Lecture 14: Localization

CS 344R/393R: Robotics

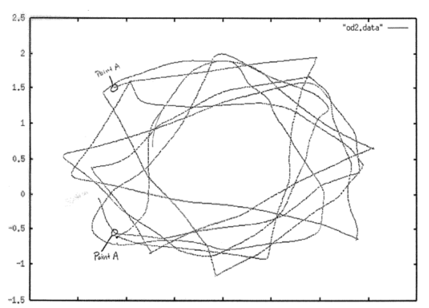
Benjamin Kuipers

Thanks to Dieter Fox for some of his figures.

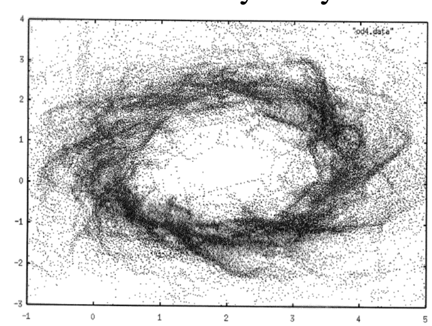
Localization: *"Where am I?"*

- The map-building method we studied assumes that the robot knows its location.
  - Precise (x,y,θ) coordinates in the same frame of reference as the occupancy grid map.
- This assumes that odometry is accurate, which is often false.
- We will need to relocalize at each step.

Odometry-Only Tracking: 6 times around a 2m x 3m area



Merging Laser Range Data Based on Odometry-Only Tracking

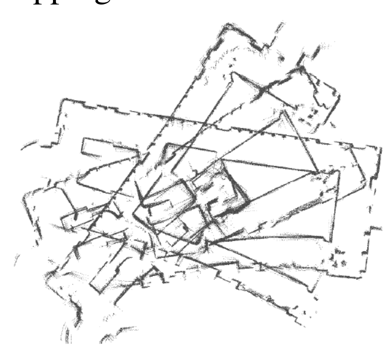


SLAM: Simultaneous Localization and Mapping

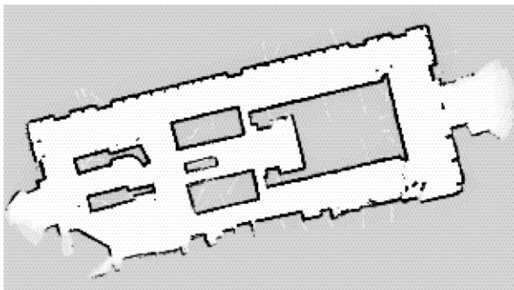
Alternate at each motion step:

1. **Localization**:
   - *Assume accurate map.*
   - Match sensor readings against the map to update location after motion.
2. **Mapping**:
   - *Assume known location in the map.*
   - Update map from sensor readings.

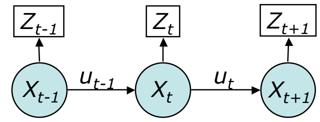
Mapping Without Localization



Mapping With Localization



Modeling Action and Sensing

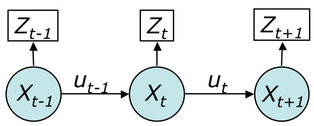


- Action model: $P(x_t \mid x_{t-1}, u_{t-1})$
- Sensor model: $P(z_t \mid x_t)$
- What we want to know is *Belief*:

  $$Bel(x_t) = P(x_t \mid u_1, z_2 \ldots, u_{t-1}, z_t)$$

  the posterior probability distribution of x_t, given the past history of actions and sensor inputs.

The Markov Assumption

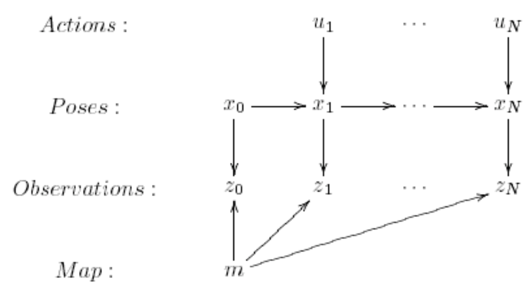


- *Given the present, the future is independent of the past.*
- Given the state x_t, the observation z_t is independent of the past.

  $$P(z_t \mid x_t) = P(z_t \mid x_t, u_1, z_2 \ldots, u_{t-1})$$

Dynamic Bayesian Network

- The well-known DBN for local SLAM.



Law of Total Probability

(*marginalizing*)

Discrete	**Continuous case**
$\sum_y P(y) = 1$	$\int p(y)\, dy = 1$
$P(x) = \sum_y P(x, y)$	$p(x) = \int p(x, y)\, dy$
$P(x) = \sum_y P(x \mid y) P(y)$	$p(x) = \int p(x \mid y) p(y)\, dy$

Bayes Law

- We can treat the denominator in Bayes Law as a normalizing constant:

$$P(x \mid y) = \frac{P(y \mid x)\; P(x)}{P(y)} = \eta\; P(y \mid x)\, P(x)$$

$$\eta = P(y)^{-1} = \frac{1}{\sum_x P(y \mid x) P(x)}$$

- We will apply it in the following form:

$$Bel(x_t) = P(x_t \mid u_1, z_2 \ldots, u_{t-1}, z_t)$$

$$= \eta\; P(z_t \mid x_t, u_1, z_2, \ldots, u_{t-1})\; P(x_t \mid u_1, z_2, \ldots, u_{t-1})$$

Bayes Filter

$$Bel(x_t) = P(x_t \mid u_1, z_2 \ldots, u_{t-1}, z_t)$$

Bayes $\quad = \eta \; P(z_t \mid x_t, u_1, z_2, \ldots, u_{t-1}) \; P(x_t \mid u_1, z_2, \ldots, u_{t-1})$

