Foundations of AI

15. Robotics

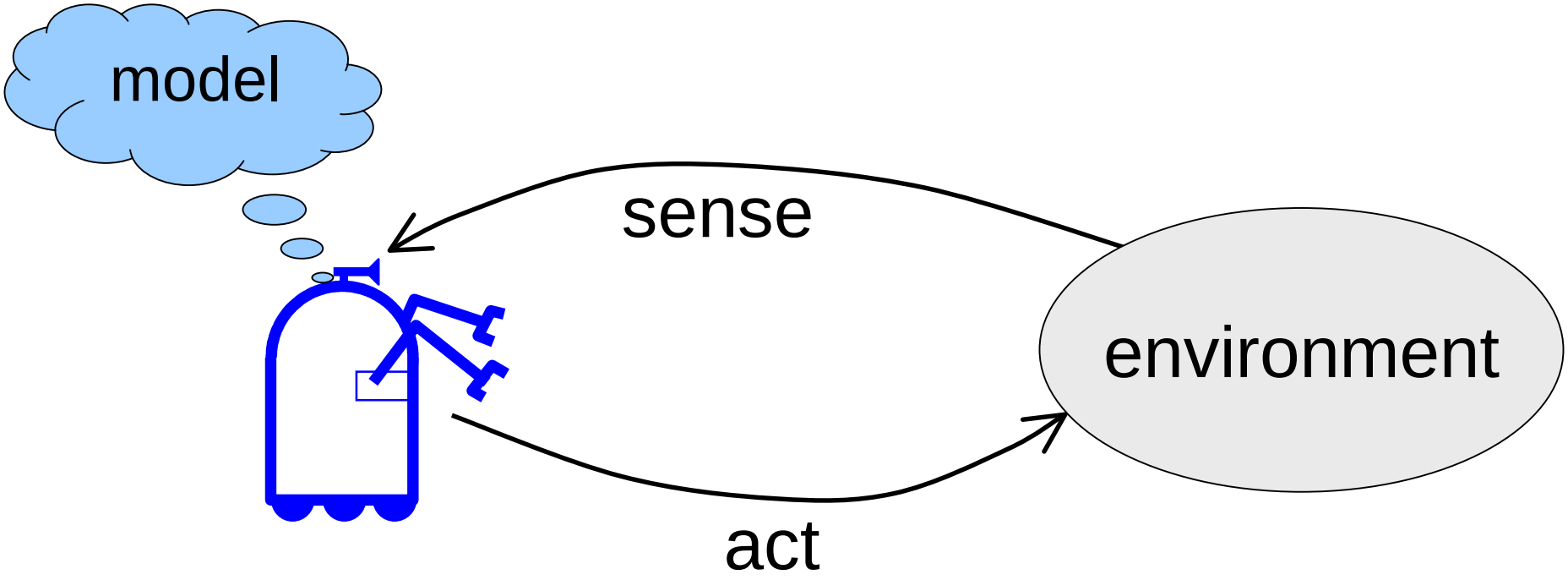
State Estimation and Filtering

Wolfram Burgard, Bernhard Nebel, and Luc De Raedt

Robots

Physical agents that

- perceive their environment and
- generate actions to achieve their goals



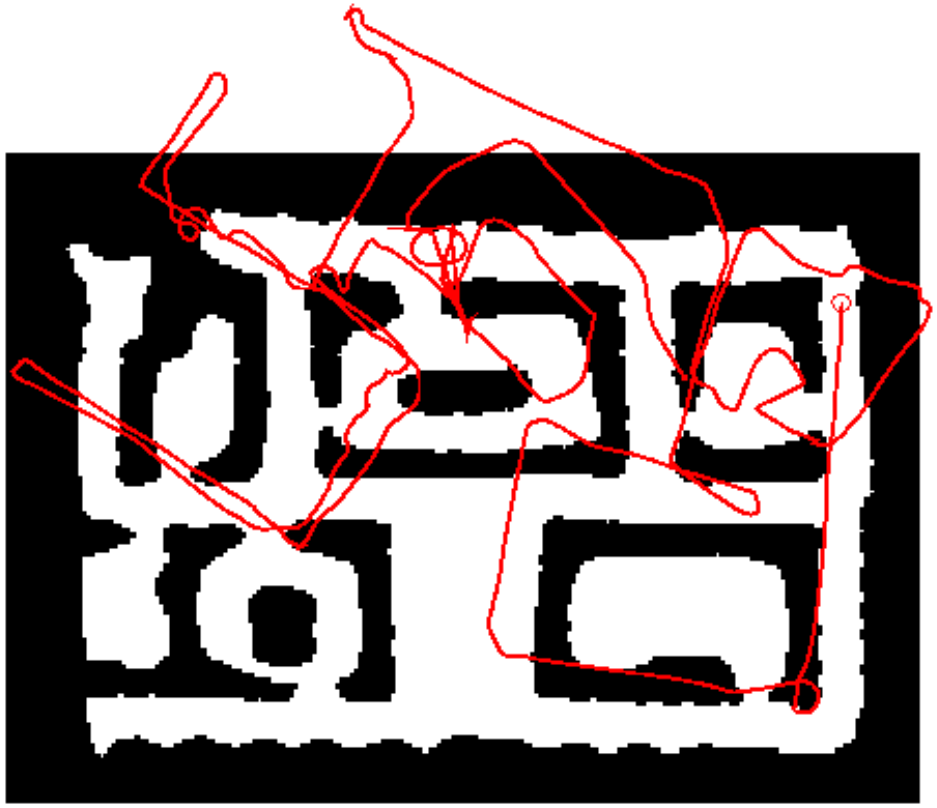
Physical Agents are Inherently Uncertain

Uncertainty arises from four major factors:

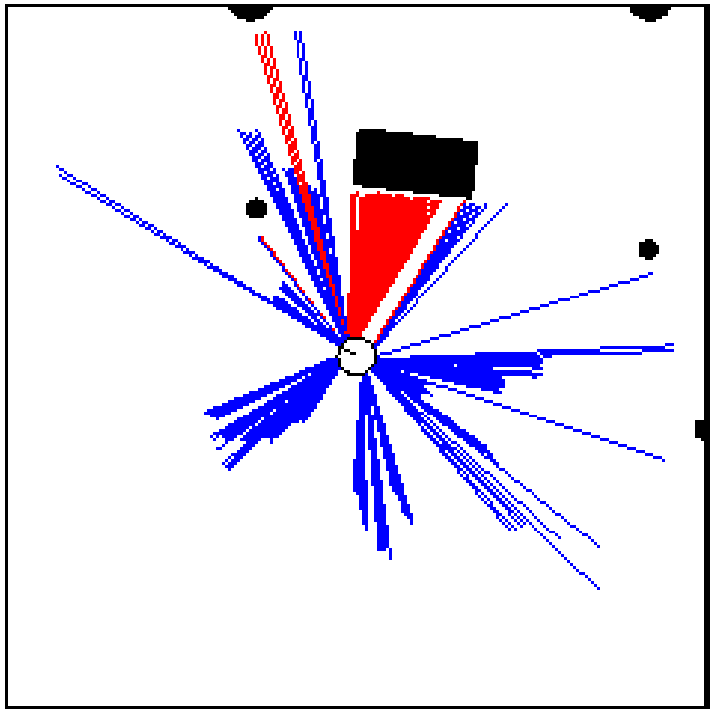
- Environment stochastic, unpredictable
- Robot stochastic
- Sensor limited, noisy
- Models inaccurate

Example: Rhino

Nature of Sensor Data



Odometry Data



Range Data

Probabilistic Techniques for Physical Agents

Key idea:
Explicit representation of uncertainty using the calculus of probability theory

