Mobile Robotics, Localization: Introduction to recursive state estimation

Material based on the book Probabilistic Robotics (Thrun, Burgard, Fox) [PR]; Chapter 1,2

Part of the material is based on lectures from Cyrill Stachniss

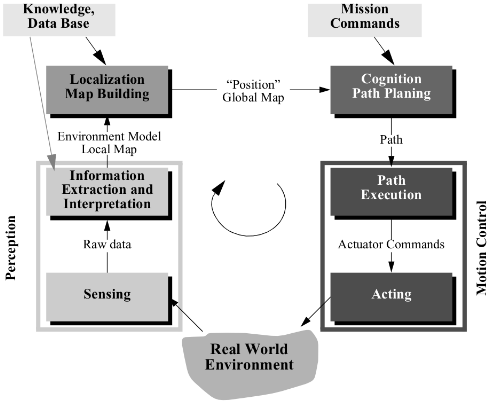
Summary

- Introduction to localization
- Recursive State Estimation using Bayes Filters [Chapter 2]

Introduction to Localization

Mobile Robot Localization: Introduction

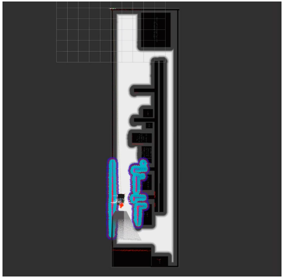
◇ Localization: estimate position based on observations



General control scheme for mobile robot systems (Source [AMR, Siegwart, Nourbakhsh, Scaramuzza])

Localization: definition

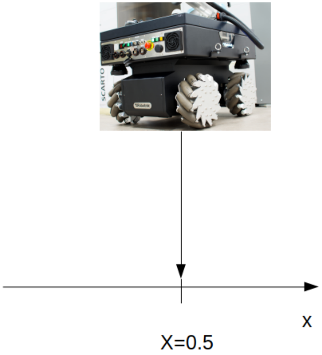
- Robot estimates its position based on perceived informations and a map
  - Map is given: Localization
  - Map is built in parallel: Simultaneous Localization and Mapping (SLAM)

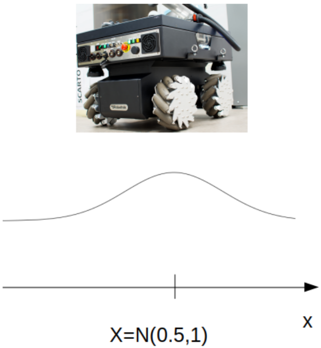


Kairos platform localizing in the ICE Lab

Localization: challenges

- Measurements (proprioceptive, exteroceptive) are inherently noisy
- Map can not be perfect
- Must deal with uncertain information
  - Probabilistic approach to represent uncertainty





Quick probability refresher

- X **random variable**, x is a realization for X
- Discrete
  - $x \in \{x_1, x_2, ..., x_n\}$
  - $P(X = x_i)$ probability that X takes value x_i
  - e.g., Pr(room) = <0.7,0.2,0.08,0.02>
- Continuous
  - X takes value in the continuum
  - $P(X = x)$ or $P(x)$ probability density function
  - e.g., $Pr(x \in (a, b)) = \int_a^b p(x)dx$

Joint and conditional probability

- Joint probability $P(X = x \text{ and } Y = y) = P(x, y)$
- If X and Y are **independent** then $P(x, y) = P(x)P(y)$
- $P(x|y)$ is the probability of X **given** Y
  - $P(x|y) = \frac{P(x,y)}{P(y)}$
  - $P(x, y) = P(x|y)P(y)$
- If X and Y are independent then $P(x|y) = P(x)$
- Marginals and total probability
  - **Discrete** $\sum_x P(x) = 1,\ P(x) = \sum_y P(x, y),\ P(x) = \sum_y P(x|y)P(y)$
  - **Continuous** $\int_x P(x)dx = 1,\ P(x) = \int_y P(x, y)dy,\ P(x) = \int_y P(x|y)P(y)dy$

Bayes Formula

- $P(x,y) = P(x|y)P(y) = P(y|x)P(x)$
- $P(x|y) = \frac{P(y|x)P(x)}{P(y)} \Rightarrow \frac{\text{likelihood}\cdot\text{prior}}{\text{evidence}}$
- $P(x|y) = \eta P(y|x)P(x)$, $\eta = \frac{1}{\sum_x P(y|x)P(x)}$
- Incorporating previous knowledge
  - $P(x|y,z) = \eta P(y|x,z)P(x|z)$, $\eta = \frac{1}{\sum_x P(y|x,z)P(x|z)}$

