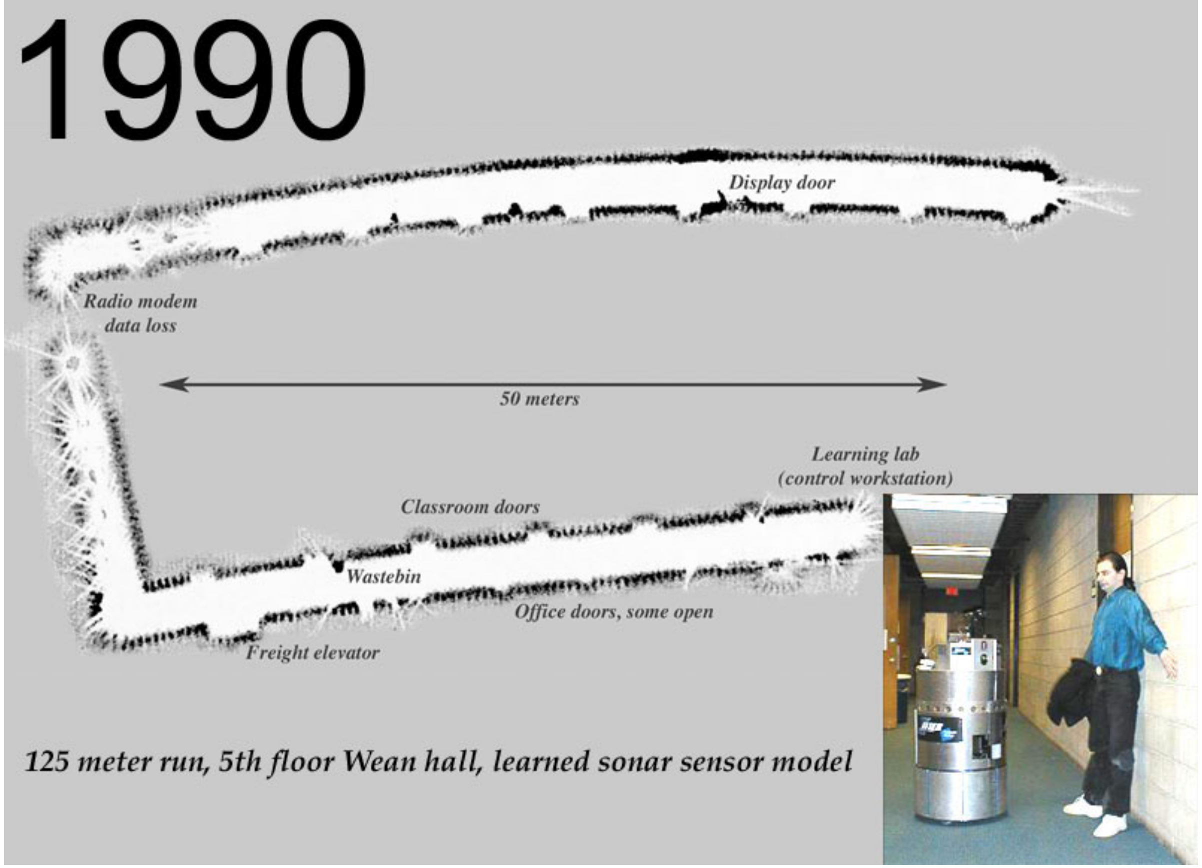
1. Wall as Lines
    1. Line Extraction
    2. Segmentation
        - Split and Merge
        - Split and Merge – Fixed Endpoint
        - RANSAC
2. Walls as Grid Cells
    1. Evidence Grid
    2. Log Likelihood

Walls as Grid Cells

- Evidence Grids
  - AKA Occupancy Grids
  - Workspace is discritized into grid cells
  - Each grid cell is assigned a likelihood of occupation $p_{ij} \in [0,1]$