Markov $\quad = \eta \; P(z_t \mid x_t) \; P(x_t \mid u_1, z_2, \ldots, u_{t-1})$

Total prob. $\quad = \eta \; P(z_t \mid x_t) \int P(x_t \mid u_1, z_2, \ldots, u_{t-1}, x_{t-1}) \; P(x_{t-1} \mid u_1, z_2, \ldots, u_{t-1}) \; dx_{t-1}$

Markov $\quad = \eta \; P(z_t \mid x_t) \int P(x_t \mid u_{t-1}, x_{t-1}) \; P(x_{t-1} \mid u_1, z_2, \ldots, u_{t-1}) \; dx_{t-1}$

$$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_{t-1}, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}$$

Markov Localization

$$Bel(x_t) = \eta \; P(z_t \mid x_t) \int P(x_t \mid u_{t-1}, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}$$

- $Bel(x_{t-1})$ and $Bel(x_t)$ are prior and posterior probabilities of location x.
- $P(x_t \mid u_{t-1}, x_{t-1})$ is the action model, giving the probability distribution over result of u_{t-1} at x_{t-1}.
- $P(z_t \mid x_t)$ is the sensor model, giving the probability distribution over sense images z_t at x_t.
- η is a normalization constant, ensuring that total probability mass over x_t is 1.

Markov Localization

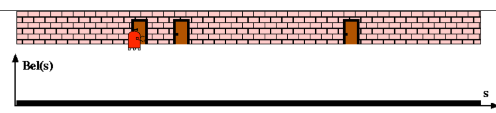
- Evaluate $Bel(x_t)$ for every possible state x_t.
- **Prediction** phase:

$$Bel^-(x_t) = \int P(x_t \mid u_{t-1}, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}$$

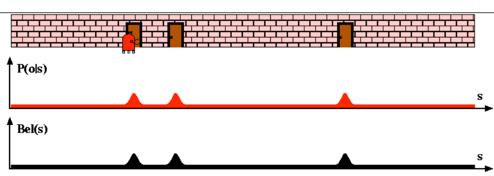
  - Integrate over every possible state x_{t-1} to apply the probability that action u_{t-1} could reach x_t from there.
- **Correction** phase:

$$Bel(x_t) = \eta \; P(z_t \mid x_t) \; Bel^-(x_t)$$

  - Weight each state x_t with likelihood of observation z_t.

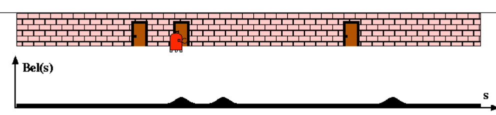


Uniform prior probability $Bel^-(x_0)$



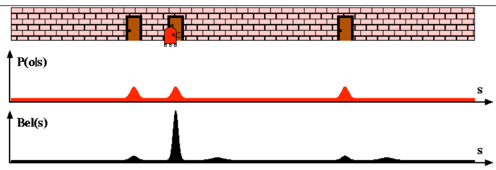
Sensor information $P(z_0 \mid x_0)$

$$Bel(x_0) = \eta \; P(z_0 \mid x_0) \; Bel^-(x_0)$$



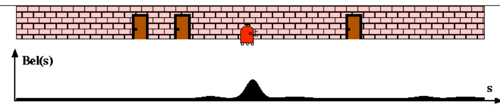
Apply the action model $P(x_1 \mid u_0, x_0)$

$$Bel^-(x_1) = \int P(x_1 \mid u_0, x_0) \; Bel(x_0) \; dx_0$$



Combine with observation $P(z_1 \mid x_1)$

$$Bel(x_1) = \eta\, P(z_1 \mid x_1)\, Bel^-(x_1)$$



Action model again: $P(x_2 \mid u_1, x_1)$

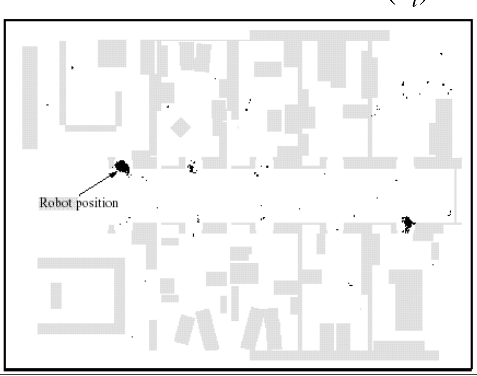
$$Bel^-(x_2) = \int P(x_2 \mid u_1, x_1)\, Bel(x_1)\, dx_1$$

Local and Global Localization

- Most localization is *local*:
  - Incrementally correct belief in position after each action.
- *Global* localization is more dramatic.
  - Where in the entire environment am I?
- The “kidnapped robot problem”
  - Includes detecting that I am lost.

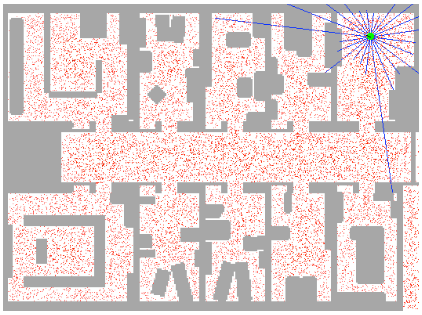
Initial belief $Bel(x_0)$

Intermediate Belief $Bel(x_t)$



Final Belief $Bel(x_t)$

Global Localization Movie



Future Attractions

- Sensor and action models
- Particle filtering
  - elegant, simple algorithm
  - Monte Carlo simulation