Conditioning and conditional independence

- Conditioning
  - $P(x) = \int_z P(x,z)dz$
  - $P(x) = \int_z P(x|z)P(z)dz$
  - $P(x|y) = \int_z P(x|y,z)P(z|y)dz$
- X **conditionally independent** from Y **given** Z
  - $P(x,y|z) = P(x|z)P(y|z)$
  - $P(x|z) = P(x|z,y)$
  - $P(y|z) = P(y|z,x)$

Simple example for state estimation

- Robot in front of a door, **State**: Door open or closed ?
- Observations:
  - Light sensor: high or low
  - Range sensor: near or far
- Our model:
  - $P(Light = high|Door = open) = 0.6$, $P(Light = high|Door = closed) = 0.3$
  - $P(Range = near|Door = open) = 0.5$, $P(Range = near|Door = closed) = 0.6$
  - $P(Door = open) = P(Door = closed) = 0.5$
- Assume the robot receives **a first** observation $Light = high$, compute $P(Door = open|Light = High)$
- Assume the robot receives **a second** observation $Range = near$, compute $P(Door = open|Light = High, Range = Near)$

Incorporating Actions

- World is dynamic:
  - Robots' actions change the state of the environment
  - External events (e.g., other agents) may change the state of the environment
- Most of the time actions do not have deterministic outcomes
- Actions **increase** uncertainty of the state
- Incorporate actions in the conditional probability distribution of the state
- Transition model: probability that action u changes the state from x' to x

$$P(x|u, x')$$

- Integrating actions to compute probability of **current** state x:
  - **Continuous state** $P(x|u) = \int_{x'} P(x|u, x')P(x')dx'$
  - **Discrete state** $P(x|u) = \sum_{x} P(x|u, x')P(x')$

Simple example for incorporating actions

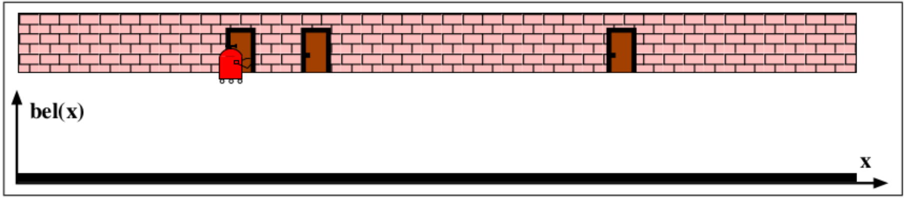
- Robot can close the door
  - $P(x = open | u = close, x' = open) = 0.1$
  - $P(x = closed | u = close, x' = open) = 0.9$
  - $P(x = open | u = close, x' = closed) = 0$
  - $P(x = closed | u = close, x' = closed) = 1$
- Compute probability of $P(x = closed)$ after executing the action $u = close$
- Assume $P(x = open) = \frac{5}{8}$ and $P(x = closed) = \frac{3}{8}$

Recursive State Estimation using Bayes Filters

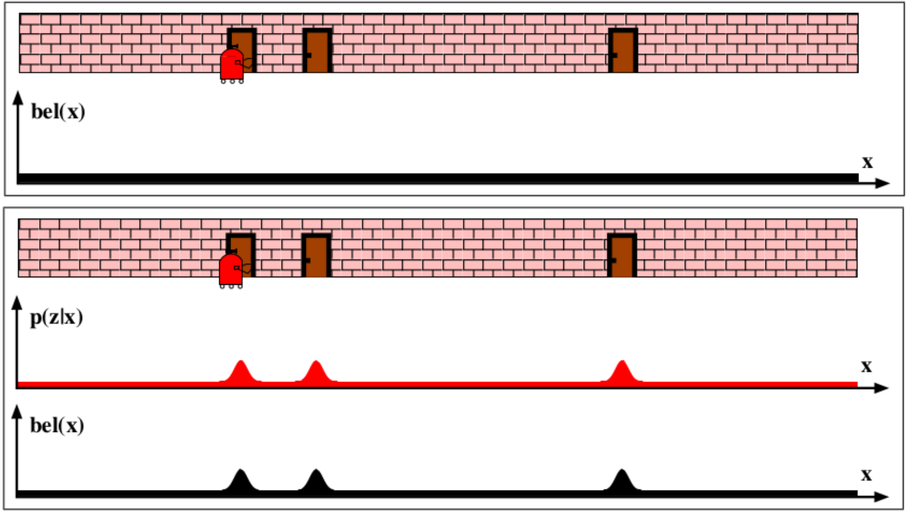
- Bayes Filter Framework
- Given
  - Streams (or history) of observations z and actions u, $h_t = u_1, z_1, \ldots, u_t, z_t$
  - **Sensor (observation) model** $P(z|x)$
  - **Action (motion) model** $P(x|u, x')$
  - **Prior** probability of system state $P(x)$
- Goal
  - Estimate of the state X of a **dynamical system**
  - The **posterior** of the state also clled **Belief**

