

Recursive Bayes Filtering

CS485
Autonomous Robotics
Amarda Shehu
Fall 2013

Physical Agents are Inherently Uncertain

- Uncertainty arises from four major factors:
 - Environment stochastic, unpredictable
 - Robot stochastic
 - Sensor limited, noisy
 - Models inaccurate

Example: Museum Tour-Guide Robots



Rhino, 1997



Minerva, 1998

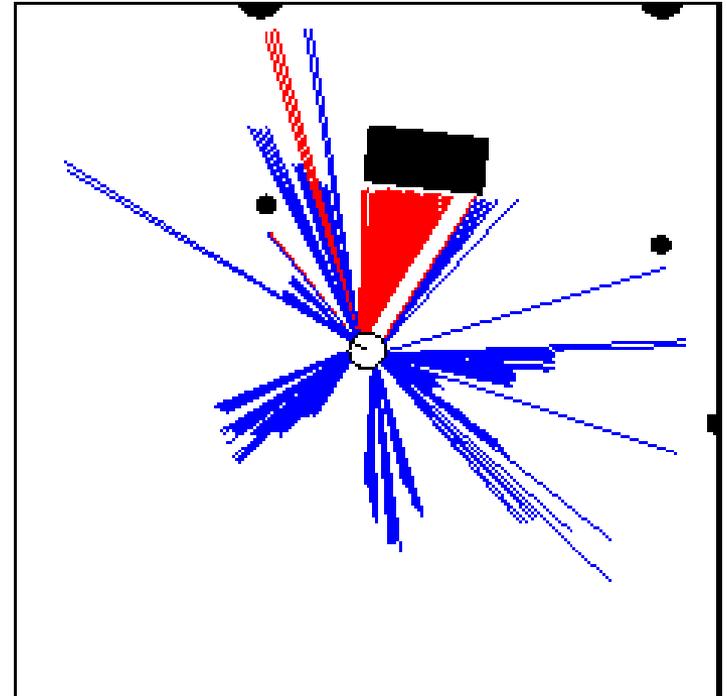
Technical Challenges

- Navigation
 - Environment crowded, unpredictable
 - Environment unmodified
 - “Invisible” hazards
 - Walking speed or faster
 - High failure costs
- Interaction
 - Individuals and crowds
 - Museum visitors’ first encounter
 - Age 2 through 99
 - Spend less than 15 minutes

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

Outline

- Introduction
- Probabilistic State Estimation
- Robot Localization
- Probabilistic Decision Making
 - Planning
 - Between MDPs and POMDPs
 - Exploration
- Conclusions

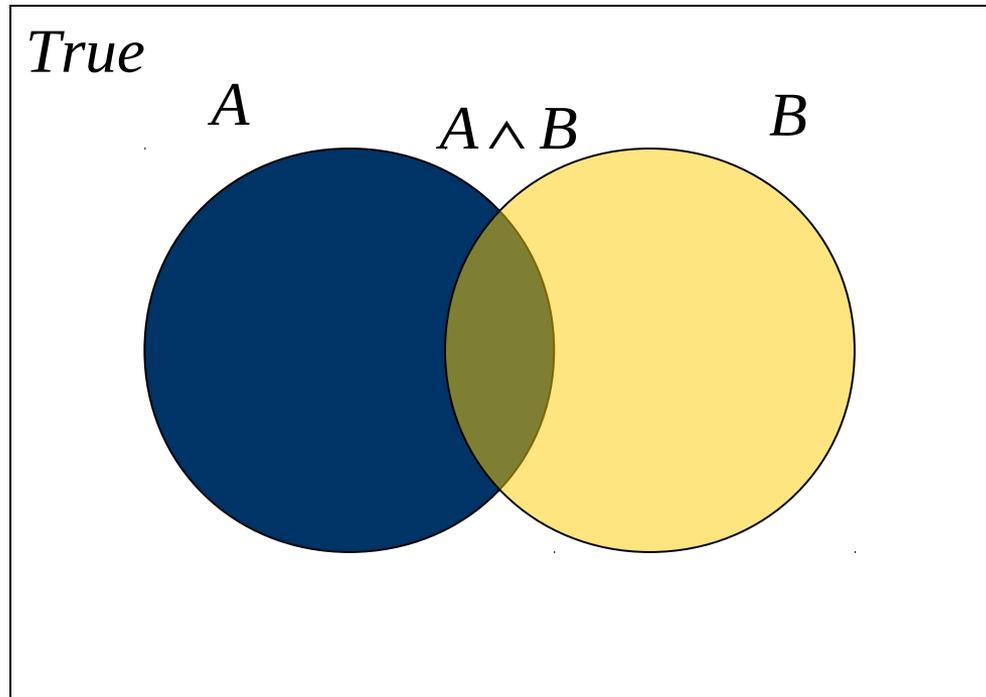
Axioms of Probability Theory

$\Pr(A)$ denotes probability that proposition A is true.

- $0 \leq \Pr(A) \leq 1$
- $\Pr(\textit{True}) = 1$ $\Pr(\textit{False}) = 0$
- $\Pr(A \vee B) = \Pr(A) + \Pr(B) - \Pr(A \wedge B)$

A Closer Look at Axiom 3

$$\Pr(A \vee B) = \Pr(A) + \Pr(B) - \Pr(A \wedge B)$$



Using the Axioms

$$\Pr(A \vee \neg A) = \Pr(A) + \Pr(\neg A) - \Pr(A \wedge \neg A)$$

$$\Pr(\textit{True}) = \Pr(A) + \Pr(\neg A) - \Pr(\textit{False})$$

$$1 = \Pr(A) + \Pr(\neg A) - 0$$

$$\Pr(\neg A) = 1 - \Pr(A)$$

Discrete Random Variables

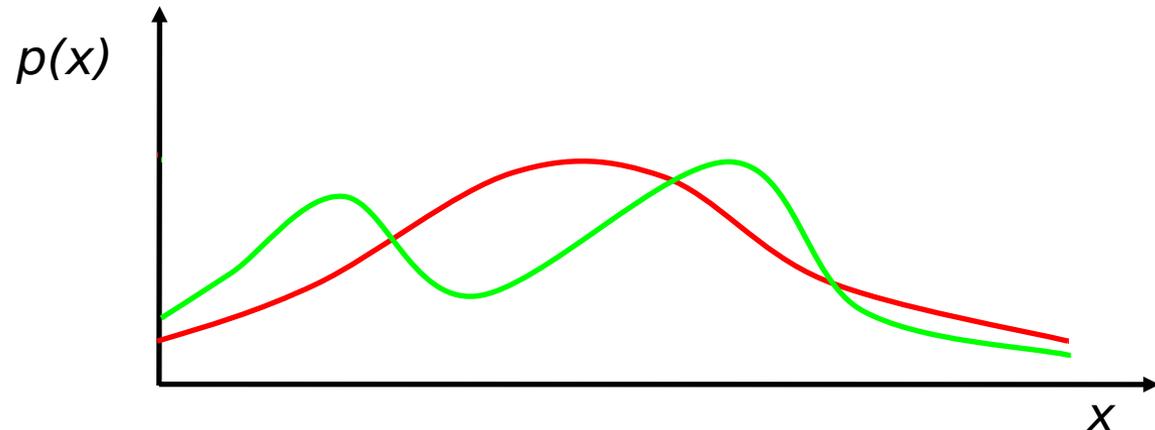
- X denotes a **random variable**.
- X can take on a finite number of values in $\{x_1, x_2, \dots, x_n\}$.
- $P(X=x_i)$, or $P(x_i)$, is the **probability** that the random variable X takes on value x_i .

Continuous Random Variables

- X takes on values in the continuum.
- $p(X=x)$, or $p(x)$, is a probability density function.

$$\Pr(x \in [a, b]) = \int_a^b p(x) dx$$

- E.g.



Joint and Conditional 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**
$$P(x | y) = P(x,y) / P(y)$$
$$P(x,y) = P(x | y) P(y)$$
- If X and Y are **independent** then
$$P(x | y) = P(x)$$

Law of Total Probability, Marginals

Discrete case

$$\sum_x P(x) = 1$$

$$P(x) = \sum_y P(x, y)$$

$$P(x) = \sum_y P(x | y) P(y)$$

Continuous case

$$\int p(x) dx = 1$$

$$p(x) = \int p(x, y) dy$$

$$p(x) = \int p(x | y) p(y) dy$$

Bayes Formula

$$P(x, y) = P(x | y)P(y) = P(y | x)P(x)$$

\Rightarrow

$$P(x | y) = \frac{P(y | x) P(x)}{P(y)} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}$$

Normalization

$$P(x|y) = \frac{P(y|x) P(x)}{P(y)} = \eta P(y|x) P(x)$$

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

Algorithm:

$$\forall x: \text{aux}_{x|y} = P(y|x) P(x)$$

$$\eta = \frac{1}{\sum_x \text{aux}_{x|y}}$$

$$\forall x: P(x|y) = \eta \text{aux}_{x|y}$$

Conditioning

- Total probability:

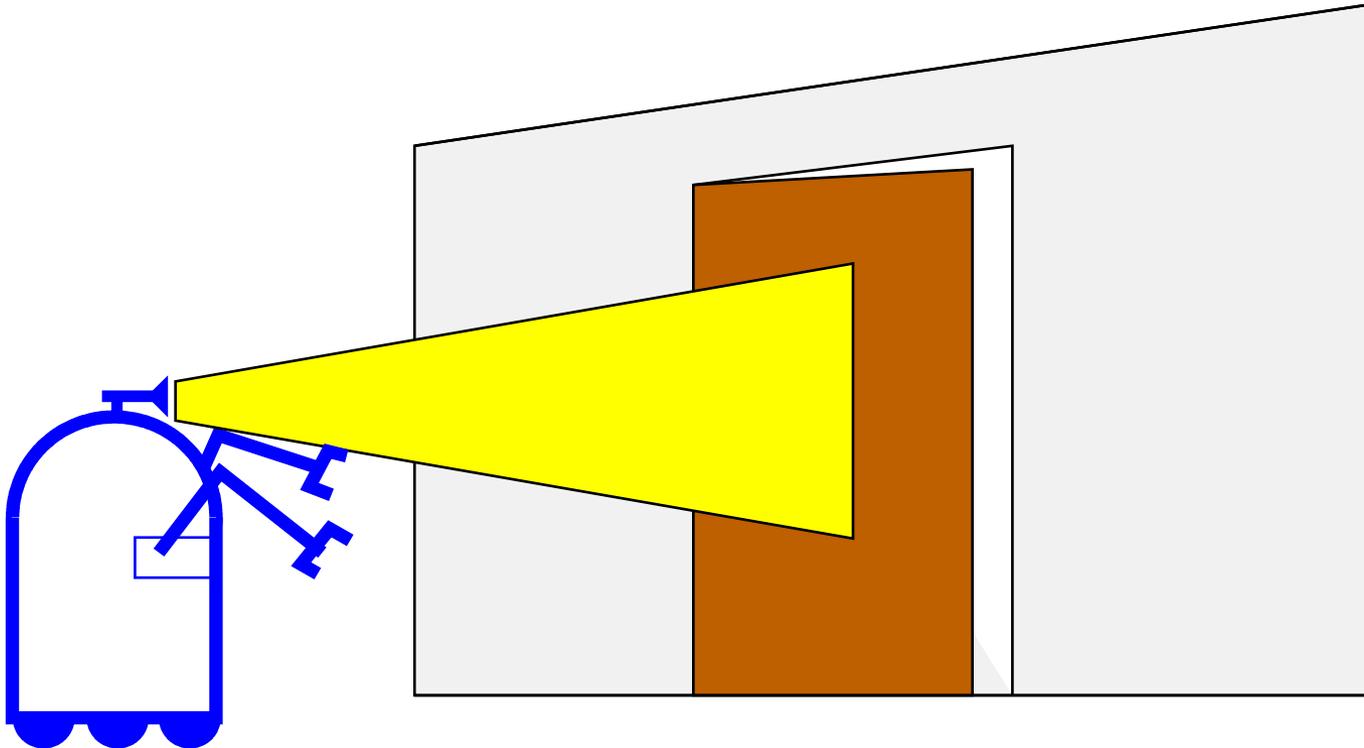
$$P(x|y) = \int P(x | y, z) P(z | y) dz$$

- Bayes rule and background knowledge:

$$P(x | y, z) = \frac{P(y | x, z) P(x | z)}{P(y | z)}$$

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:

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

Causal vs. Diagnostic Reasoning

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

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

Example

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

- $P(open) = P(\neg open) = 0.5$

$$P(open | z) = ?$$

Example

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

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

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

- 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 $P(x | z_1 \dots z_n)$?

Recursive Bayesian Updating

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

Recursive Bayesian Updating

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

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

Recursive Bayesian Updating

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

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

$$\begin{aligned} P(x | z_1, \dots, z_n) &= \frac{P(z_n | x) P(x | z_1, \dots, z_{n-1})}{P(z_n | z_1, \dots, z_{n-1})} \\ &= \eta P(z_n | x) P(x | z_1, \dots, z_{n-1}) \\ &= \eta_{1\dots n} \prod_{i=1\dots n} P(z_i | x) P(x) \end{aligned}$$

Example: Second Measurement

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

$P(open | z_2, z_1) = ?$

Example: Second Measurement

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

$$\begin{aligned} P(open | z_2, z_1) &= \frac{P(z_2 | open) P(open | z_1)}{P(z_2 | open) P(open | z_1) + P(z_2 | \neg open) P(\neg open | z_1)} \\ &= \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625 \end{aligned}$$

- z_2 lowers the probability that the door is open.

Actions

- Often the world is **dynamic** since
 - **actions carried out by the robot,**
 - **actions carried out by other agents,**
 - or just the **time** passing bychange the world.

- How can we **incorporate** such **actions**?

Typical Actions

- The robot **turns its wheels** to move
- The robot **uses its manipulator** to grasp an object
- Plants grow over **time**...

- Actions are **never carried out with absolute certainty**.
- In contrast to measurements, **actions generally increase the uncertainty**.

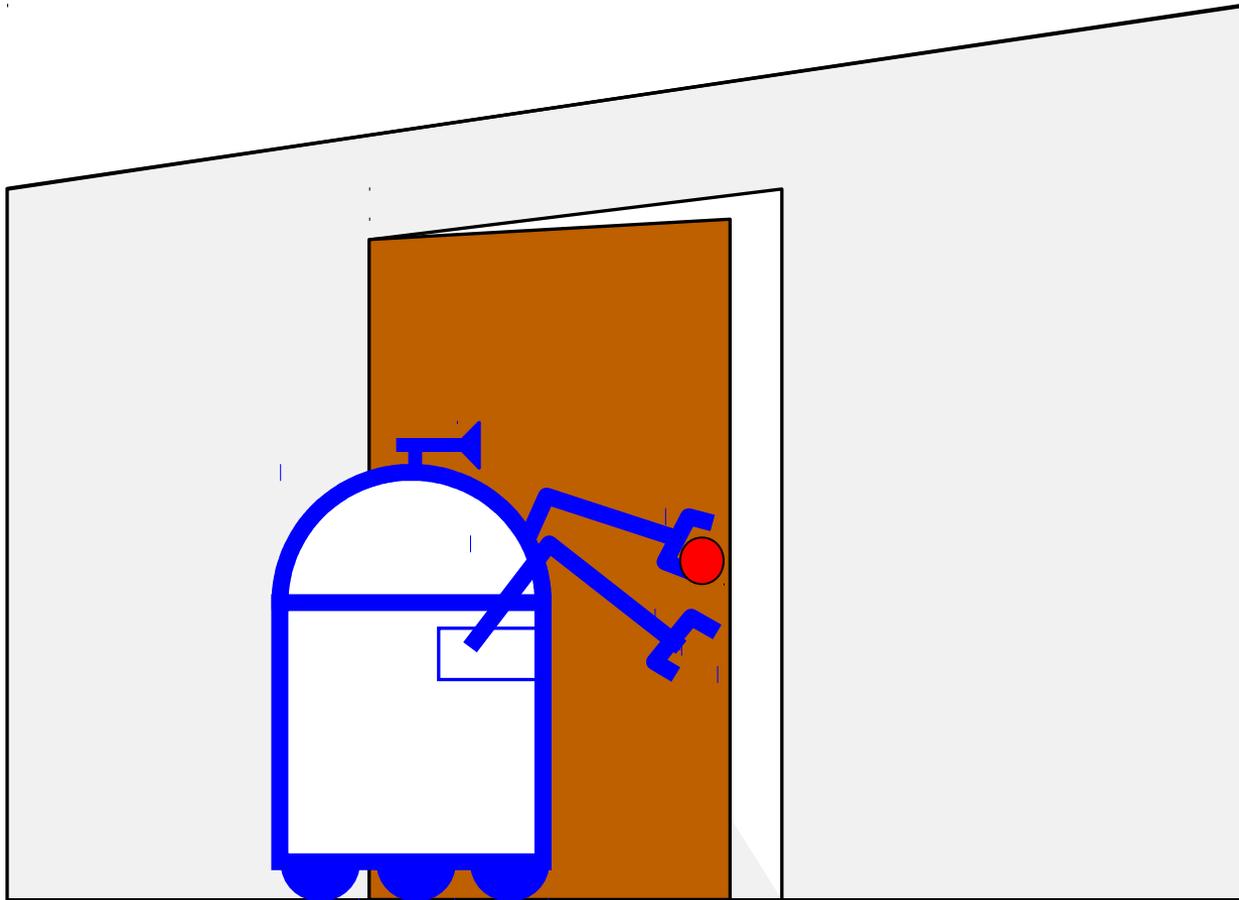
Modeling Actions

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

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

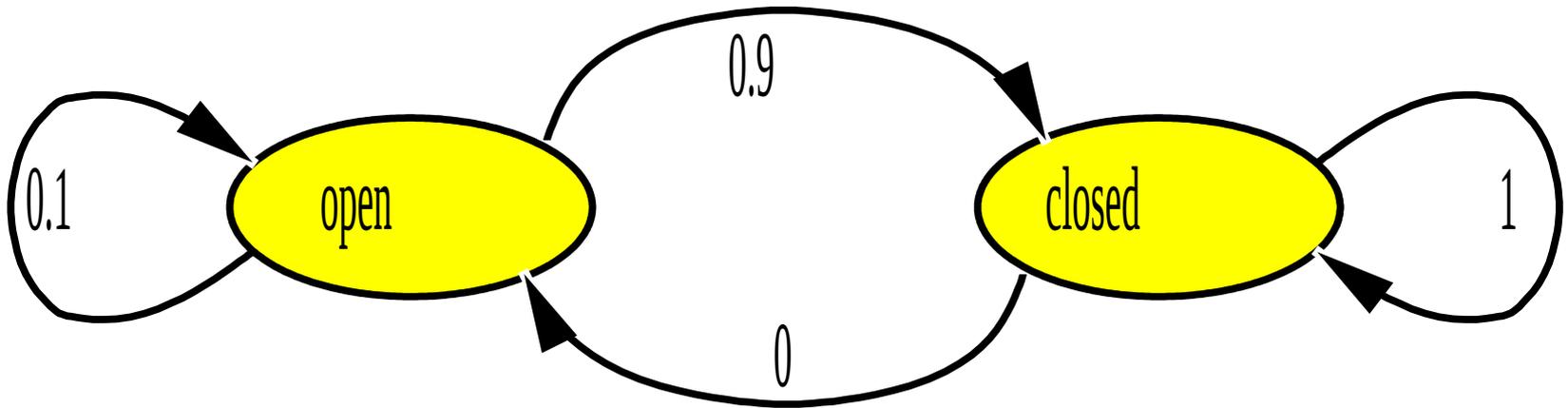
- This term specifies the pdf that **executing u changes the state from x' to x .**

Example: Closing the door



State Transitions

$P(x|u, x')$ for $u = \text{“close door”}$:



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

Integrating the Outcome of Actions

Continuous case:

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

Discrete case:

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

Example: The Resulting Belief

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

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

Bayes Filters: Framework

■ Given:

- Stream of observations z and action data u :

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

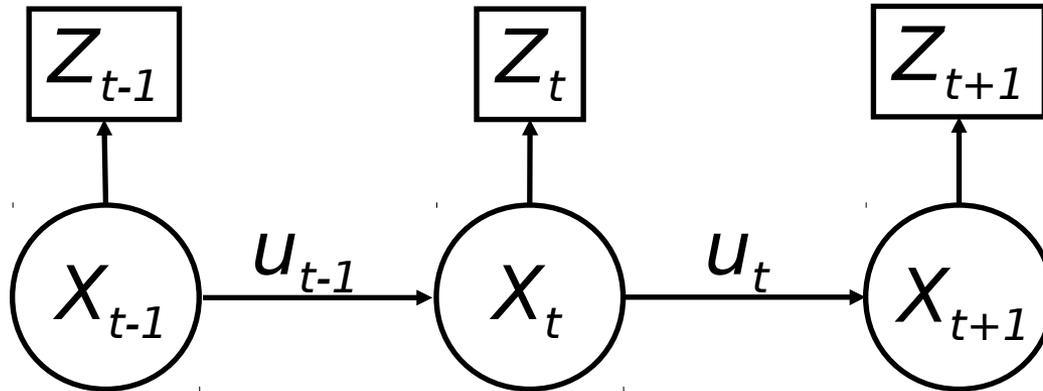
- Sensor model $P(z|x)$.
- Action model $P(x|u, x')$.
- Prior probability of the system state $P(x)$.

