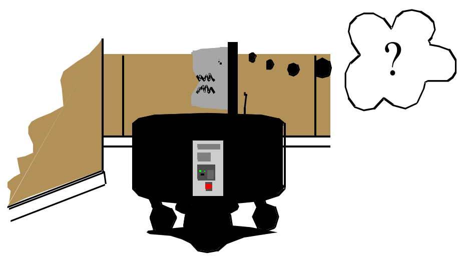
Probability & Localization

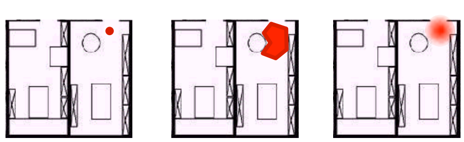
where am I (most likely)?

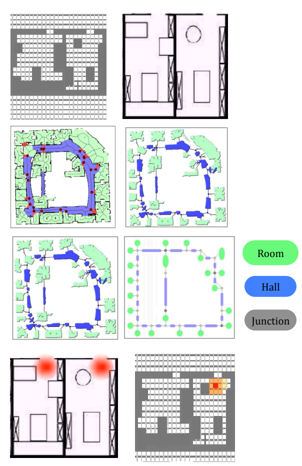


Representation Review

3

- Characterizing maps
  - Discrete vs. continuous
  - Geometric vs. topological
  - Semantically labeled vs. unlabeled
- Characterizing beliefs (location)
  - Discrete vs. continuous
  - Single vs. multiple hypothesis
  - Point, bounding box, probability function

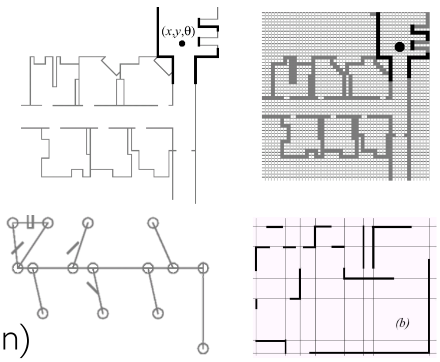


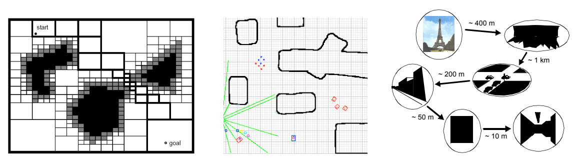


Map Representation Review

4

- Precision
- Types of representation:
  - Continuous line
  - Grid
  - Topological
  - Line extraction
- Decomposition (discretization)
  - Grid-based
  - Topological

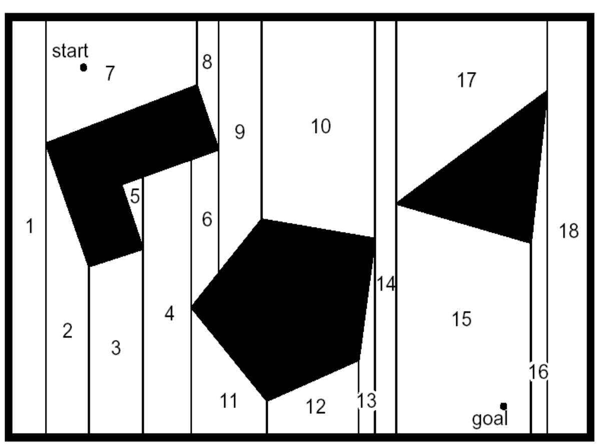




One More Decomposition

5

- Exact cell decomposition



Current Challenges

6

- 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
- …

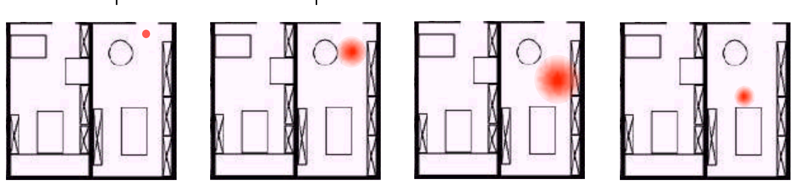
Probabilistic Map-Based Localization

7

- Mobile robot moving in known environment
  1. Starts from known location
  2. Keeps track (maintains a belief) using odometry
  3. Over time, uncertainty about position grows
     → Observe environment for new estimate
  4. Fuse estimate with odometric estimation
     →
  6. Best possible belief update

