Robots and People

Vlad Estivill-Castro (2016)

Robots for People

--- A project for intelligent integrated systems

Uncertainty representation

Localization

Chapter 5 (textbook)

What is the course about?

- textbook
- **Introduction to Autonomous Mobile Robots**
- second edition

Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza

Localization

- Where are we?
  - Crucial for mobile robotics
  - What is the best action depends clearly on where we are
- When do we need to localize?
  - Not always
    - Shoot if in possession of the ball and the opponent goal is on sight [Robotic soccer, color-coded teams for goals]
    - Walk keeping a wall to the left [can recognize target; like the exit of a maze], but slow
      - Behavior-based navegation

Localization (role in navigation)

- May be necessary to avoid obstacles
- To get from A to B (B is recognized by localization)
- In contrast to behavior based navigation, ***map based navigation*** relies on a map
  - Assuming that the map is known, at every time step the robot has to localize itself in the map.
    - How?
      - If we know the start position, we can use wheel odometry or dead reckoning.
      - Is this enough?
      - What else can we use?
        - But how do we represent the map for the robot
        - And how do we represent the position of the robot in the map?

Issues in Localization (and navigation)

- Types of localization and examples of localization systems
- Noise: odometric position estimation
- Belief representation: how to represent the robot position
- Map representation: continuous, discrete, topological
- Introduction to probabilistic map based localization

Localization

- Global localization
  - The robot is not told its initial position
  - Its position must be estimated from scratch
- Position Tracking
  - A robot knows its initial position and "only" has to accommodate small errors in its odometry as it moves

How to localize

- Localization based on external sensors, beacons or landmarks
- Odometry
- Map Based Localization
  - without external sensors or artificial landmarks, just use robot onboard sensors
  - Example: Probabilistic Map Based Localization

Beacon Based Localization Systems: Triangulation

- Ex 1: Poles with highly reflective surface and a laser for detecting them
- Ex 2: Coloured beacons and an omnidirectional camera for detecting them
  - (example: robocup or autonomous robots in tennis fields)

Beacon based localization

- **Warehouse Robots at Work**
- http://www.youtube.com/watch?v=lWsMdN7HMuA
- **Brisbane container terminal simulation**
- http://www.youtube.com/watch?v=0xhECD3UDjI

Localization Cycle

- Improving belief state by moving
- Improving belief state by sensing

- Cycle between (SEE and ACT)
  - While I was moving, I'm blind until the sensor measurement arrives
    - update my belief by a motion model
    - followed by an update by the sensor model

Human navigation

- Topological and conceptual maps
  - the dimensions of Australia, and what is close, north, south
  - number of buildings vs name of buildings
- Relations at different scales

The Map Categories

1. recognizable landmarks
2. topological maps (graph model)
   - games like snakes and ladders
3. metric topological maps
   - edges have weight equal to distance
4. Full metric maps
   - nodes are geo-positioned

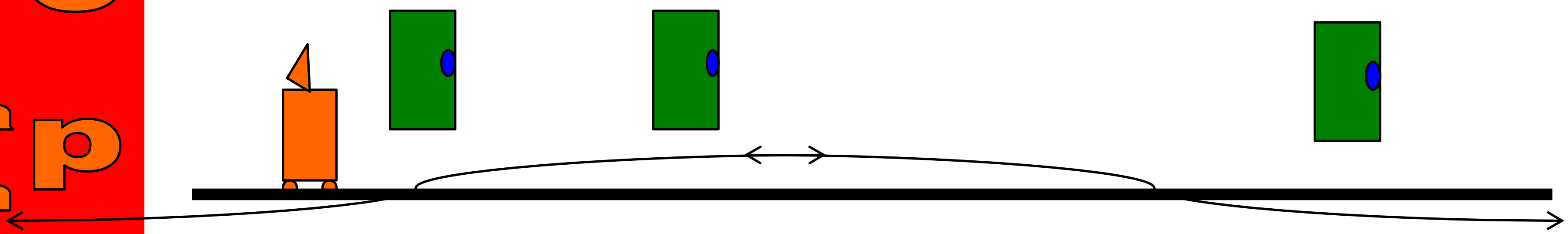
Understanding --- Probabilistic reasoning (Bayesian inference)

- Reasoning in the presence of uncertainties and incomplete information
- Combining preliminary information and models with learning from observed data
- $P(A|B) = P(B|A)\,P(A)/P(B)$
  - the probability that the new situation is A once I observed B is related to the chances of noticing B when in situation A and the ratio of observing A with the chances of observing B

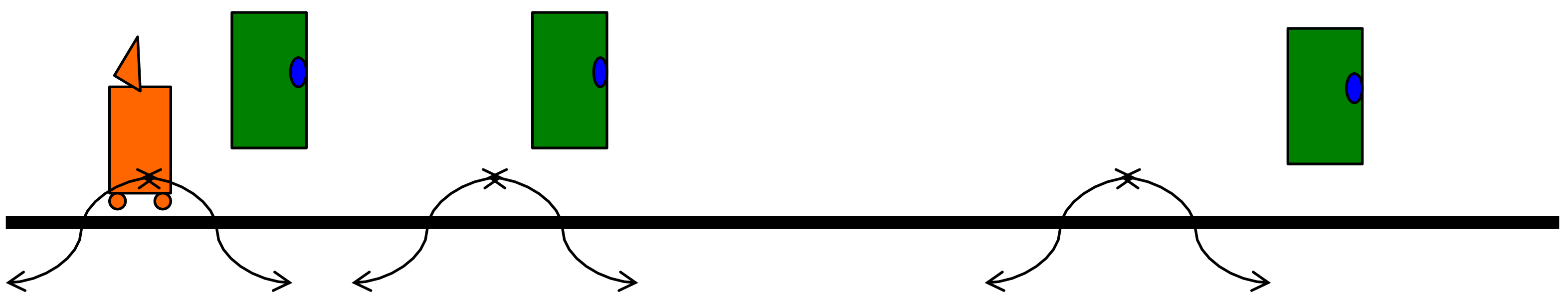
Reducing Uncertainties

- Where can I be?