■ Wanted:

- Estimate of the state X of a dynamical system.
- The posterior of the state is also called **Belief**:

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

Markov Assumption



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

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

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

Underlying Assumptions

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

z = observation
 u = action
 x = state

Bayes Filters

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

z = observation
 u = action
 x = state

Bayes Filters

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

Bayes

$$= \eta P(z_t | x_t, u_1, z_2, \dots, u_{t-1}) P(x_t | u_1, z_2, \dots, u_{t-1})$$

z = observation
 u = action
 x = state

Bayes Filters

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

Bayes $= \eta P(z_t | x_t, u_1, z_2, \dots, u_{t-1}) P(x_t | u_1, z_2, \dots, u_{t-1})$

Markov $= \eta P(z_t | x_t) P(x_t | u_1, z_2, \dots, u_{t-1})$

z = observation
 u = action
 x = state

Bayes Filters

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

Bayes $= \eta P(z_t | x_t, u_1, z_2, \dots, u_{t-1}) P(x_t | u_1, z_2, \dots, u_{t-1})$

Markov $= \eta P(z_t | x_t) P(x_t | u_1, z_2, \dots, u_{t-1})$

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

z = observation
 u = action
 x = state

Bayes Filters

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

Bayes $= \eta P(z_t | x_t, u_1, z_2, \dots, u_{t-1}) P(x_t | u_1, z_2, \dots, u_{t-1})$

Markov $= \eta P(z_t | x_t) P(x_t | u_1, z_2, \dots, u_{t-1})$

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

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

z = observation
 u = action
 x = state

Bayes Filters

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

Bayes $= \eta P(z_t | x_t, u_1, z_2, \dots, u_{t-1}) P(x_t | u_1, z_2, \dots, u_{t-1})$

Markov $= \eta P(z_t | x_t) P(x_t | u_1, z_2, \dots, u_{t-1})$

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

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

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

z = observation
u = action
x = state

Bayes Filters

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

Bayes $= \eta P(z_t | x_t, u_1, z_2, \dots, u_{t-1}) P(x_t | u_1, z_2, \dots, u_{t-1})$

Markov $= \eta P(z_t | x_t) P(x_t | u_1, z_2, \dots, u_{t-1})$

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

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

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

Bayes Filter Algorithm

- Algorithm **Bayes_filter**($Bel(x), d$):
- $\eta=0$
- if d is a **perceptual** data item z then
- For all x do
- $Bel'(x) = P(z | x)Bel(x)$
- $\eta = \eta + Bel'(x)$
- For all x do
- $Bel'(x) = \eta^{-1}Bel'(x)$
- else if d is an **action** data item u then
- For all x do
- $Bel'(x) = \int P(x | u, x') Bel(x') dx'$
- return $Bel'(x)$

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

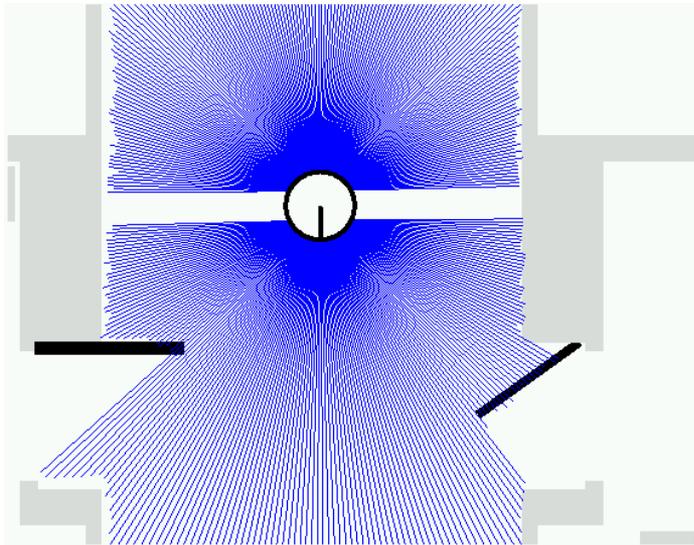
Bayes Filters are Familiar!

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

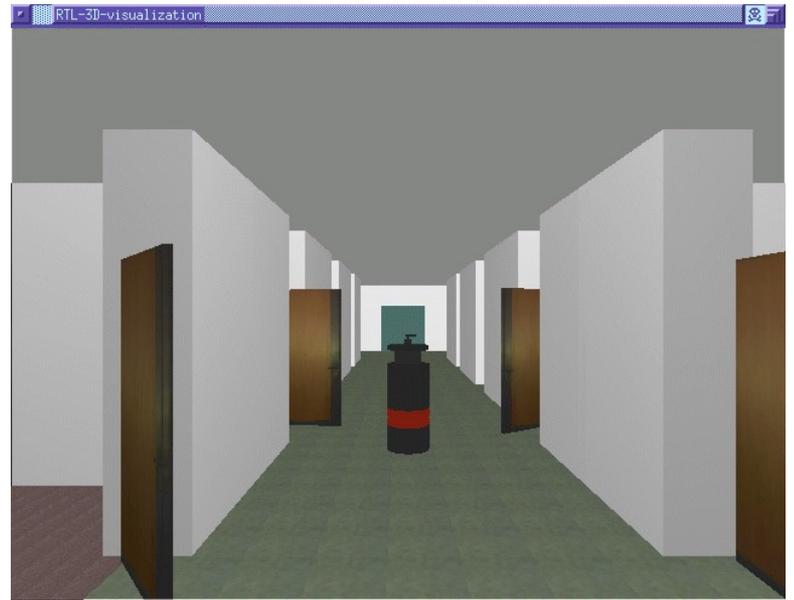
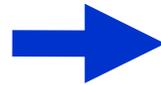
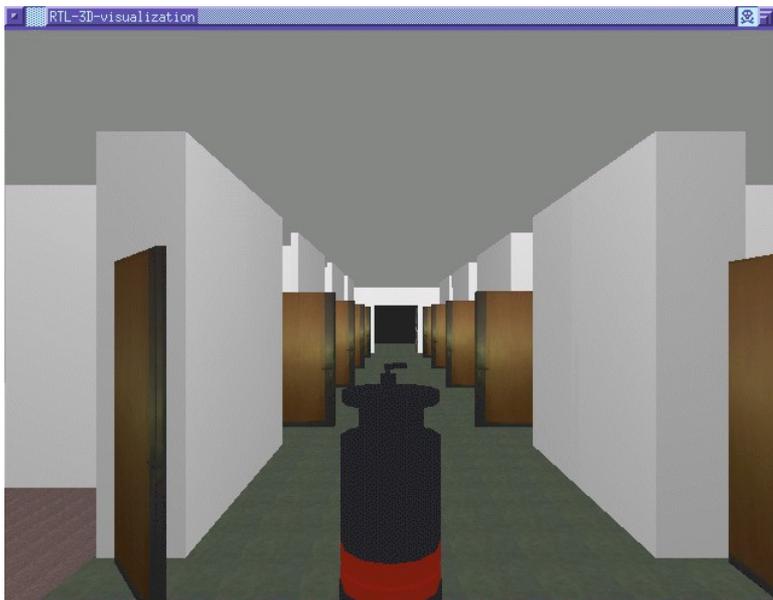
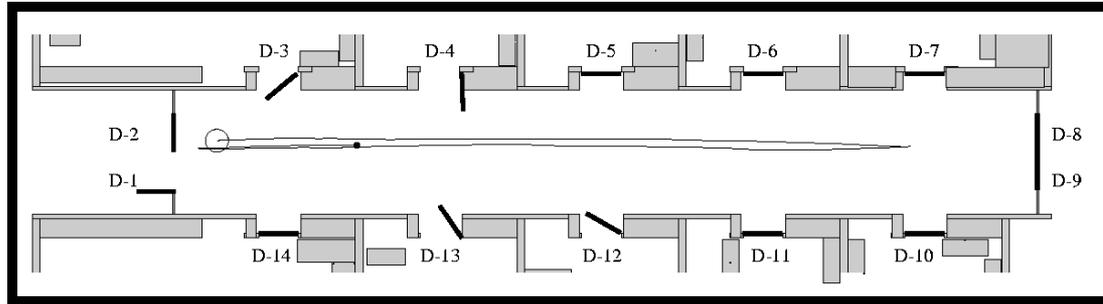
- Kalman filters
- Particle filters
- Hidden Markov models
- Dynamic Bayes networks
- Partially Observable Markov Decision Processes (POMDPs)

Application to Door State Estimation

- Estimate the opening angle of a door
- and the state of other dynamic objects
- using a laser-range finder
- from a moving mobile robot and
- based on Bayes filters.



Result



Lessons Learned

- Bayes rule allows us to compute probabilities that are hard to assess otherwise.
- Under the Markov assumption, recursive Bayesian updating can be used to efficiently combine evidence.
- Bayes filters are a probabilistic tool for estimating the state of dynamic systems.

Tutorial Outline

- Introduction
- Probabilistic State Estimation
- Localization

The Localization Problem

“Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities.” [Cox '91]

■ Given

- Map of the environment.
- Sequence of sensor measurements.

■ Wanted

- Estimate of the robot's position.

■ Problem classes

- Position tracking
- Global localization
- Kidnapped robot problem (recovery)

Representations for Bayesian Robot Localization

Discrete approaches ('95)

- Topological representation ('95)
 - uncertainty handling (POMDPs)
 - occas. global localization, recovery
- Grid-based, metric representation ('96)
 - global localization, recovery

Particle filters ('99)

- sample-based representation
- global localization, recovery

AI

Kalman filters (late-80s?)

- Gaussians
- approximately linear models
- position tracking

Robotics

Multi-hypothesis ('00)

- multiple Kalman filters
- global localization, recovery

Localization with Bayes Filters

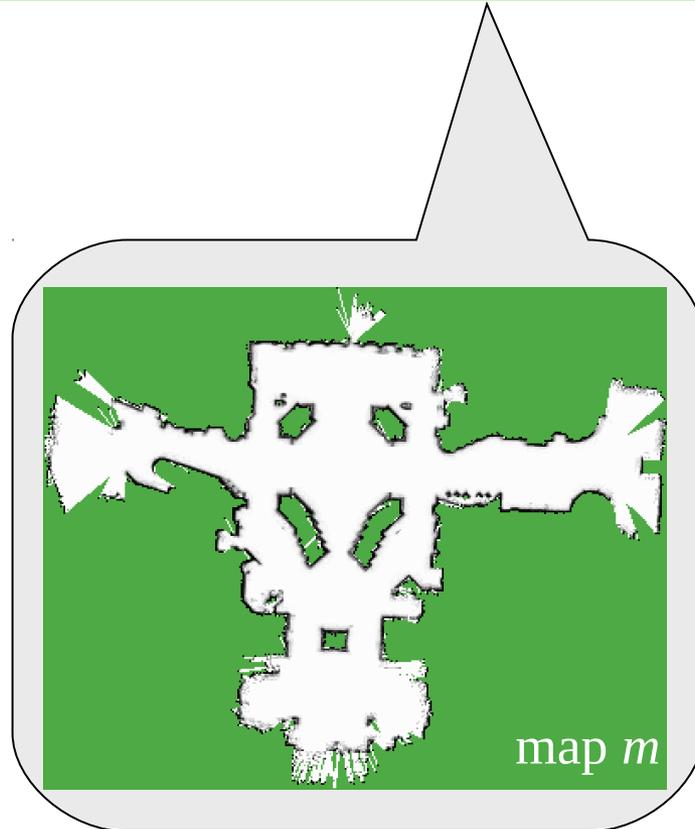
$$bel(x_t) = \eta p(s_t | x_t) \int p(x_t | x_{t-1}, a_{t-1}) bel(x_{t-1}) dx_{t-1}$$

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

Localization with Bayes Filters

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

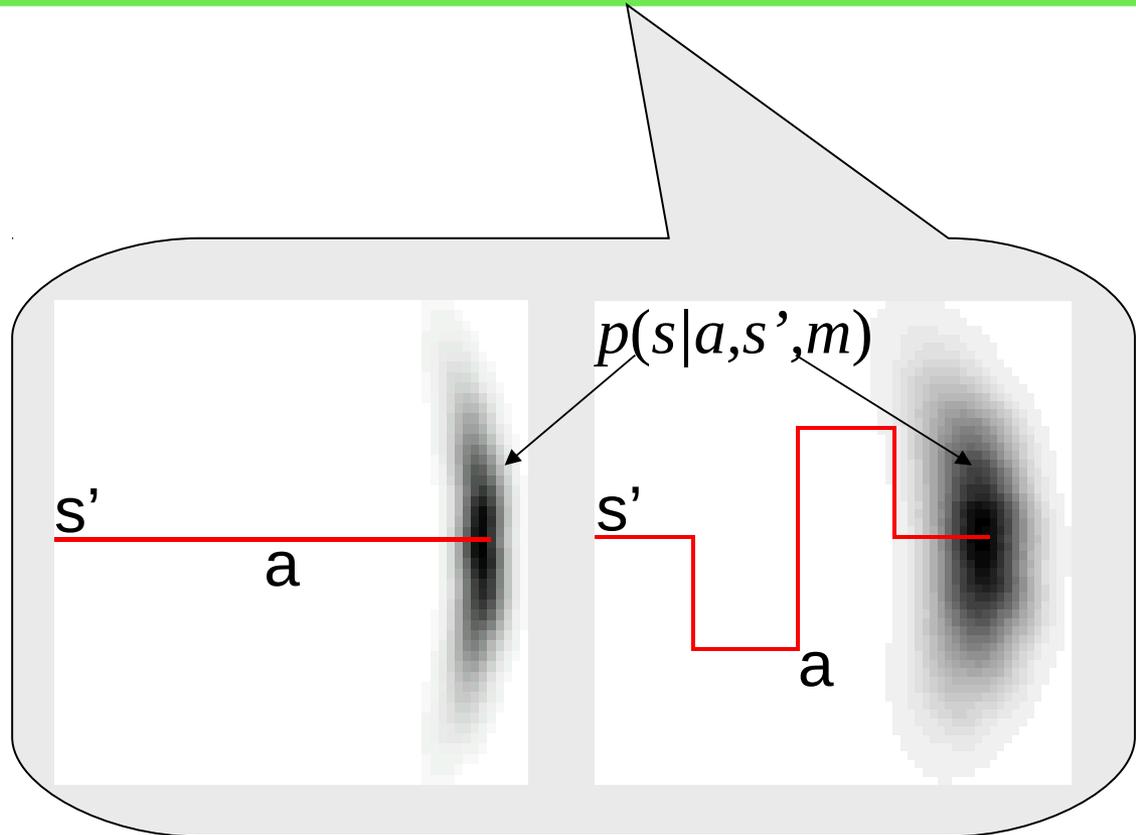
$$\Rightarrow bel(x_t | m) = \eta p(s_t | x_t, m) \int p(x_t | x_{t-1}, a_{t-1}, m) bel(x_{t-1} | m) dx_{t-1}$$



Localization with Bayes Filters

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

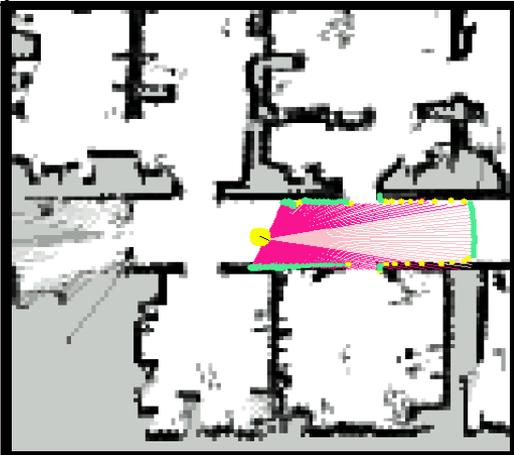
$$\Rightarrow bel(x_t | m) = \eta p(s_t | x_t, m) \int p(x_t | x_{t-1}, a_{t-1}, m) bel(x_{t-1} | m) dx_{t-1}$$



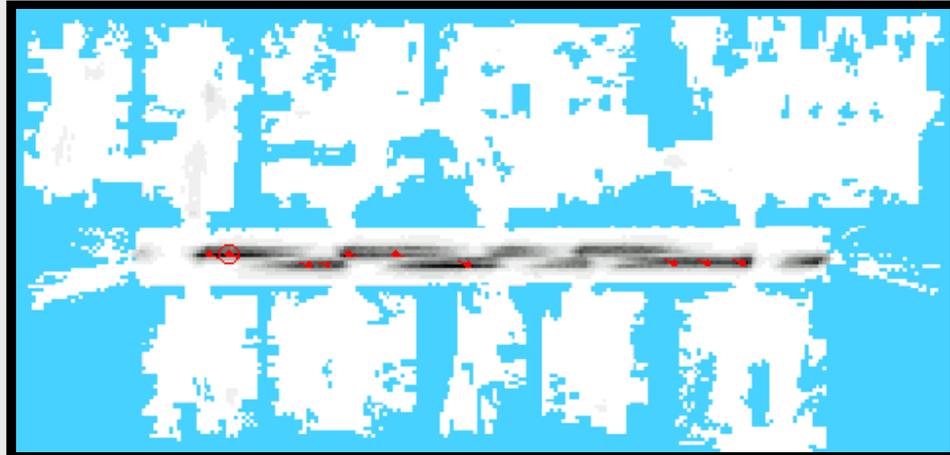
Localization with Bayes Filters

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

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



observation o

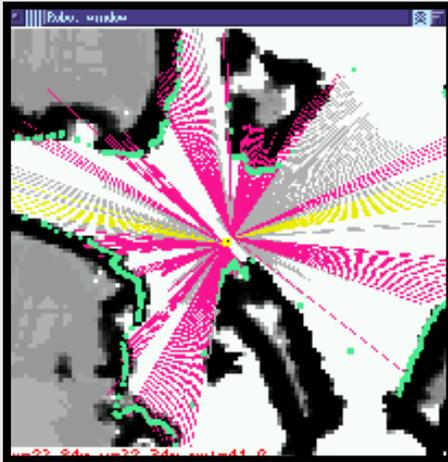


$p(o|s,m)$

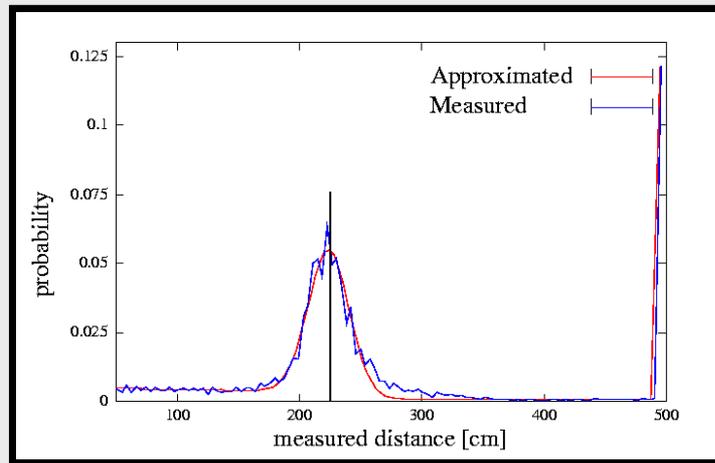
Localization with Bayes Filters

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

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



laser data



$p(o|s,m)$

Localization with Bayes Filters

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

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

Motion:

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

Perception:

$$bel(x_t) = \eta p(s_t | x_t) bel^-(x_t)$$

... is optimal under the Markov assumption

What is the Right Representation?