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

Recursive Bayes Filter: Example I

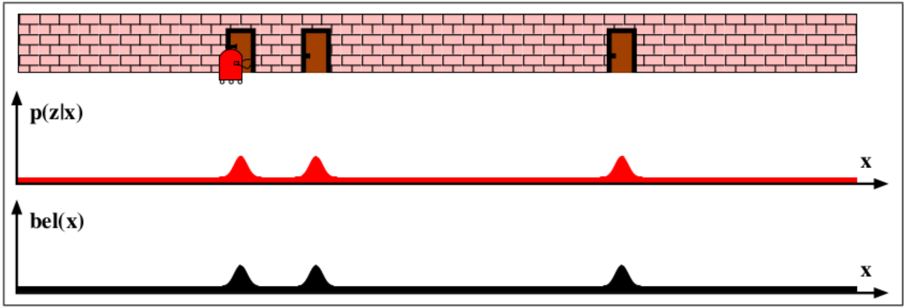


Recursive Bayes Filter: Example I

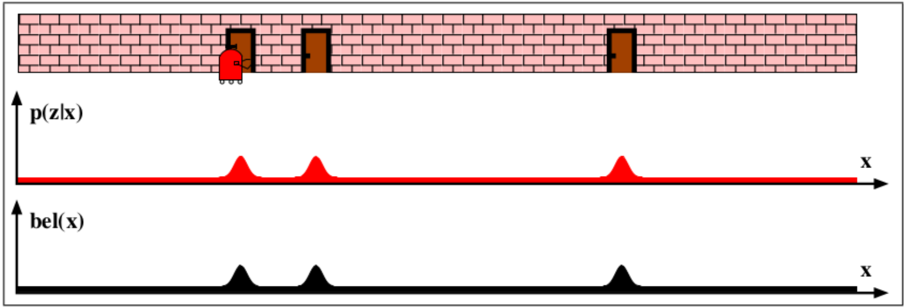


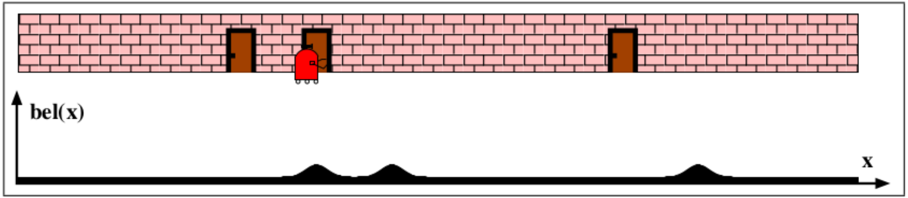
Source [PR]

