Introduction to Localization

CS 3630

Localization

Consider a scenario in which the role of perception is to solve the ***localization*** problem, i.e., to determine an estimate of x_t, the robot's state at time t.

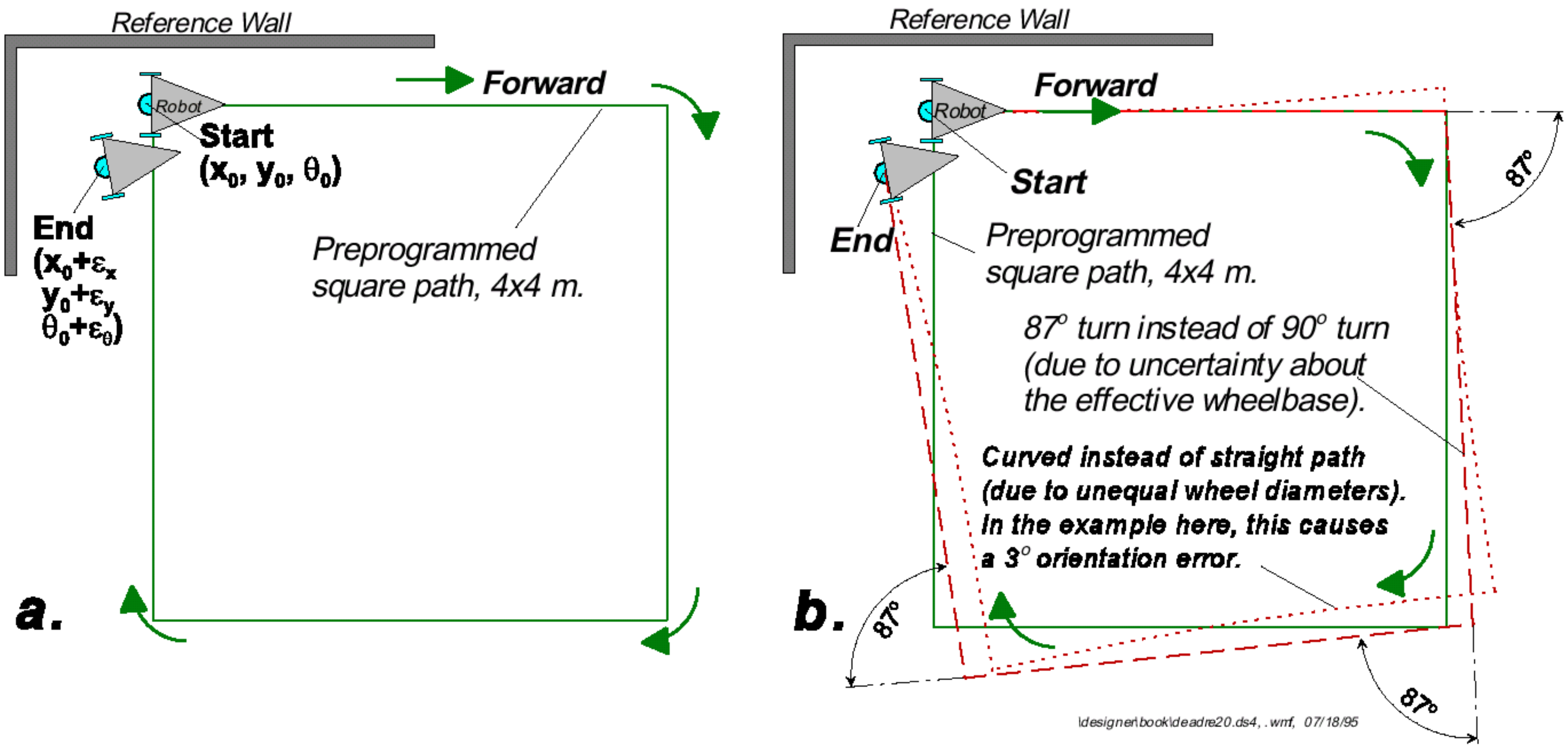
- Mathematically, the problem is to estimate the state x_t, given the action history $u_1 \ldots u_n$ and sensing history $z_1 \ldots z_n$ such that

$$P(x_t | u_1, z_1, u_2 \ \ldots z_{t-1}, u_{t-1}, z_t)$$

- We will assume the robot has access to an accurate map, though it may not know where it started.