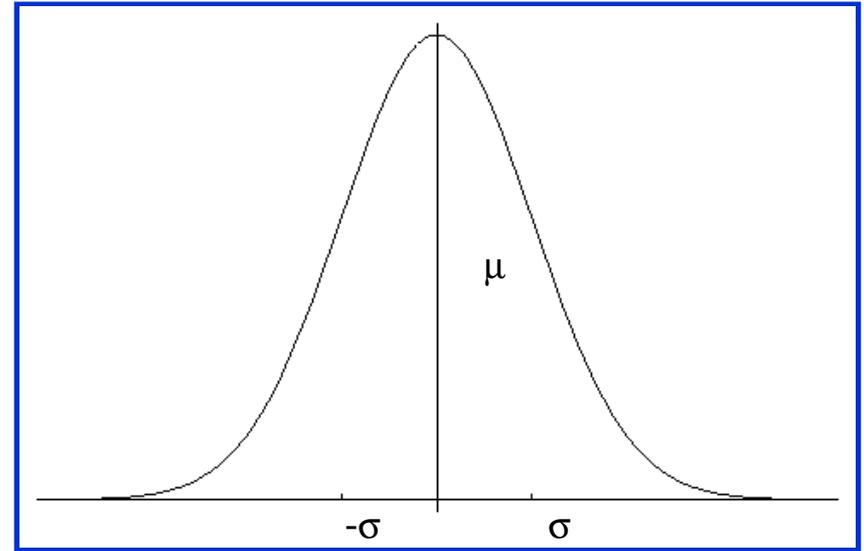
- Kalman filters
- Multi-hypothesis tracking
- Grid-based representations
- Topological approaches
- Particle filters

Gaussians

$p(x) \sim N(\mu, \sigma^2)$:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}}$$

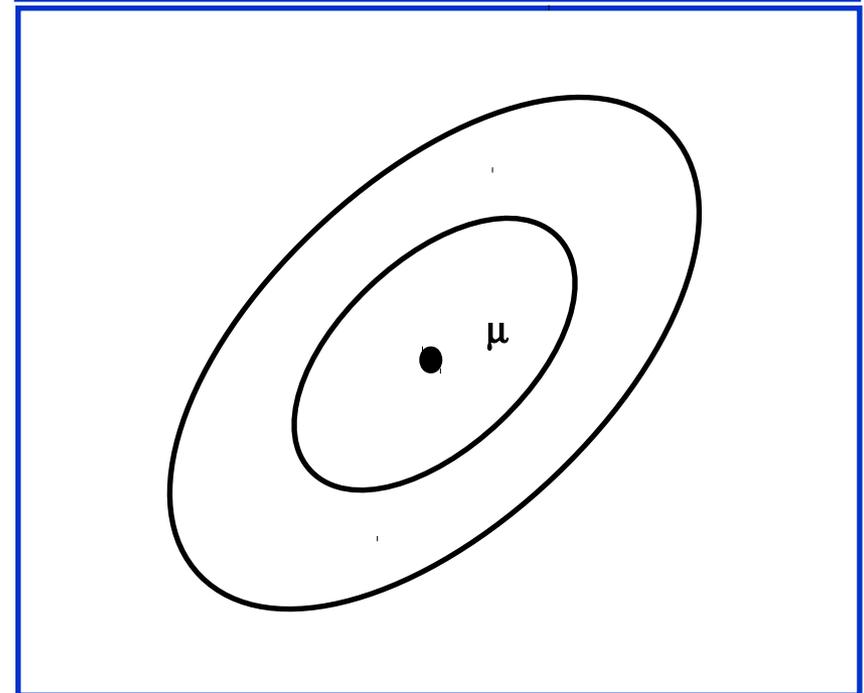
Univariate



$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2} (\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

Multivariate



Kalman Filters

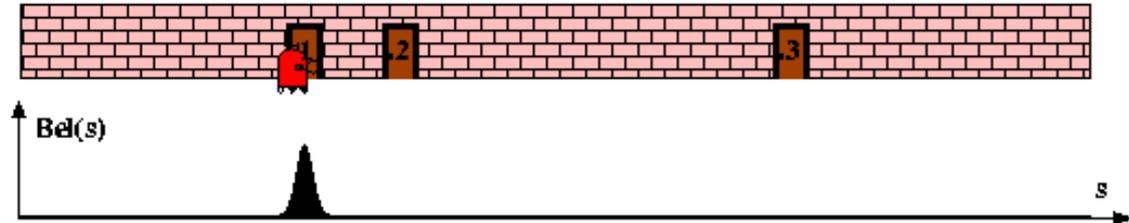
Estimate the state of processes that are governed by the following linear stochastic difference equation.

$$x_{t+1} = Ax_t + Bu_t + v_t$$

$$z_t = Cx_t + w_t$$

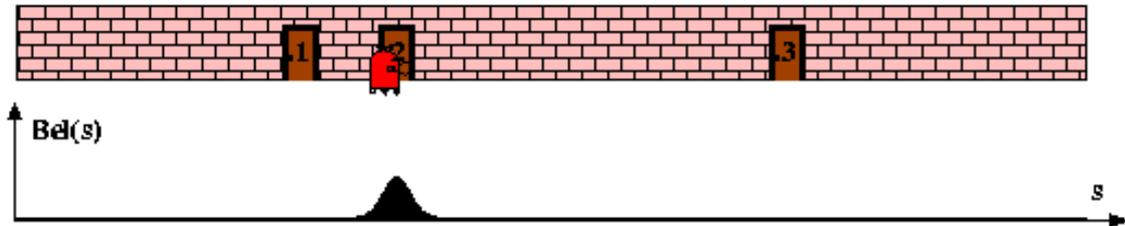
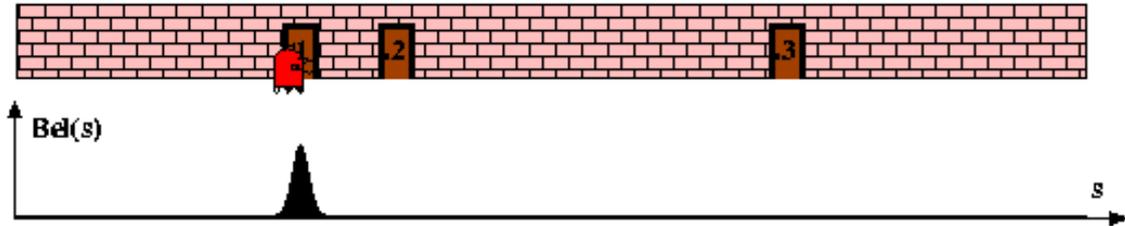
The random variables v_t and w_t represent the process measurement noise and are assumed to be independent, white and with normal probability distributions.

Kalman Filters



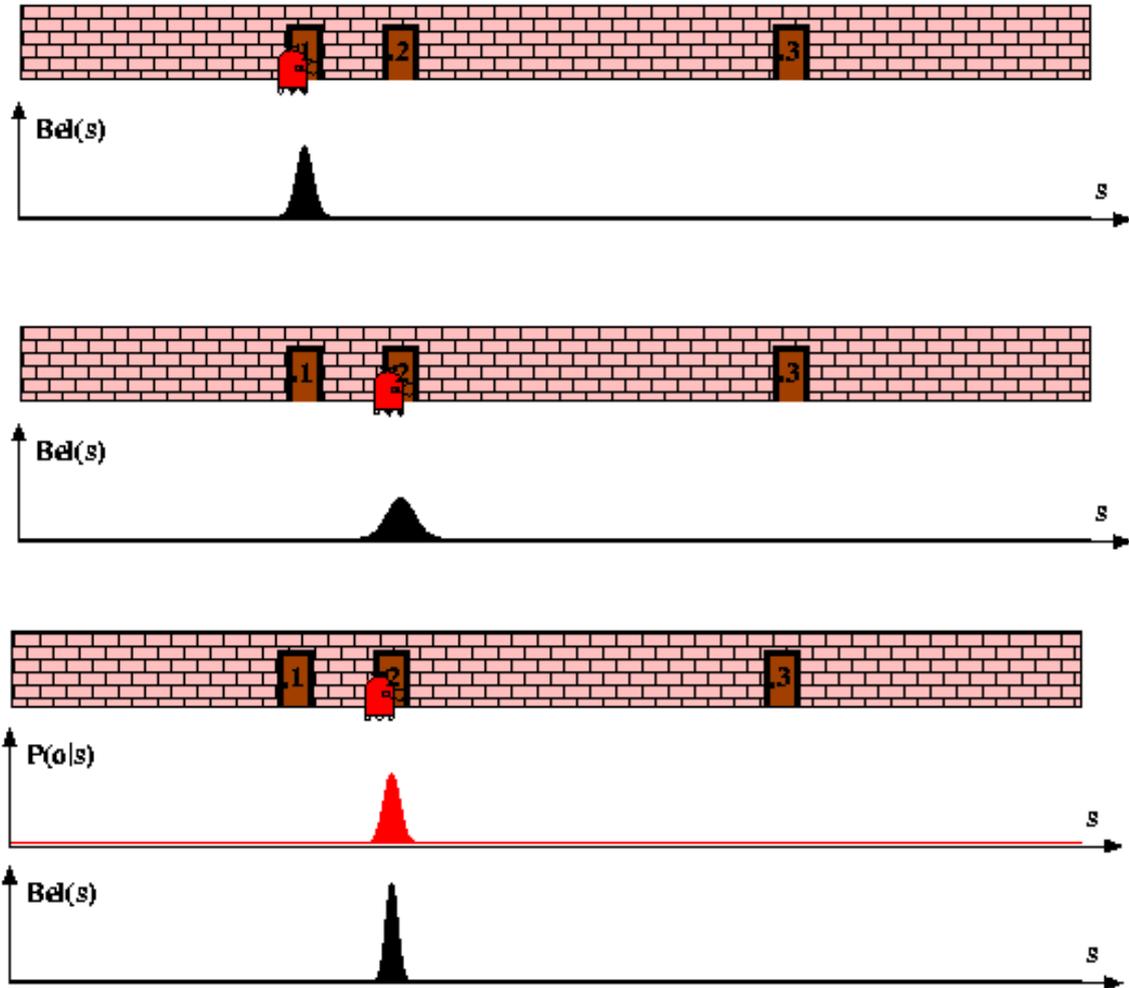
[Schiele et al. 94], [Weiß et al. 94],
[Borenstein 96],
[Gutmann et al. 96, 98], [Arras 98]

Kalman Filters



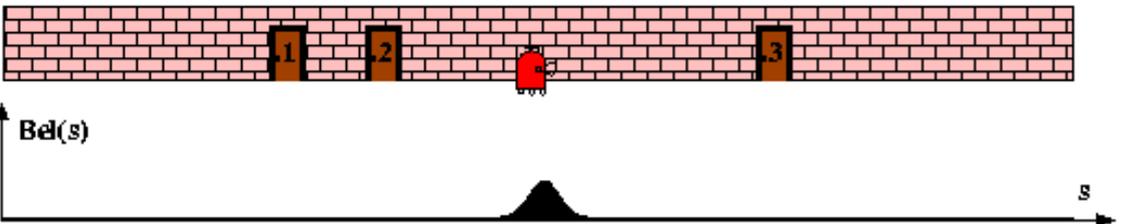
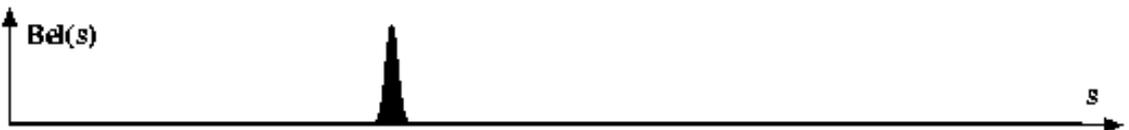
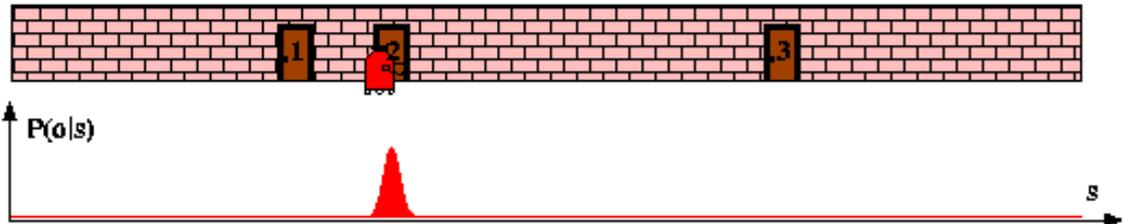
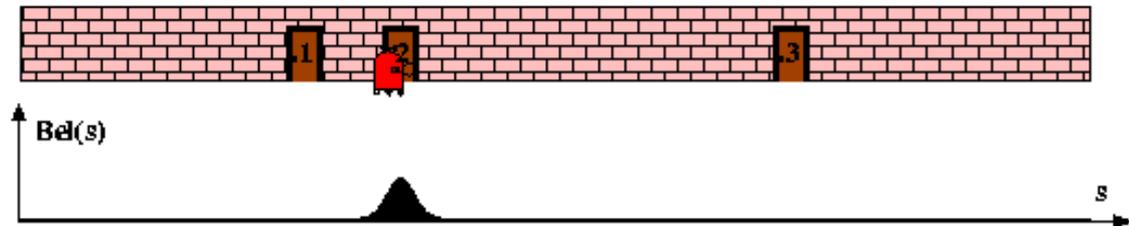
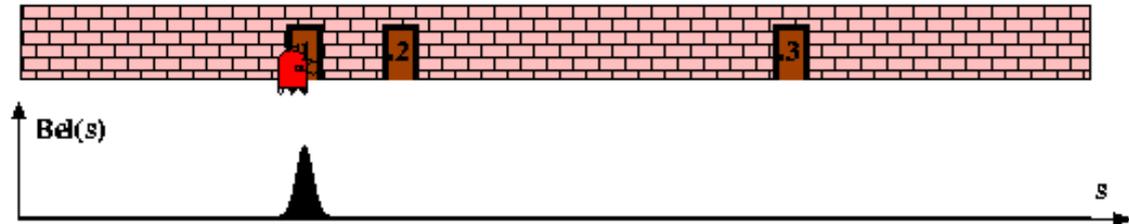
[Schiele et al. 94], [Weiß et al. 94],
[Borenstein 96],
[Gutmann et al. 96, 98], [Arras 98]

Kalman Filters



[Schiele et al. 94], [Weiß et al. 94],
[Borenstein 96],
[Gutmann et al. 96, 98], [Arras 98]

Kalman Filters



[Schiele et al. 94], [Weiß et al. 94],
[Borenstein 96],
[Gutmann et al. 96, 98], [Arras 98]

Kalman Filter Algorithm

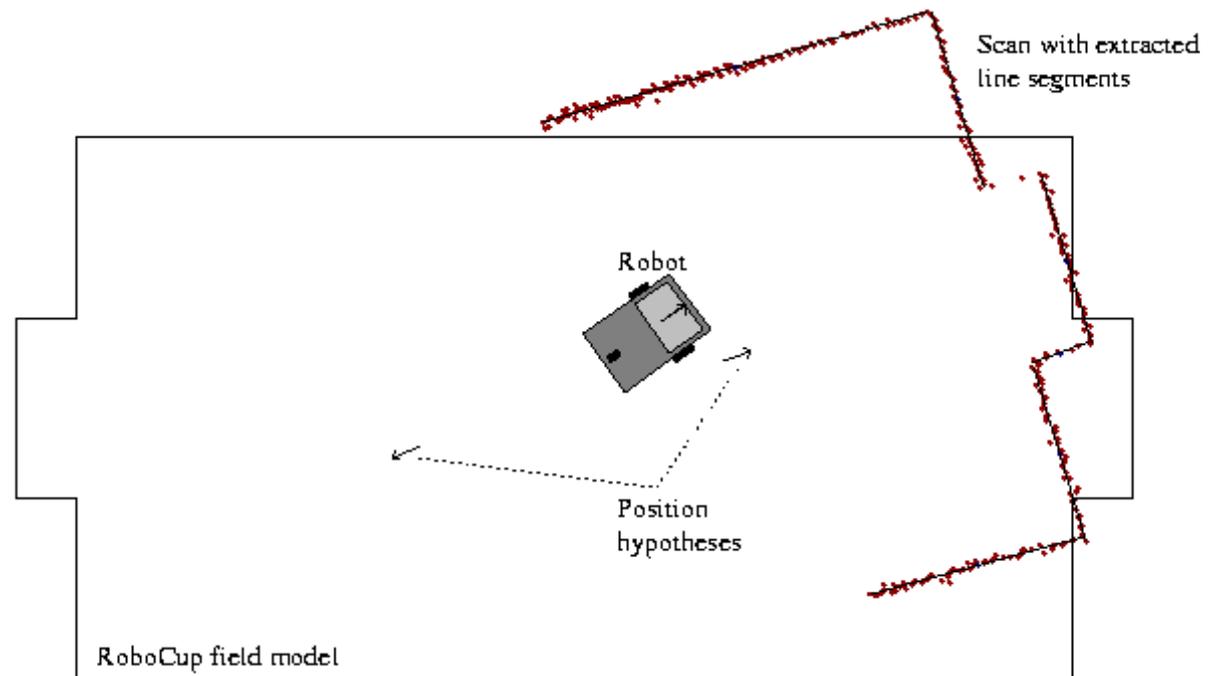
- Algorithm **Kalman_filter**($\langle \mu, \Sigma \rangle, d$):
- If d is a **perceptual** data item z then
- $$K = \Sigma C^T (C \Sigma C^T + \Sigma_{obs})^{-1}$$
- $$\mu = \mu + K(z - C\mu)$$
- $$\Sigma = (I - KC)\Sigma$$
- Else if d is an **action** data item u then
- $$\mu = A\mu + Bu$$
- $$\Sigma = A\Sigma A^T + \Sigma_{act}$$
- Return $\langle \mu, \Sigma \rangle$

Non-linear Systems

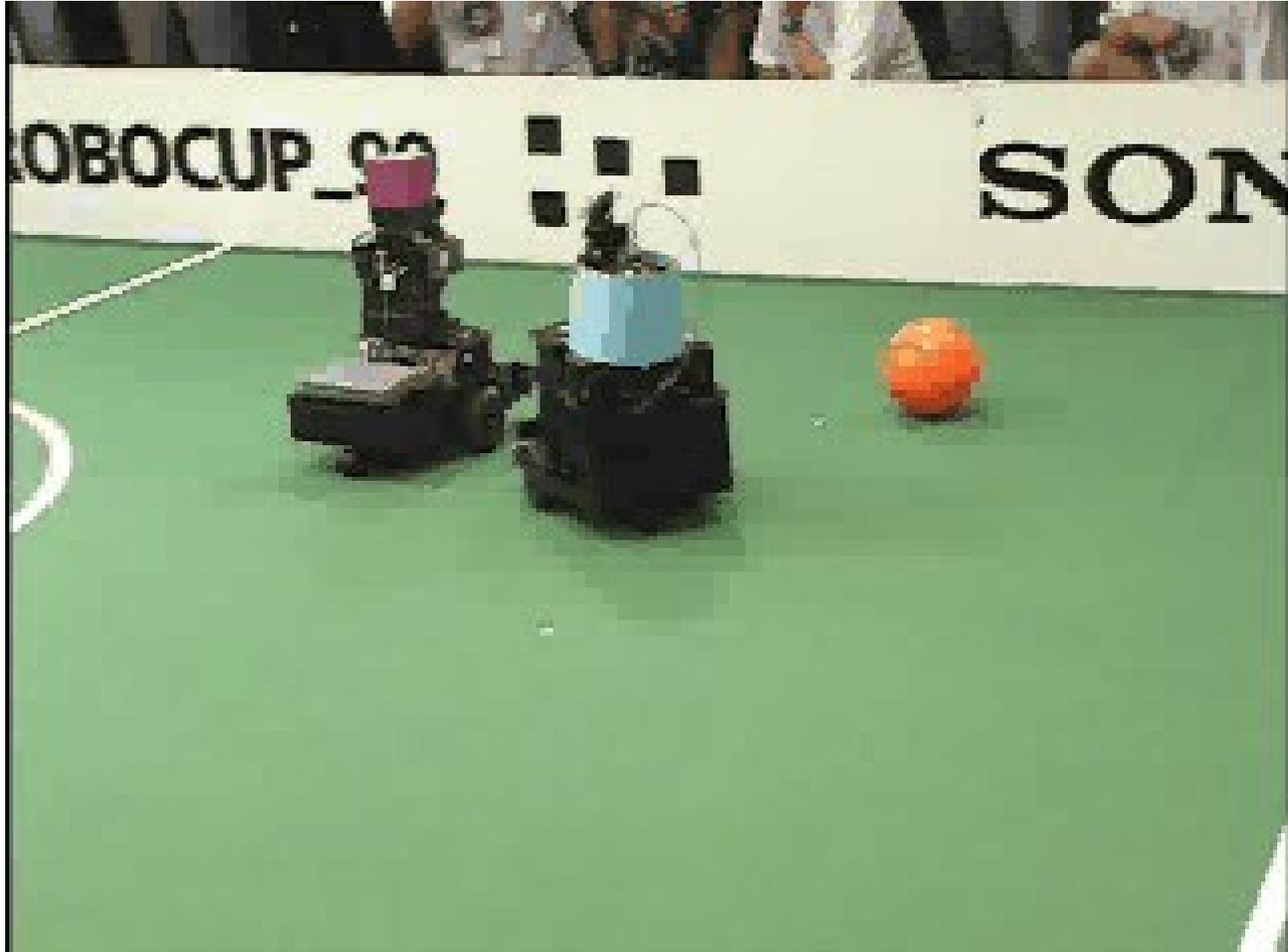
- Very strong assumptions:
 - Linear state dynamics
 - Observations linear in state
- What can we do if system is not linear?
 - Linearize it: **EKF**
 - Compute the Jacobians of the dynamics and observations at the current state.
 - Extended Kalman filter works surprisingly well even for highly non-linear systems.