Recursive Bayes Filter: Example II



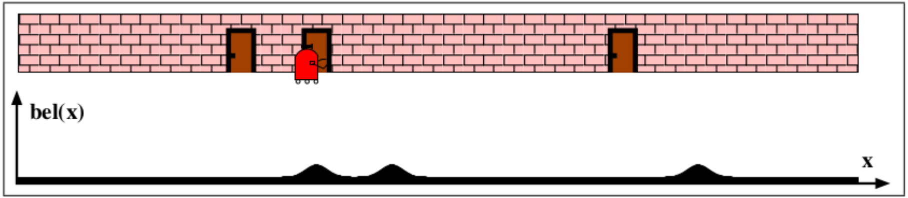
Recursive Bayes Filter: Example II



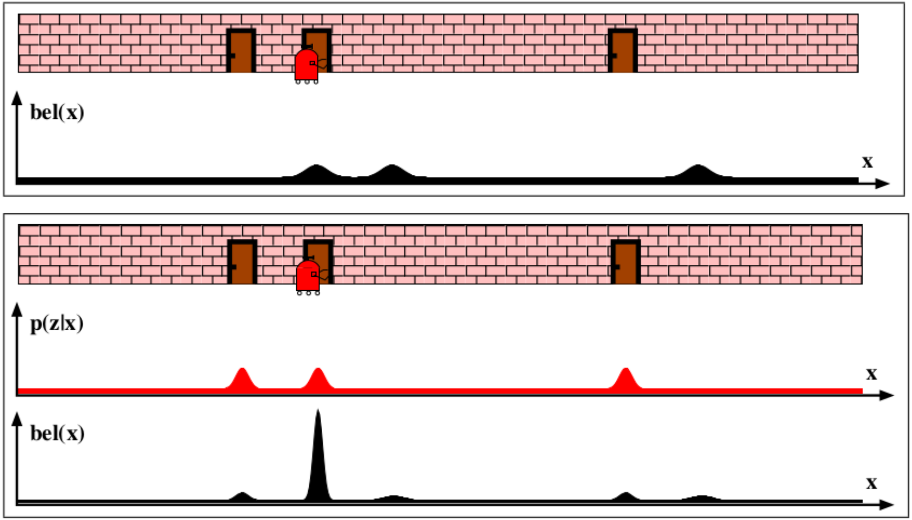


Source [PR]

Recursive Bayes Filter: Example III

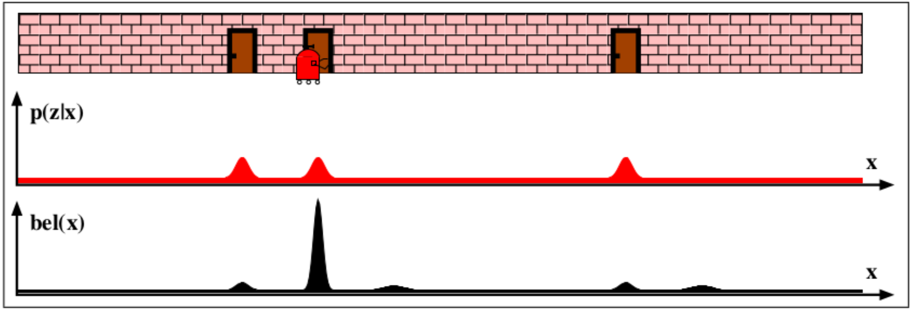


Recursive Bayes Filter: Example III

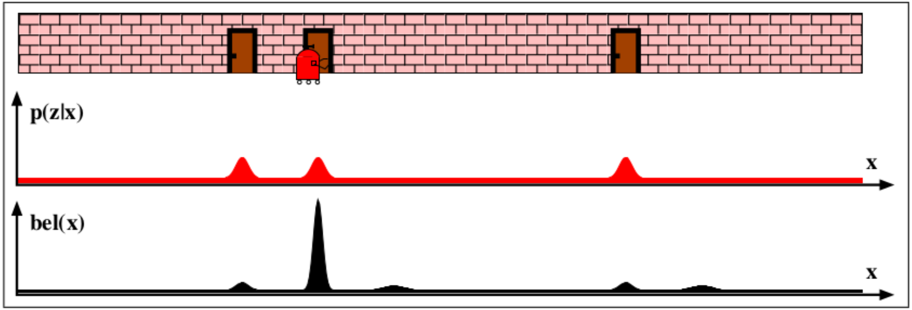


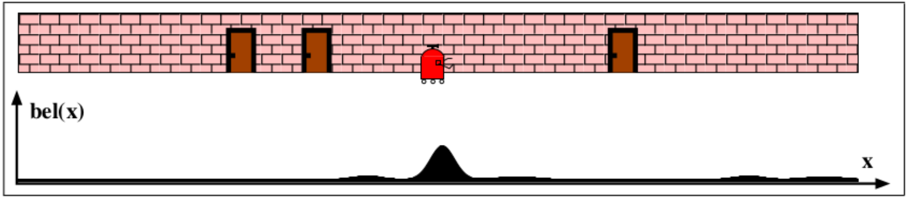
Source [PR]

Recursive Bayes Filter: Example IV



Recursive Bayes Filter: Example IV





Source [PR]