- Anywhere



- Observe a door

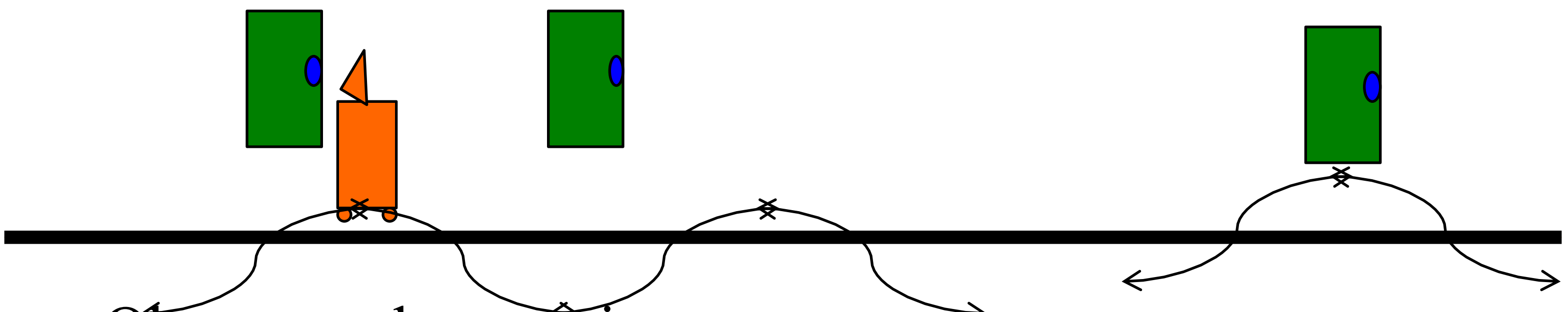


Robots and People

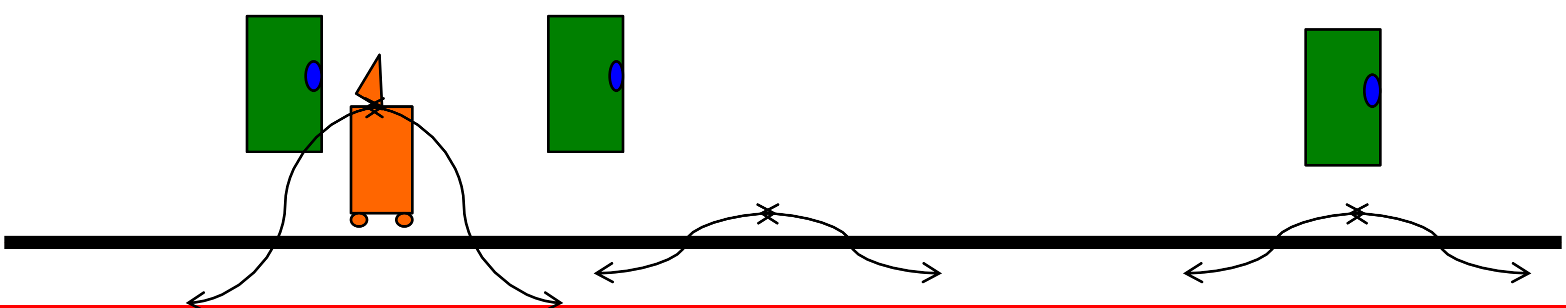
Reducing Uncertainties

- Observe a door

- Walk a bit



- Observe a door again



Robots and people

Metric Navigation: *Probabilistic Position Estimation (Kalman Filter)*

- Continuous, recursive and very compact
  - ACT PHASE
    - state prediction via a motion model
      - obtain $x(k+1 \mid k)$ and $P(k+1 \mid k)$ given $x(k|k)$ $P(k|k)$
  - SEE PHASE
    - Balance the advice of where I am (result of the motion model) and the advice of where I am (result of the sensor)
    - Depending on how much I trust each

State prediction: odometry

- Incrementally (dead reckoning / open loop control)
    - There is a discrepancy into the real trajectory and the belief of where I have been
    - DRIFT
        - cumulative error
- Under no error, current position is accurate

Methods for Localization

Quantitative Metric Approach

1. A priori, know your environment
   - in graph, map
2. Extract features (line segments, landmarks)
3. Match them with location in map
4. Estimate your position
   - Kalman filter, Markov filters (particle filters)
   - Balance observations with your belief
   - issues:
     - estimation of uncertainties
     - weights to prior statistics

Challenges of Localization

- Knowing the absolute position (e.g. GPS) is not sufficient
- Localization in human-scale in relation with environment
- Planning in the *Cognition* step requires more than only position as input
- Perception and motion plays an important role
  - Exteroceptive sensor noise
  - Effector noise
  - Odometric position estimation

Robots and people

Exteroceptive Sensor Noise

- Sensor noise is mainly influenced by environment e.g. surface, illumination …
- and by the measurement principle itself e.g. interference between ultrasonic sensors
- Sensor noise drastically reduces the useful information of sensor readings. The solution is:
  - to model sensor noise appropriately
  - to take multiple readings into account
  - employ temporal and/or multi-sensor fusion

Effector Noise: Odometry, Deduced Reckoning

- Odometry and dead reckoning:
  Position update is based on proprioceptive sensors
  - Odometry: wheel sensors only
  - Dead reckoning: also heading sensors
- The movement of the robot, sensed with wheel encoders and/or heading sensors is integrated to the position.
  - Pros: Straight forward, easy
  - Cons: Errors are integrated -> unbound
- Using additional heading sensors (e.g. gyroscope) might help to reduce the cumulated errors, but the main problems remain the same

Derivation of Error Models for Motion

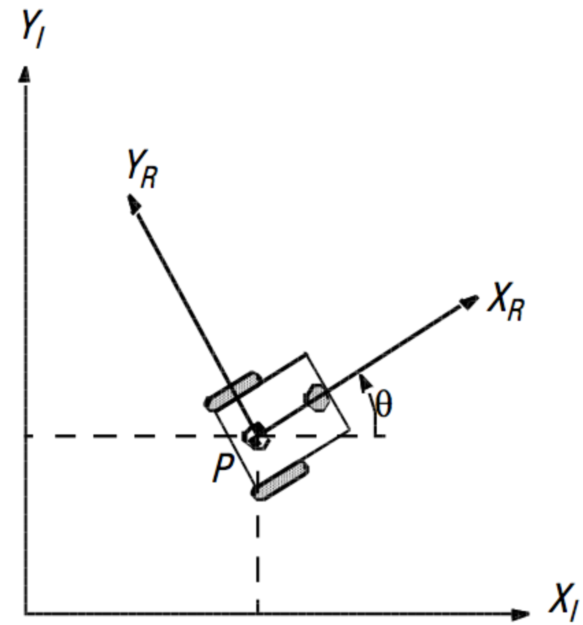
An analytical model

We want some formula/algorithms that tells me when the robot moves, where will it end up (and estimate the error/confidence of this new position)

Robots and People

Odometry: Example, the differential robot

- If we are in a point $p=(x,y,\theta)$ and we move a little bit to $p'=p+(\Delta x,\Delta y,\Delta\theta)$
  - Can we build an analytical model of where the robot is?
  - p' and the error
  - Based on measurements
    - Of the robot



Differential wheel robot (Lect 03)

- φ_1 speed right wheel, φ_2 sped of left wheel; *r* diameter of wheels, *s* half separation of wheels
- $x_R = r\,\varphi_1/2 + r\,\varphi_2/2$
- $y_R = 0$
- Change in rotational velocity ω_1 when only forward spin of right wheel
  - robot moves clockwise $\omega_1 = r\,\varphi_1/2s$
- *Change in rotational velocity ω_2 when only the left wheel moves forward*
  - *robot moves counterclockwise* $\omega_2 = - r\,\varphi_2/2s$
  - Total rotational velocity $r\,\varphi_1/2s - r\,\varphi_2/2s$

Robots and People

Differential wheel robot (adapt)