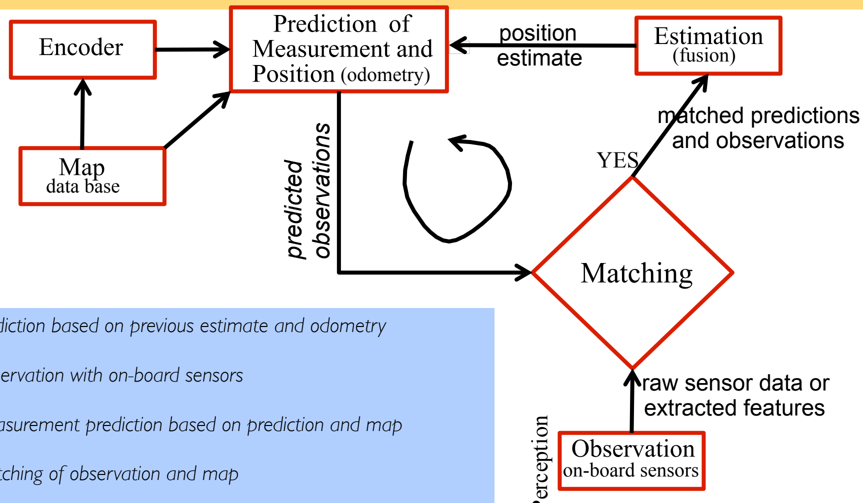
Prediction (action) update

Perception (measurement, correction) update



Map-Based Localization

10



1. Prediction based on previous estimate and odometry

2. Observation with on-board sensors

3. Measurement prediction based on prediction and map

4. Matching of observation and map

5. Estimation -> position update (posteriori position)

Markov ⇔ Kalman Filter Localization

11

- Markov localization
  - Localization starting from any unknown position
  - Recovers from ambiguous situation.
  - Update the probability of all positions within the whole state space at any time
    - Requires a discrete representation of the space (grid)
    - Memory and calculation power very important
    - Cell size matters!
- Kalman filter localization
  - Tracks robot
  - Inherently precise, efficient.
  - If uncertainty becomes too large, Kalman filter will fail and the position is definitively lost
    - e.g. collision with an object

Markov Localization (1)

12

- Markov localization: explicit, discrete representation for probability of each position in the state space
- Usually represents environment as:
  - Grid or
  - Topological graph
  - …with finite number of possible states (positions).
- During each update, the probability for each state (element) of the entire space is updated

Markov Localization (2)

13

- P(A): Probability that A is true.
  - Example:

    $p(r_t = l)$: probability that the robot r is at position l at time t
- Compute the probability of each robot position given
  - Actions and
  - Sensor readings
- P(A|B): Conditional probability of A, given that we know B.
  - Example:

    $p(r_t = l \mid i_t)$: probability robot is at position l given sensor input i_t

Markov Localization (3)

14

- Product rule:

$$p(A \wedge B) = p(A|B)p(B)$$

$$p(A \wedge B) = p(B|A)p(A)$$

- Bayes rule:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

Markov Localization (4)

15

- Bayes rule: $$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$
  - Map from belief state and sensor input to a refined belief state:

$$p(l|i) = \frac{p(i|l)p(l)}{p(i)}$$

  - p(l): belief state before perceptual update process
  - p(i | l): probability to get measurement i when at position l
    - Consult map
    - Identify probability of a sensor reading for each possible position
  - p(i): normalization factor so that sum over all l for L equals 1.