Markov assumption

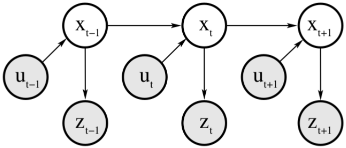
◇ Conditional independence of current data from data history **given** state

◇ Conditional independence of current observation from history (previous observations, commands) **given** current state

$$P(z_t|x_t, z_{1:t-1}, u_{1:t}) = P(z_t|x_t)$$

◇ Conditional independence of current state from history (previous observations, commands) **given** previous state and current command

$$P(x_t|x_{t-1}, z_{1:t-1}, u_{1:t}) = P(x_t|x_{t-1}, u_t)$$



Dynamic Bayesian network for the evolution of commands, states, and observations (Source [PR])

Recursive Bayes Filter derivation

$$Bel(x_t) = P(x_t | u_{1:t}, z_{1:t})$$

$$= \eta P(z_t | u_{1:t}, z_{1:t-1}, x_t) P(x_t | u_{1:t}, z_{1:t-1}) \text{ Bayes}$$

Recursive Bayes Filter derivation

$$Bel(x_t) = P(x_t | u_{1:t}, z_{1:t})$$

$$= \eta P(z_t | u_{1:t}, z_{1:t-1}, x_t) P(x_t | u_{1:t}, z_{1:t-1}) \text{ Bayes}$$

$$= \eta P(z_t | x_t) P(x_t | u_{1:t}, z_{1:t-1}) \text{ Markov (1)}$$

Recursive Bayes Filter derivation

$$Bel(x_t) = P(x_t | u_{1:t}, z_{1:t})$$

$$= \eta P(z_t | u_{1:t}, z_{1:t-1}, x_t) P(x_t | u_{1:t}, z_{1:t-1}) \text{ Bayes}$$

$$= \eta P(z_t | x_t) P(x_t | u_{1:t}, z_{1:t-1}) \text{ Markov (1)}$$

$$= \eta P(z_t | x_t) \int P(x_t | u_{1:t}, z_{1:t-1}, x_{t-1}) P(x_{t-1} | u_{1:t}, z_{1:t-1}) dx_{t-1} \text{ Total prob.}$$

Recursive Bayes Filter derivation

$$Bel(x_t) = P(x_t | u_{1:t}, z_{1:t})$$

$$= \eta P(z_t | u_{1:t}, z_{1:t-1}, x_t) P(x_t | u_{1:t}, z_{1:t-1}) \text{ Bayes}$$

$$= \eta P(z_t | x_t) P(x_t | u_{1:t}, z_{1:t-1}) \text{ Markov (1)}$$

$$= \eta P(z_t | x_t) \int P(x_t | u_{1:t}, z_{1:t-1}, x_{t-1}) P(x_{t-1} | u_{1:t}, z_{1:t-1}) dx_{t-1} \text{ Total prob.}$$

$$= \eta P(z_t | x_t) \int P(x_t | u_t, x_{t-1}) P(x_{t-1} | u_{1:t}, z_{1:t-1}) dx_{t-1} \text{ Markov (2)}$$

Recursive Bayes Filter derivation

$$Bel(x_t) = P(x_t|u_{1:t}, z_{1:t})$$

$$= \eta P(z_t|u_{1:t}, z_{1:t-1}, x_t)P(x_t|u_{1:t}, z_{1:t-1}) \text{ Bayes}$$

$$= \eta P(z_t|x_t)P(x_t|u_{1:t}, z_{1:t-1}) \text{ Markov (1)}$$

$$= \eta P(z_t|x_t) \int P(x_t|u_{1:t}, z_{1:t-1}, x_{t-1})P(x_{t-1}|u_{1:t}, z_{1:t-1})dx_{t-1} \text{ Total prob.}$$

$$= \eta P(z_t|x_t) \int P(x_t|u_t, x_{t-1})P(x_{t-1}|u_{1:t}, z_{1:t-1})dx_{t-1} \text{ Markov (2)}$$

$$= \eta P(z_t|x_t) \int P(x_t|u_t, x_{t-1})P(x_{t-1}|u_{1:t-1}, z_{1:t-1})dx_{t-1} \text{ Independence from } u_t$$