- φ_1 speed right wheel, φ_2 sped of left wheel; r diameter of wheels, b=2s separation of wheels
- $x_R = r\,\varphi_1/2 + r\,\varphi_2/2 = (r\,\varphi_1 + r\,\varphi_2)/2$
- $y_R = 0$
- Change in rotational velocity ω_1 when only forward spin of right wheel
  - robot moves clockwise $\omega_1 = r\,\varphi_1/2s$
- *Change in rotational velocity ω_2 when only the left wheel moves forward*
  - *robot moves counterclockwise* $\omega_2 = - r\,\varphi_2/2s$
  - Total rotational velocity $r\,\varphi_1/2s - r\,\varphi_2/2s = (r\,\varphi_1 - r\,\varphi_2)/b$

Some definitions

- $p=[x\ y\ \theta]^T$ the starting position
- Δx, displacement in the environment in x direction
- Δy, displacement in the environment in y direction
- $\Delta\theta$, change in spin / orientation in the environment
  - All after some small time interval Δt
- Δs_r, displacement in right wheel
- Δs_l, displacement in left wheel

Because diferential robot

- $\Delta\theta$, change in spin / orientation in the environment
  - All after some small time interval Δt
- Δs_r, displacement in right wheel
- Δs_l, displacement in left wheel
- $\Delta\theta = (\Delta s_r - \Delta s_l)/b$
- And if Δs is the displacement of the robot
  - $\Delta s = (\Delta s_r + \Delta s_l)/2$

- Remember b is the separation between the wheels

In the referential frame of the environment

- Δx, displacement in the environment in x direction
  - $\Delta x = \Delta s \cos(\theta + \Delta\theta/2) = \cos(\theta + \Delta\theta/2)(\Delta s_r + \Delta s_l)/2$
- Δy, displacement in the environment in y direction
  - $\Delta y = \Delta s \sin(\theta + \Delta\theta/2)$
- $\Delta\theta$, change in spin / orientation in the environment
  - $\Delta\theta = (\Delta s_r - \Delta s_l)/b$

Robots and People

Re-write the model

- $p' = p + (\Delta x, \Delta y, \Delta\theta)$

$p' = p + (\cos(\theta + \Delta\theta/2)(\Delta s_r + \Delta s_l)/2,$

$\sin(\theta + \Delta\theta/2)(\Delta s_r + \Delta s_l)/2,$

$(\Delta s_r - \Delta s_l)/b)$

$p' = (x, y, \theta) + (\cos(\theta + \Delta\theta/2)(\Delta s_r + \Delta s_l)/2,$

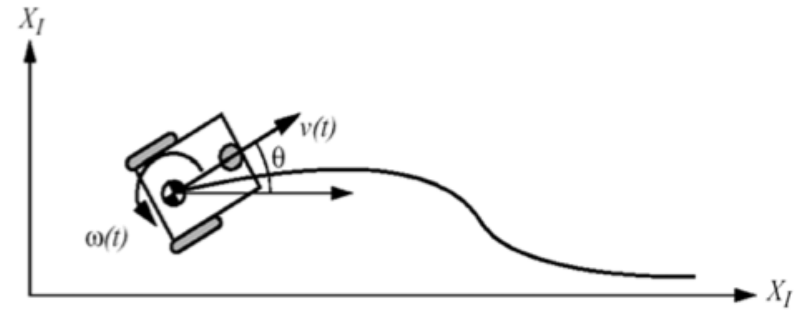
$\sin(\theta + \Delta\theta/2)(\Delta s_r + \Delta s_l)/2,$

$(\Delta s_r - \Delta s_l)/b)$

$p' = f(x, y, \theta, \Delta s_r, \Delta s_l)$

Odometry in the robots frame of reference

- b is the separation between the wheels
- Kinematics:



$$\Delta x = \Delta s \cos(\theta + \Delta\theta / 2)$$

$$\Delta y = \Delta s \sin(\theta + \Delta\theta / 2)$$

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b}$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2}$$

$$p' = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$

We still have to make some assumptions

- For the position Σ_p initially can be all zeros
- Estimate an (initial) covariance matrix

$$\Sigma_{\Delta} = \begin{bmatrix} k_r\,|\Delta s_r| & 0 \\ 0 & k_l\,|\Delta s_{rl}| \end{bmatrix}$$

  - The two errors from the wheels are independent
  - The variance of the errors in the wheels is proportional to the absolute value of the travel distance
    - The constants k_l k_r determined experimentally

In higher dimensions (from Lect 08)

- It can be shown that the output covariance matrix C_Y is given by the error propagation law:
  - $C_Y = F_X C_X F_X^T$
  - where
    - C_X:covariance matrix representing the input uncertainties
    - C_Y:covariance matrix representing the propagated output uncertainties
    - is the Jacobian matrix (the transpose off the gradient of $f(X)$ defined as

$$\begin{bmatrix} \delta f_1/\delta X_1 & \cdots & \delta f_1/\delta X_n \\ \vdots & & \\ \delta f_m/\delta X_1 & & \delta f_m/\delta X_n \end{bmatrix}$$

Applying the error propagation law

- $p' = f(x, y, \theta, \Delta s_r, \Delta s_l)$
- Two Jacobians
  - $F_p = \nabla_p f = \nabla_p (f^T) = (\delta f/\delta x \;\; \delta f/\delta y \;\; \delta f/\delta \theta)$
  - $F_{\Delta rl} = \nabla_{\Delta rl} f = (\delta f/\delta \Delta s_r \;\; \delta f/\delta \Delta s_l)$
- *Update the confidence/trust on the error*
  - $\Sigma_p = \nabla_p f \, \Sigma_p \, \nabla_p f^T + \nabla_{\Delta rl} f \, \Sigma_\Delta \, \nabla_{\Delta rl} f^T$

Applying the error propagation law

- $Fp = \nabla_p f = \nabla_p (f^T) = (\delta f/\delta x \;\; \delta f/\delta y \;\; \delta f/\delta \theta)$

$$f = \begin{bmatrix} x + \cos(\theta + \Delta\theta/2)(\Delta s_r + \Delta s_l)/2 \\ y + \sin(\theta + \Delta\theta/2)(\Delta s_r + \Delta s_l)/2 \\ \theta + (\Delta s_r - \Delta s_l)/b \end{bmatrix}$$

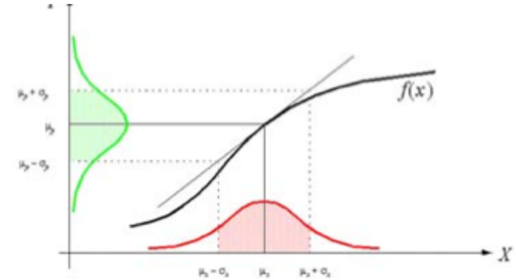
Robots and people

Robots and People

Odometry in the robots frame of reference

- b is the separation between the wheels
- (Motion) Error Model:

$$\Sigma_\Delta = covar(\Delta s_r, \Delta s_l) = \begin{bmatrix} k_r|\Delta s_r| & 0 \\ 0 & k_l|\Delta s_l| \end{bmatrix}$$