Perception	=	state estimation
Action	=	utility optimization

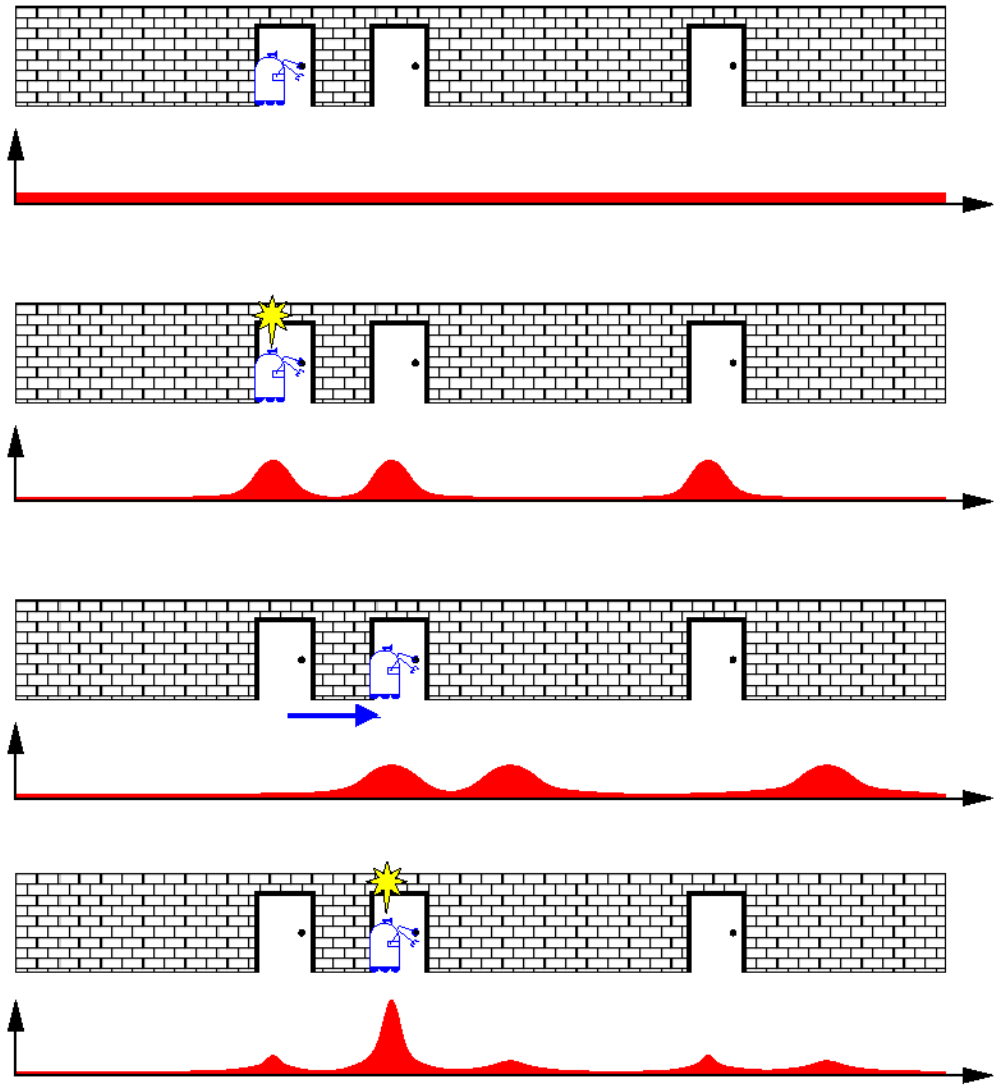
Advantages of Probabilistic Paradigm

- Can accommodate inaccurate models
- Can accommodate imperfect sensors
- Robust in real-world applications
- Best known approach to many hard robotics problems

Pitfalls

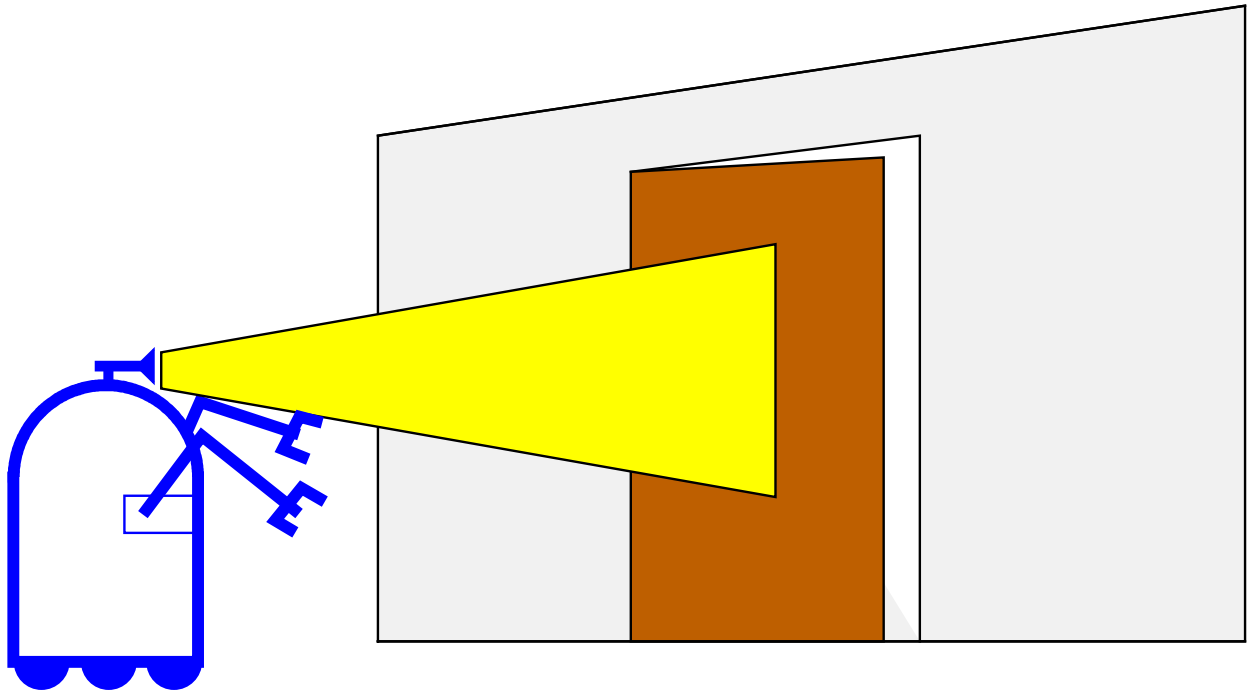
- Computationally demanding
- False assumptions
- Approximate

Probabilistic Localization



Simple Example of State Estimation

- Suppose a robot obtains measurement z
- What is *P(open|z)?*



Causal vs. Diagnostic Reasoning

- *P(open|z)* is diagnostic.
- *P(z|open)* is causal.
- Often causal knowledge is easier to obtain.
- Bayes rule allows us to use causal knowledge:

count frequencies!

$$P(open \mid z) = \frac{P(z \mid open)P(open)}{P(z)}$$

Example

- $P(z|open) = 0.6$ $\quad\quad$ $P(z|\neg open) = 0.3$
- $P(open) = P(\neg open) = 0.5$

$$P(open \mid z) = \frac{P(z \mid open)P(open)}{P(z \mid open)\,p(open) + P(z \mid \neg open)\,p(\neg open)}$$

$$P(open \mid z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67$$

z raises the probability that the door is open.

Combining Evidence

- Suppose our robot obtains another observation z_2.
- How can we integrate this new information?
- More generally, how can we estimate $P(x| z_1,\ldots, z_n)$?

Recursive Bayesian Updating

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x, z_1, \ldots, z_{n-1})\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

Markov assumption: z_n is independent of $z_1,\ldots,z_{n-1}$ if we know x.

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

$$= \eta\, P(z_n \mid x)\, P(x \mid z_1, \ldots, z_{n-1})$$

$$= \eta_{1\ldots n} \prod_{i=1\ldots n} P(z_i \mid x)\, P(x)$$

Example: Second Measurement

- $P(z_2|open) = 0.5$ $\quad\quad$ $P(z_2|\neg open) = 0.6$
- $P(open|z_1)=2/3$

$$P(open \mid z_2, z_1) = \frac{P(z_2 \mid open)\, P(open \mid z_1)}{P(z_2 \mid open)\, P(open \mid z_1) + P(z_2 \mid \neg open)\, P(\neg open \mid z_1)}$$

$$= \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625$$

z_2 lowers the probability that the door is open.