First, an alternative to localization: dead reckoning

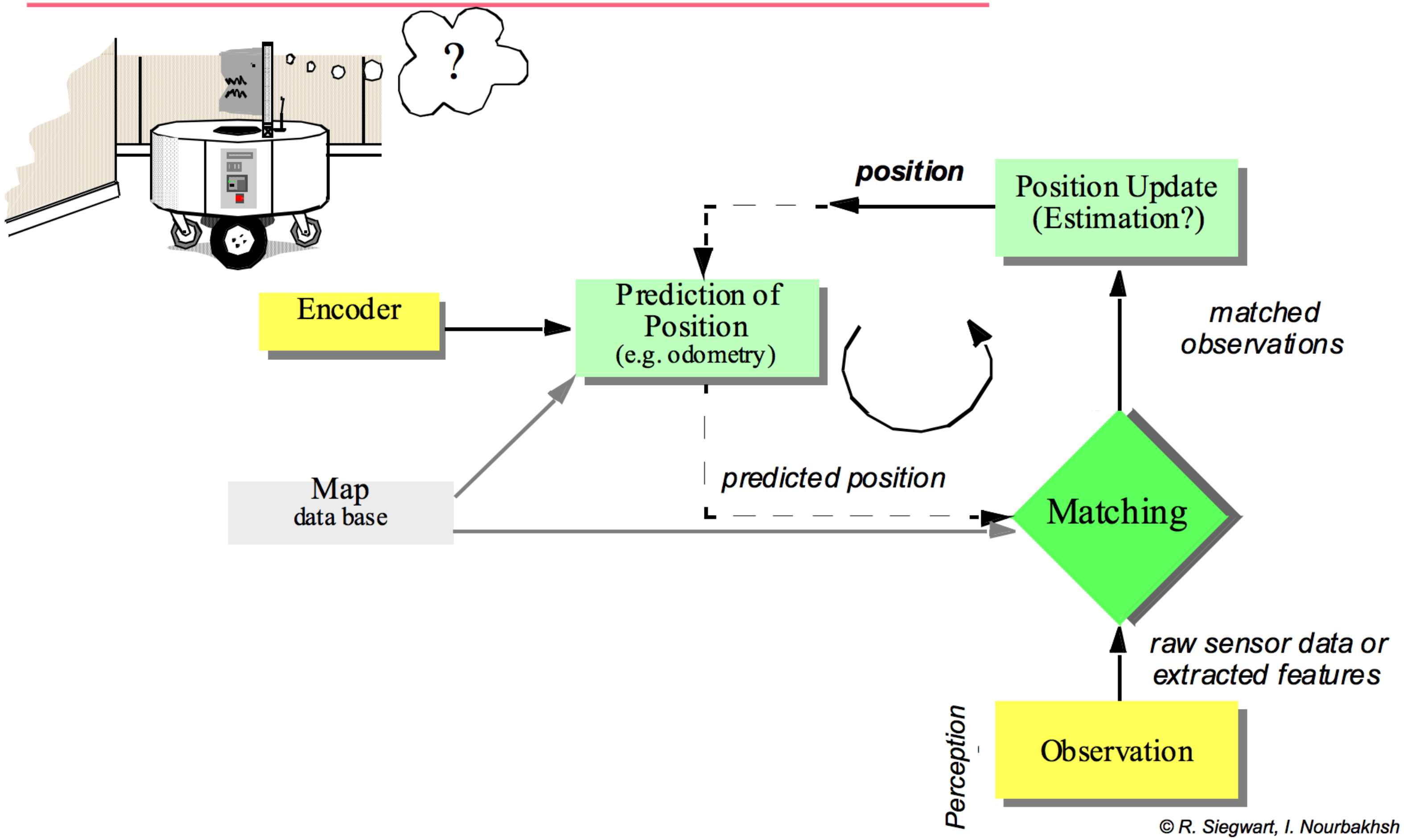
Dead reckoning is the process of calculating vehicle's current position by using a previously determined position and estimated speeds over the elapsed time.