Kalman Filter-based Systems (1)

- [Gutmann et al. 96, 98]:
 - Match LRF scans against map
 - Highly successful in RoboCup mid-size league

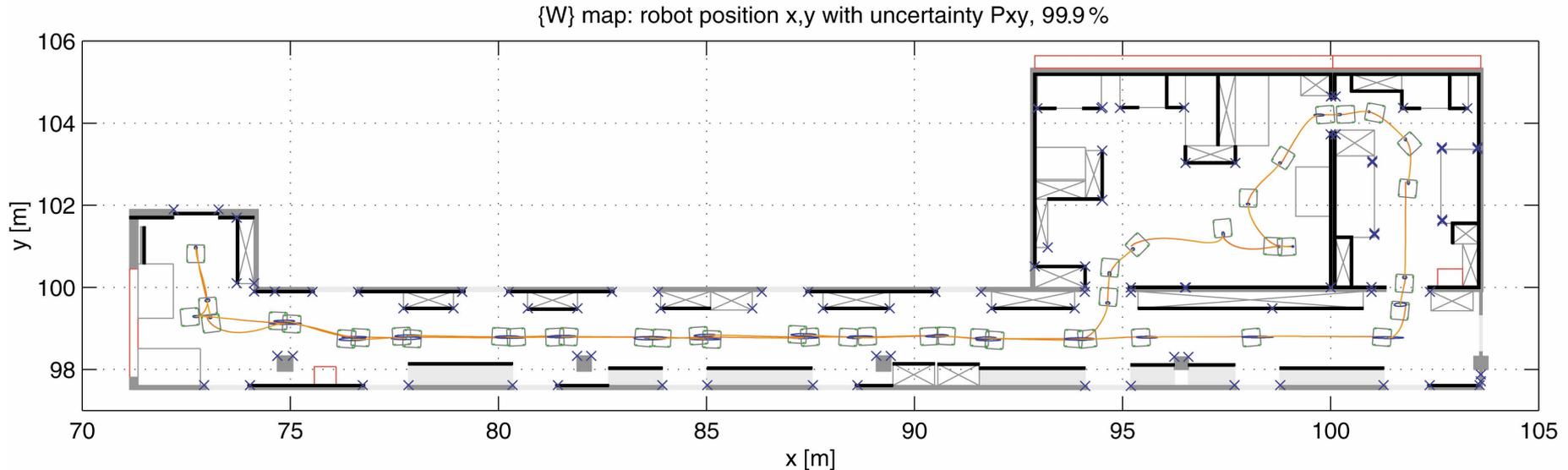


Kalman Filter-based Systems (2)



Kalman Filter-based Systems (3)

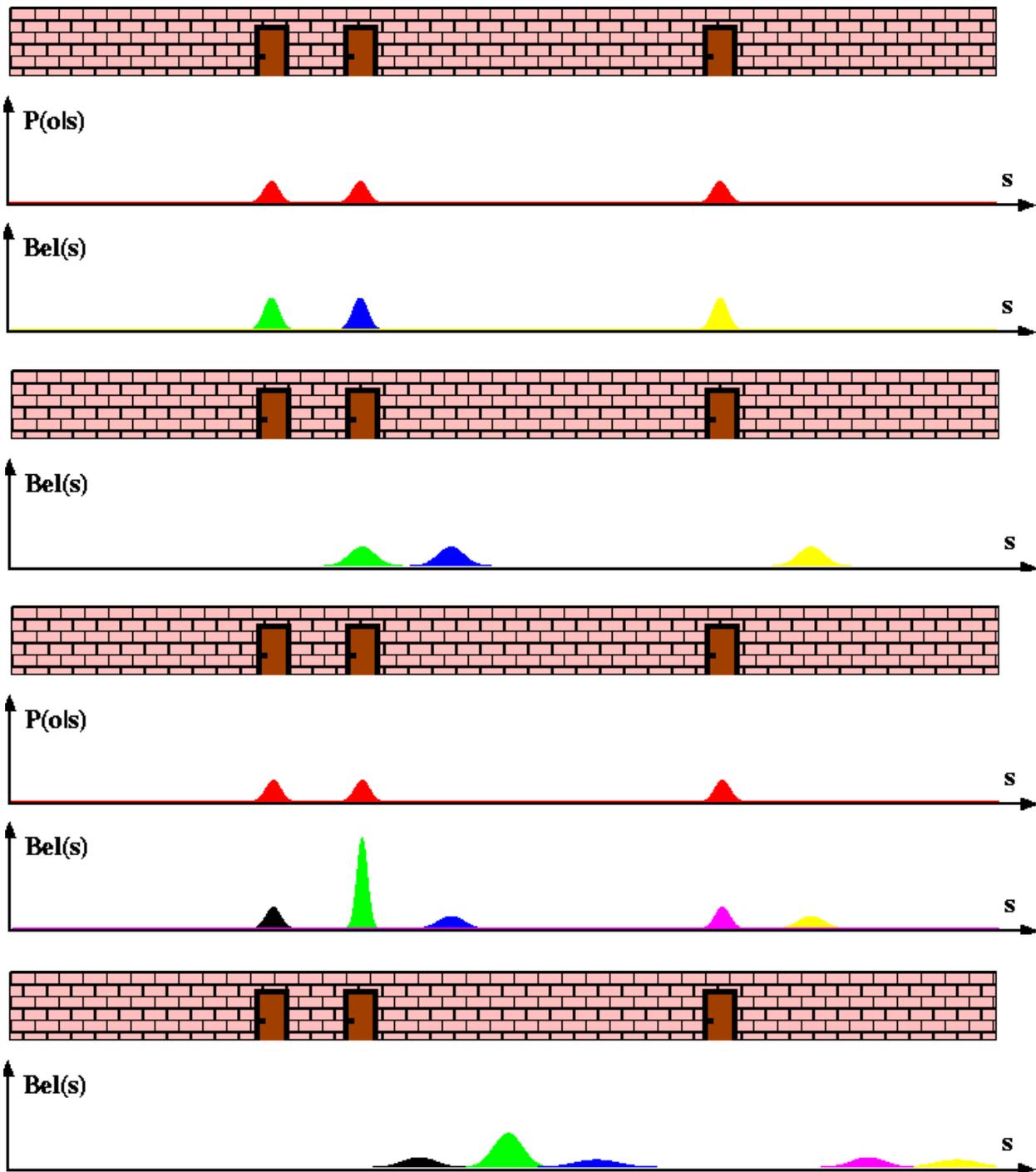
- [Arras et al. 98]:
 - Laser range-finder and vision
 - High precision (<1cm accuracy)



Localization Algorithms - Comparison

	Kalman filter
Sensors	Gaussian
Posterior	Gaussian
Efficiency (memory)	++
Efficiency (time)	++
Implementation	+
Accuracy	++
Robustness	-
Global localization	No

Multi-hypothesis Tracking



[Cox 92], [Jensfelt, Kristensen 99]

Localization With MHT

- Belief is represented by multiple hypotheses
- Each hypothesis is tracked by a Kalman filter
- **Additional problems:**
 - **Data association:** Which observation corresponds to which hypothesis?
 - **Hypothesis management:** When to add / delete hypotheses?
- Huge body of literature on target tracking, motion correspondence etc.

MHT: Implemented System (1)

- [Jensfelt and Kristensen 99,01]
 - Hypotheses are extracted from LRF scans
 - Each hypothesis has probability of being the correct one:

$$H_i = \{\hat{x}_i, \Sigma_i, P(H_i)\}$$

- Hypothesis probability is computed using Bayes' rule

$$P(H_i | s) = \frac{P(s | H_i)P(H_i)}{P(s)}$$

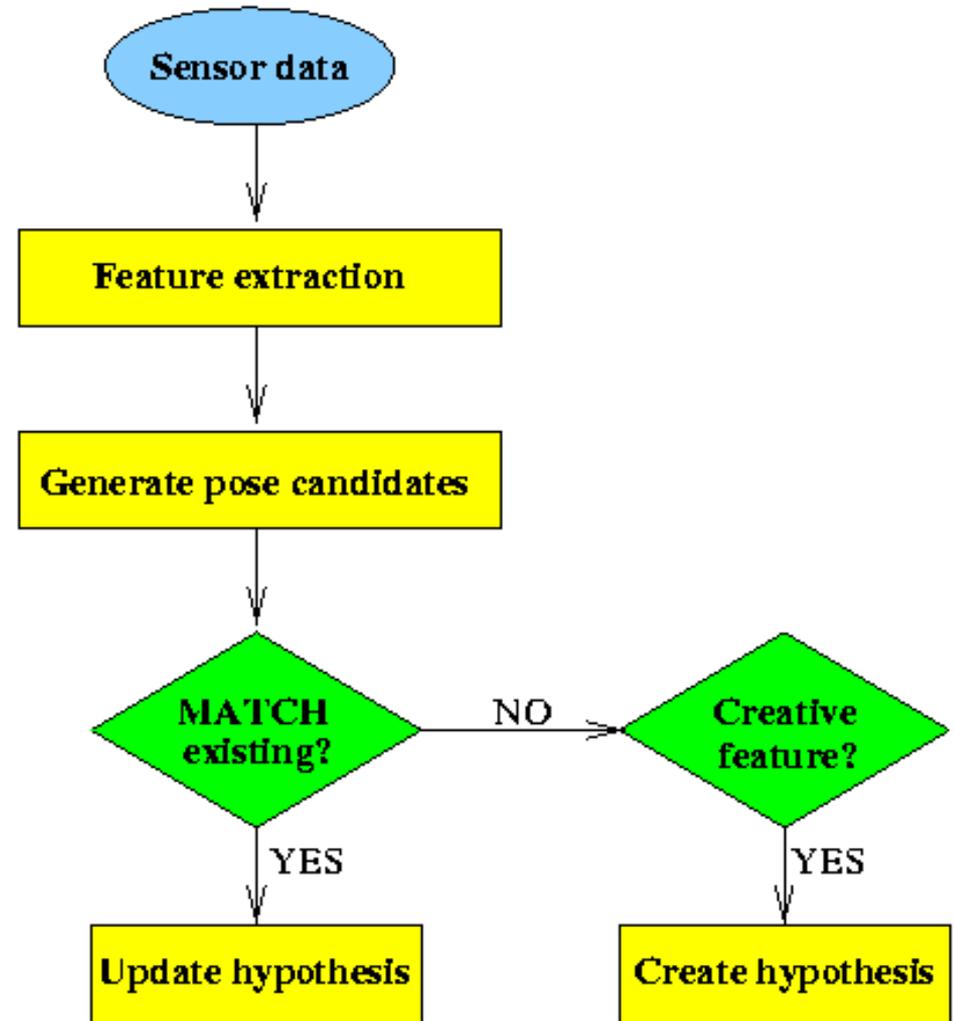
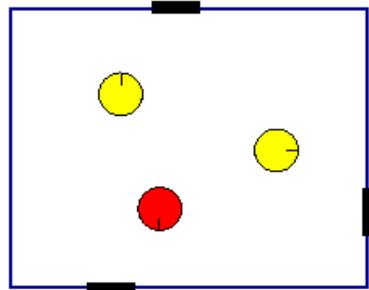
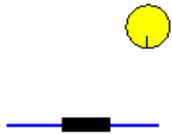
- Hypotheses with low probability are deleted
- New candidates are extracted from LRF scans

$$C_j = \{z_j, R_j\}$$

MHT: Implemented System (2)

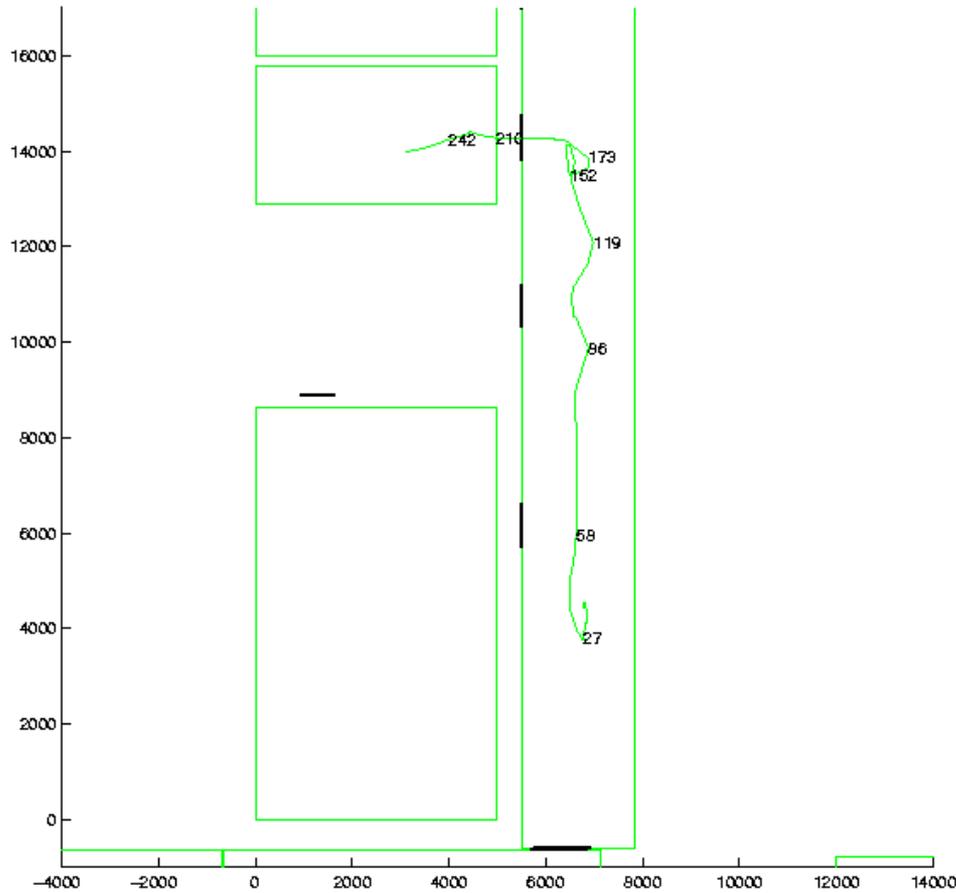
Robot view

Pose candidates

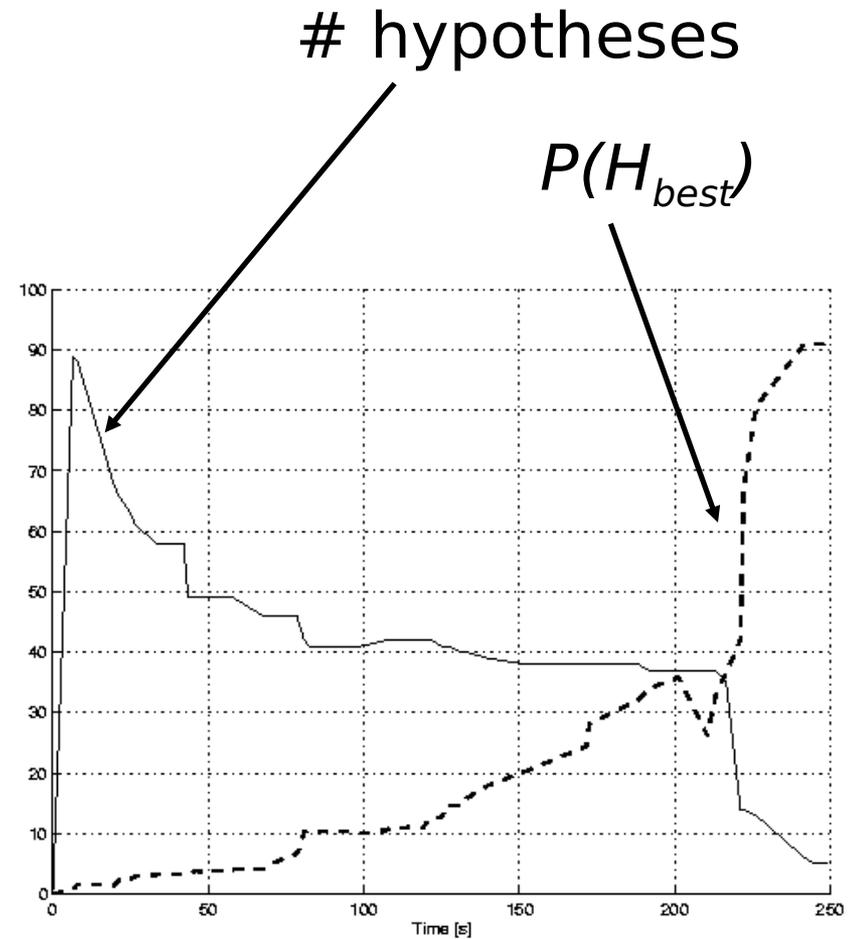


MHT: Implemented System (3)

Example run



Map and trajectory

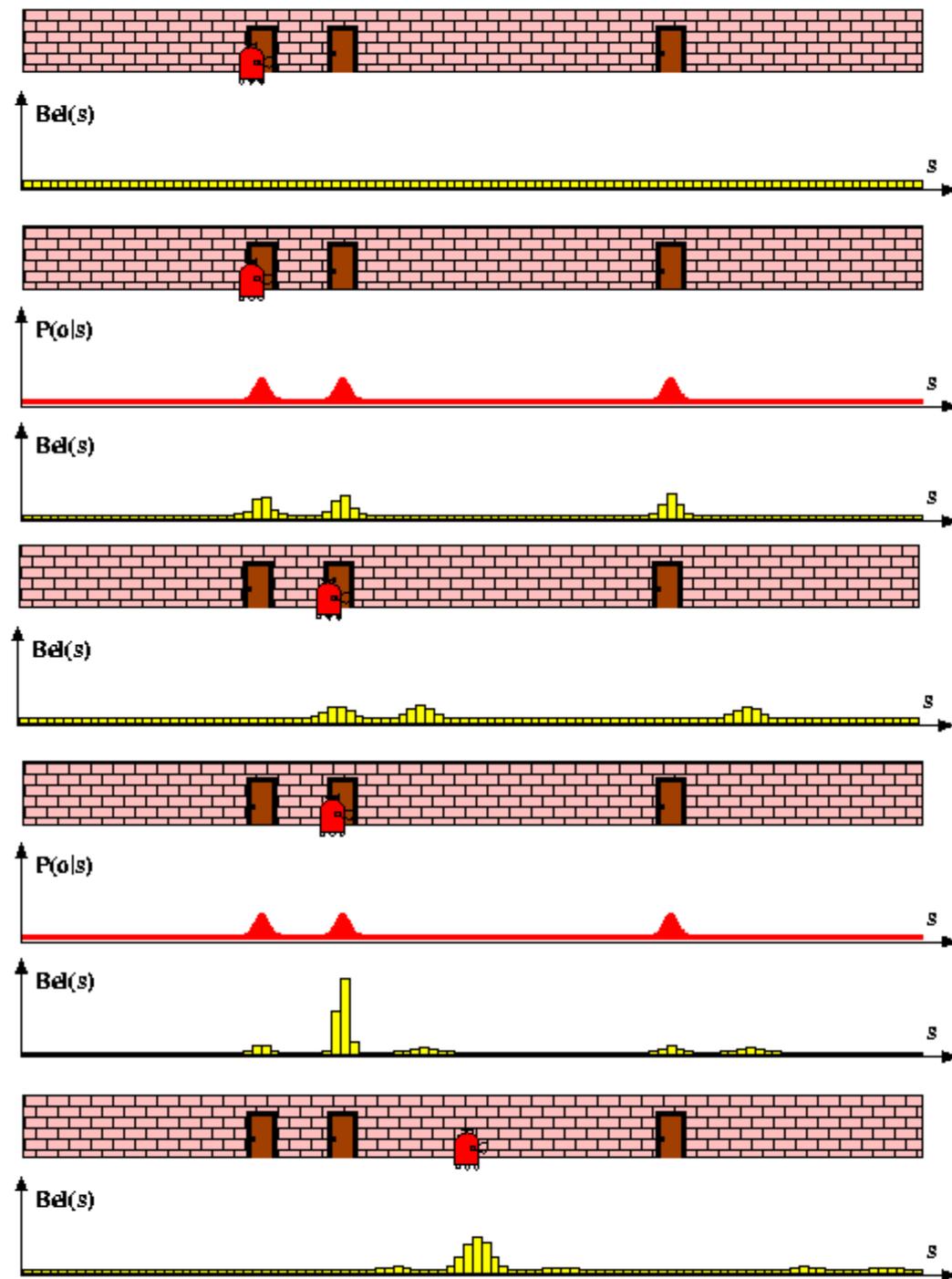


Hypotheses vs. time

Localization Algorithms - Comparison

	Kalman filter	Multi-hypot hesis tracking
Sensors	Gaussian	Gaussian
Posterior	Gaussian	Multi-modal
Efficiency (memory)	++	++
Efficiency (time)	++	++
Implementation	+	0
Accuracy	++	++
Robustness	-	+
Global localization	No	Yes

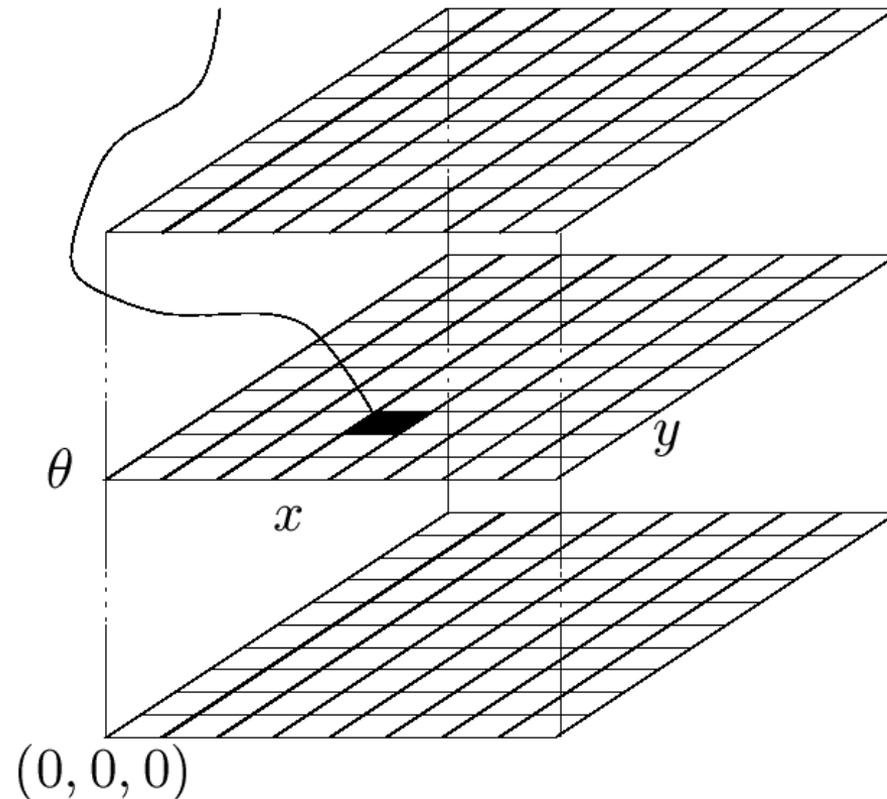
Piecewise Constant



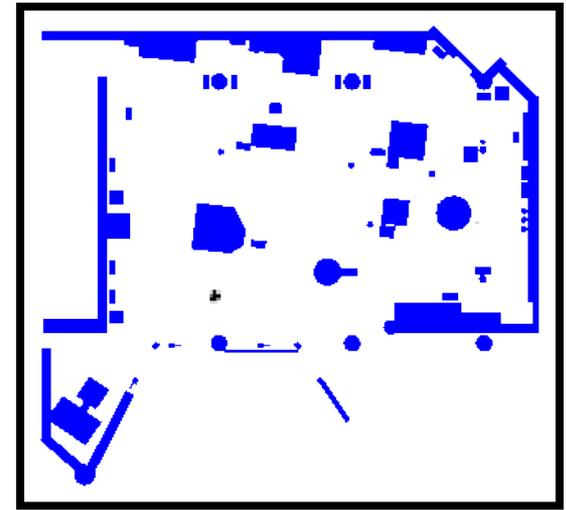
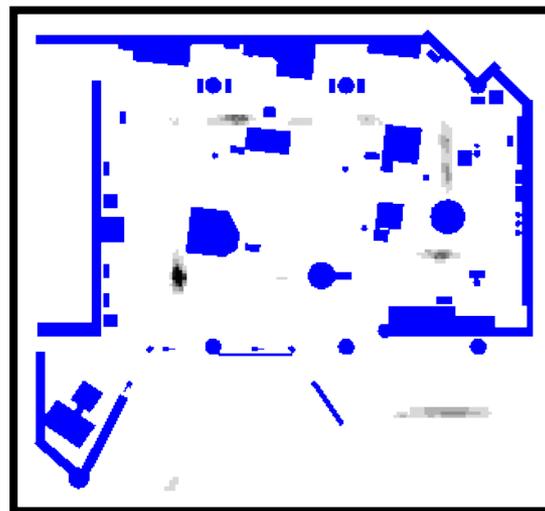
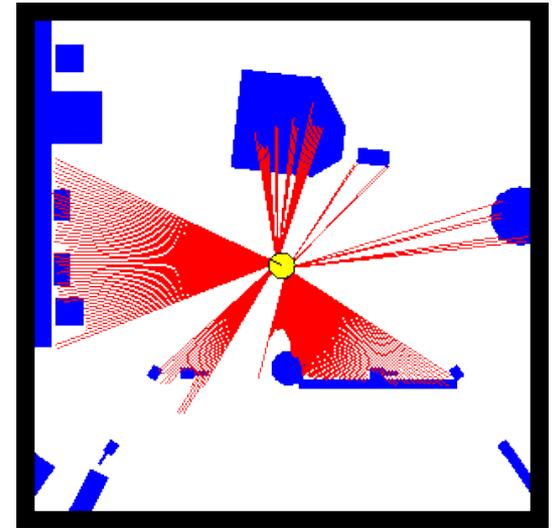
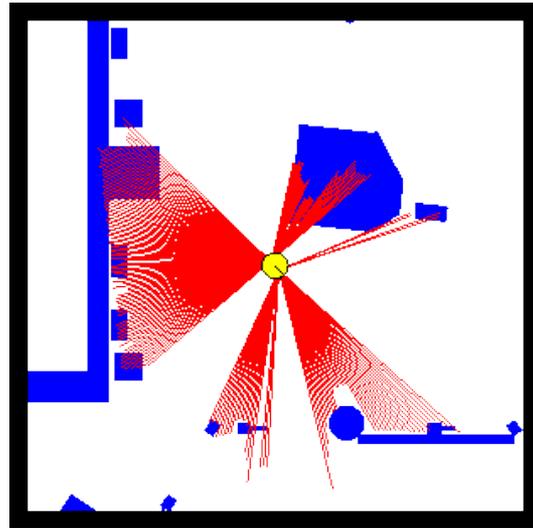
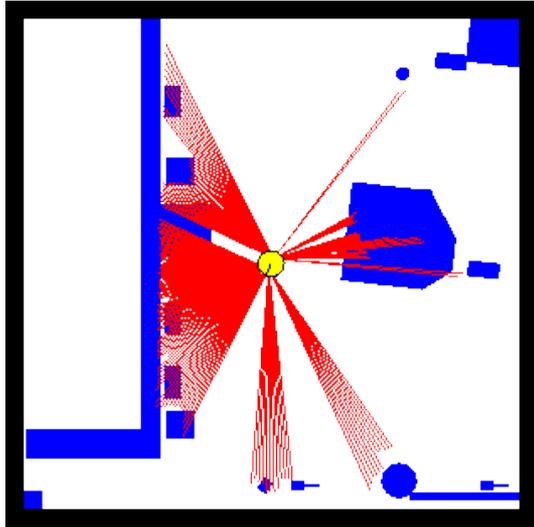
[Burgard et al. 96,98], [Fox et al. 99],
[Konolige et al. 99]