Actions

- Often the world is dynamic because of
  - actions carried out by the robot,
  - actions carried out by other agents,
  - or just the time passing by change the world.

- How can we incorporate such actions?

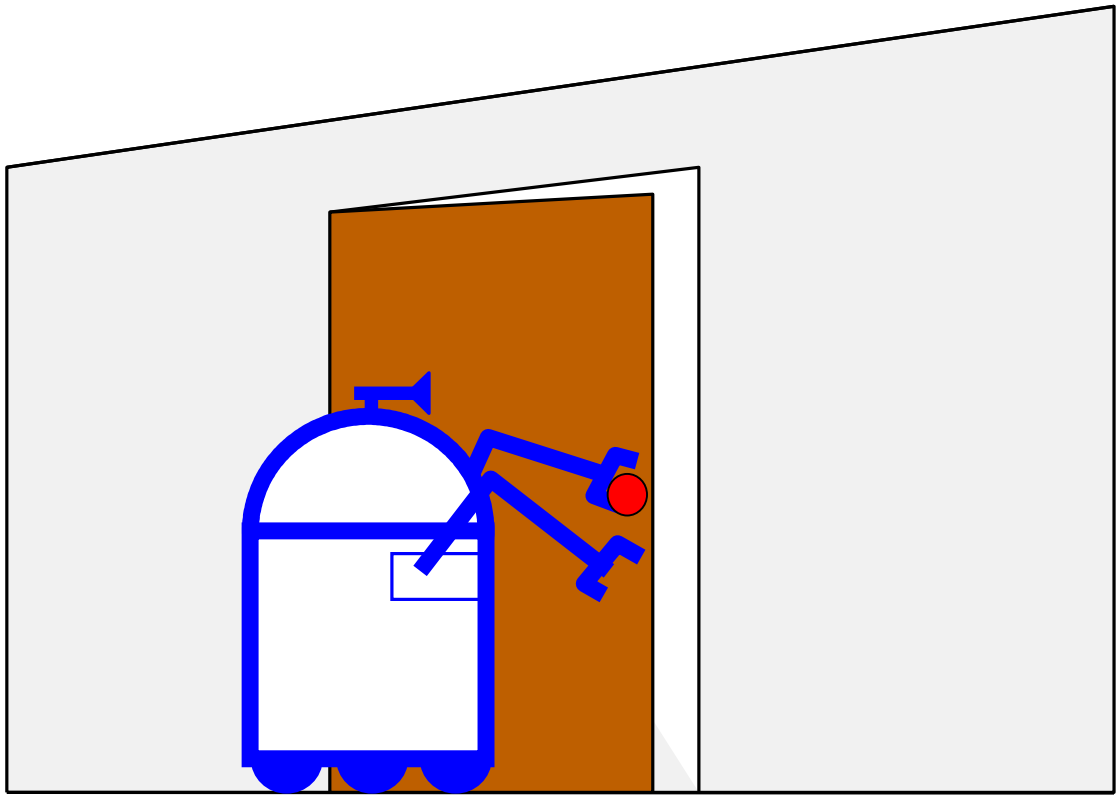
Modeling Actions

- To incorporate the outcome of an action *u* into the current "belief", we use the conditional pdf

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

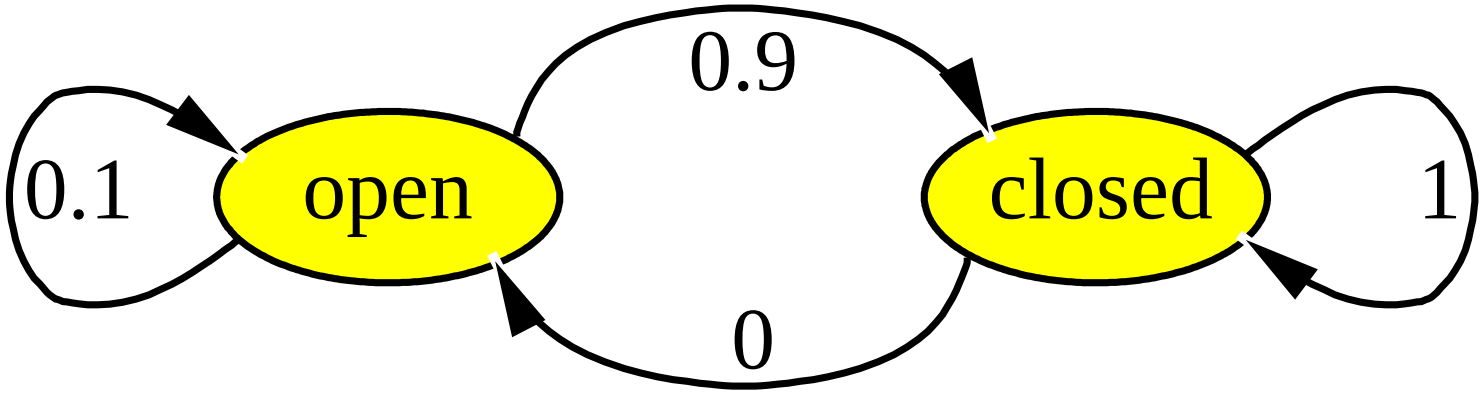
- This term specifies the pdf that executing *u* changes the state from *x' to x*.

Example: Closing the Door



State Transitions

$P(x|u,x')$ for u = "close door":



If the door is open, the action "close door" succeeds in 90% of all cases.

Integrating the Outcome of Actions

Continuous case:

$$P(x \mid u) = \int P(x \mid u, x') P(x') dx'$$

Discrete case:

$$P(x \mid u) = \sum P(x \mid u, x') P(x')$$

Example: The Resulting Belief

$$
\begin{aligned}
P(closed \mid u) &= \sum P(closed \mid u, x')P(x') \\
&= P(closed \mid u, open)P(open) \\
&\quad + P(closed \mid u, closed)P(closed) \\
&= \frac{9}{10} * \frac{5}{8} + \frac{1}{1} * \frac{3}{8} = \frac{15}{16} \\
P(open \mid u) &= \sum P(open \mid u, x')P(x') \\
&= P(open \mid u, open)P(open) \\
&\quad + P(open \mid u, closed)P(closed) \\
&= \frac{1}{10} * \frac{5}{8} + \frac{0}{1} * \frac{3}{8} = \frac{1}{16} \\
&= 1 - P(closed \mid u)
\end{aligned}
$$

Bayes Filters: Framework

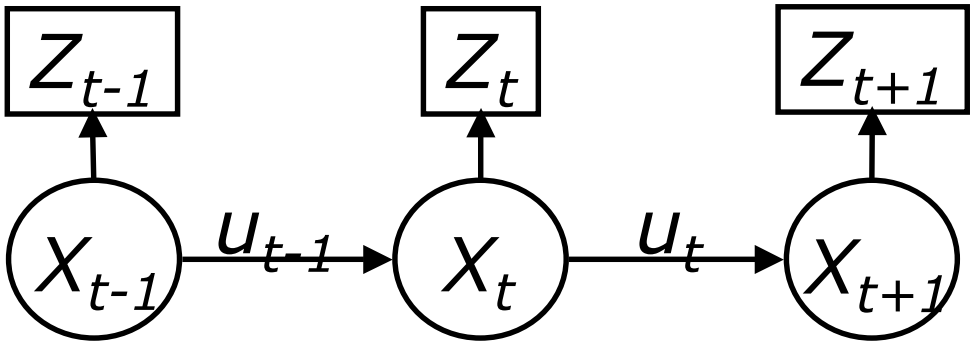
- Given:
  - Stream of observations *z* and action data *u:*

$$d_t = \{u_1, z_2 \text{😐}, u_{t-1}, z_t\}$$

  - Sensor model *P(z|x).*
  - Action model *P(x|u,x')*.
  - Prior probability of the system state *P(x).*
- Wanted:
  - Estimate of the state *X* of a dynamical system.
  - The posterior of the state is also called Belief:

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

Markov Assumption



$$p(d_t, d_{t-1}, \ldots d_0 \mid x_t, d_{t+1}, d_{t+2}, \ldots) = p(d_t, d_{t-1}, \ldots d_0 \mid x_t)$$

$$p(d_t, d_{t+1}, \ldots \mid x_t, d_1, d_2, \ldots d_{t-1}) = p(d_t, d_{t+1}, \ldots \mid x_t)$$

$$p(x_t \mid u_{t-1}, x_{t-1}, d_{t-2}, \ldots, d_0) = p(x_t \mid u_{t-1}, x_{t-1})$$

Underlying Assumptions

- Static world
- Independent noise
- Perfect model, no approximation errors

z = observation
u = action
x = state

Bayes Filters

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

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

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

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

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

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

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

1. Algorithm **Bayes_filter**(*Bel(x),d*):
2. η=0
3. if *d* is a perceptual data item *z* then
4. For all *x* do
5. $Bel'(x) = P(z \mid x) Bel(x)$
6. $\eta = \eta + Bel'(x)$
7. For all *x* do
8. $Bel'(x) = \eta^{-1} Bel'(x)$
9. else if *d* is an action data item *u* then
10. For all *x* do
11. $Bel'(x) = \int P(x \mid u, x')\ Bel(x')\ dx'$
12. return *Bel'(x)*

Bayes Filters are Familiar!

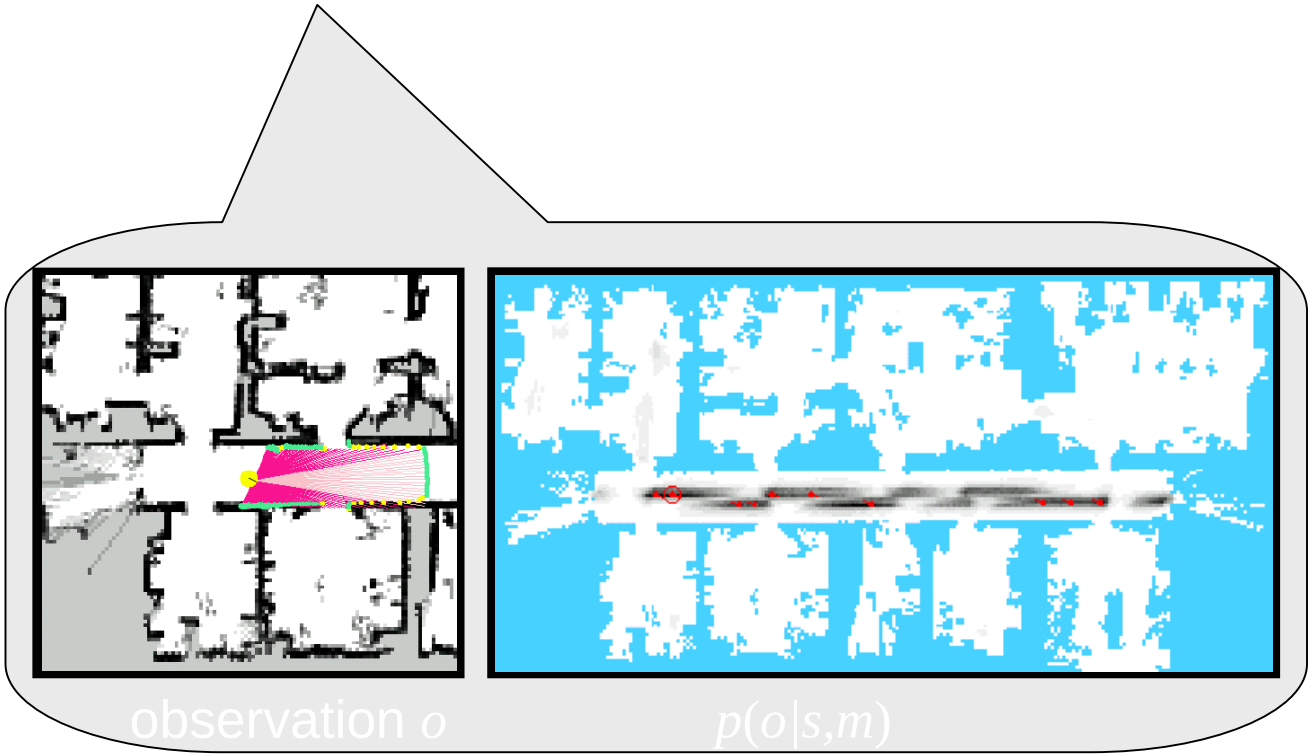
$$Bel(x_t) = \eta\ P(z_t \mid x_t) \int P(x_t \mid u_{t-1}, x_{t-1})\ Bel(x_{t-1})\ dx_{t-1}$$

- Kalman filters
- Particle filters
- Hidden Markov models
- Dynamic Bayes networks
- Partially Observable Markov Decision Processes (POMDPs)

Localization with Bayes Filters (1)

$$bel(x_t) = \eta\, p(s_t \mid x_t) \int p(x_t \mid x_{t-1}, a_{t-1})\, bel(x_{t-1})\, dx_{t-1}$$

$$\Rightarrow bel(x_t \mid m) = \eta\, p(s_t \mid x_t, m) \int p(x_t \mid x_{t-1}, a_{t-1}, m)\, bel(x_{t-1} \mid m)\, dx_{t-1}$$



Localization with Bayes Filters (2)

$$bel(x_t) = \eta\, p(s_t \mid x_t) \int p(x_t \mid x_{t-1}, a_{t-1})\, bel(x_{t-1})\, dx_{t-1}$$

$$\Rightarrow bel(x_t \mid m) = \eta\, p(s_t \mid x_t, m) \int p(x_t \mid x_{t-1}, a_{t-1}, m)\, bel(x_{t-1} \mid m)\, dx_{t-1}$$

Motion: $$bel^{-}(x_t) = \int p(x_t \mid x_{t-1}, a_{t-1})\, bel(x_{t-1})\, dx_{t-1}$$

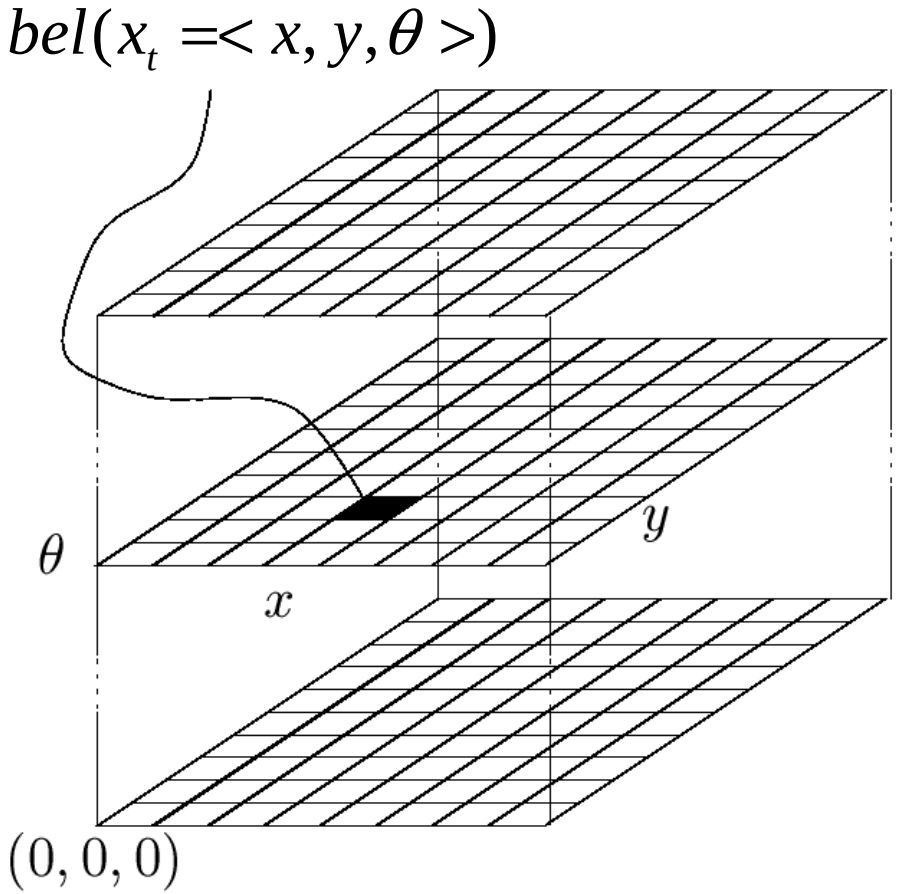
Perception: $$bel(x_t) = \eta\, p(s_t \mid x_t)\, bel^{-}(x_t)$$