Example of warehouse robots using dead reckoning successfully, but only for 1m of travel distance

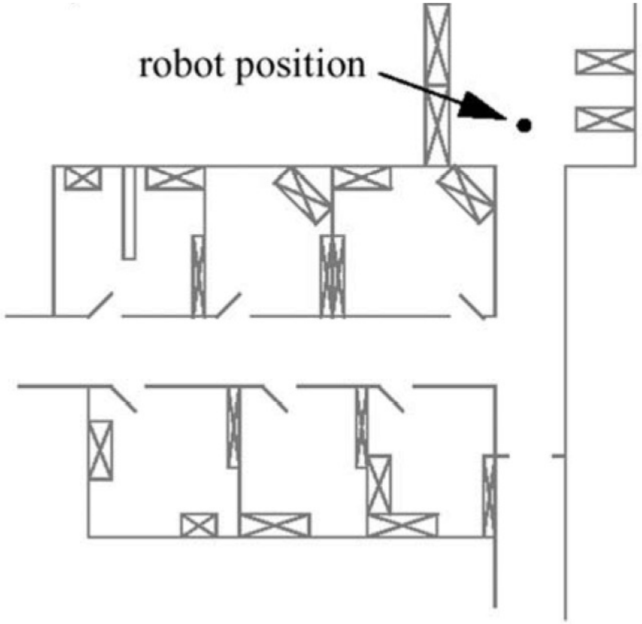
a.k.a. estimating the current *state* of the robot, or *state estimation*

Localization, Where am I?

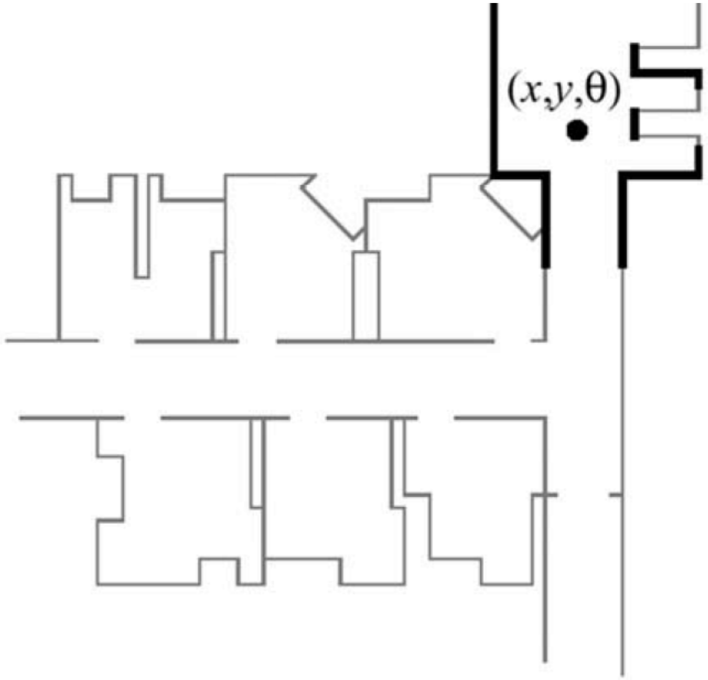


Given that we need localization, how do we represent the robot's world and the robot's location?