Walls as Grid Cells



www.frc.ri.cum/~hpm/talks/cevo.slides/seeqrid.html

Walls as Grid Cells

www.frc.ri.cum/~hpm/talks/cevo.slides/seeqrid.html

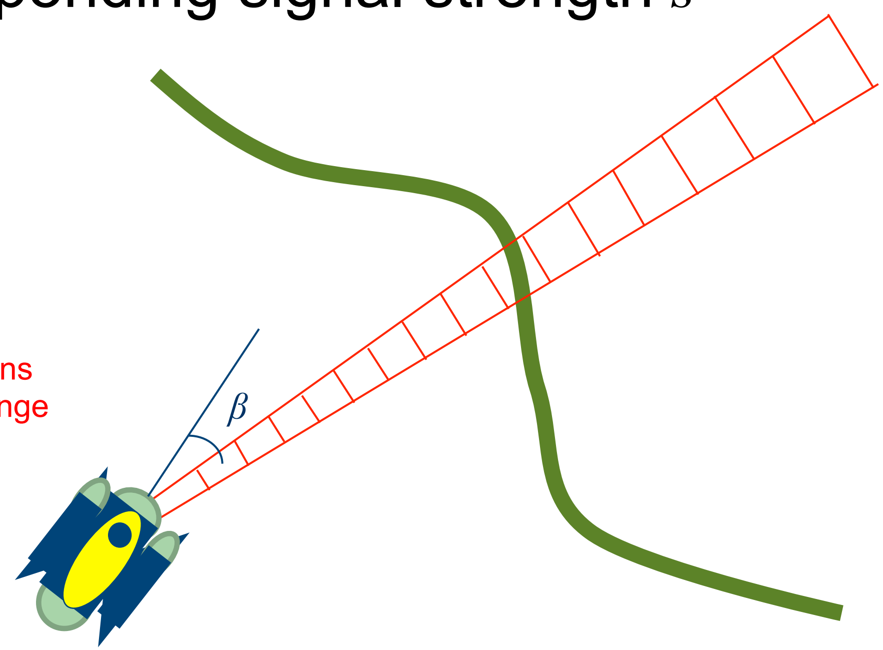
Walls as Grid Cells

- Updating with a Sensor Model (example)
  - For a maximum range R, there are B range values - each with a corresponding signal strength s^i

$$z = [\ \beta\ s^0\ s^1\ \dots\ s^B]$$

sonar angle

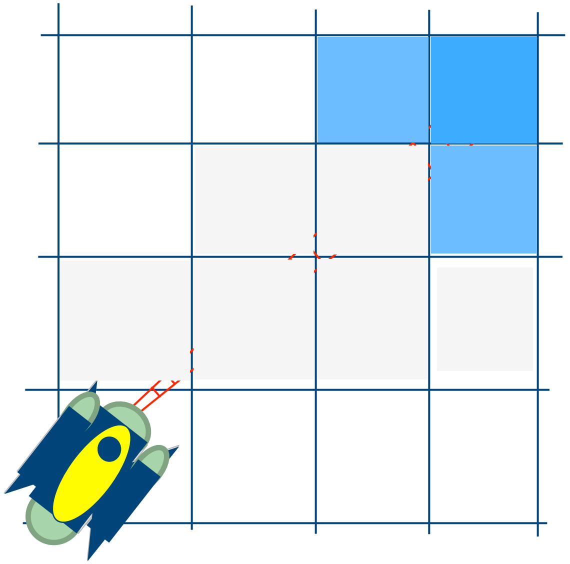
Strength of returns for increasing range



Walls a Grid Cells

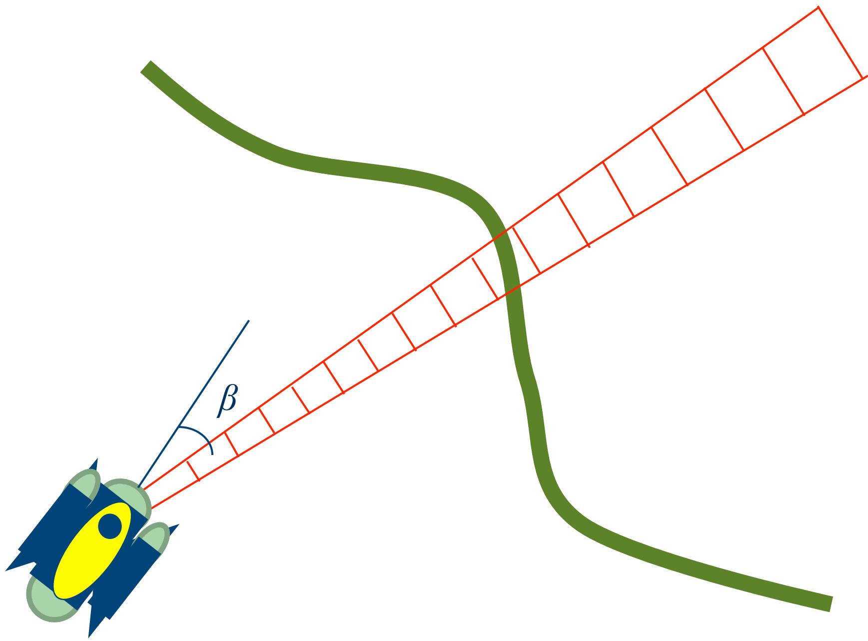
- Updating the Grid
  - Using geometry, the corresponding grid cell for each each sonar sensor bin must be determined.
  - Several bins could correspond with a single grid cell