… is optimal under the Markov assumption

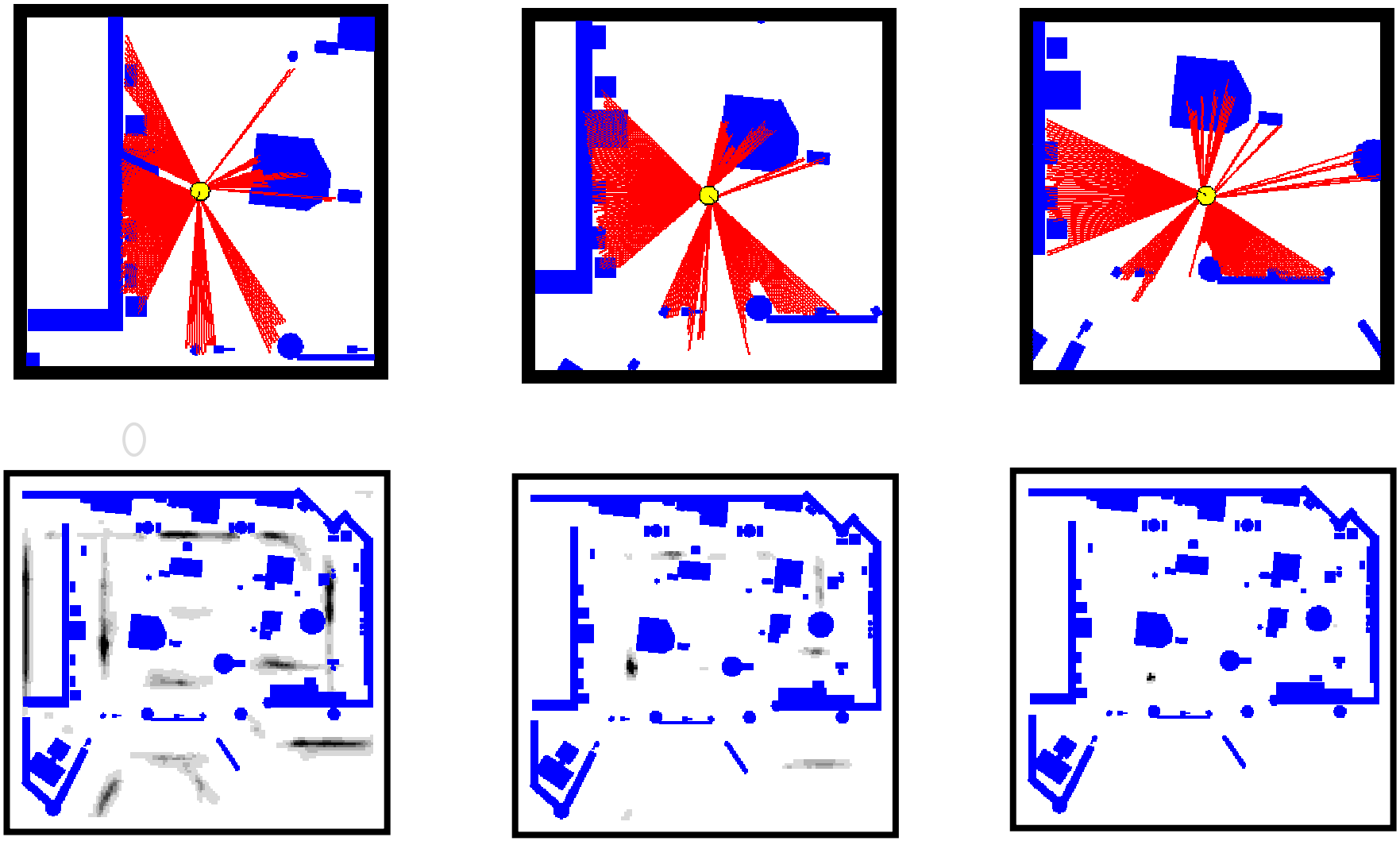
What is the Right Representation?

- Kalman filters
- Multi-hypothesis tracking
- Grid-based representations
- Topological approaches
- Particle filters

Piecewise Constant Representation (Histograms)



Grid-based Localization



Particle Filters

- Represent density by random samples
- Estimation of non-Gaussian, nonlinear processes

- Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter, Particle filter

- Filtering: [Rubin, 88], [Gordon et al., 93], [Kitagawa 96]
- Computer vision: [Isard and Blake 96, 98]
- Dynamic Bayesian Networks: [Kanazawa et al., 95]

Monte-Carlo Localization

- Set of N samples $\{<x_1,w_1>, \ldots <x_N,w_N>\}$ containing a state x and an importance weight w
- Initialize sample set according to prior knowledge
- For each motion u do:
  - Sampling: Generate from each sample a new pose according to the motion model $p(x \mid u, x')$

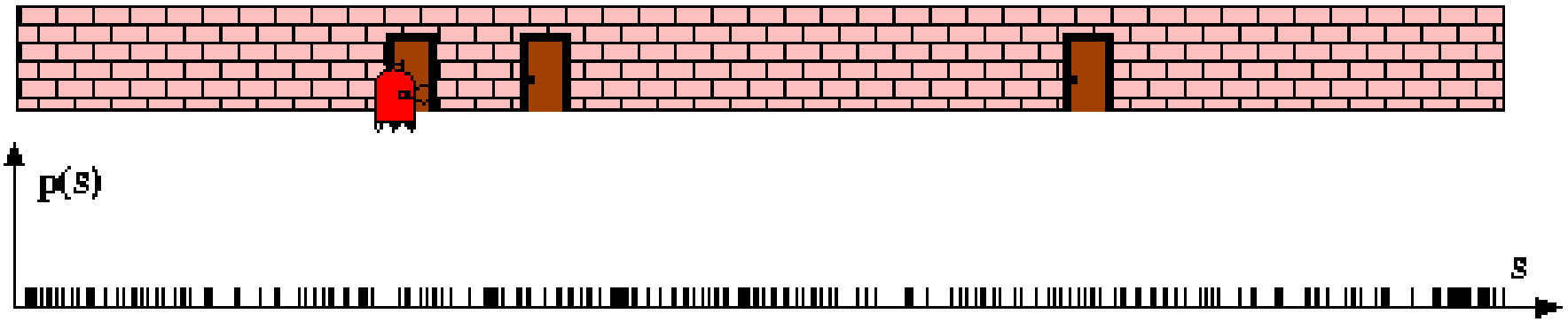
$$x_n \leftarrow x_n + u'$$

- For each observation z do:
  - Importance sampling: weigh each sample with the likelihood

$$w_n \leftarrow \alpha \, p(z \mid x_n)$$

  - Re-sampling: Draw N *new* samples from the sample set according to the importance weights