Map Representations

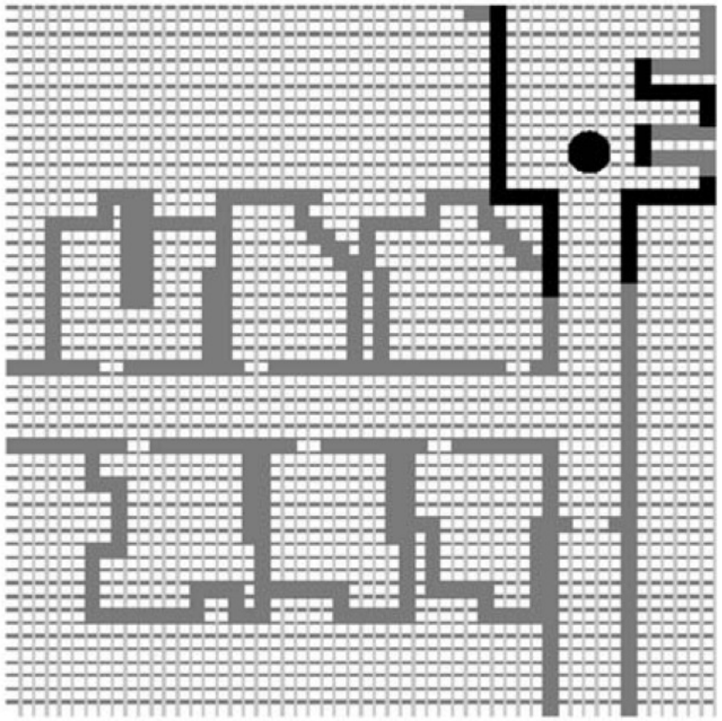


Floorplan with walls, doors and furniture.

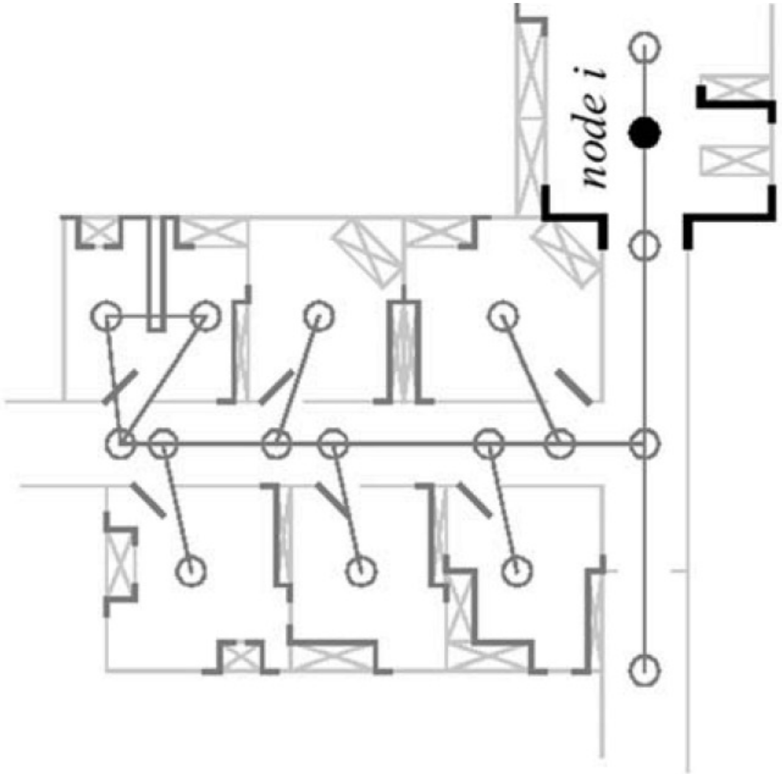


Line-based map (~100 lines)

Map Representations



Occupancy grid map (3000 cells, each 50cm x 50cm)



Topological map (50 features, 18 nodes)