Piecewise Constant Representation

$$bel(x_t = \langle x, y, \theta \rangle)$$

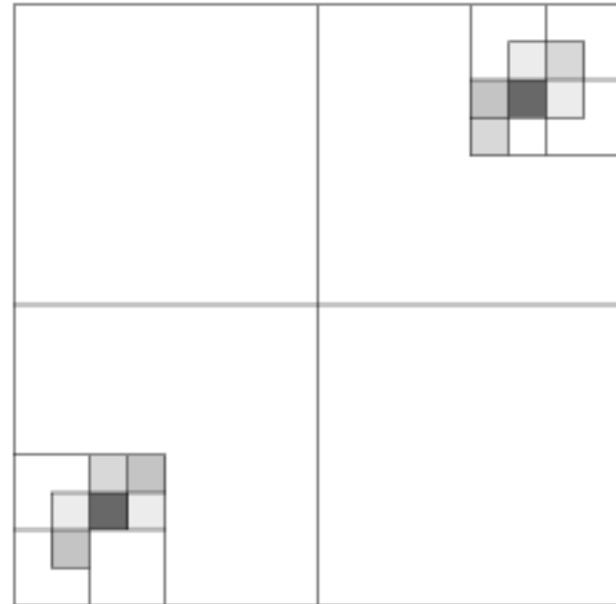
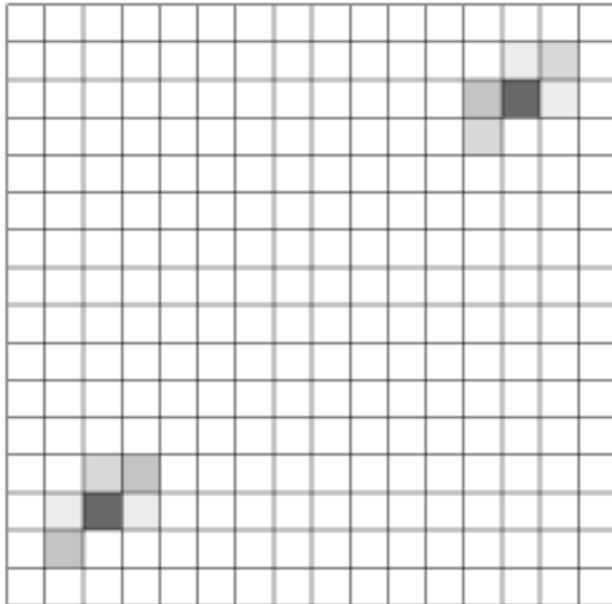


Grid-based Localization



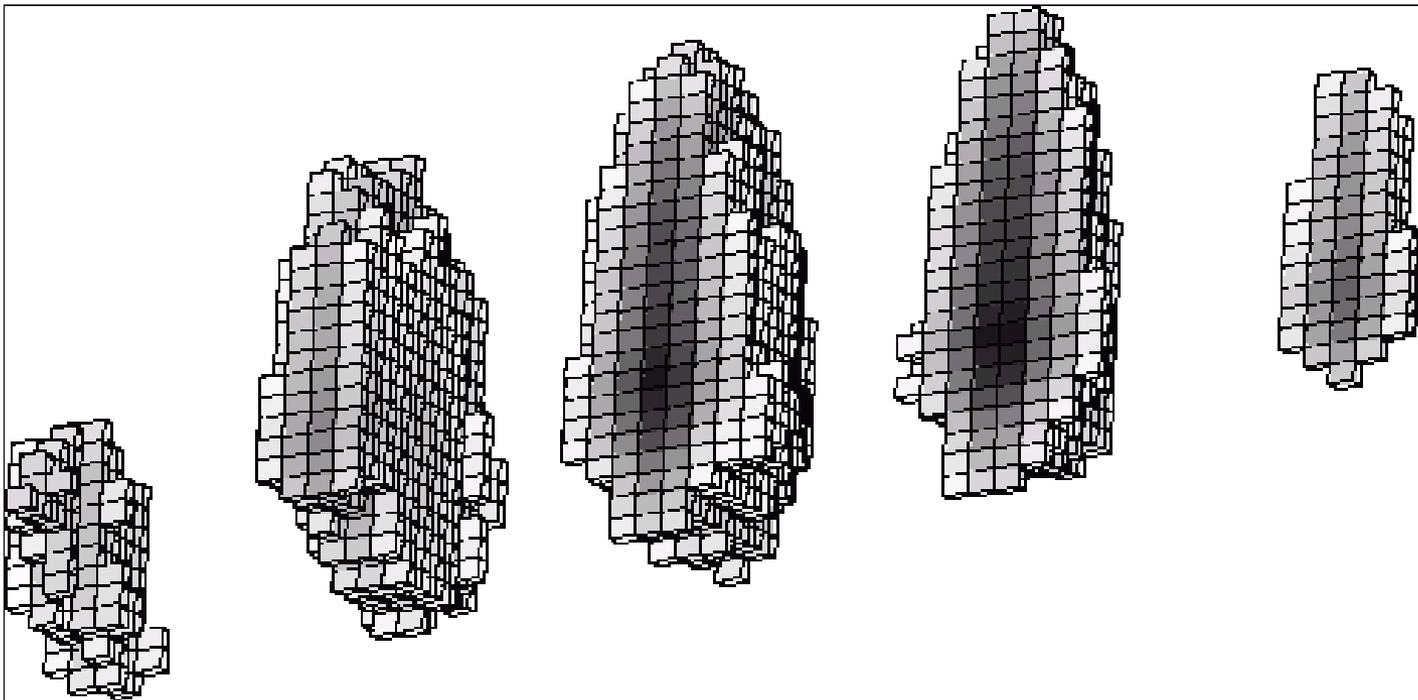
Tree-based Representations (1)

Idea: Represent density using a variant of Octrees



Tree-based Representations (2)

- Efficient in space and time
- Multi-resolution



Localization Algorithms - Comparison

	Kalman filter	Multi-hypot hesis tracking	Grid-based (fixed/variable)
Sensors	Gaussian	Gaussian	Non-Gaussian
Posterior	Gaussian	Multi-modal	Piecewise constant
Efficiency (memory)	++	++	-/+
Efficiency (time)	++	++	o/+
Implementation	+	o	+/o
Accuracy	++	++	+/>++
Robustness	-	+	++
Global localization	No	Yes	Yes

Localization Algorithms - Comparison

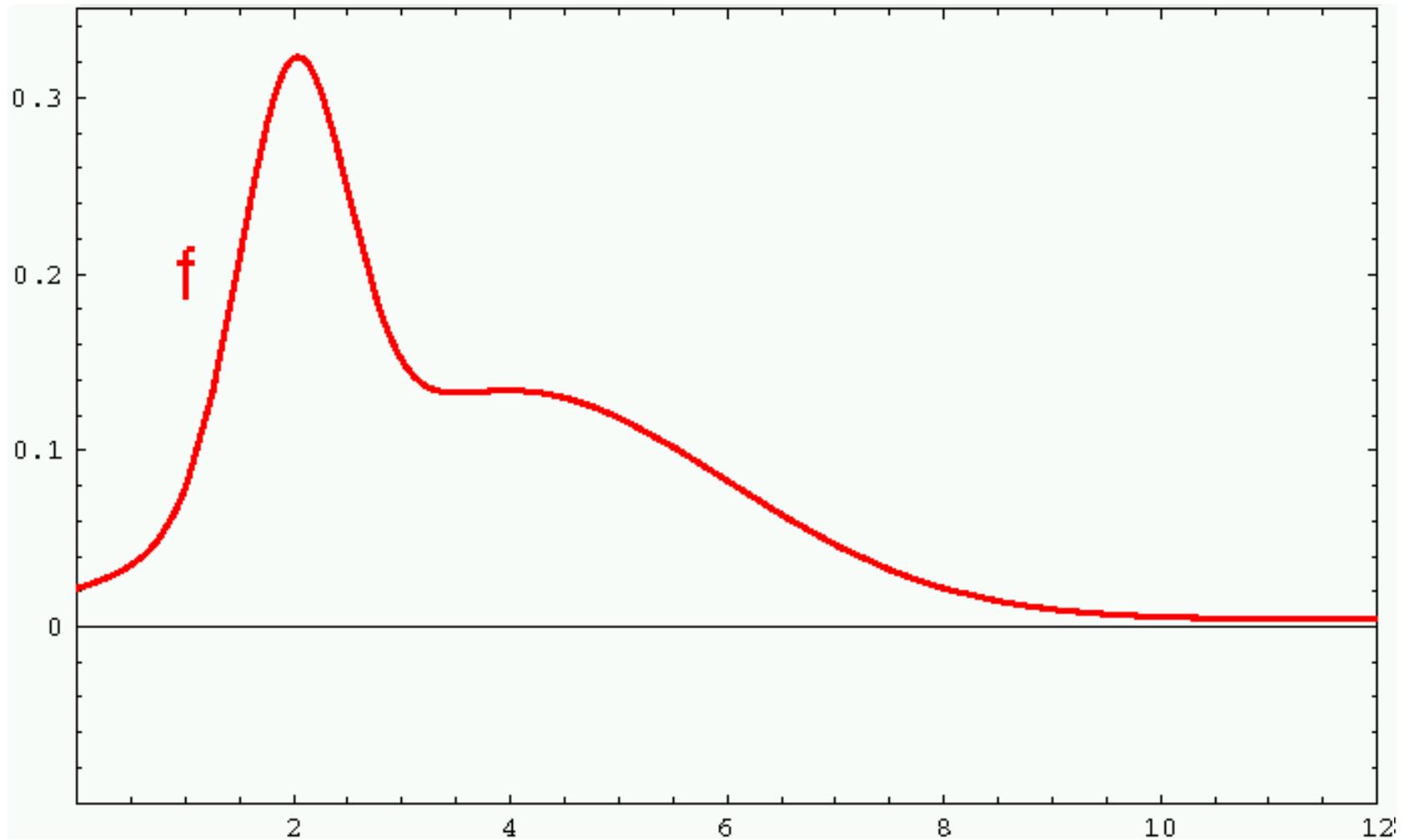
	Kalman filter	Multi-hypot hesis tracking	Grid-based (fixed/variable)	Topological maps
Sensors	Gaussian	Gaussian	Non-Gaussian	Features
Posterior	Gaussian	Multi-modal	Piecewise constant	Piecewise constant
Efficiency (memory)	++	++	-/+	++
Efficiency (time)	++	++	o/+	++
Implementation	+	o	+/o	+/o
Accuracy	++	++	+/>++	-
Robustness	-	+	++	+
Global localization	No	Yes	Yes	Yes

Particle Filters

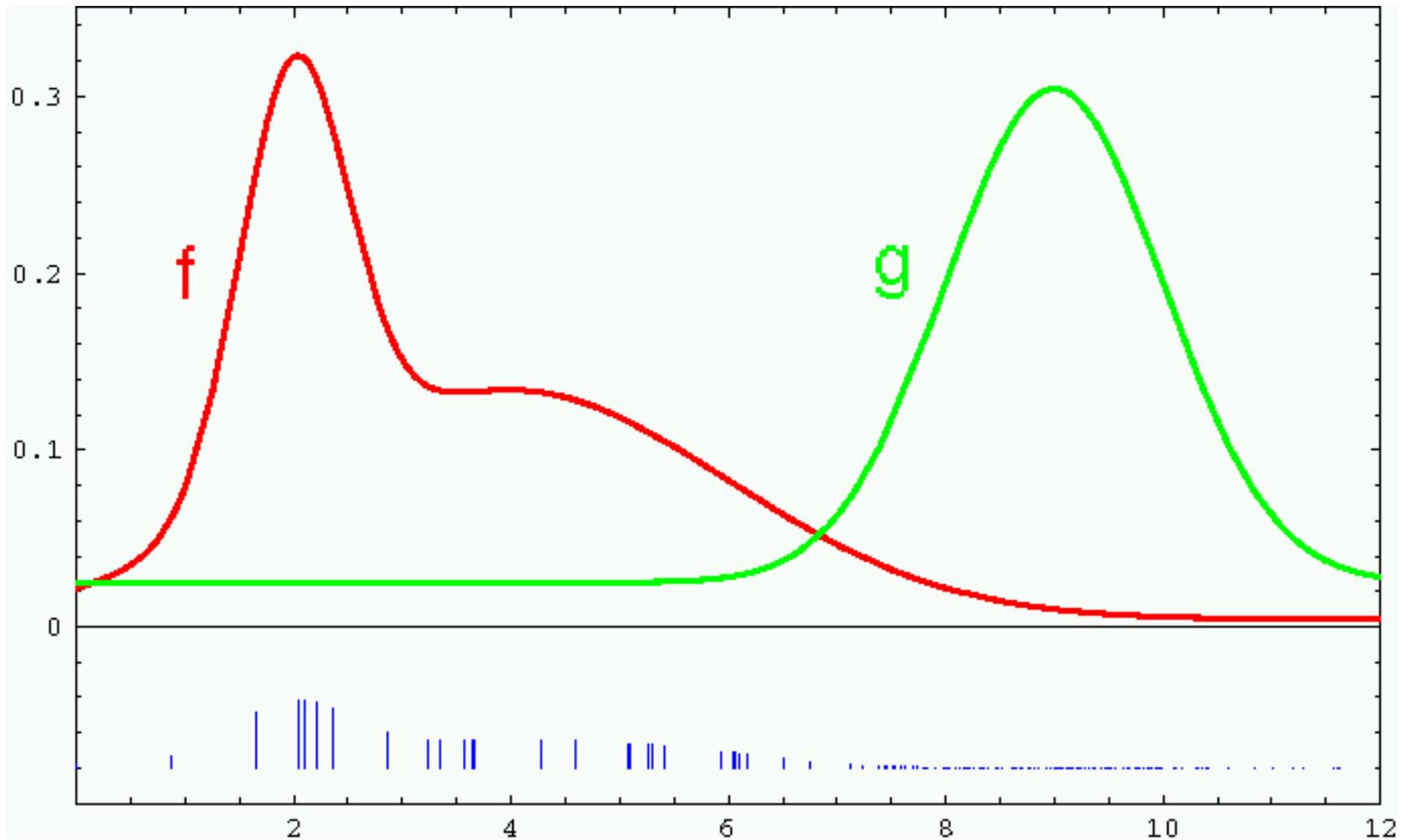
- Represent density by random **samples**
- Estimation of **non-Gaussian, nonlinear** processes
- Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter, Particle filter
- Filtering: [Rubin, 88], [Gordon et al., 93], [Kitagawa 96]
- Computer vision: [Isard and Blake 96, 98]
- Dynamic Bayesian Networks: [Kanazawa et al., 95]

Monte Carlo Localization (MCL)

Represent Density Through Samples

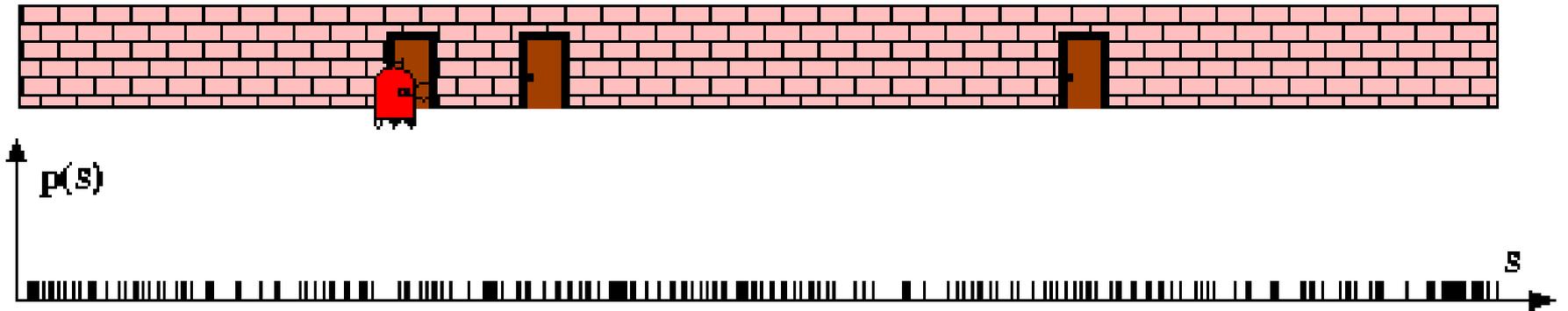


Importance Sampling



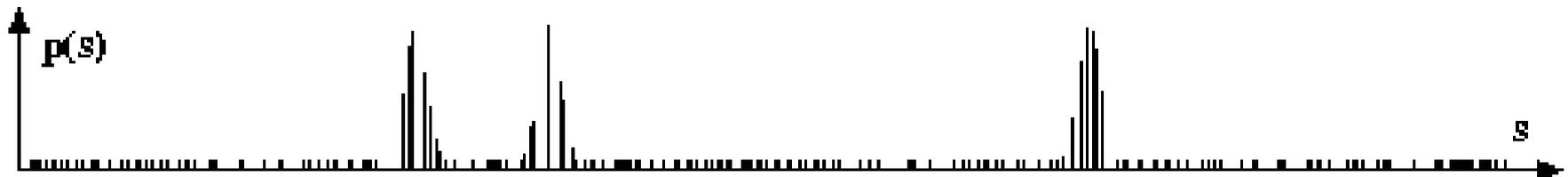
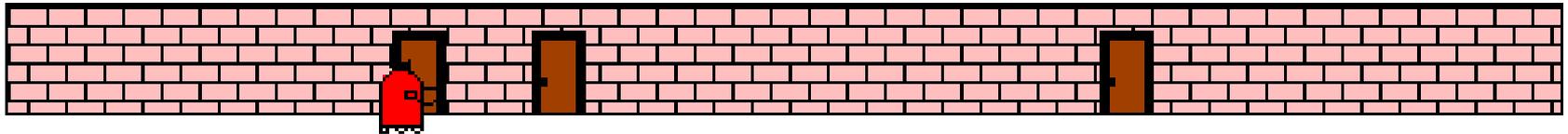
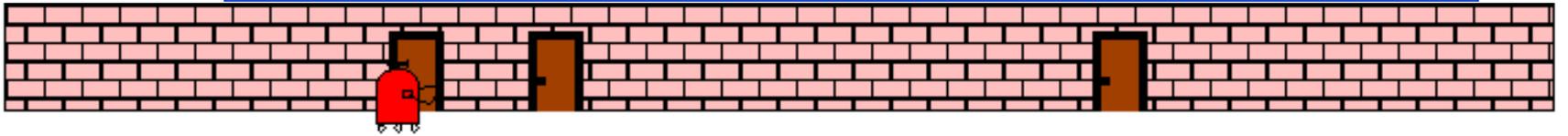
Weight samples: $w = \frac{f}{g}$

