Autonomous Mobile Robots

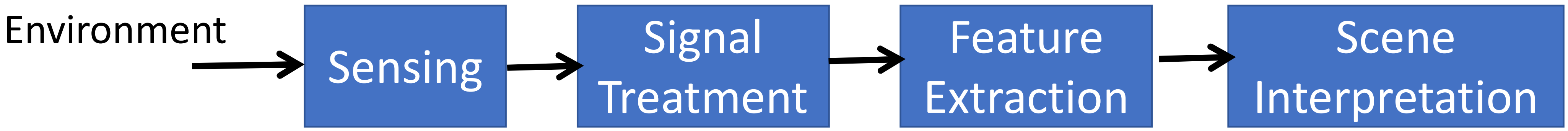
Module 18: Feature extraction based on range data

Feature extraction

- Sensors are imperfect, measurements will have errors and uncertainty
- Sensor inputs must be used in such a way that robot should be able to operate inspite of the uncertainty
- Uncertain sensors inputs can be used in two ways
  - Use measurement as a raw and individual value
  - Extract higher-level information from one or more sensor readings
- This second approach is called as feature-extraction
- Ex: sensor in front of a wall object

Feature extraction

- Feature extraction need not always be used by robots for every activity
- Ex: If the robot is in front of an object, it needs to make an emergency stop, it is better to use a raw sensor reading and act in this case
- Ex: For map-building and precise navigation, range sensor and vision sensor measurements may go through the entire perceptual pipeline to minimize individual sensor uncertainty



Feature extraction - Low and High Level

- Definition of a feature: Features are recognizable elements in the environment
- They can be extracted from measurements and can be mathematically described
- Good features are easily detectable from the environment
- Low-level features (or geometric primitives) are things like lines, circles, polygons
- High-level features (objects) are things like edges, doors, tables

Feature extraction - Volume of data

- Raw sensor readings provide a large volume of data
- All information is conserved
- Low-level features provide a lower volume of data
- Expectation when using low level features is that the poor or useless data is filtered out
- But there is also a chance that some useful information may be lost

Feature extraction - Volume of data

- High level features provide the least volume of data
- But there is much higher risk of filtering away important information from the data
- In general, we may need to keep some amount of raw data and low and high-level features to interact with the environment in an optimal manner
- Features have some spatial locality. Ex: corner or edge is present at a particular location in the world
- Features need to be chosen in an appropriate manner based on the environment that the robot is working in

Factors that determine choice of features

1. Target Environment: Target geometries must be readily detected in actual environment

- Line features are really useful in house, office, building environments, but do not have any use if the robot is being used in Mars

2. Available sensors: The sensor that is being used and its uncertainty determines if the feature can be extracted from it

- A laser rangefinder can produce data at very good angular and depth resolution. A sonar, in comparison, does not have the same resolution
- Hence if corner feature is to be extracted, it is better done with laser rangefinder

Factors that determine choice of features

3. Computational Power: Vision-based feature extraction can be computationally heavy, requiring powerful processors

4. Environment Representation: Feature extraction is a preceding step for scene interpretation

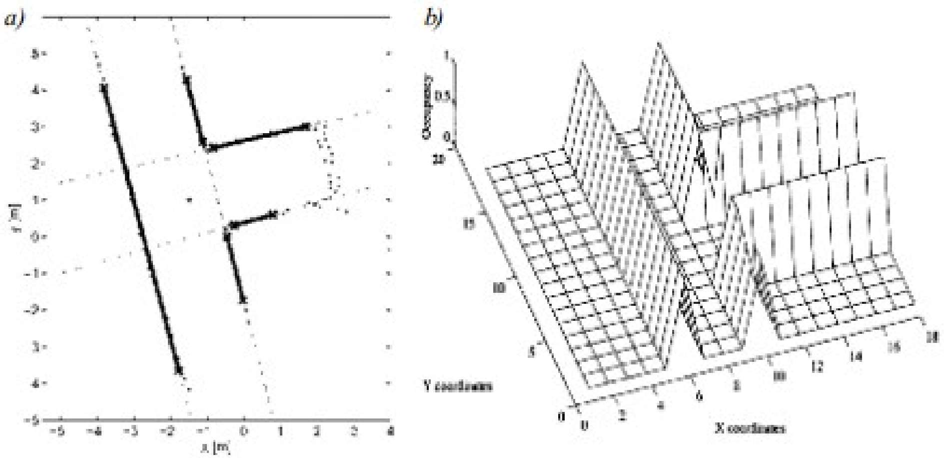


Figure 4.35
Environment representation and modeling: (a) feature based (continuous metric); (b) occupancy grid (discrete metric). Courtesy of Sjur Vestli.