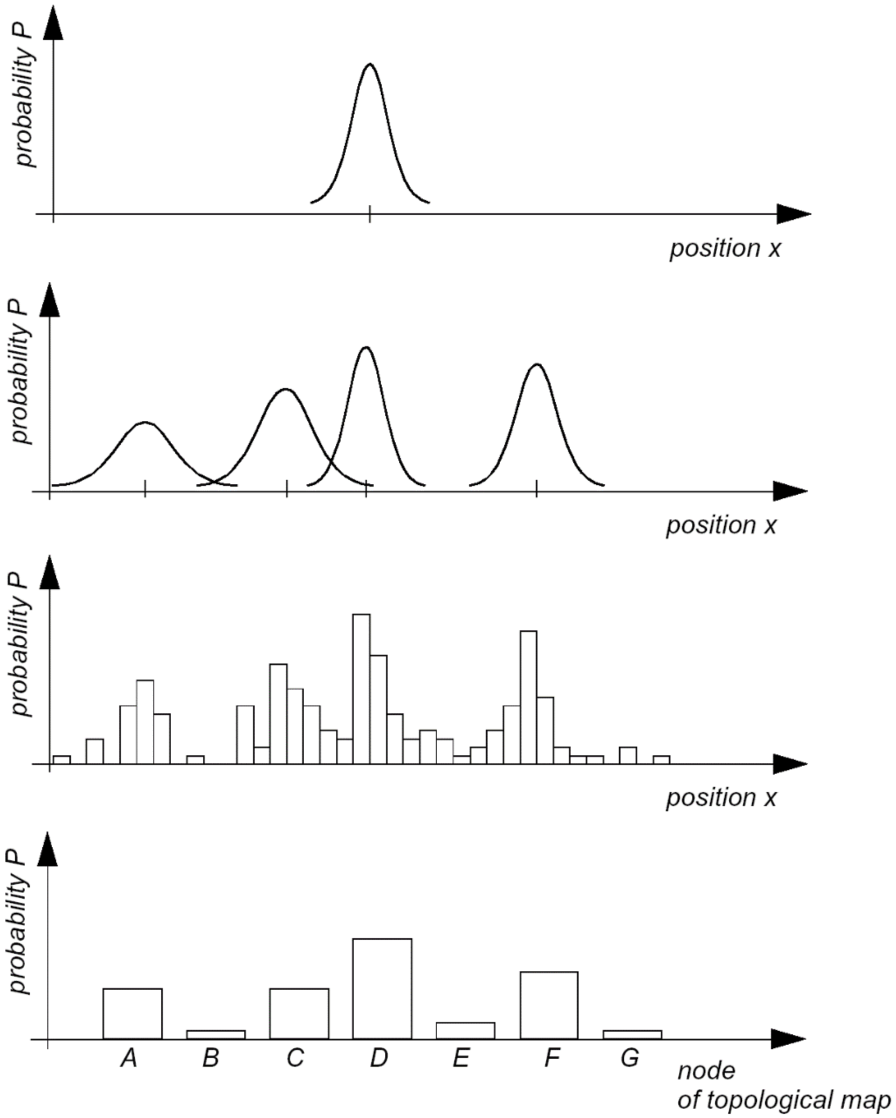
Belief representation: how do we represent our belief (hypothesis) of where the robot is located?

Continuous map with **single hypothesis** probability distribution

Continuous map with **multiple hypotheses** probability distribution

Discretized map with **multiple hypotheses** probability distribution

Discretized topological map with with **multiple hypotheses** probability distribution



Belief representation

- **Single-hypothesis belief:** The robot's belief about its position is expressed as a single point on a map
  - Advantage: no ambiguity, simplifies planning and decision making
  - Disadvantage: does not represent ambiguity/uncertainty

- **Multi-hypothesis belief**: allows the robot to track (possibly infinitely) many possible positions.

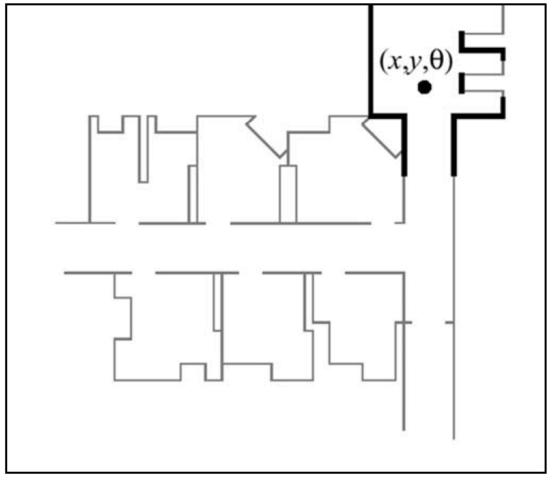
In both cases, the beliefs are represented as *probabilities*