Markov Localization (5)

16

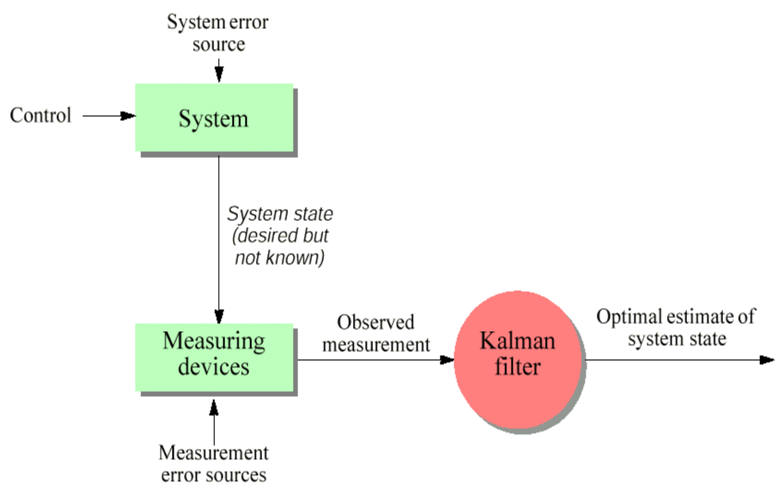
- Bayes rule: $p(A|B) = \frac{p(B|A)p(A)}{p(B)}$
  - Map from a belief state and a action to new belief state:

$$p(l_t|o_t) = \int p(l_t|l'_{t-1}, o_t)p(l'_{t-1})dl'_{t-1}$$

  - Summing over all possible ways in which the robot may have reached l.
- Markov assumption: Update only depends on previous state and its most recent actions and perception.

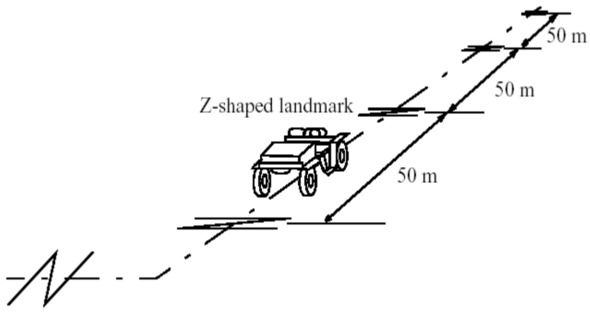
Kalman Filter Localization

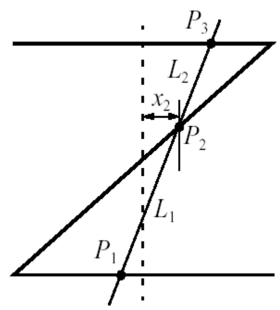
32



Based on Artificial Landmarks

50





Artificial Landmarks

51

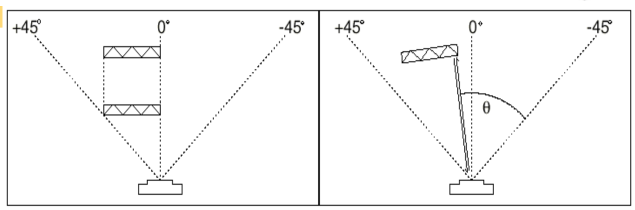


Figure 6.11: a. The perceived width of a retroreflective target of known size is used to calculate range; b. while the elapsed time between sweep initiation and leading edge detection yields target bearing. (Courtesy of NAMCO Controls).

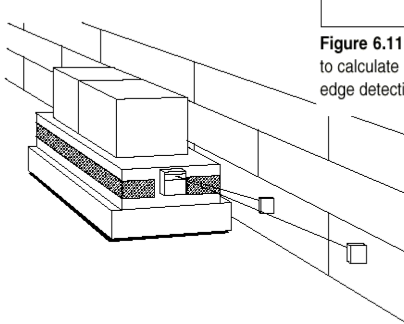


Figure 6.10: The LASERNET system can be used with projecting wall-mounted targets to guide an AGV at a predetermined offset distance. (Courtesy of NAMCO Controls.)