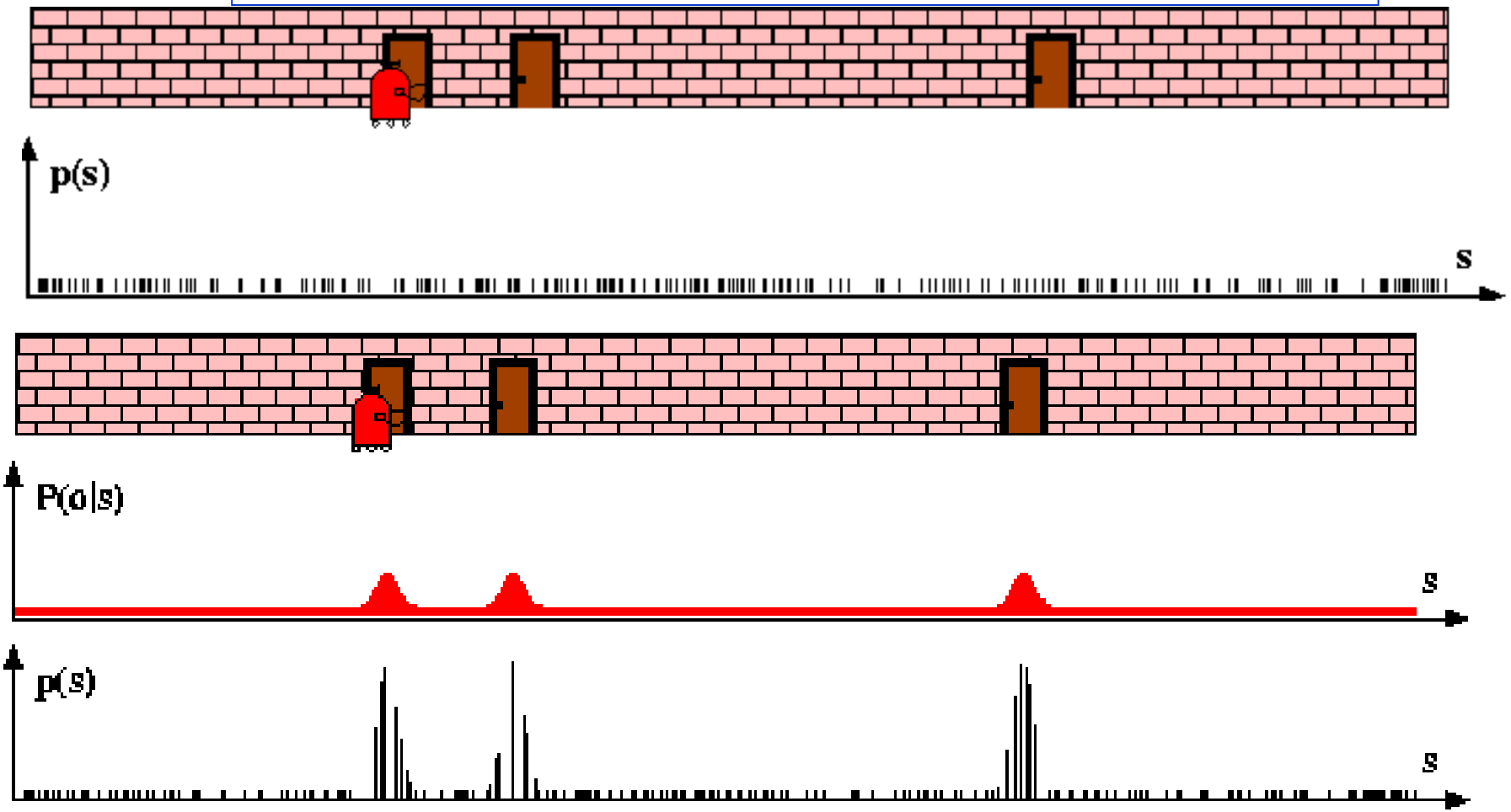
MCL: Global Localization



MCL: Sensor Update

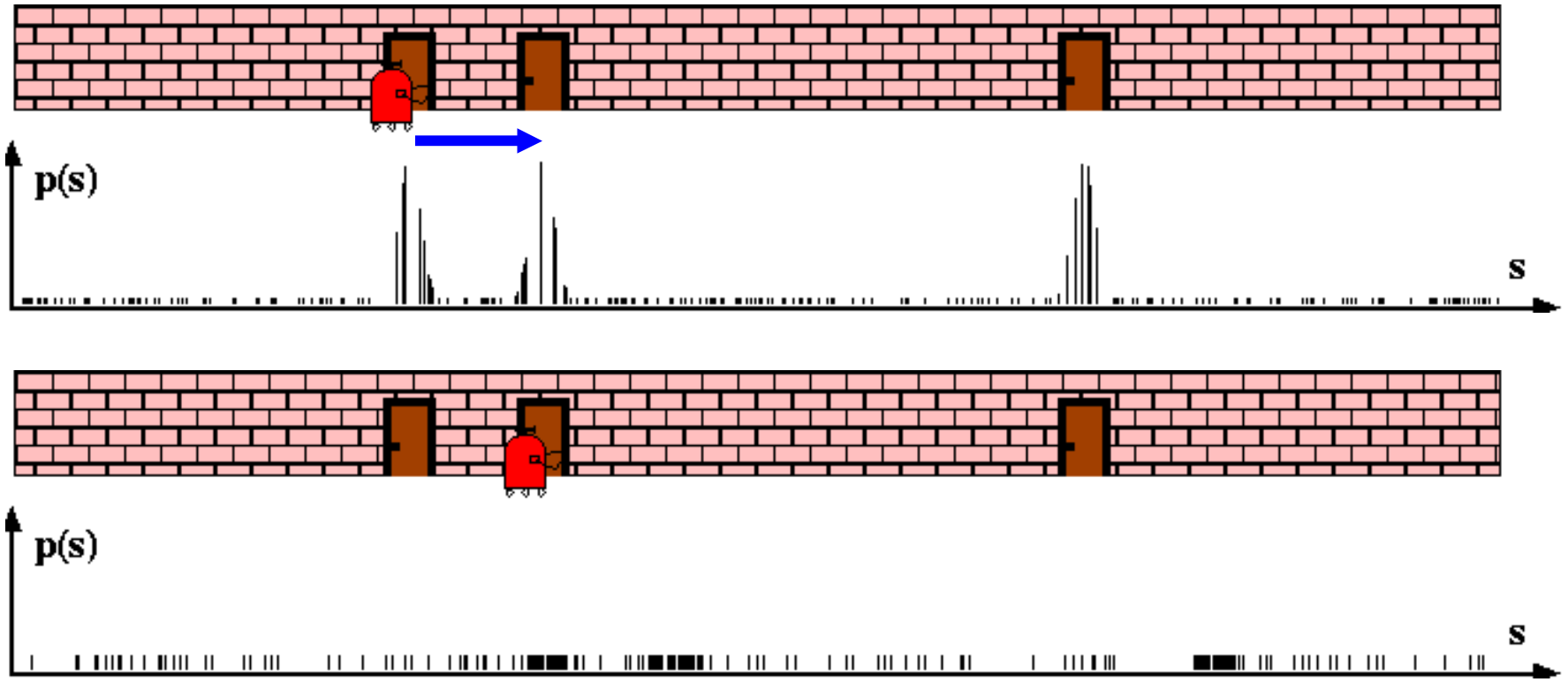
$$Bel(x) \leftarrow \alpha\ p(z \mid x)\ Bel^{-}(x)$$

$$w \leftarrow \frac{\alpha\ p(z \mid x)\ Bel^{-}(x)}{Bel^{-}(x)} = \alpha\ p(z \mid x)$$