MCL: Global Localization



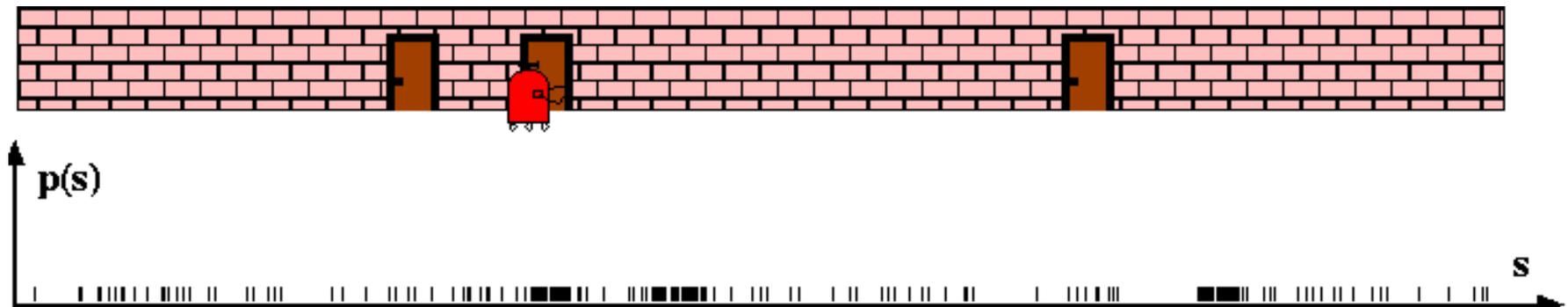
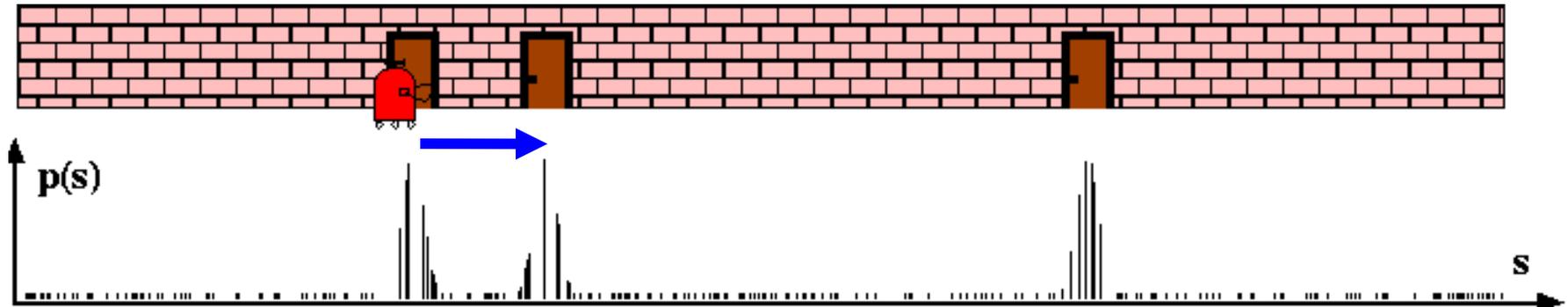
MCL: Sensor Update

$$\begin{aligned} Bel(x) &\leftarrow \alpha p(z | x) Bel^-(x) \\ w &\leftarrow \frac{\alpha p(z | x) Bel^-(x)}{Bel^-(x)} = \alpha p(z | x) \end{aligned}$$



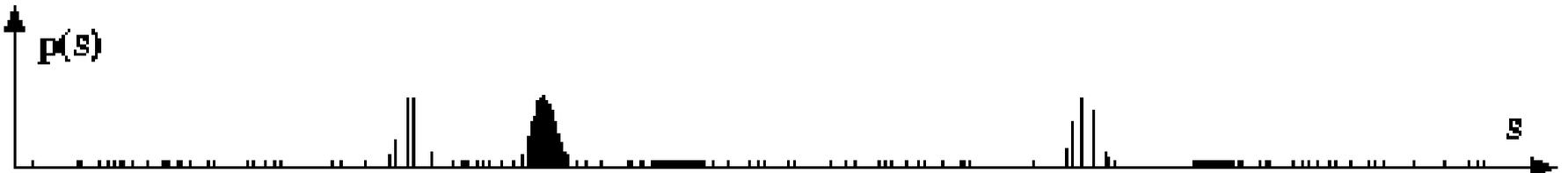
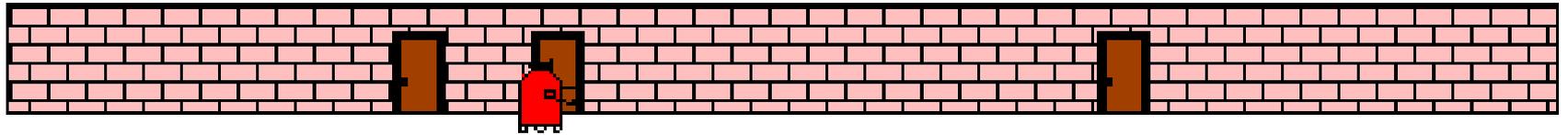
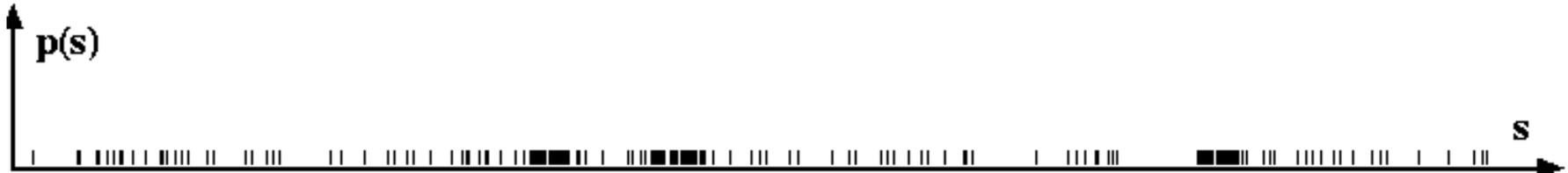
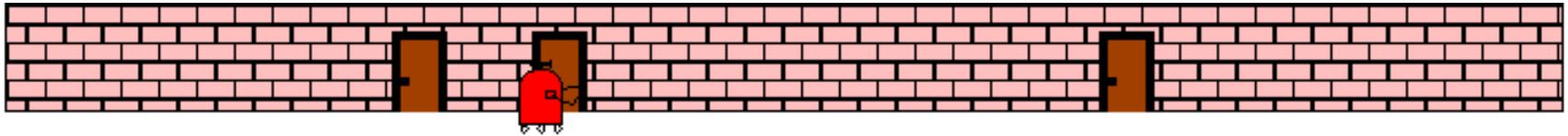
MCL: Robot Motion

$$Bel^-(x) \leftarrow \int p(x|u, x') Bel(x') dx'$$



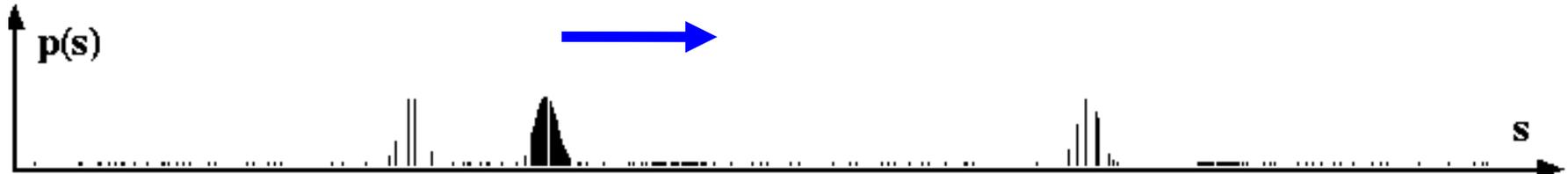
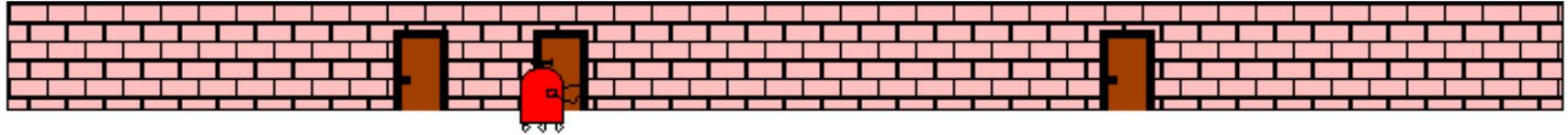
MCL: Sensor Update

$$\begin{aligned} Bel(x) &\leftarrow \alpha p(z | x) Bel^-(x) \\ w &\leftarrow \frac{\alpha p(z | x) Bel^-(x)}{Bel^-(x)} = \alpha p(z | x) \end{aligned}$$



MCL: Robot Motion

$$Bel^-(x) \leftarrow \int p(x|u, x') Bel(x') dx'$$



Particle Filter Algorithm

1. Algorithm **particle_filter**(S_{t-1}, u_{t-1}, z_t):

2. $S_t = \emptyset, \eta = 0$

3. **For** $i = 1 \dots n$ *Generate new samples*

Sample index $j(i)$ from the discrete distribution given by w_{t-1}

1. Sample x_t^i from $p(x_t | x_{t-1}, u_{t-1})$ using $x_{t-1}^{j(i)}$ and u_{t-1}

2. $w_t^i = p(z_t | x_t^i)$ *Compute importance weight*

3. $\eta = \eta + w_t^i$ *Update normalization factor*

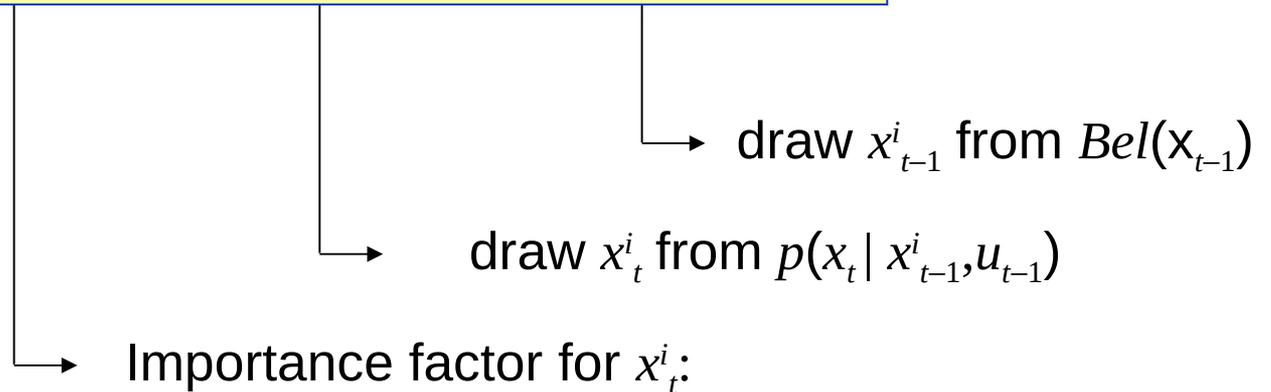
4. $S_t = S_t \cup \{ \langle x_t^i, w_t^i \rangle \}$ *Insert*

5. **For** $i = 1 \dots n$

6. $w_t^i = w_t^i / \eta$ *Normalize weights*

Particle Filter Algorithm

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



$$\begin{aligned} w_t^i &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{\eta p(z_t | x_t) p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1})}{p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1})} \\ &\propto p(z_t | x_t) \end{aligned}$$

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 Algorithm

1. Algorithm **systematic_resampling**(S, n):

1. $S' = \emptyset, c_1 = w^1$

2. **For** $i = 2 \dots n$ *Generate cdf*

3. $c_i = c_{i-1} + w^i$

4. $u_1 \sim U[0, n^{-1}], i = 1$ *Initialize threshold*

1. **For** $j = 1 \dots n$ *Draw samples ...*

2. $u_j = u_1 + n^{-1} \cdot (j - 1)$ *Advance threshold*

3. **While** ($u_j > c_i$) *Skip until next threshold reached*

4. $i = i + 1$

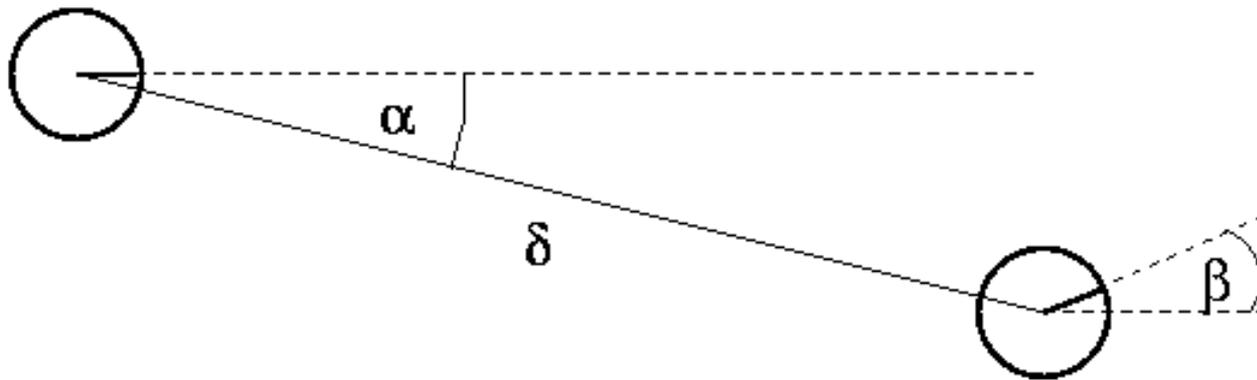
5. $S' = S' \cup \{ \langle x^i, n^{-1} \rangle \}$ *Insert*

1. **Return** S'

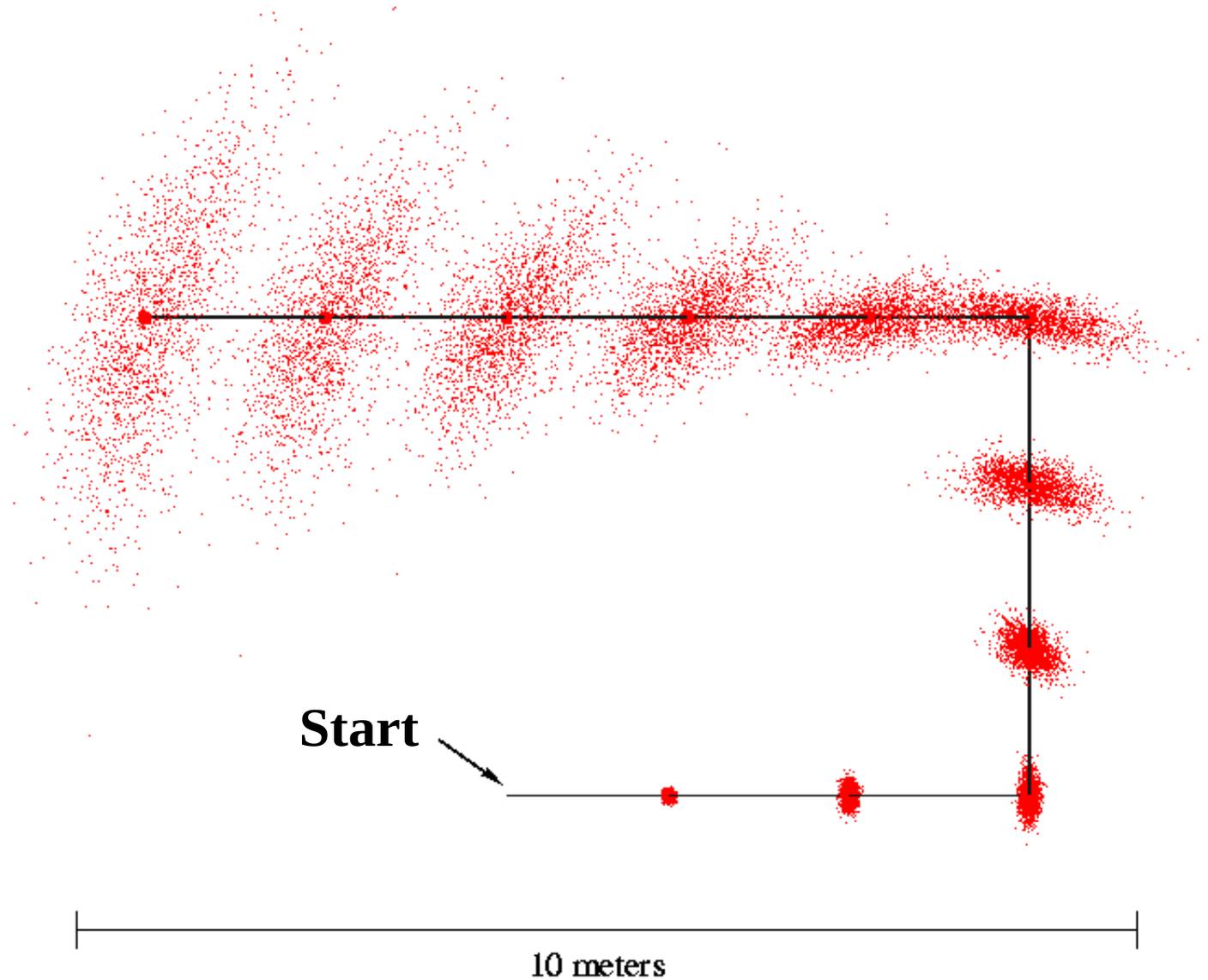
Also called **stochastic universal sampling**

Motion Model $p(x_t | a_{t-1}, x_{t-1})$

Model odometry error as Gaussian noise on α , β , and δ



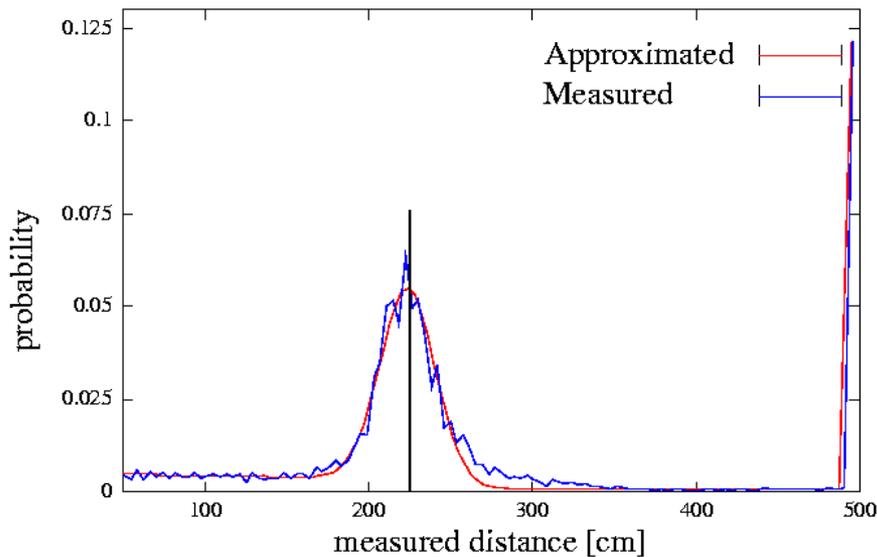
Motion Model $p(x_t | a_{t-1}, x_{t-1})$



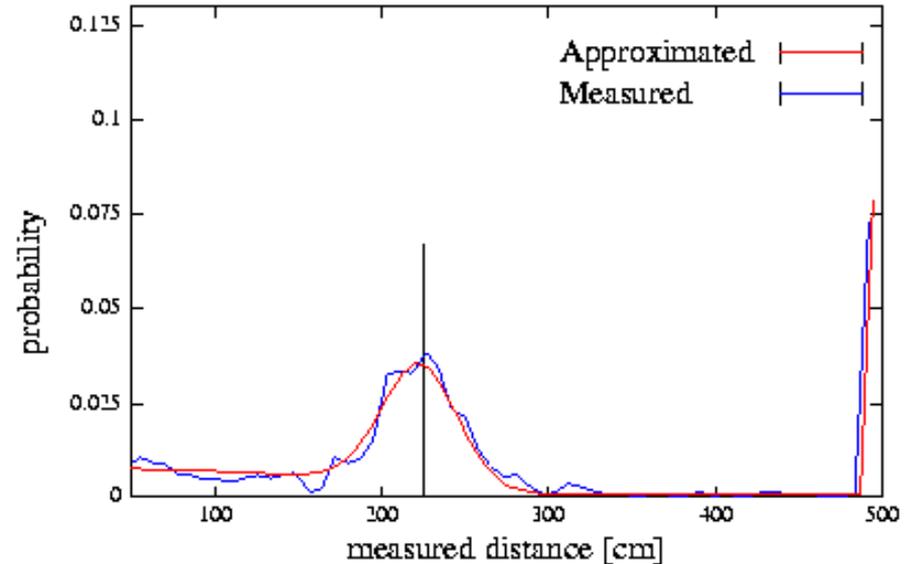
Model for Proximity Sensors

The sensor is reflected either by a **known** or by an **unknown** obstacle:

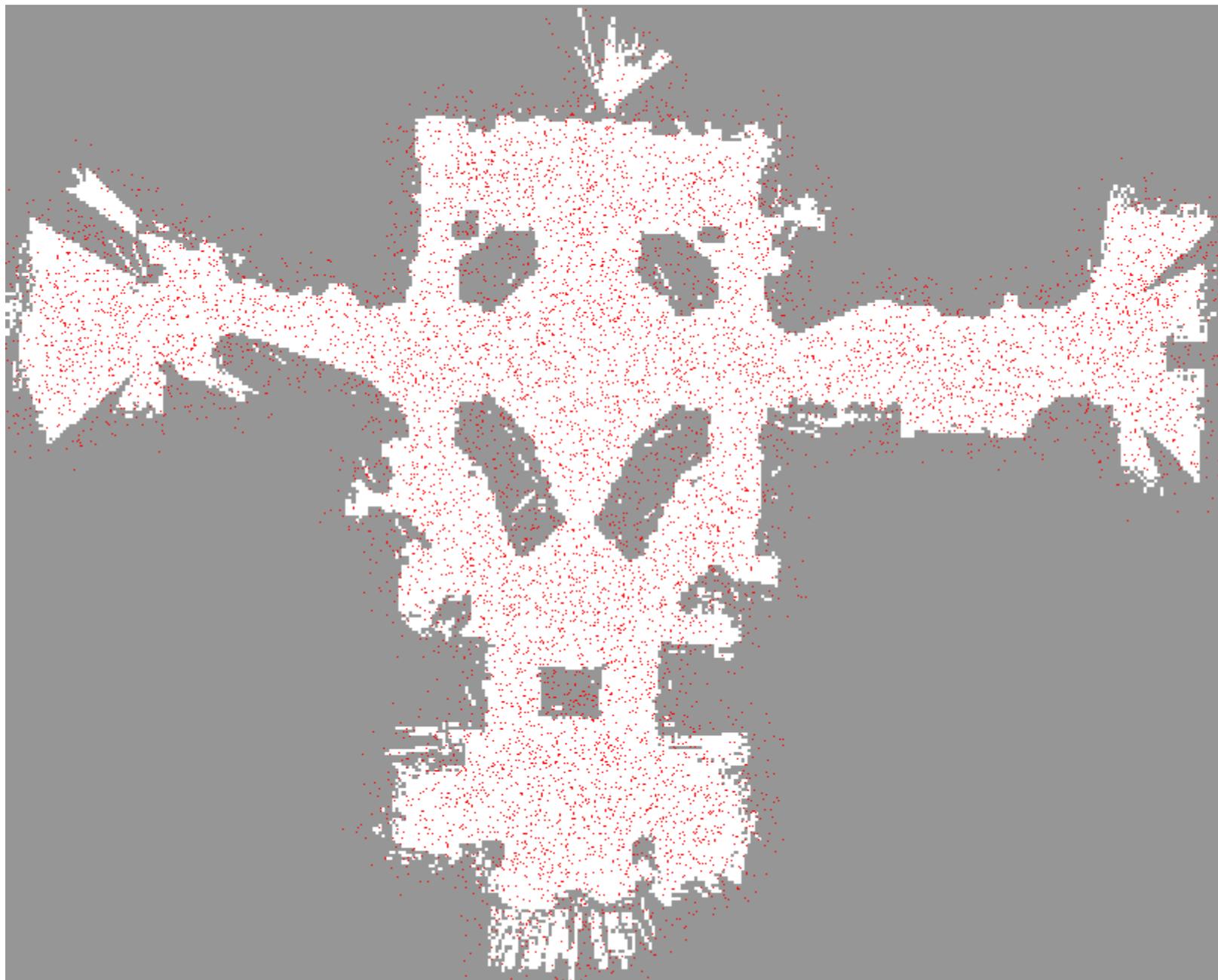
$$P(d_i | l) = 1 - (1 - (1 - \sum_{j < i} P_u(d_j)) c_d P_m(d_i | l))) \cdot (1 - (1 - \sum_{j < i} P(d_j)) c_r)$$