Beacon Systems: Triangulation

53

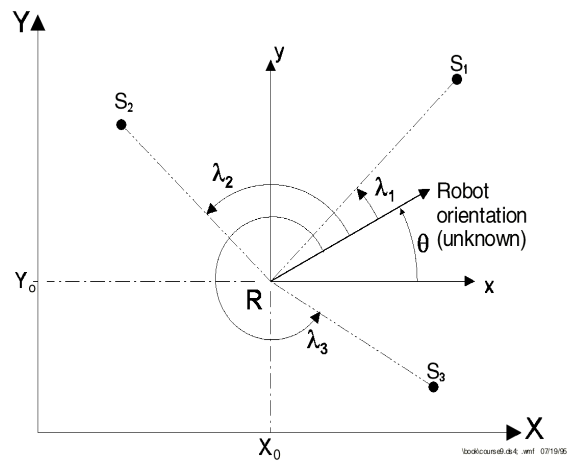
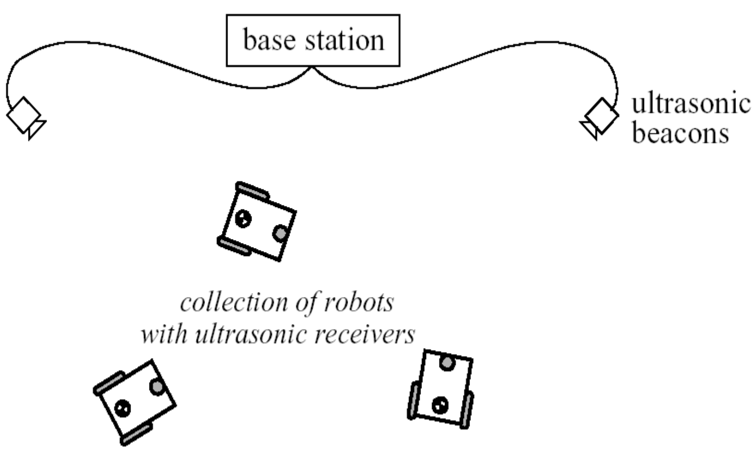


Figure 6.1: The basic triangulation problem: a rotating sensor head measures the three angles λ_1, λ_2, and λ_3 between the vehicle's longitudinal axes and the three sources S_1, S_2, and S_3.

Beacon Systems: Triangulation

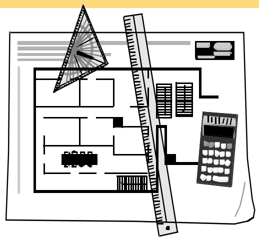
54



How to Get a Map

60

1. By Hand



2. Automatically: Map Building

The robot learns its environment

Motivation:
- by hand: hard and costly
- dynamically changing environment
- different look due to different perception

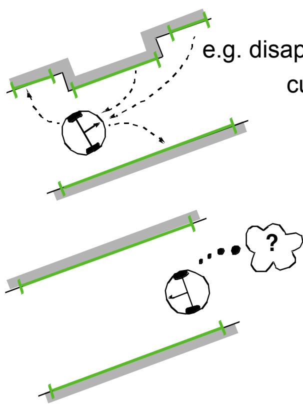
3. Basic Requirements of a Map:
  - a way to incorporate newly sensed information into the existing world model
  - information and procedures for estimating the robot's position
  - information to do path planning and other navigation task (e.g. obstacle avoidance)
- Quality of a map
  - topological correctness
  - metrical correctness
  - Predictability
- Most environments are a mixture of predictable and unpredictable features → hybrid approach

model-based vs. behaviour-based

Challenges

61

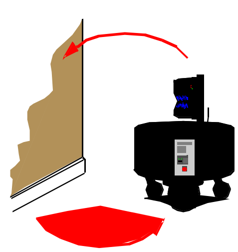
1. Map Maintaining: Keeping track of changes in the environment