    OR
  - Several grid cells could correspond with a single bin



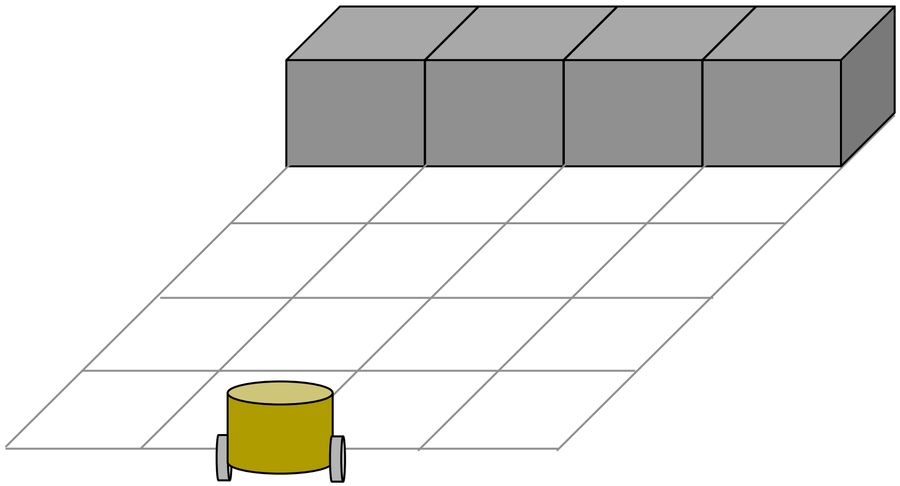
Walls as Grid Cells

- Using a Sensor Model
  - Each signal strength s^i must correspond to a likelihood of a occupancy $P(c_{ij} | z)$ in the map
  - We use a function $P(z | c_{ij})$ that must be determined experimentally.



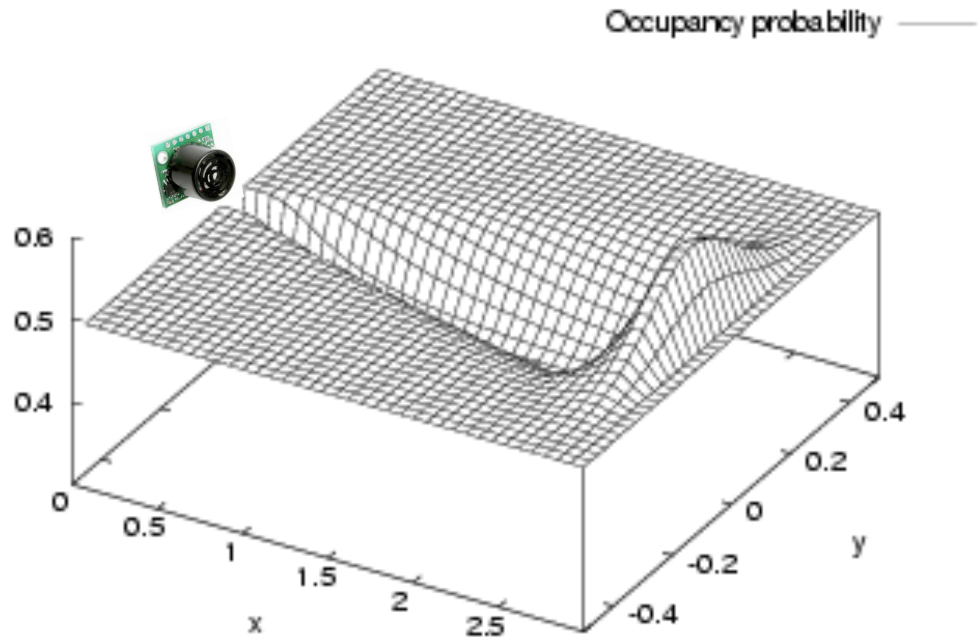
Walls a Grid Cells

- Updating the Grid
  - How do we get $P(z_t|c_{ij})$?
  - Experiments…



Walls as Grid Cells

- Using a Sensor Model
  - More sophisticated models are available for $P(z \mid c_{ij})$



Walls a Grid Cells

- Updating the Grid
  - Use Baye's rule to update each cell c_{ij}'s likelihood of occupancy for measurement z at time step t

$$P(c_{ij,t}) = P(c_{ij,t}|z_t) = \frac{P(z_t|c_{ij,t-1})P(c_{ij,t-1})}{P(z_t)}$$