$$Bel(x_t) = P(x_t | u_1, z_1, u_2 \ \ldots z_{t-1}, u_{t-1}, z_t)$$

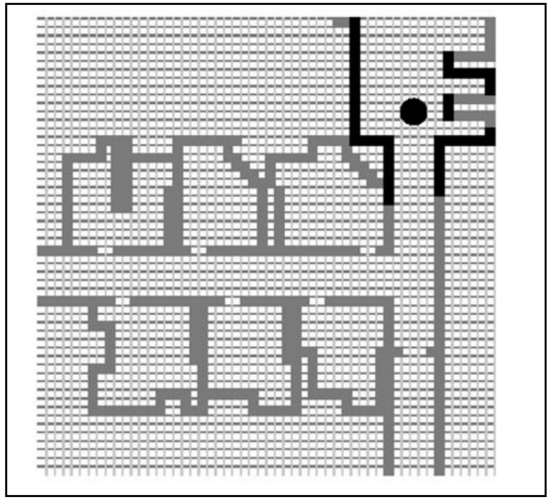
Choice of Representation

- Given the following two options:
  - a continuous representation (line map, position of the robot represented as x, y, θ)
  - a discrete representation (occupancy grid map, position of the robot represented as a grid cell)

Which would you choose?



Line Map



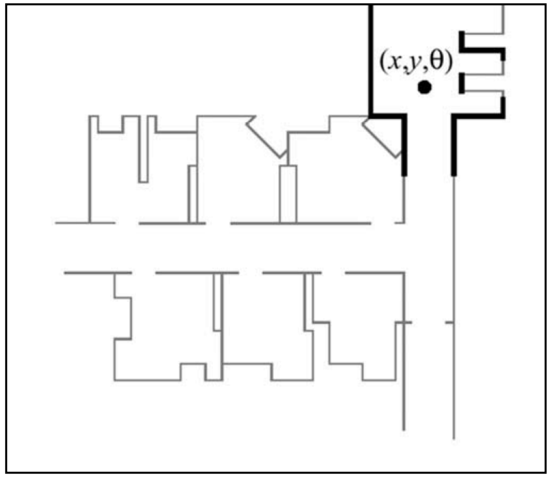
Occupancy Grid Map

Choice of Representation

- Given the following two options:
  - a continuous representation (line map, position of the robot represented as x, y, θ)
  - a discrete representation (occupancy grid map, position of the robot represented as a grid cell)

Which would you choose?

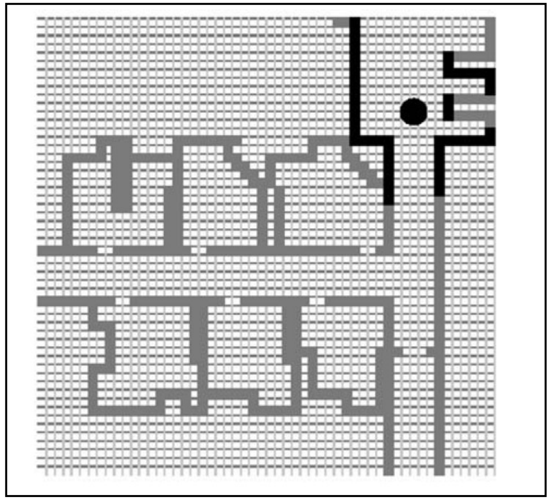
The answer somewhat depends on the application, but there are some tradeoffs to consider…