MCL: Robot Motion

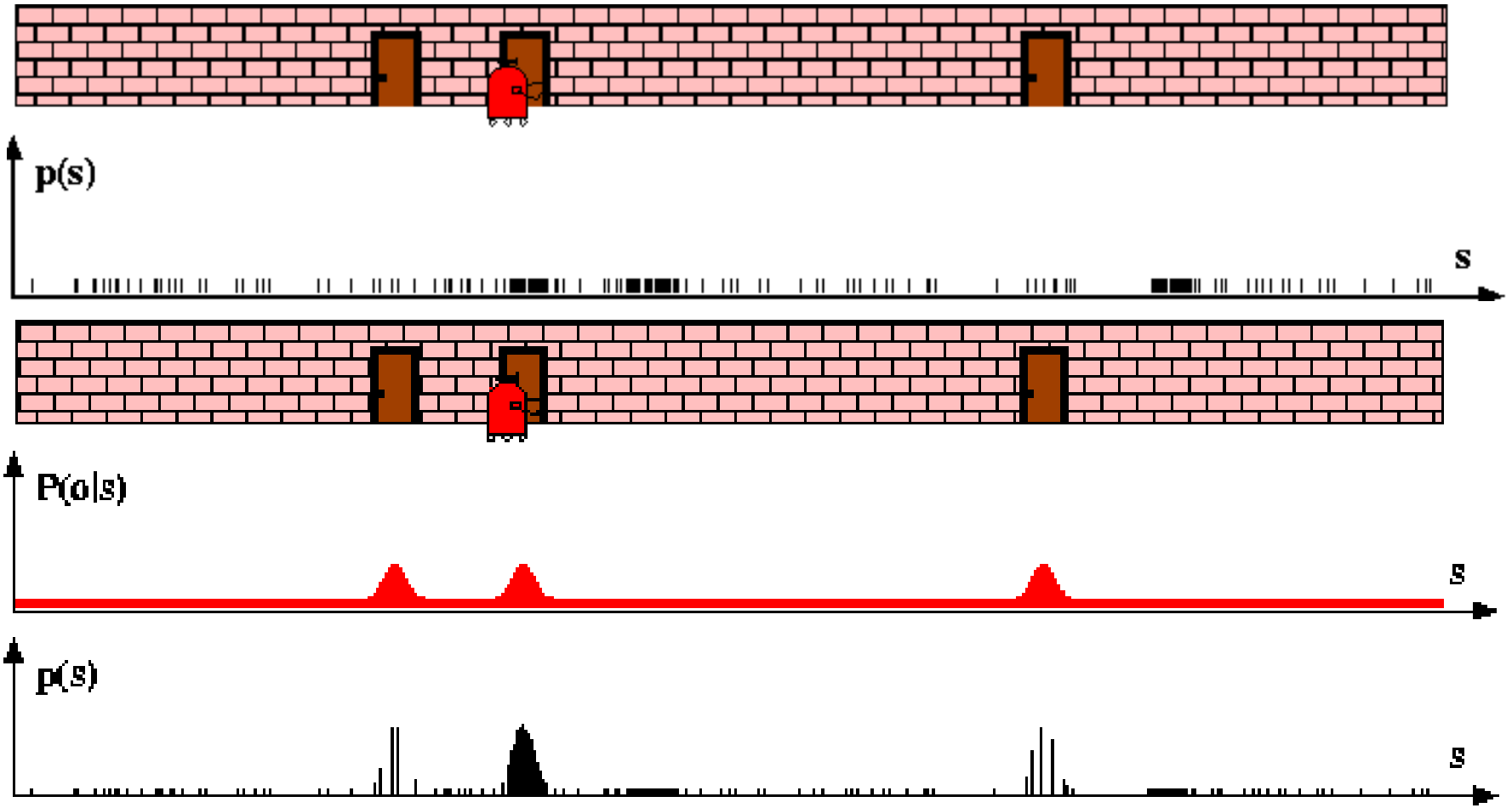
$$Bel^{-}(x) \quad \leftarrow \quad \int p(x \mid u, x') \, Bel(x') \;\; \mathrm{d}\, x'$$



MCL: Sensor Update

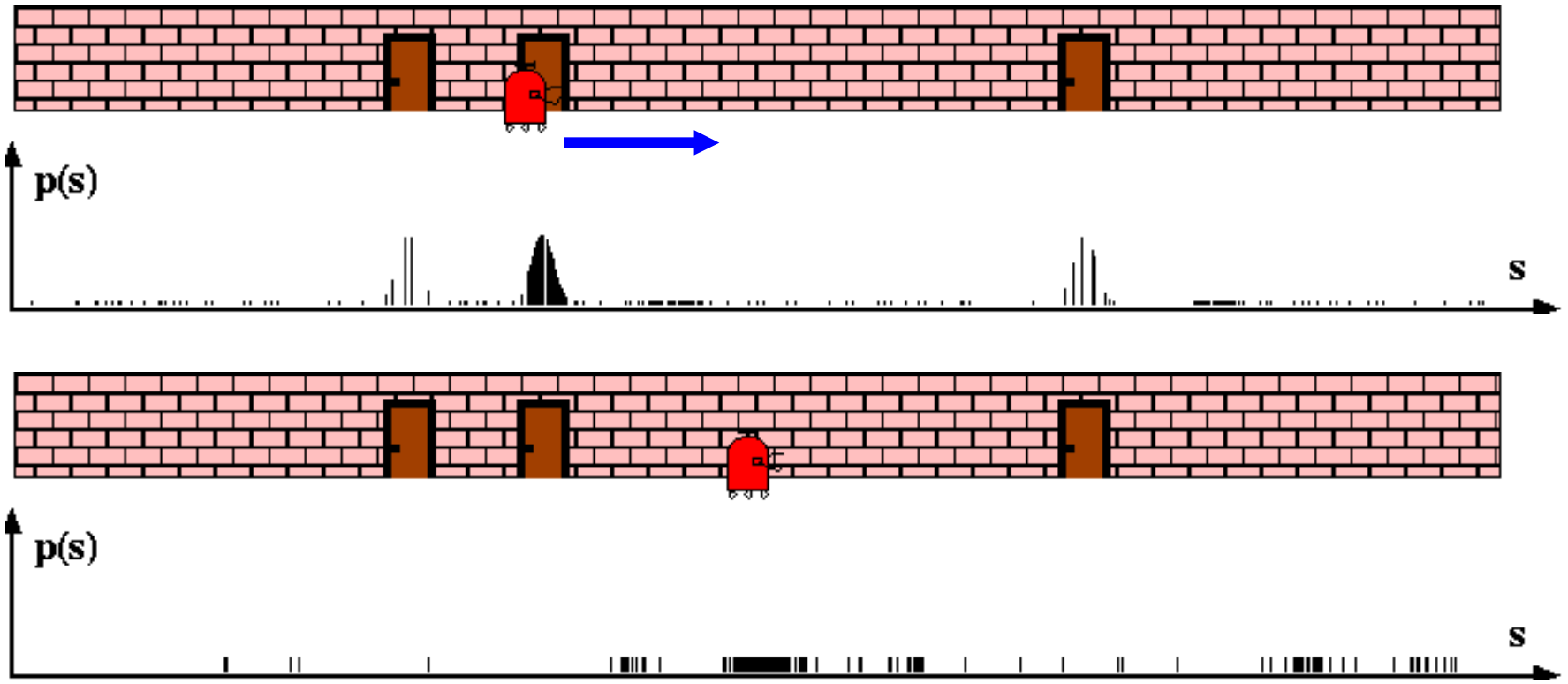
$$Bel(x) \leftarrow \alpha\ p(z \mid x)\ Bel^{-}(x)$$

$$w \leftarrow \frac{\alpha\ p(z \mid x)\ Bel^{-}(x)}{Bel^{-}(x)} = \alpha\ p(z \mid x)$$



MCL: Robot Motion

$$Bel^{-}(x) \quad \leftarrow \quad \int p(x \mid u, x') \, Bel(x') \;\; \mathrm{d}\, x'$$