- e.g. measure of belief of each environment feature

2. Representation and Reduction of Uncertainty

position of robot -> position of wall

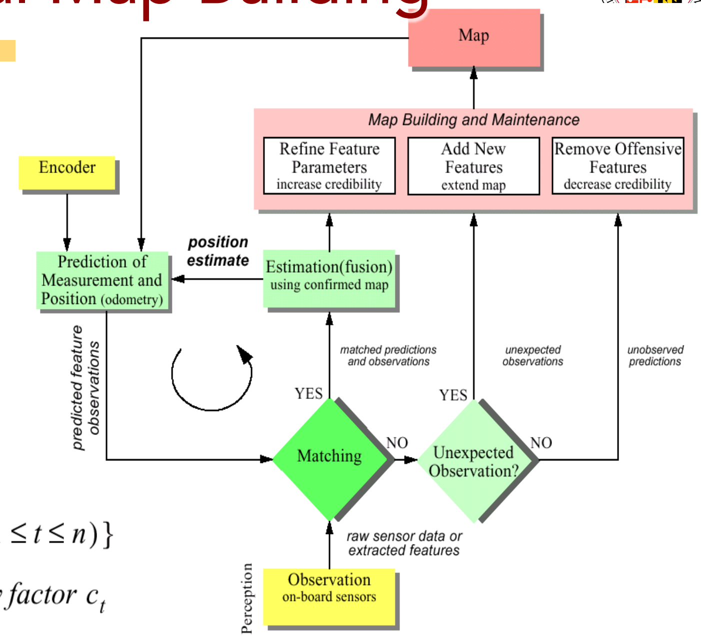


position of wall -> position of robot

- probability densities for feature positions
- additional exploration strategies

General Map Building

62



$$M = \{\hat{z}_t, \Sigma_t, c_t | (1 \leq t \leq n)\}$$

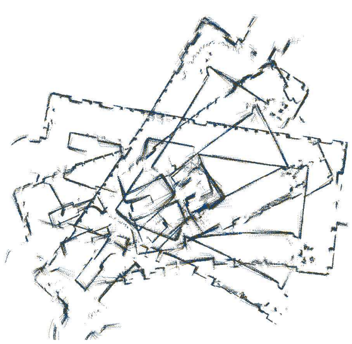
credibility factor c_t

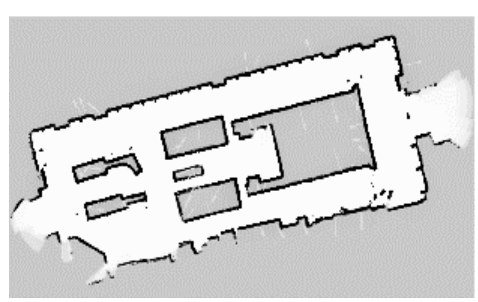
Cyclic Environments

Courtesy of Sebastian Thrun

66

- Small local error accumulate!
- This is usually irrelevant for navigation
- However, when closing loops, global error does matter

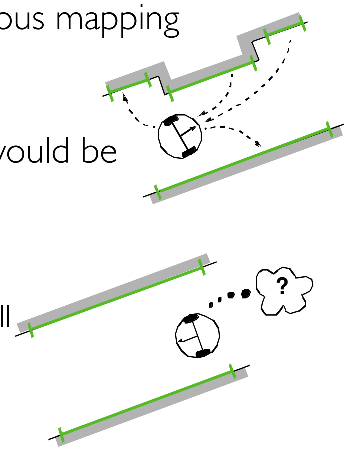




Dynamic Environments

67

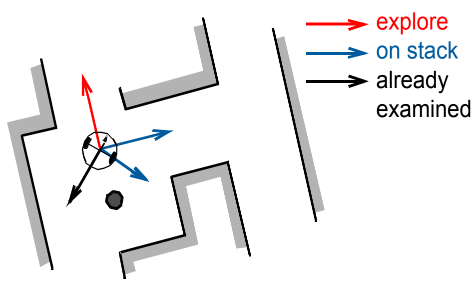
- Dynamical changes require continuous mapping
- If extraction of high-level features would be possible, the mapping in dynamic environments would become significantly more straightforward.
  - e.g. difference between human and wall
  - Environment modeling is a key factor for robustness



Exploration & Graph Construction

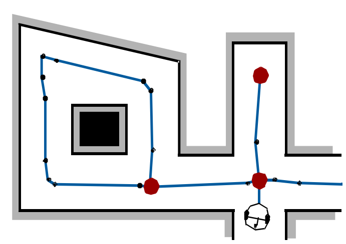
68

1. Exploration



- provides correct topology
- must recognize already visited location
- backtracking for unexplored openings

2. Graph Construction



Where to put the nodes?

- Topology-based: at distinctive locations

- Metric-based: where features disappear or get visible