Line Map

Pros:
- greater precision
- memory-efficient

Cons:
- somewhat more involved calculations



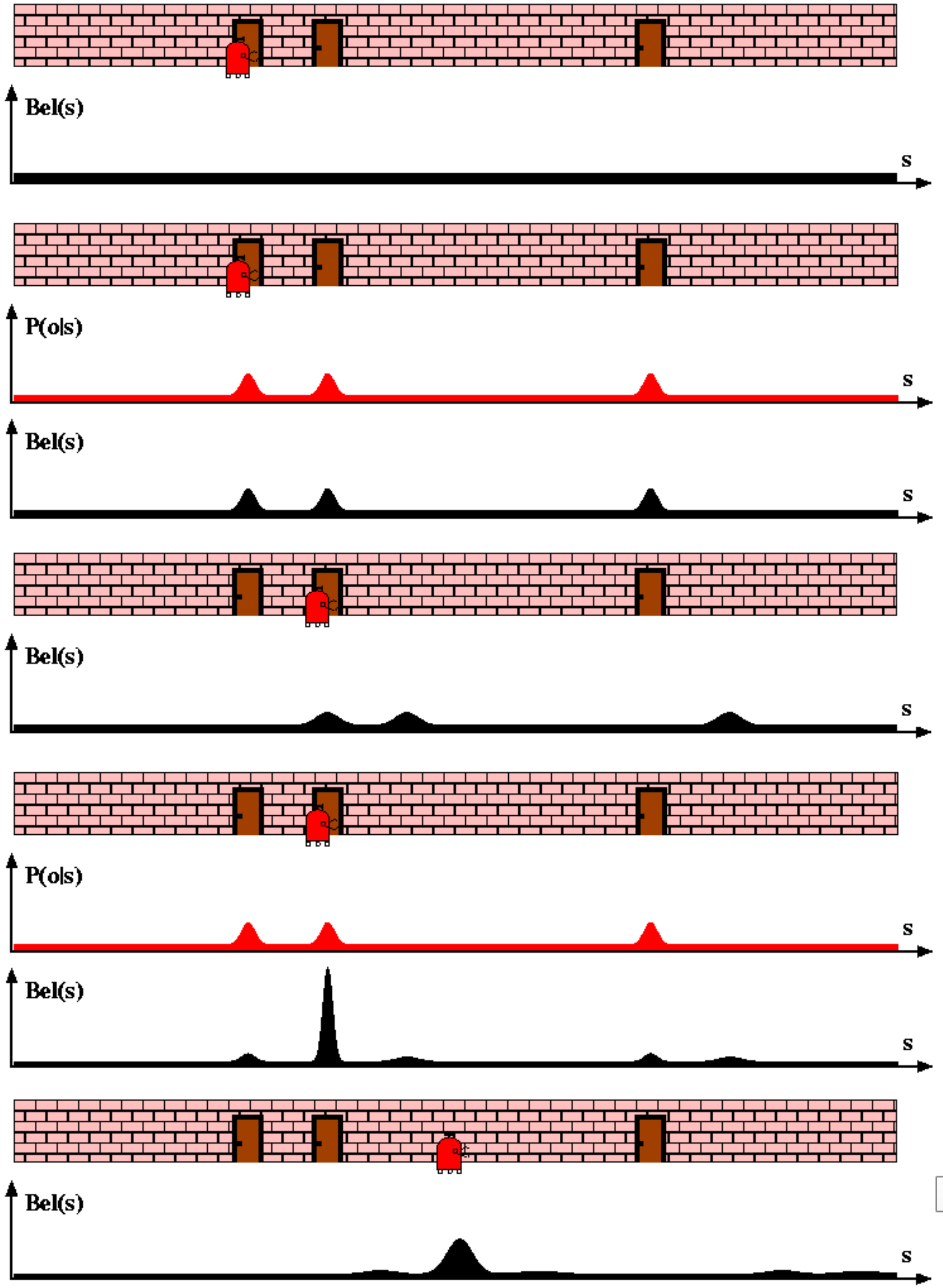
Occupancy Grid Map

Pros:
- simple and intuitive

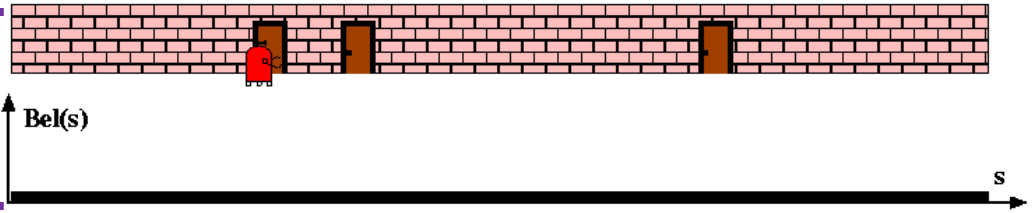
Cons:
- extreme memory demands
- sensitive to chosen size of grid cells

Given a map, how can the robot use its sensing and action information to localize within it?