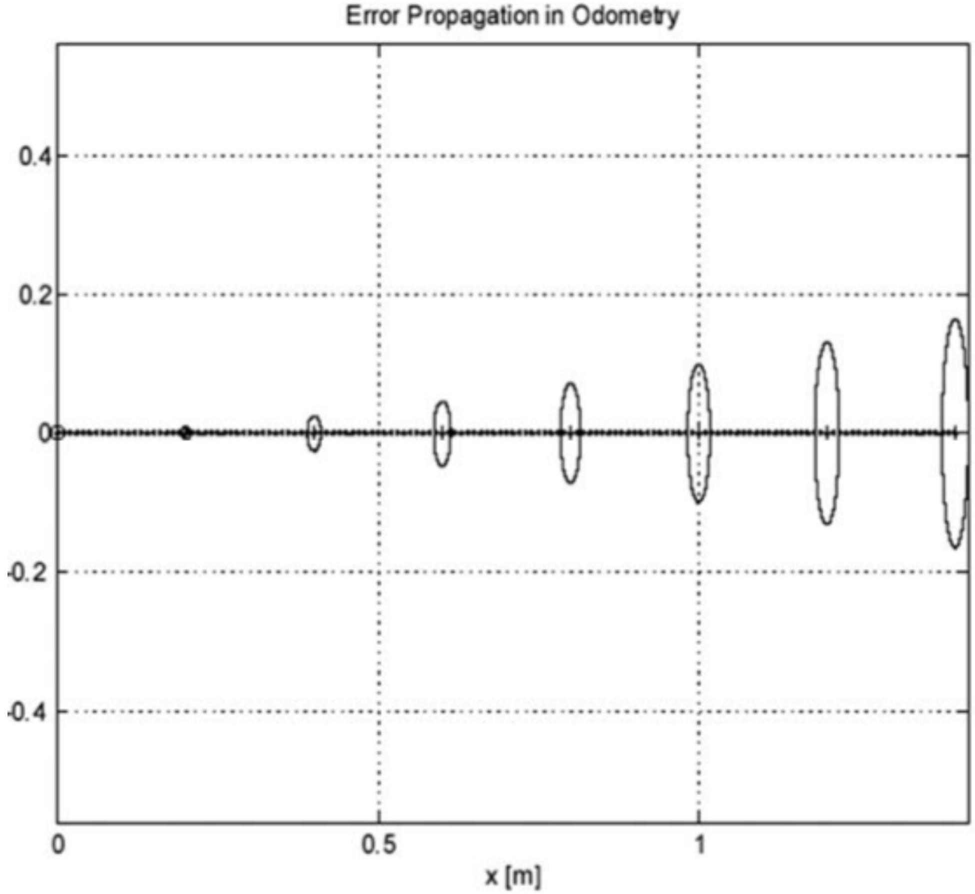
$$\Sigma_{p'} = \nabla_p f \cdot \Sigma_p \cdot \nabla_p f^T + \nabla_{\Delta_{rl}} f \cdot \Sigma_\Delta \cdot \nabla_{\Delta_{rl}} f^T$$

$$F_p = \nabla_p f = \nabla_p(f^T) = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s \sin(\theta + \Delta\theta/2) \\ 0 & 1 & \Delta s \cos(\theta + \Delta\theta/2) \\ 0 & 0 & 1 \end{bmatrix}$$

$$F_{\Delta_{rl}} = \begin{bmatrix} \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{2b}\sin\left(\theta + \frac{\Delta\theta}{2}\right) & \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2b}\sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{2}\sin\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2b}\cos\left(\theta + \frac{\Delta\theta}{2}\right) & \frac{1}{2}\sin\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{2b}\cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix}$$

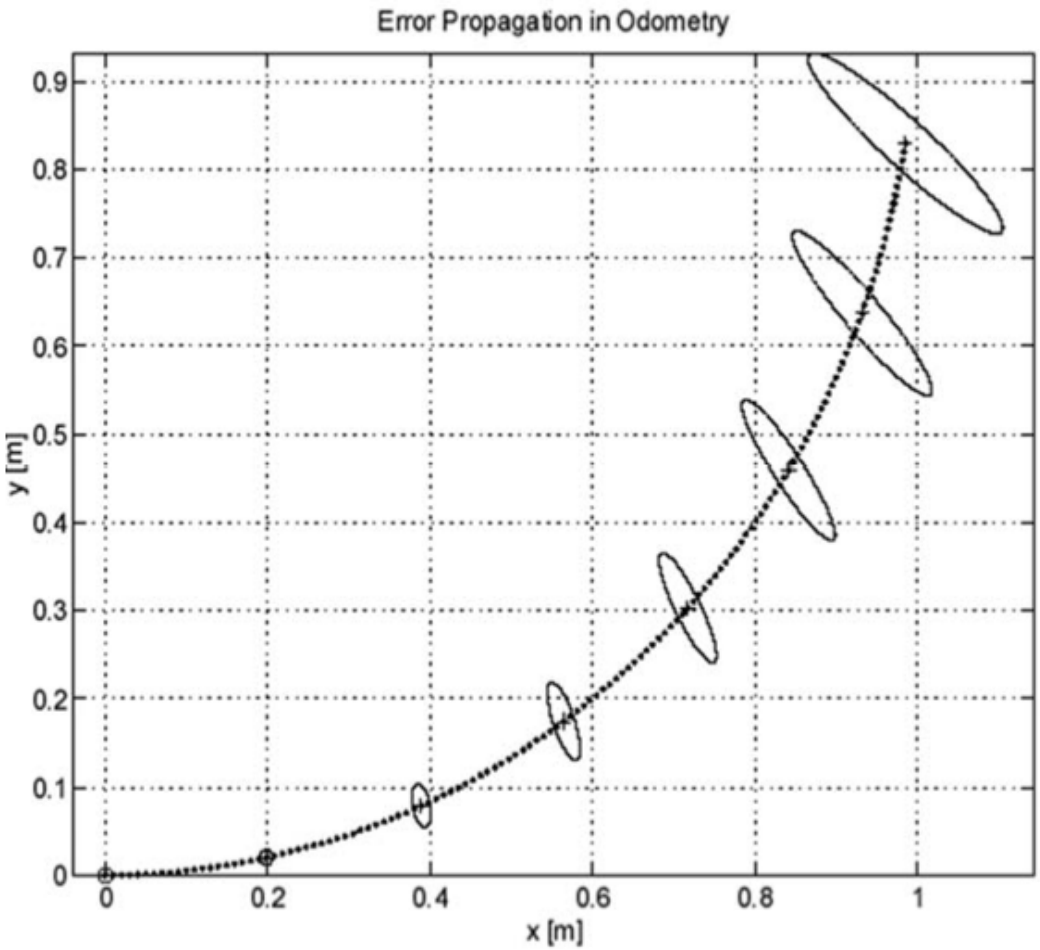
Odometry: Growth of Pose uncertainty for Straight Line Movement

- Note: Errors perpendicular to the direction of movement are growing much faster!



Odometry: Growth of Pose uncertainty for Movement on a Circle

- Note: Errors: ellipse does not remain perpendicular to the direction of movement!



Robots and people

Odometry: example of non-Gaussian error model

- Note: Errors are not shaped like ellipses!