Reproduced from: Siegwart, Roland, Illah Reza Nourbakhsh, and Davide Scaramuzza. Introduction to autonomous mobile robots. MIT press, 2011.

Factors that determine choice of features

4. Environment Representation (contd.):

- Environment can be represented by a feature-based model or an occupancy grid model
- Ex: model of an office building hallway
- Extraction of line and corner features is more relevant for representation on left
- Occupancy-grid model is more relevant for obstacle avoidance

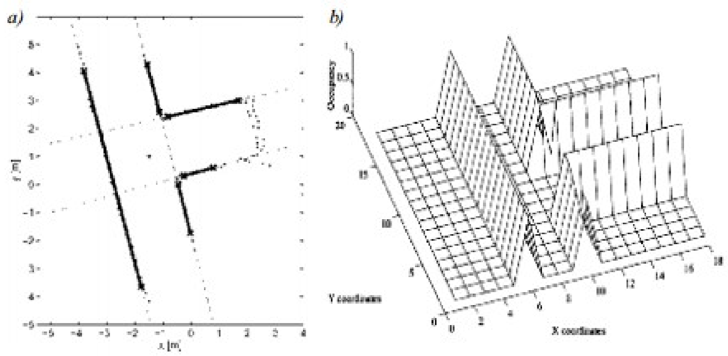


Figure 4.35
Environment representation and modeling: (a) feature based (continuous metric); (b) occupancy grid (discrete metric). Courtesy of Sjur Vestli.

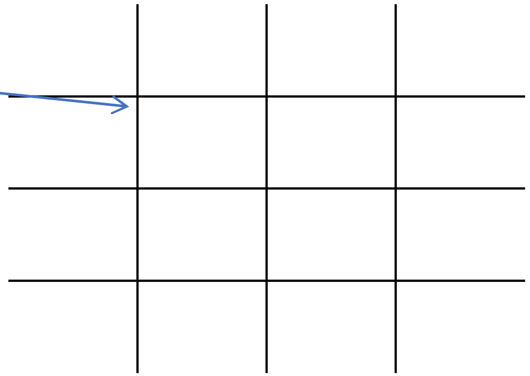
Reproduced from: Siegwart, Roland, Illah Reza Nourbakhsh, and Davide Scaramuzza. Introduction to autonomous mobile robots. MIT press, 2011.

At every grid point:

value 0 or 1 is stored

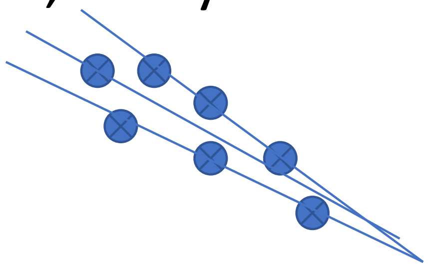
0 - unoccupied

1 - occupied



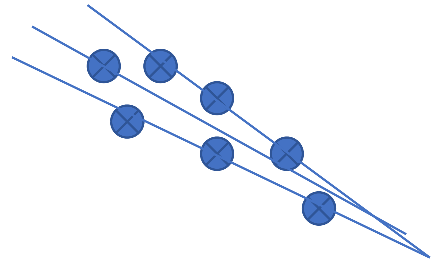
Feature extraction based on range data

- Line extraction: we are trying to determine if a line is present in the data provided by a rangefinder
- Features that are of interest for extraction here are geometric features (and not color for example)
- Data obtained from sensor should be compared and matched against a predefined template
- Line extraction is done based on Least Squares Estimation, which is an optimization-based approach
- Given a set of points, many lines can be drawn through the set of points