$P(c_{ij,t})$ *=probability cell ij is occupied at time t*

$P(z_t)$ *=probability of obtaining measurement Z at time t*

$P(z_t|c_{ij,t-1})$*=probability of Z given* o_{ij} *from the sensor model*

Walls a Grid Cells

- Updating the Grid
  - Similarly

$$P(-c_{ij,t}|z_t) = \frac{P(z_t|-c_{ij,t-1})P(-c_{ij,t-1})}{P(z_t)}$$

Walls a Grid Cells

- Updating the Grid
  - Now, the odds o of some fact A being true can be written as

$$o(A) = P(A)/P(-A)$$

  - In our case

$$o(c_{ij,t}|z_t) = P((c_{ij,t}|z_t)/P(-c_{ij,t}|z_t)$$

$$= \frac{P(z_t|c_{ij,t-1})P(c_{ij,t-1})}{P(z_t|-c_{ij,t-1})P(-c_{ij,t-1})}$$

$$= o(z_t|c_{ij,t-1})o(c_{ij,t-1})$$

Walls a Grid Cells

- Updating the Grid
  - What if we take the **log odds**

$$\log o(c_{ij,t}|z_t) = \log o(z_t|c_{ij,t-1}) + \log o(c_{ij,t-1})$$

  - Characteristics
    - The last term is equated to previous log odds of $\log o(c_{ij,t-1}|z_{t-1})$
    - No need for knowledge of $P(z)$
    - Updates can be done with **addition**, not multiplication

Walls a Grid Cells

- Updating the Grid
  - Properties of log odds

$$\gamma(p) = logit(p)$$
$$= \log\,(p/(1\text{-}p))$$
$$= \log(p) - \log(1\text{-}p)$$

  - Most often the natural logarithm is used

$$\gamma(p) = \ln(p) - \ln(1\text{-}p)$$

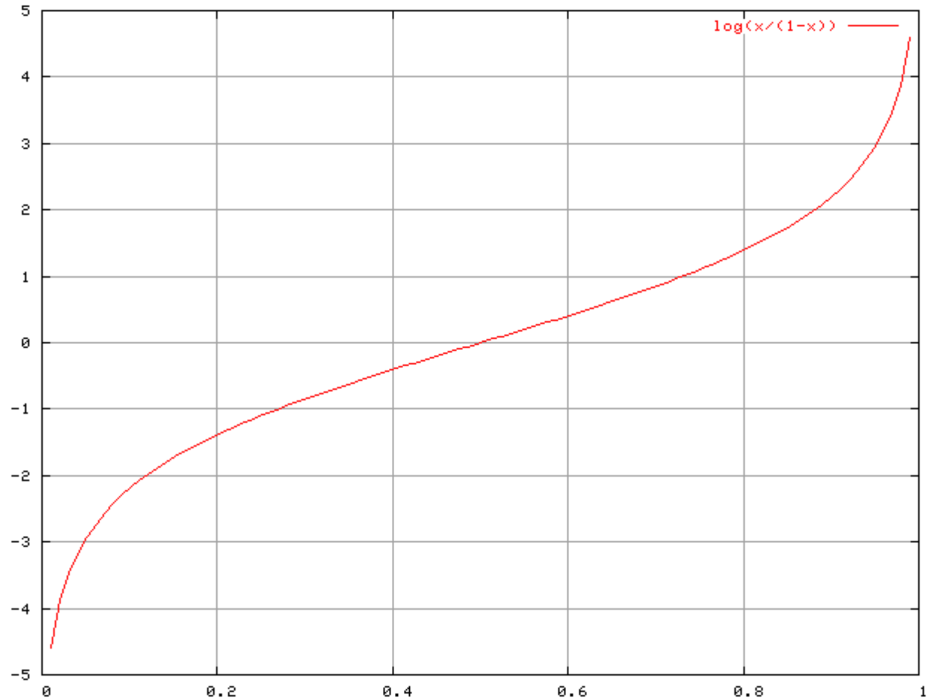
Walls a Grid Cells

- Updating the Grid
  - The *logit()* function

$\gamma(p)$ =

=

=



en.wikipedia.org

Walls a Grid Cells

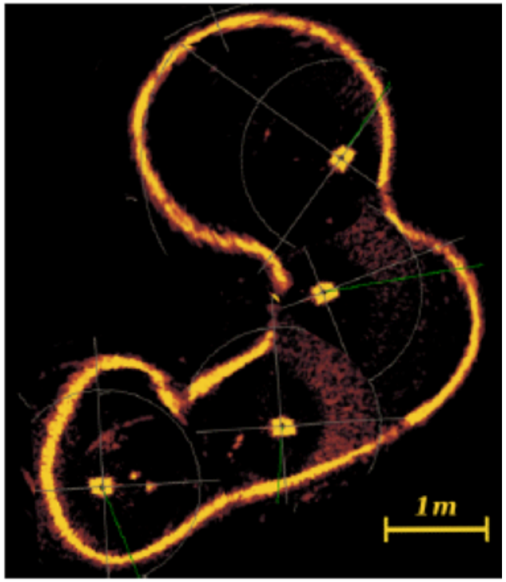
- Updating the Grid
  - The $logit^{-1}()$ function

$$p(\gamma) = logit^{-1}(\gamma)$$
$$= exp(\gamma) / (1+exp(\gamma))$$

en.wikipedia.org

Walls as Grid Cells

- Application Example



(a) Cistern sonar mosaic