[Fox, Thrun, Burgard, Dellaert, 2000]

Odometry: Error sources

- deterministic errors can be eliminated by proper calibration of the system.
  - SYSTEMATIC
- non-deterministic errors have to be described by error models and will always lead to uncertain position estimate
  - NON-SYSTEMATIC

Odometry: Error sources

- Major Error Sources in Odometry:
  - Limited resolution during integration (time increments, measurement resolution)
  - Misalignment of the wheels (deterministic)
  - Unequal wheel diameter (deterministic)
  - Variation in the contact point of the wheel
  - Unequal floor contact (slipping, not planar ...)

Odometry: Classification of Integration Errors

Robots and People

- Range error: integrated path length (distance) of the robots movement
  - sum of the wheel movements
- Turn error: similar to range error, but for turns
  - difference of the wheel motions
- Drift error: difference in the error of the wheels leads to an error in the robots angular orientation

Odometry: Classification of Integration Errors

- **Over long periods of time, turn and drift errors** far outweigh range errors!
- Consider moving forward on a straight line along the x axis.
  - The error in the y- position introduced by a move of d meters will have a component of $d\sin q$, which can be quite large as the angular error q grows.
-

Illustration

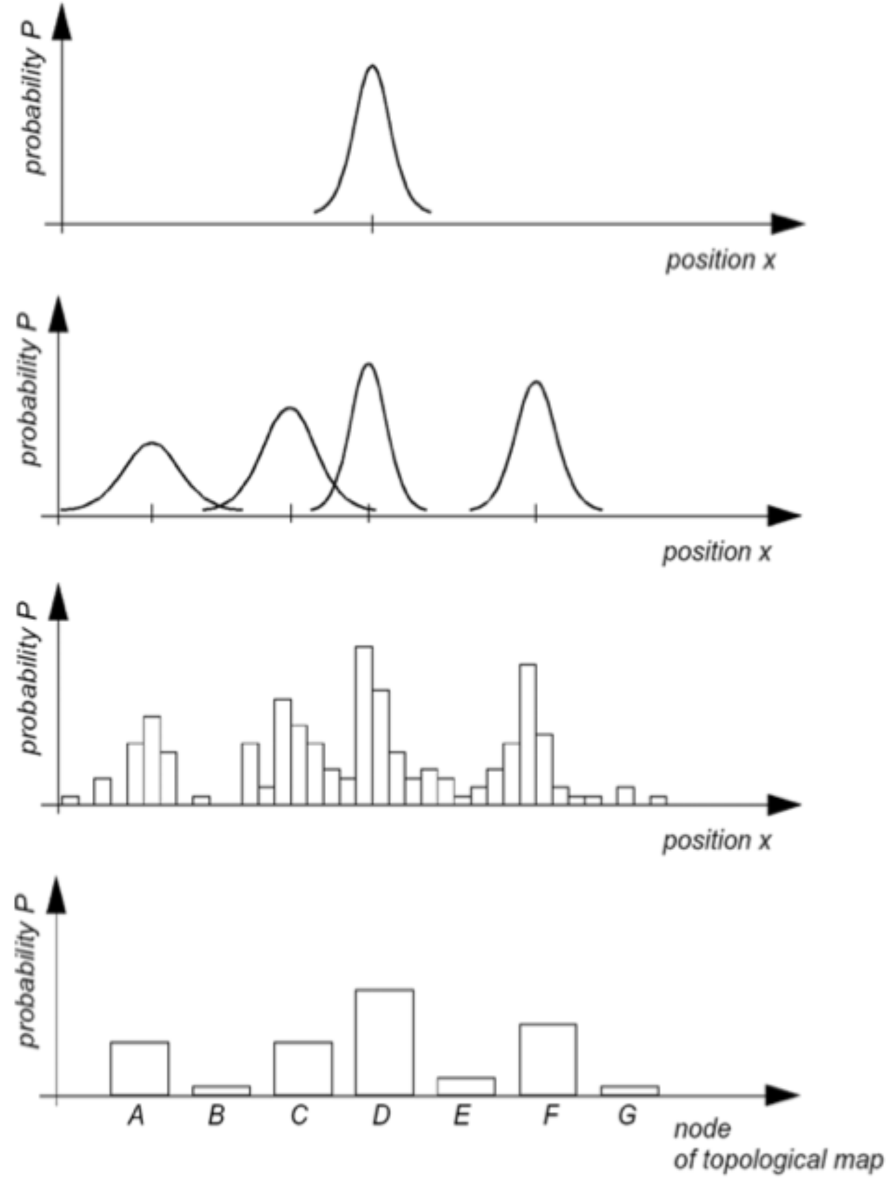
- MiPAL 2013 classification
- Sense-Plan-Act
  - (some replanning)
  - http://www.youtube.com/watch?v=cQgCrqRznCo&feature=youtu.be

Belief representation

How do we represent the robot position, where the robot ‘believes” to be?

Belief Representation

- Continuous map with single hypothesis probability distribution
- Continuous map with multiple hypothesis probability distribution
- Discretized map with probability distribution
- Discretized topological map with probability distribution



Belief Representation: Characteristics

Continuous

- Precision bound by sensor data
- Typically single hypothesis pose estimate
- Lost when diverging (for single hypothesis)
- Compact representation and typically reasonable in processing power.

Discrete

- Precision bound by resolution of discretisation
- Typically multiple hypothesis pose estimate
- Never lost (when diverges converges to another cell)
- Important memory and processing power needed. (not the case for topological maps)

Robots and People

Map representation

- Map precision vs. application
- Features precision vs. map precision
- Precision vs. computational complexity

- Continuous Representation
- Decomposition (Discretisation)

Robots and People

Representation of the Environment

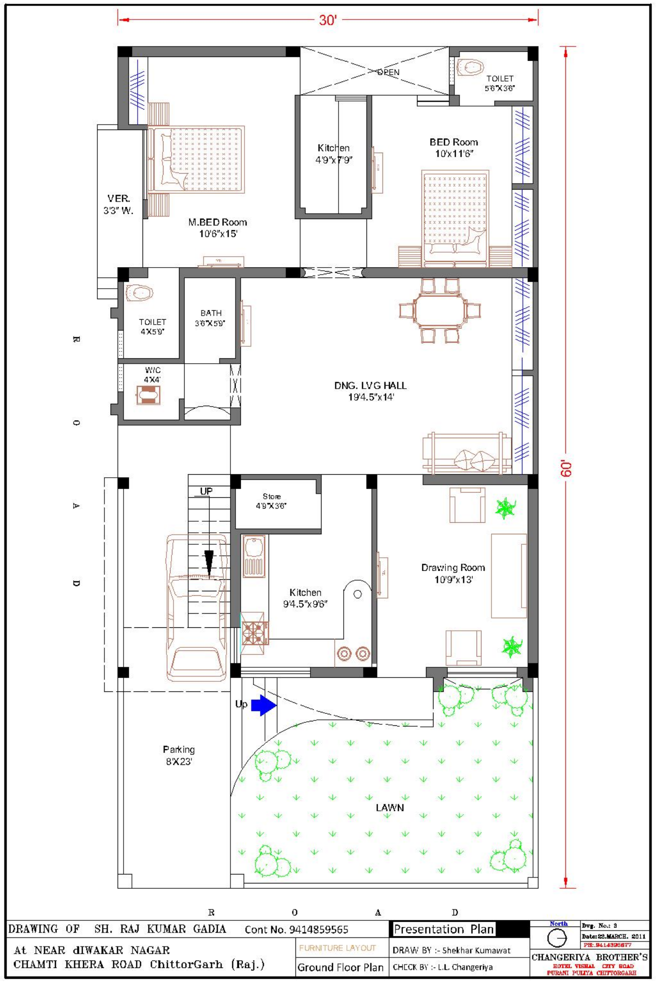
- Environment Representation
  - Continuous Metric
    - x,y,
  - Discrete Metric
    - metric grid
  - Discrete Topological
    - topological grid