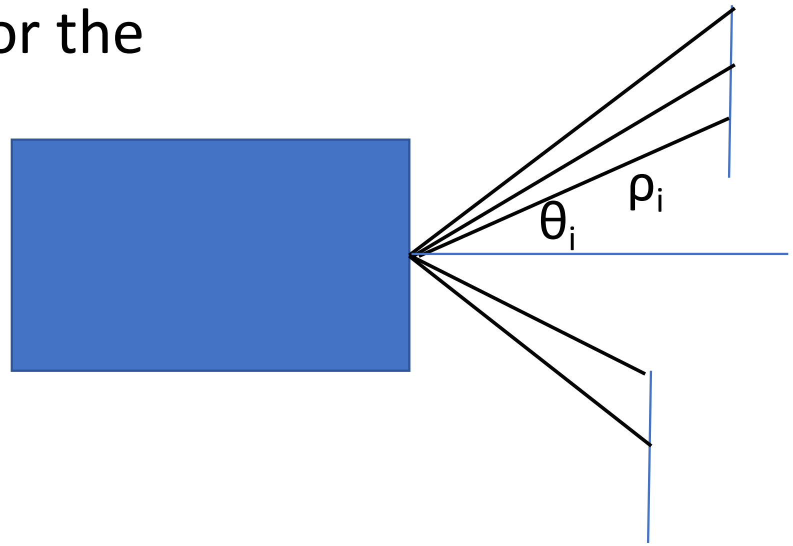


Feature extraction based on range data

- **Objective:** Probabilistic Line extraction from uncertain range sensor data



- Suppose we have *n* range measurement points.
- We represent the ranging measurement points in polar coordinates as $x_i = (\rho_i, \theta_i)$
- Choice of these coordinates is more intuitive for the readings measured by a sensor on a robot
- We can represent the *x* and *y* coordinates of a point using the polar coordinate representation

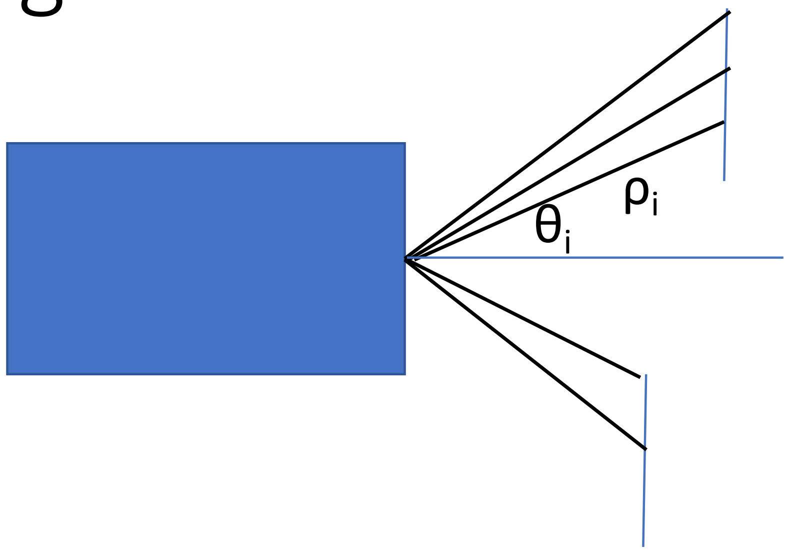


- x-coordinate : $\rho_i \cos\theta_i$,
- y-coordinate : $\rho_i \sin\theta_i$

Feature extraction based on range data

Example Readings from sensor

pointing angle of sensor θ_i [deg]	range ρ_i [m]
0	0.5197
5	0.4404
10	0.4850
15	0.4222
20	0.4132
25	0.4371
30	0.3912
35	0.3949
40	0.3919
45	0.4276

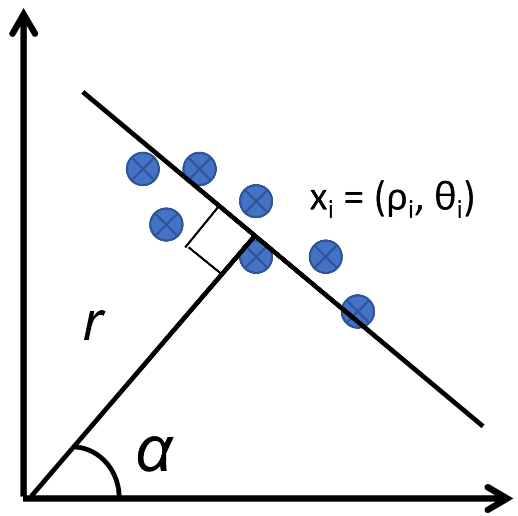


Feature extraction based on range data

- We have *n* range measurement points $x_i = (\rho_i, \theta_i)$
- There is uncertainty associated with each measurement made (in both ρ_i and θ_i)
- A typical line equation is $y = mx+c$
- A line can alternatively be represented as

$$x\cos\alpha + y\sin\alpha = r$$

- Instead of parameters (m,c) we make use of (α, r)
- (α, r) are used to represent the line of best-fit



- Refer Class Notes for further equations