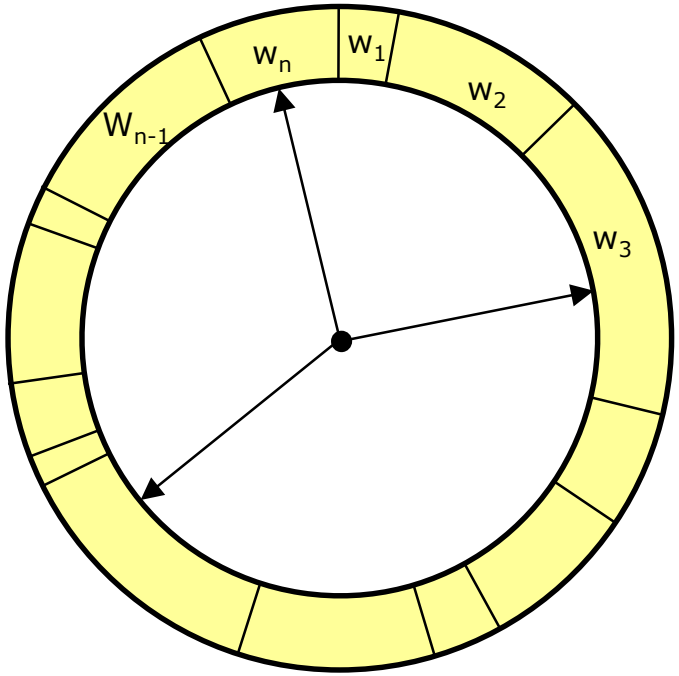
Particle Filter Algorithm

1. Algorithm **particle_filter**(S_{t-1}, u_{t-1} z_t):
2. $S_t = \emptyset, \quad \eta = 0$
3. **For** $i = 1 \text{😐} n$ ***Generate new samples***
4. Sample index *j(i)* from the discrete distribution given by w_{t-1}
5. Sample x_t^i from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x_{t-1}^{j(i)}$ and u_{t-1}
6. $w_t^i = p(z_t \mid x_t^i)$ ***Compute importance weight***
7. $\eta = \eta + w_t^i$ ***Update normalization factor***
8. $S_t = S_t \cup \{< x_t^i, w_t^i >\}$ ***Insert***
9. **For** $i = 1 \text{😐} n$
10. $w_t^i = w_t^i / \eta$ ***Normalize weights***

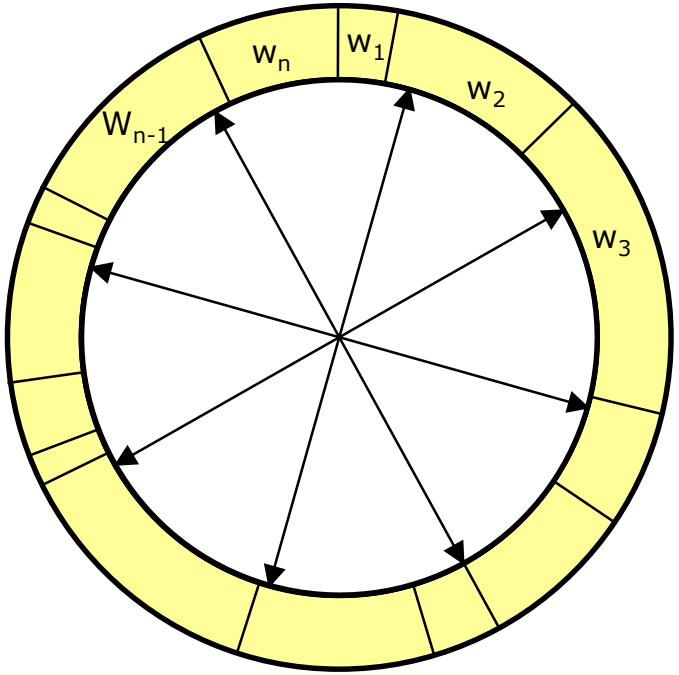
Resampling

- Given: Set S of weighted samples.
- Wanted : Random sample, where the probability of drawing x_i is given by w_i.
- Typically done n times with replacement to generate new sample set S'.

Resampling

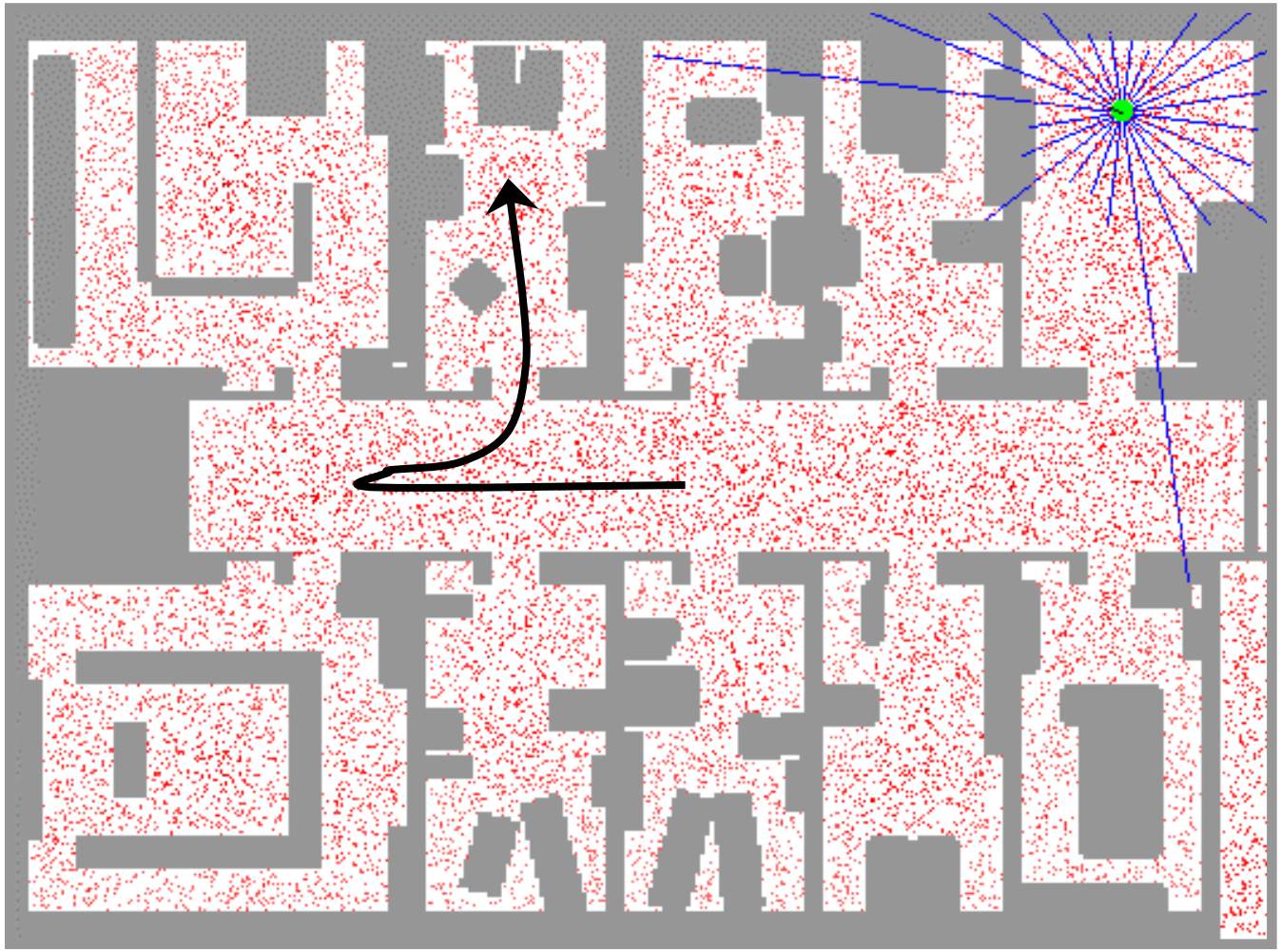


- Roulette wheel
- Binary search, log n



- Stochastic universal sampling
- Systematic resampling
- Linear time complexity
- Easy to implement, low variance

MCL: Global Localization (Sonar)



Summary

- Bayes filters are a probabilistic tool for estimating the state of dynamic systems.
- It can be used in the context of mobile robotics to robustly estimate the pose of a vehicle.
- Particle filters are an efficient implementation of Recursive Bayesian filtering.