Environment Modeling

- Raw sensor data, e.g. laser range data, gray-scale images
  - large volume of data, low distinctiveness on the level of individual values
  - makes use of all acquired information
- Low level features, e.g. line other geometric features
  - medium volume of data, average distinctiveness
  - filters out the useful information, still ambiguities
- High level features, e.g. doors, a car, the Eiffel tower
  - low volume of data,
  - high distinctiveness
  - filters out the useful information, few/no ambiguities, not enough information

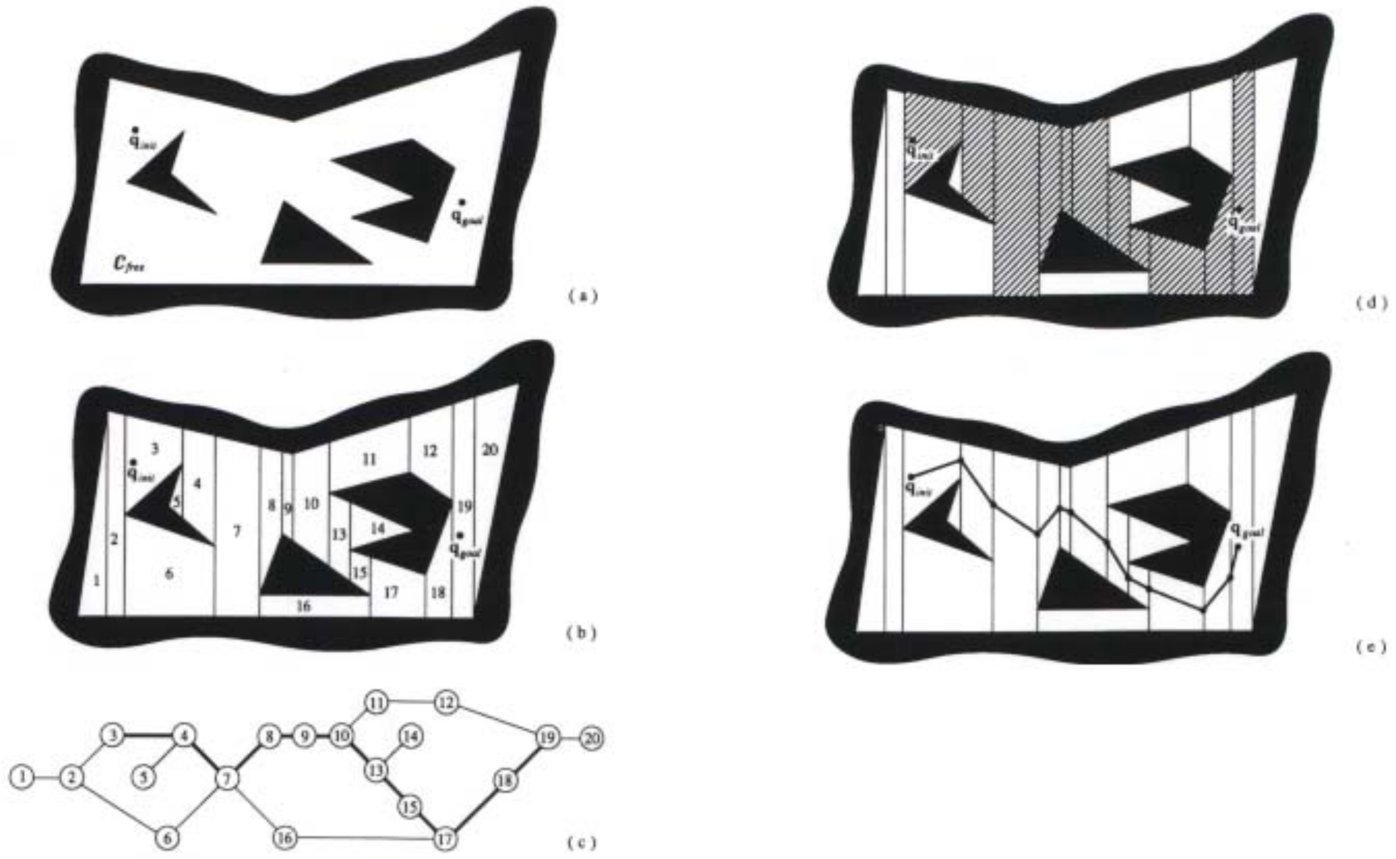
Map Representation

- Architecture map
- Representation with set of finite or infinite lines



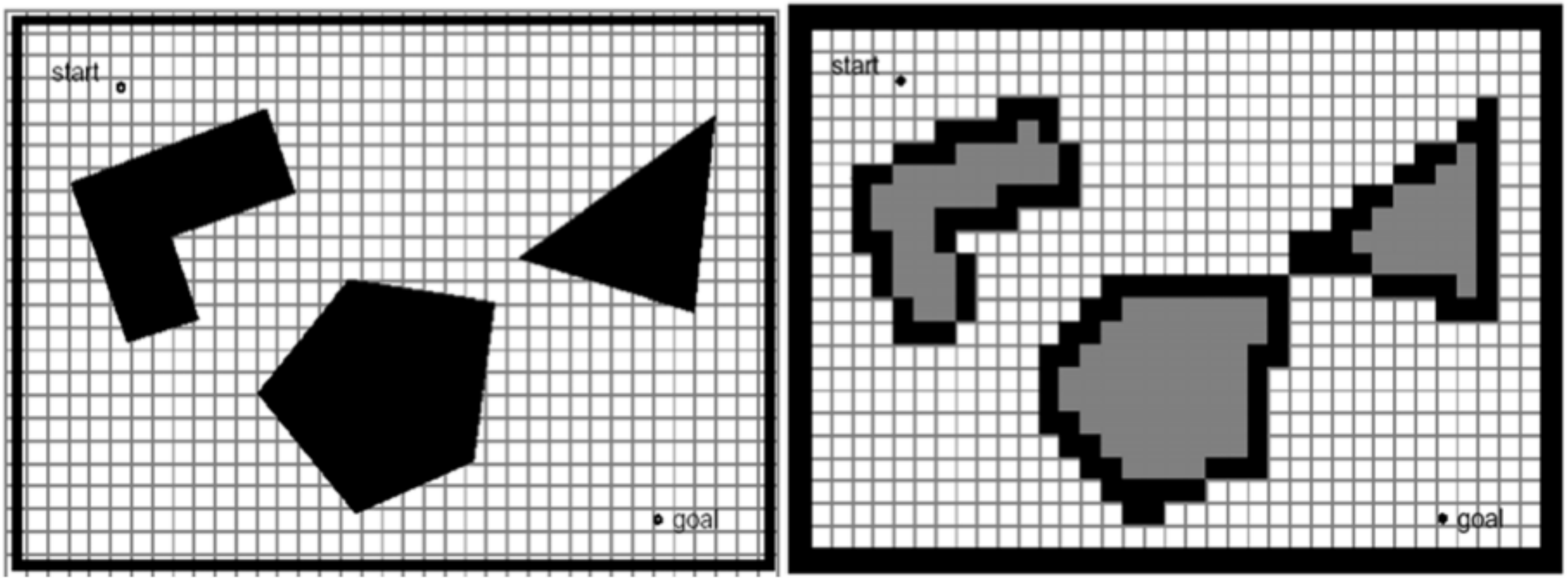
Map Representation

- Exact cell decomposition - Polygons



Map Representation

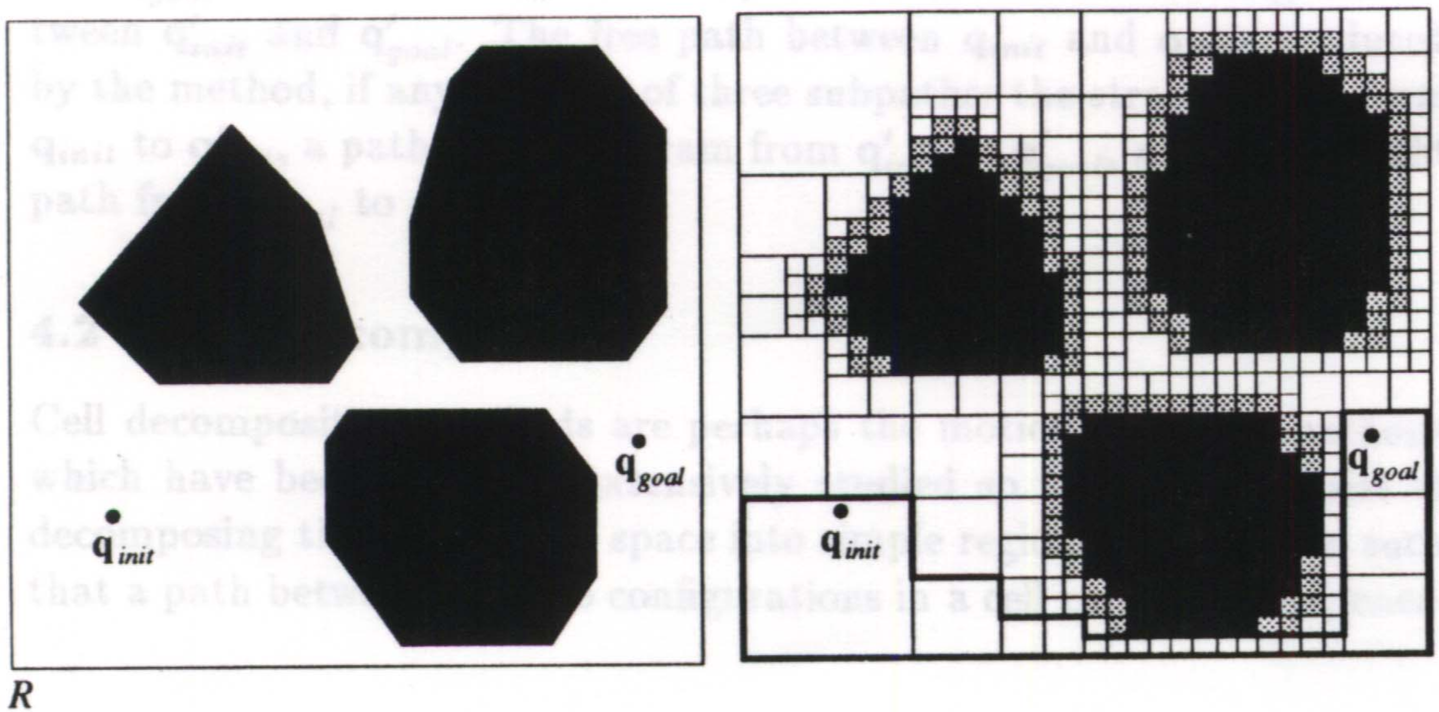
- Approximate cell decomposition
  - Fixed cell decomposition
    - Narrow passages disappear



Robots and people

Map Representation

- Adaptive cell decomposition



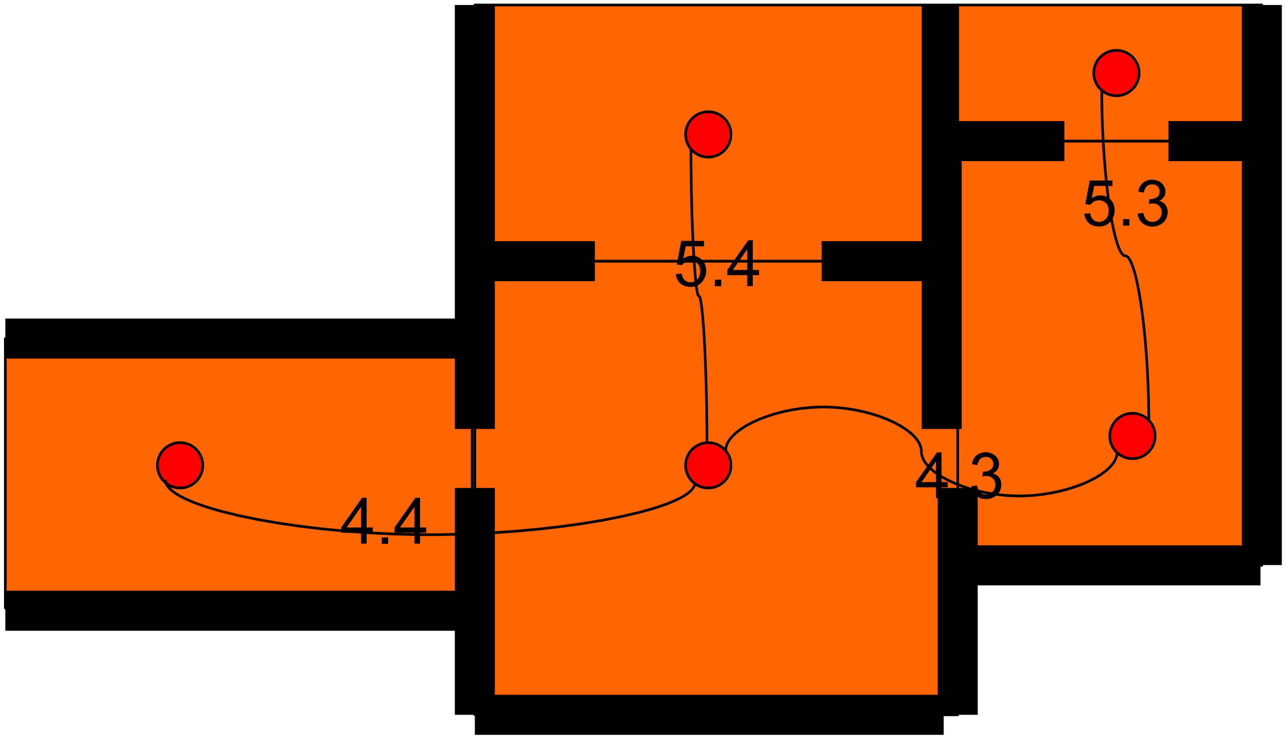
(a) (b)