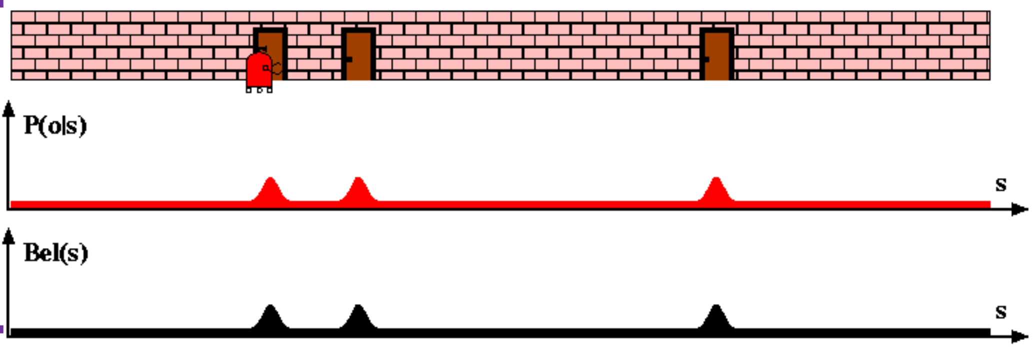
Bayes Filters for Robot Localization



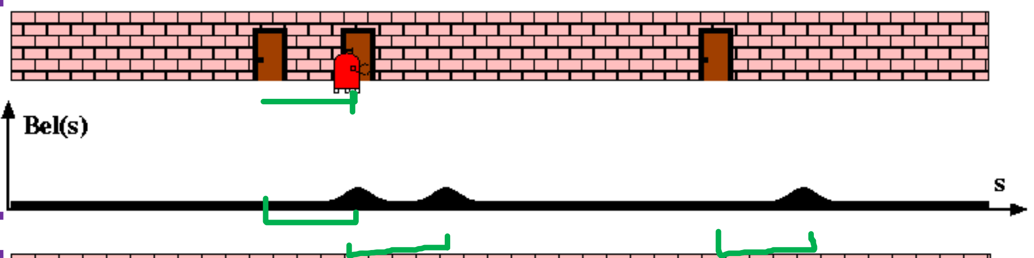
Initial Guess: Could be anywhere…



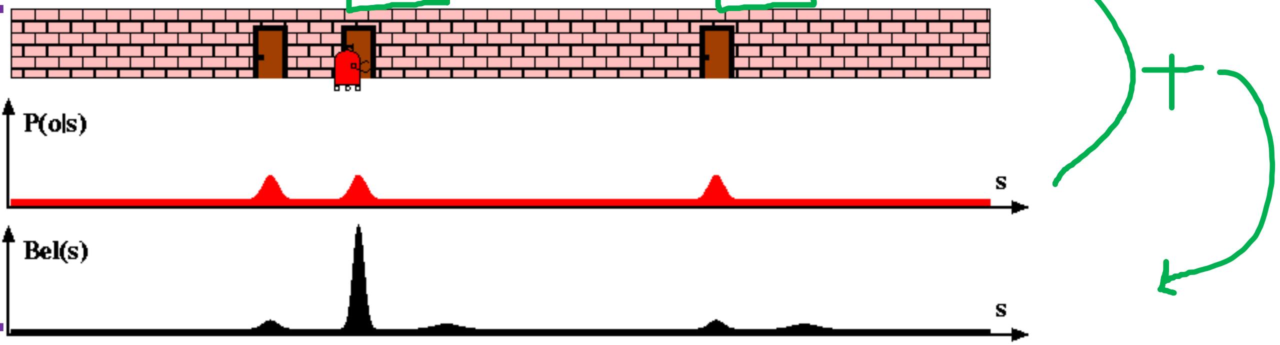
Take a measurement: we’re probably in front of a door…



Execute an action – move to the right by about a meter… probability mass “spreads out”



Take another measurement. It seems we’re in front of a door again (red). Given what we believed before about position, the most likely place now is the second door.



Execute an action – move to the right by about a meter… probability mass “spreads out”