Recursive Bayes Filter derivation

$$Bel(x_t) = P(x_t|u_{1:t}, z_{1:t})$$

$$= \eta P(z_t|u_{1:t}, z_{1:t-1}, x_t)P(x_t|u_{1:t}, z_{1:t-1}) \text{ Bayes}$$

$$= \eta P(z_t|x_t)P(x_t|u_{1:t}, z_{1:t-1}) \text{ Markov (1)}$$

$$= \eta P(z_t|x_t) \int P(x_t|u_{1:t}, z_{1:t-1}, x_{t-1})P(x_{t-1}|u_{1:t}, z_{1:t-1})dx_{t-1} \text{ Total prob.}$$

$$= \eta P(z_t|x_t) \int P(x_t|u_t, x_{t-1})P(x_{t-1}|u_{1:t}, z_{1:t-1})dx_{t-1} \text{ Markov (2)}$$

$$= \eta P(z_t|x_t) \int P(x_t|u_t, x_{t-1})P(x_{t-1}|u_{1:t-1}, z_{1:t-1})dx_{t-1} \text{ Independence from } u_t$$

$$= \eta P(z_t|x_t) \int P(x_t|u_t, x_{t-1})Bel(x_{t-1})dx_{t-1} \textbf{ Recursive}$$

Bayes Filter as a two step process

- Prediction and correction step

$$Bel(x_t) = \eta P(z_t|x_t) \int P(x_t|u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

- Prediction Step:

$$\overline{Bel(x_t)} = \int P(x_t|u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

- Correction Step:

$$Bel(x_t) = \eta P(z_t|x_t) \overline{Bel(x_t)}$$

Bayes Filter, different realizations

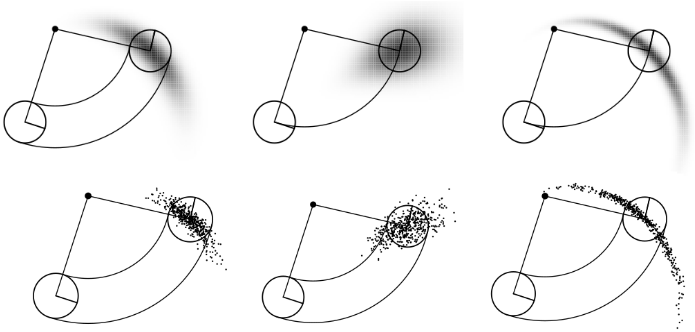
- Bayes filter is a **framework** for recursive state estimation
- Many **different realizations**
- Various properties/assumptions
  - Linear vs. non linear motion and observation models
  - Gaussian distribution for the belief
  - parametric vs. non parametric filters
  - ...

Popular Filters

- Kalman filters and EKF
  - Gaussians
  - Linear of linearized models
- Particle filters
  - Non-parametric
  - Arbitrary models (sampling required)

Example of motion models

$$\overline{Bel(x_t)} = \int P(x_t | u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

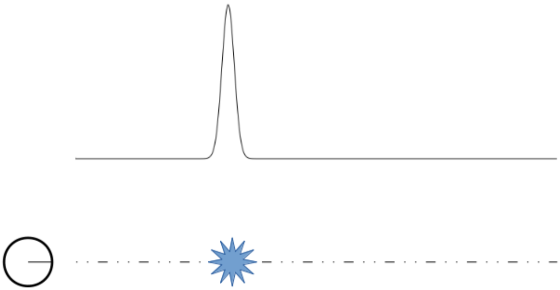


Different types of motion models (source [PR])

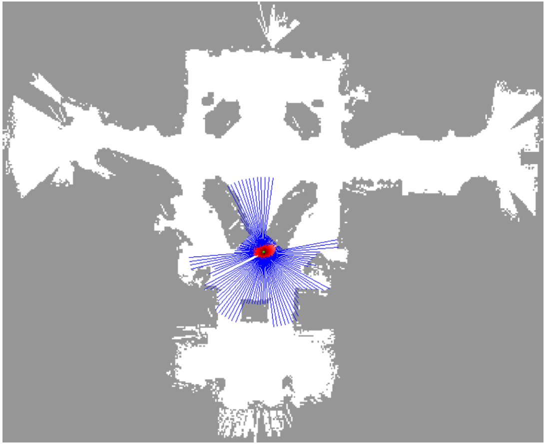
Example of observation model

$$Bel(x_t) = \eta P(z_t|x_t)\overline{Bel(x_t)}$$

- Range sensor estimating distance to closest object
- Gaussian noise in range reading



Bayes Filter in Action



Monte Carlo Localization in the Smithsonian Museum (Courtesy of Sebastian Thrun)