Map Representation: Occupancy grid

- Fixed cell decomposition: occupancy grid example
  - In occupancy grids, each cell may have a counter where 0 indicates that the cell has not been hit by any ranging measurements and therefore it is likely free-space. As the number of ranging strikes increases, the cell value is incremented and, above a certain threshold, the cell is deemed to be an obstacle
- The values of the cells are discounted when a ranging strike travels through the cell. This allows us to represent "transient" (dynamic) obstacles

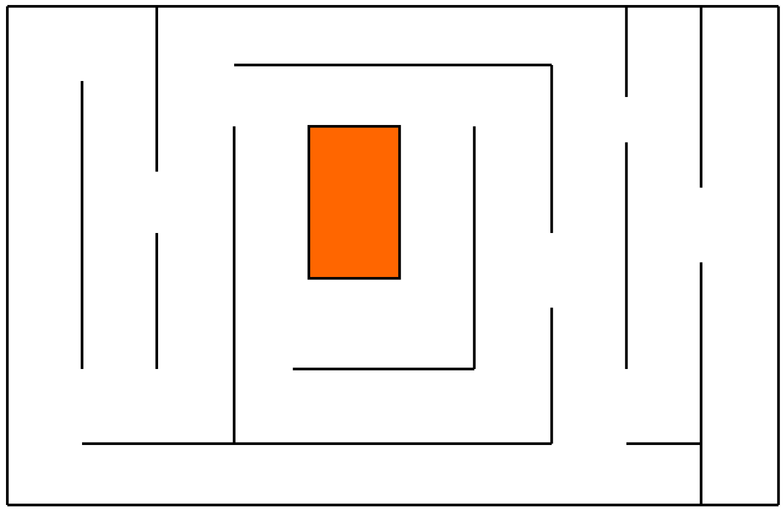
Topological Decomposition

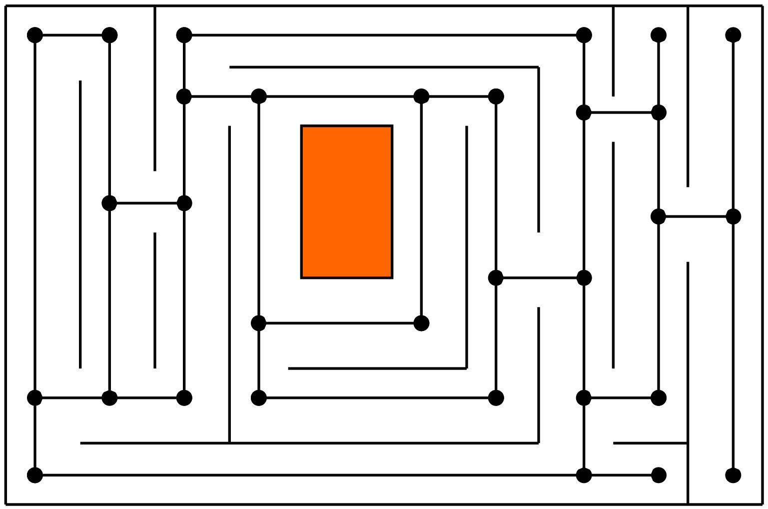
- With weights

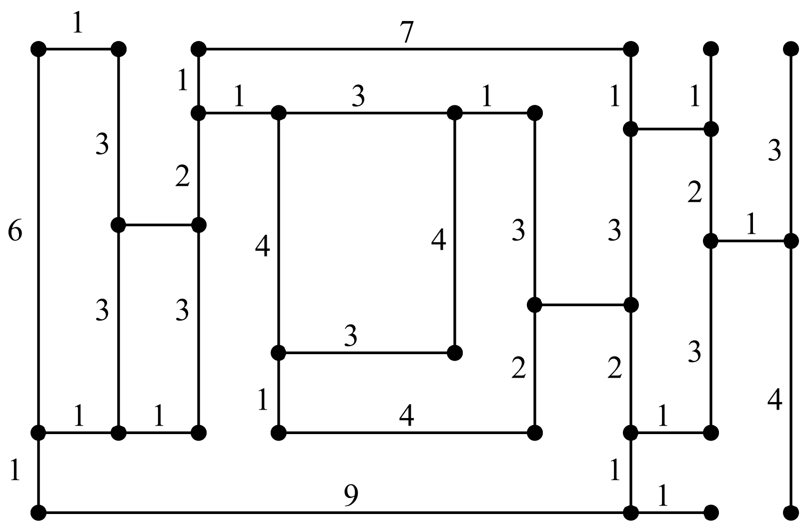


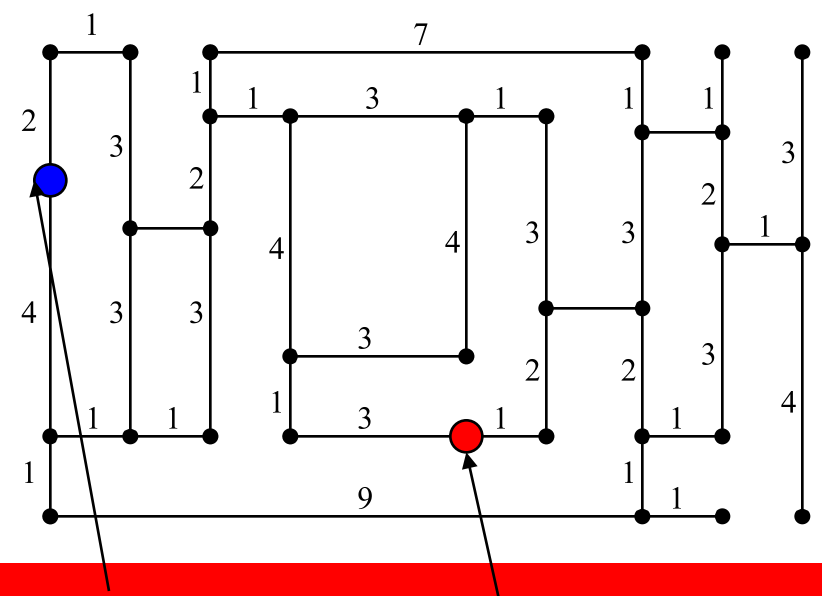
Robots and people

Transformation









Summary of Map Representation

- Metric maps
  - Continuous
    - For example line based, point based, or plane based
  - Discrete
    - Exact cell decomposition
    - Approximate cell decomposition
      - Fixed cell decomposition (also occupancy grids)
      - Adaptive cell decomposition
- Topological
- Hybrid (mixture of metric and topological)

State-of-the-Art: Current Challenges in Map Representation

- Real world is dynamic
- Perception is still a major challenge
  - Error prone
  - Extraction of useful information difficult
- Traversal of open space
- How to build up topology (boundaries of nodes)
- Sensor fusion
- 2D...3D

Robots and people

THANK YOU