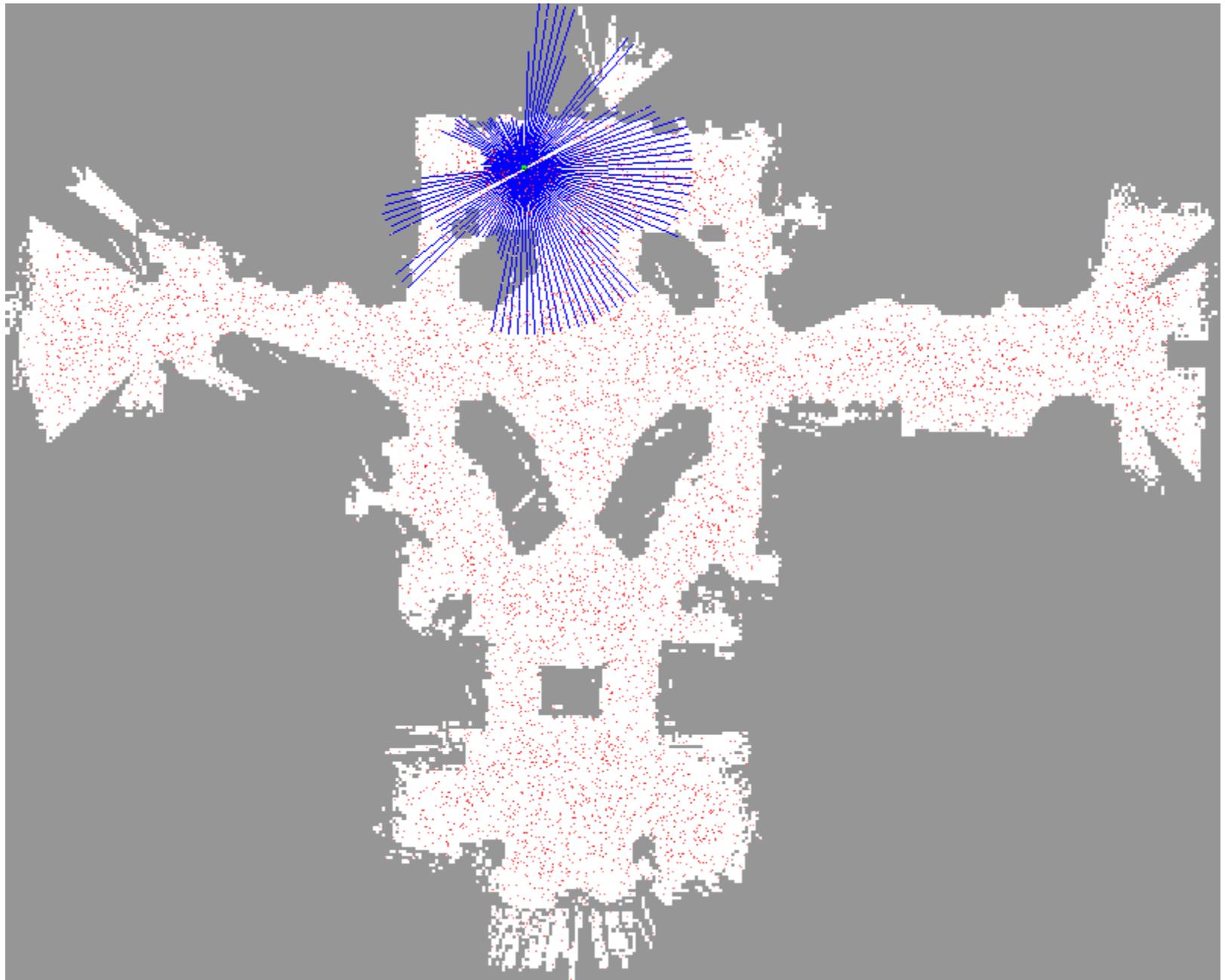


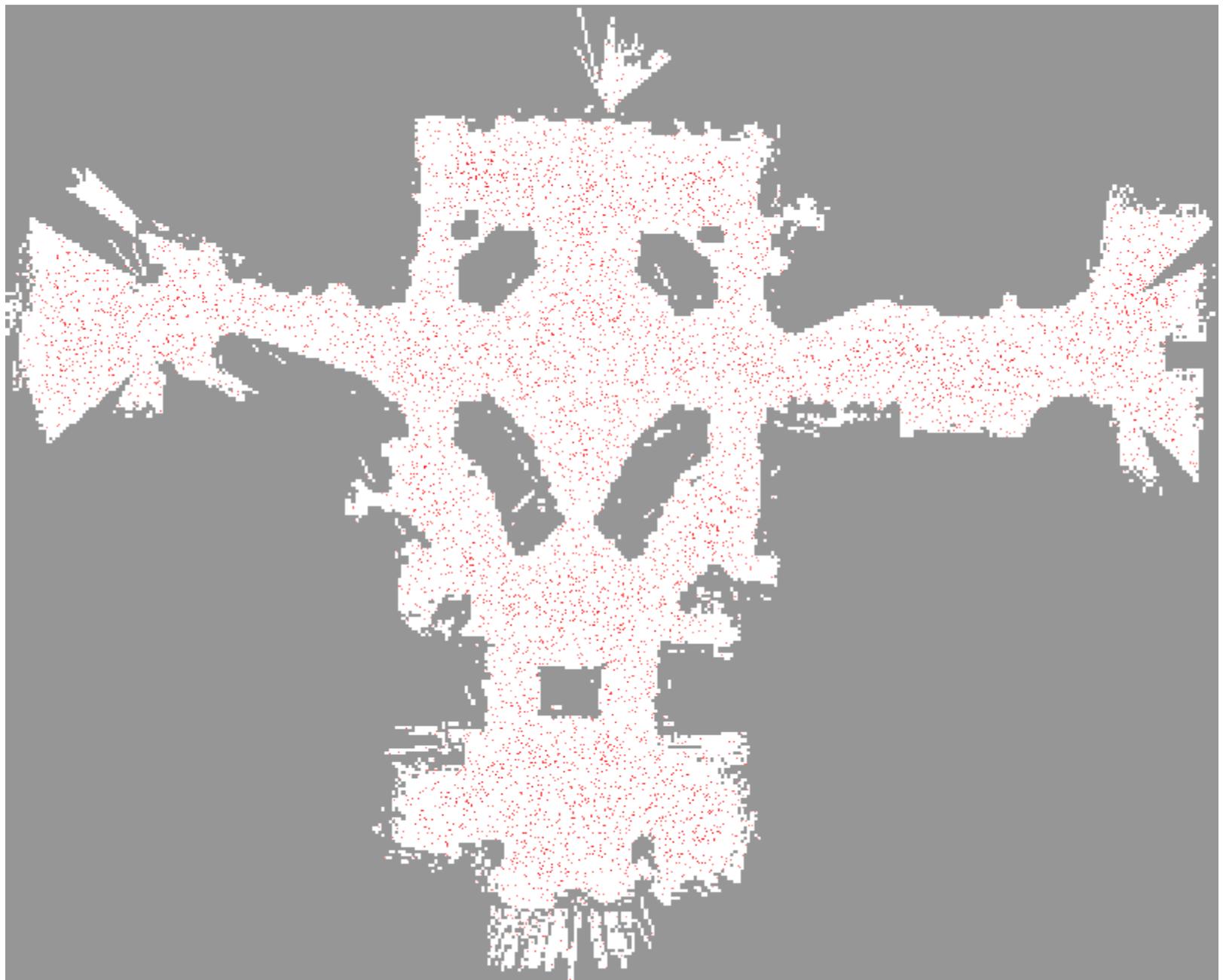
Laser sensor

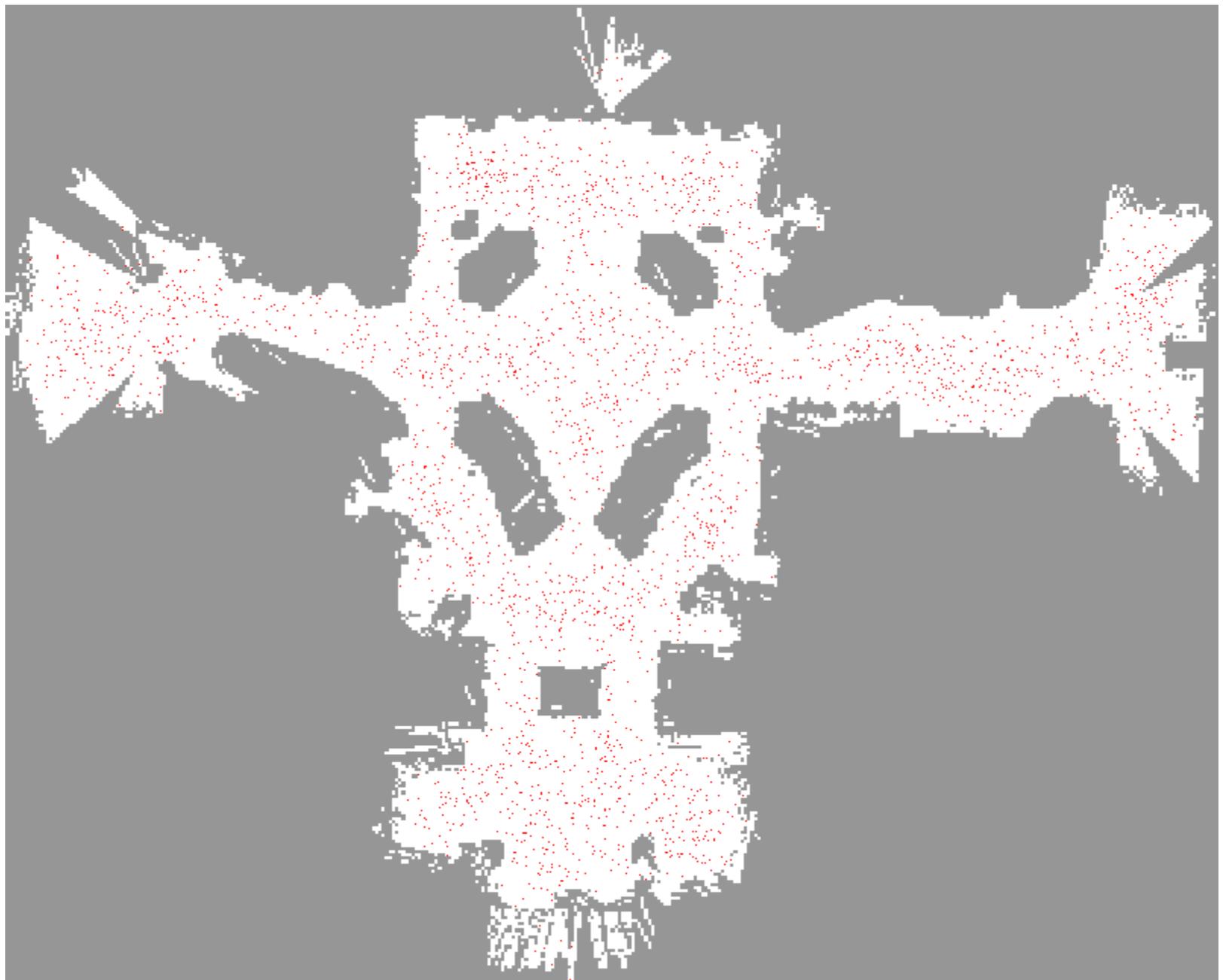


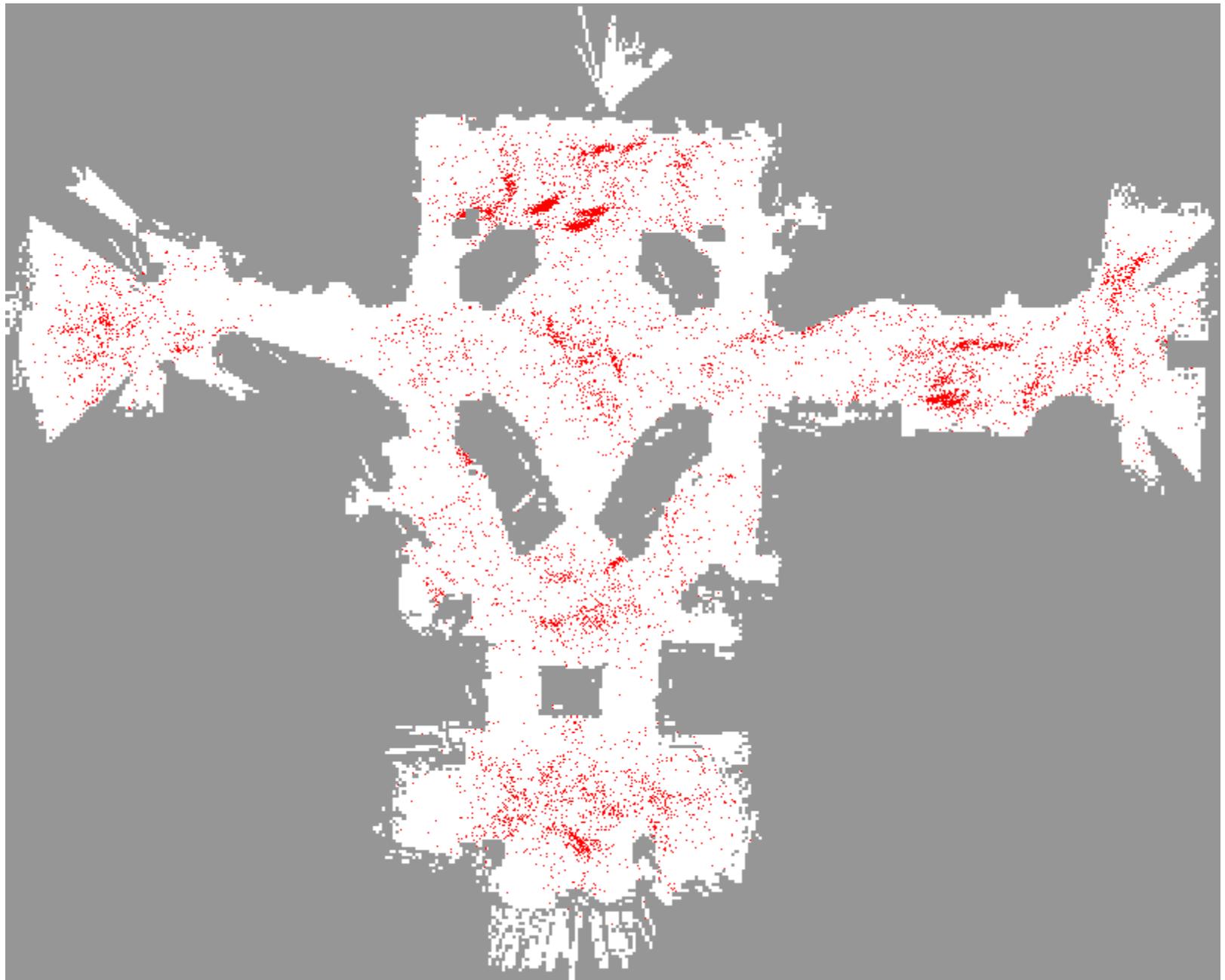
Sonar sensor

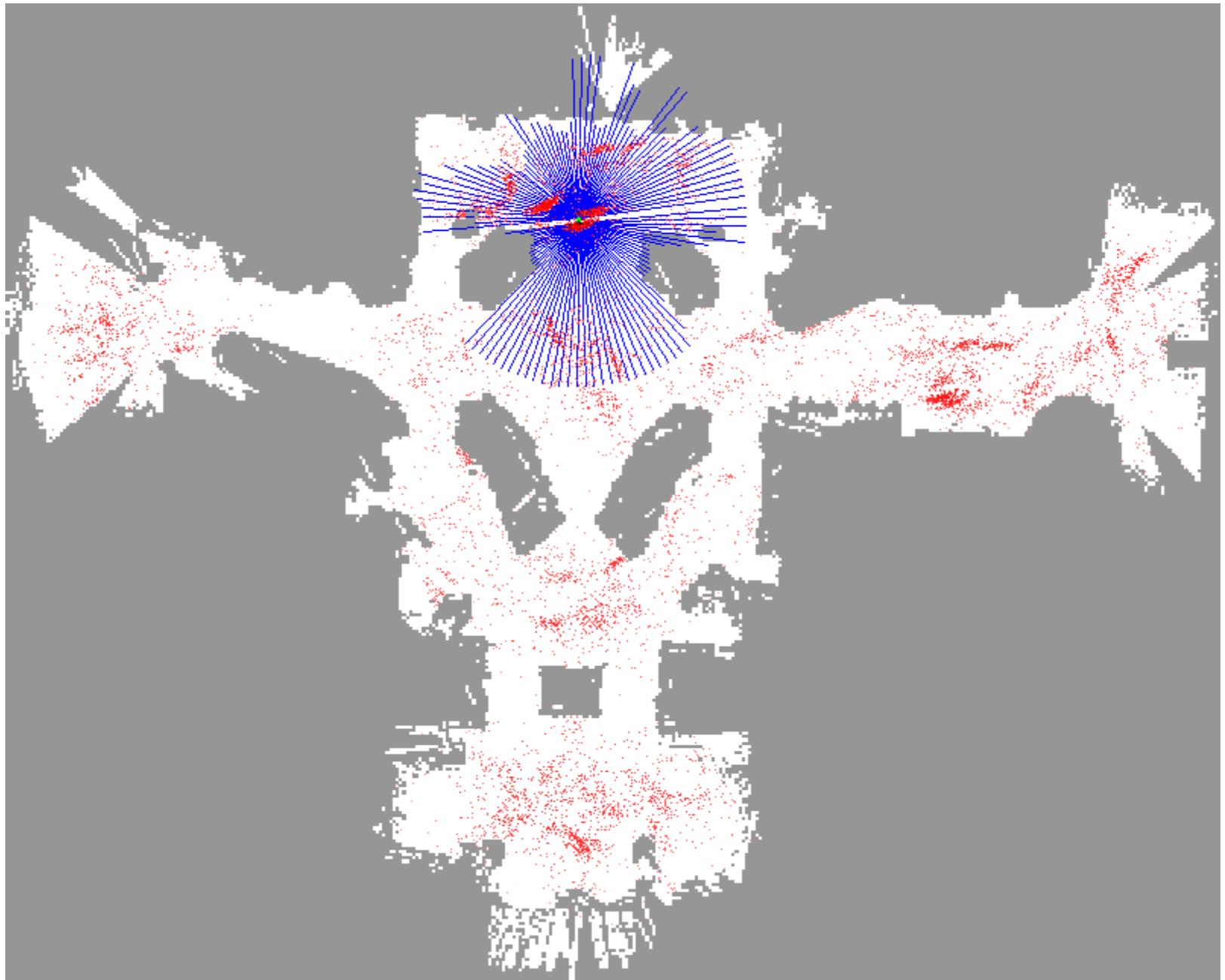


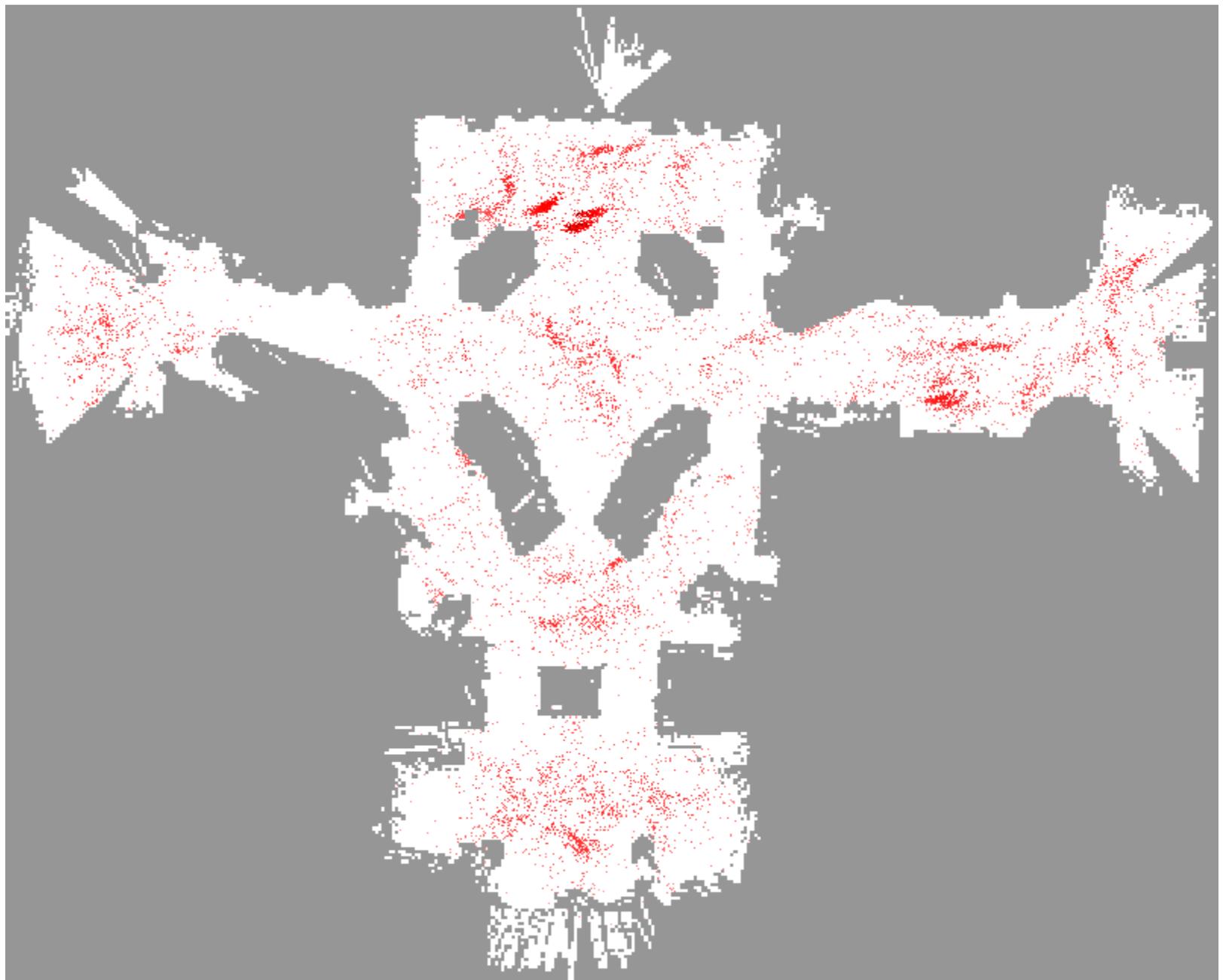


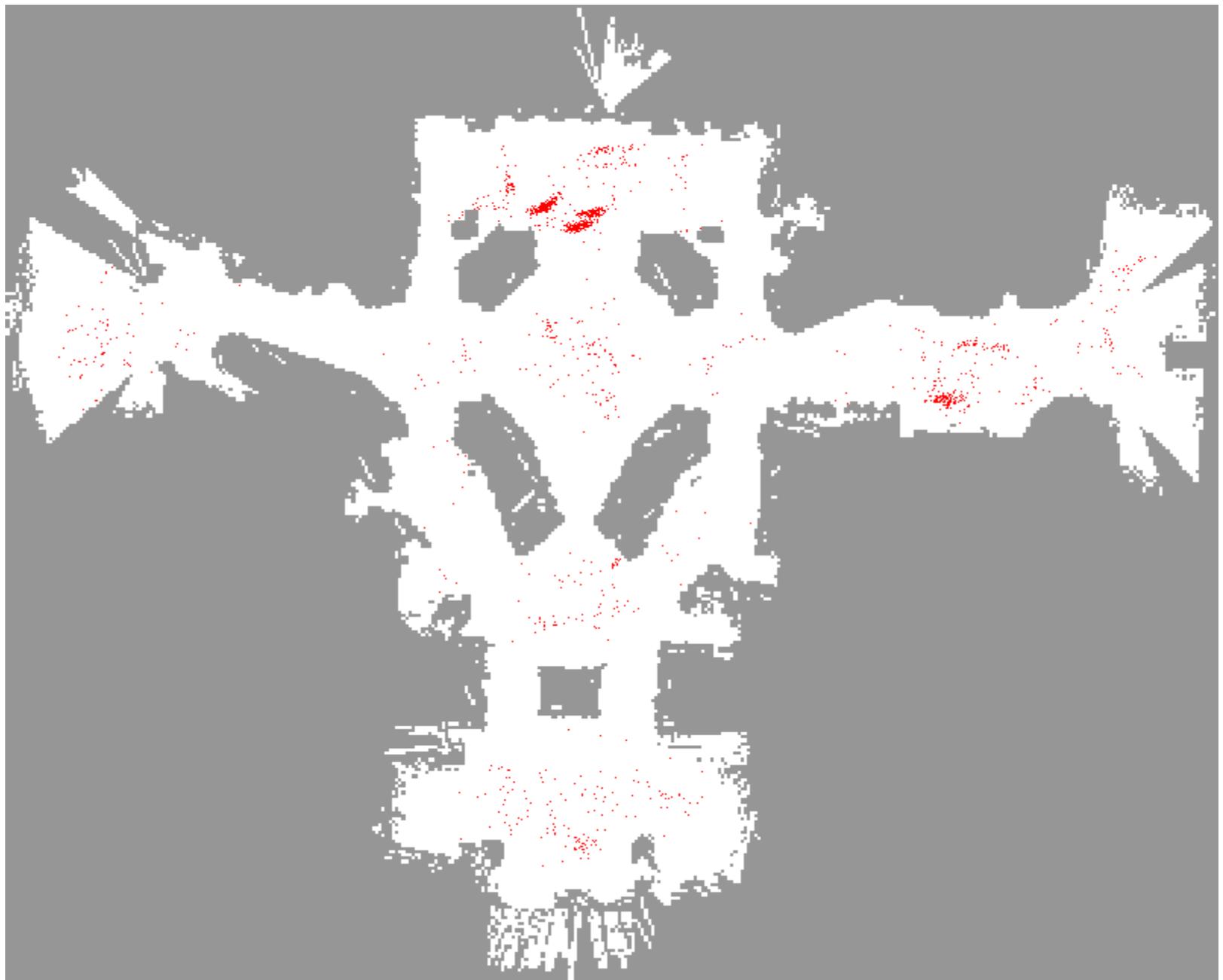




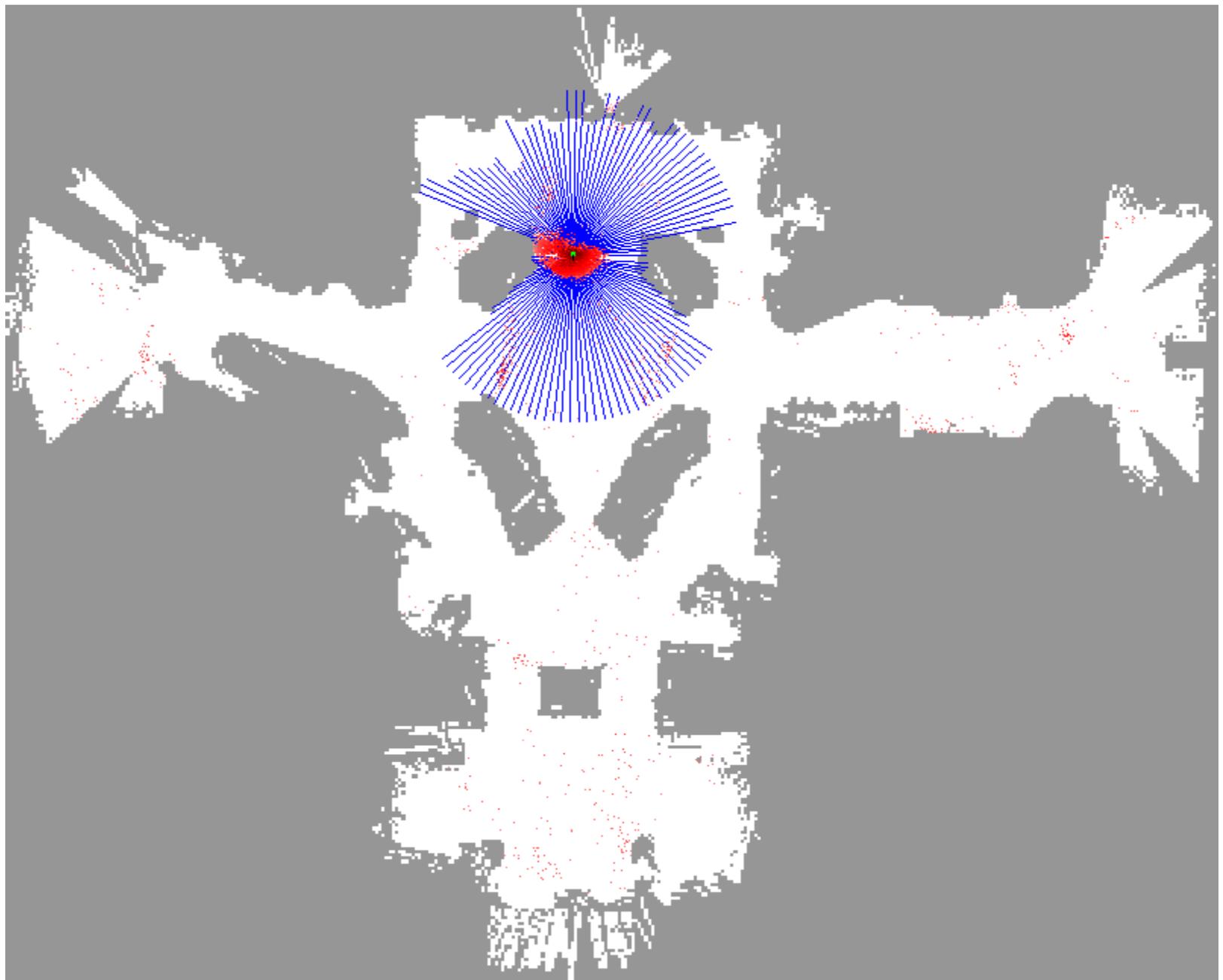




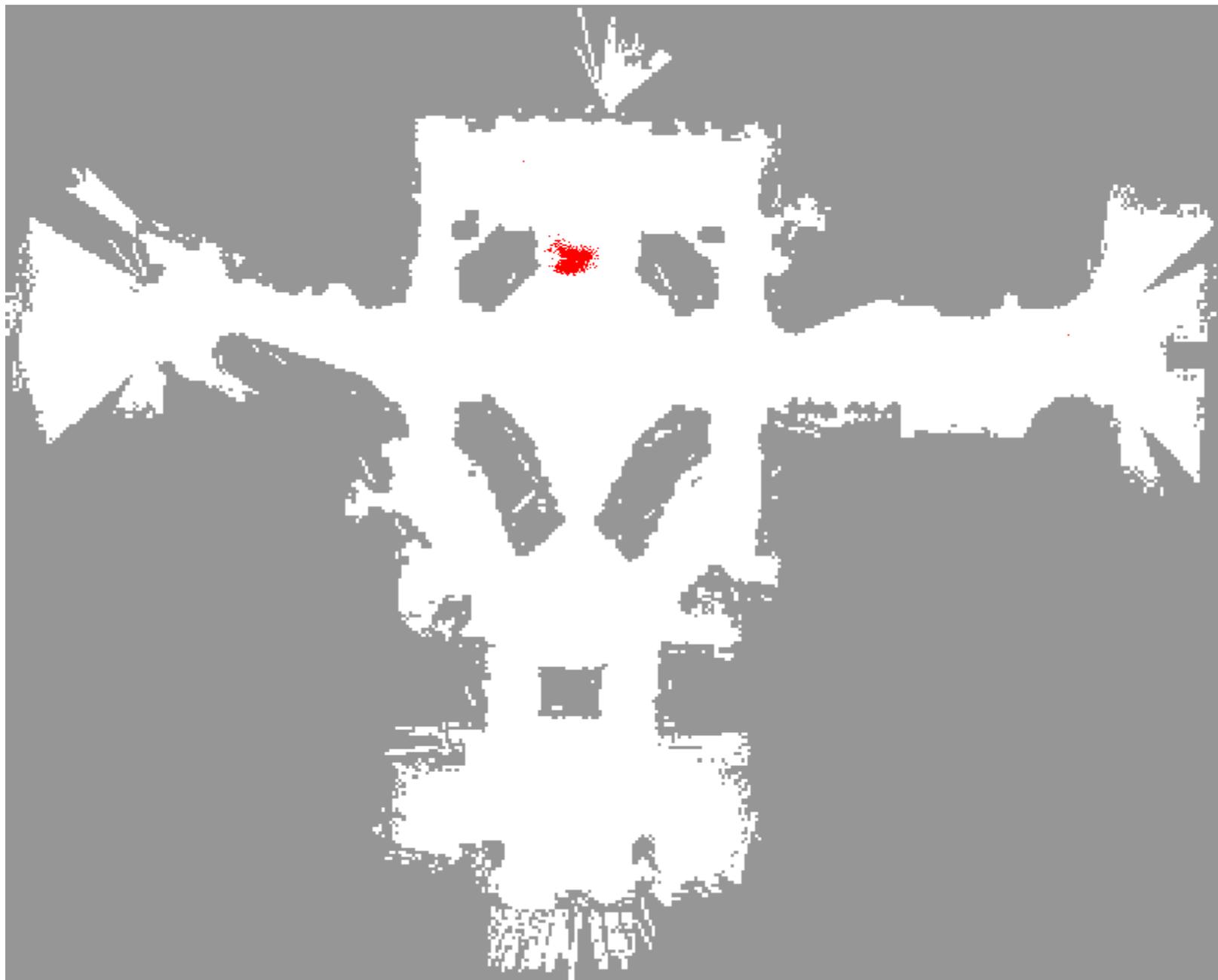


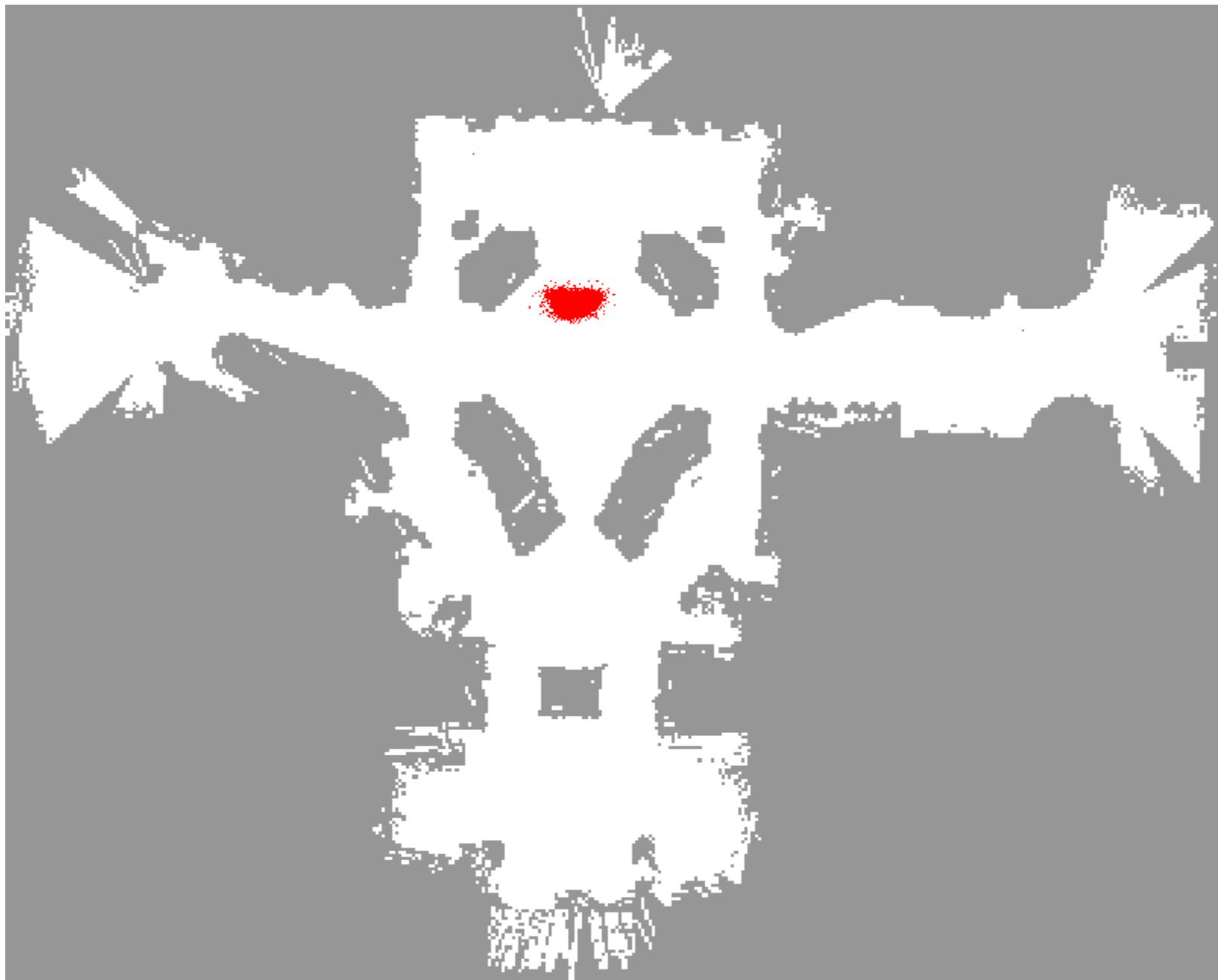


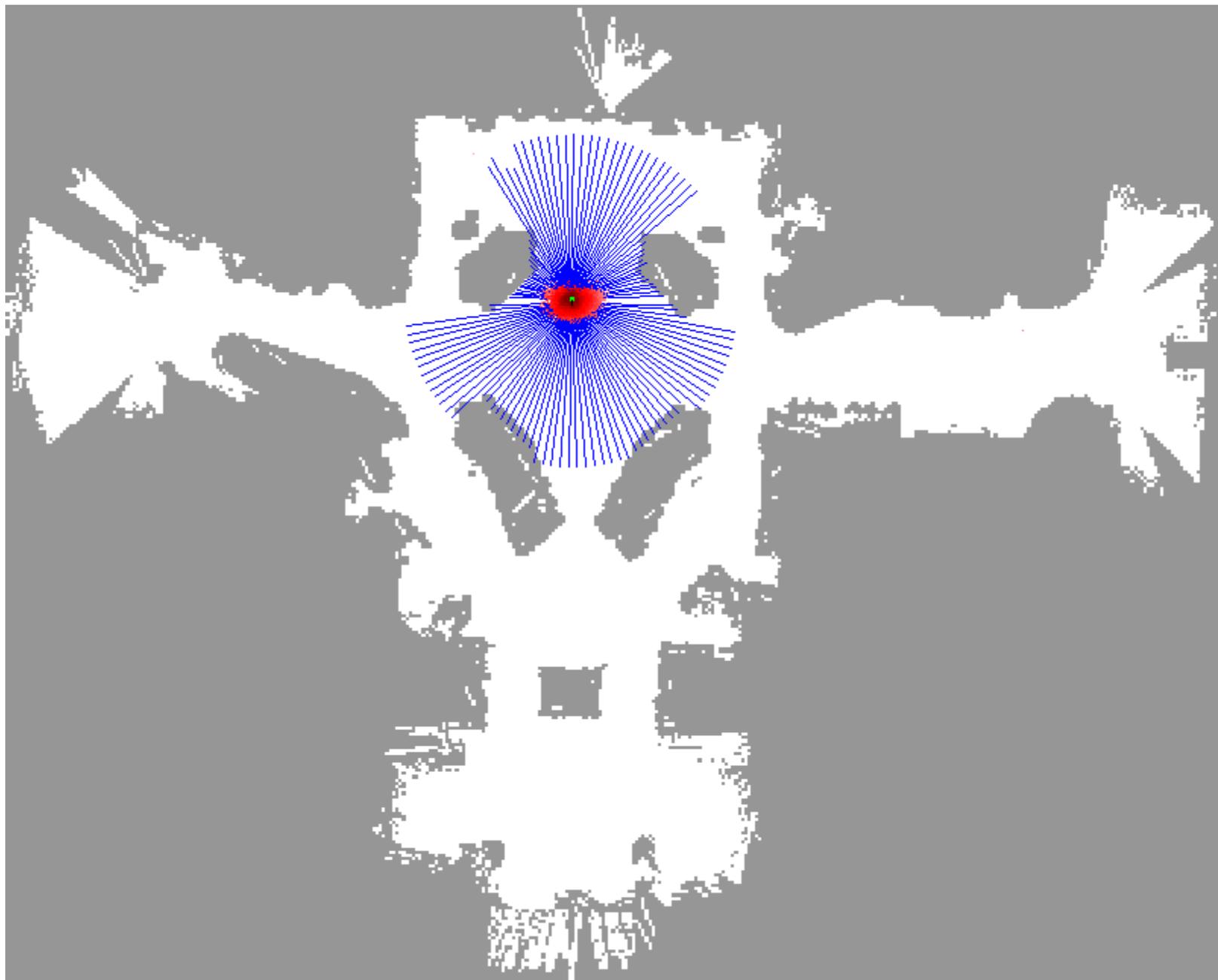


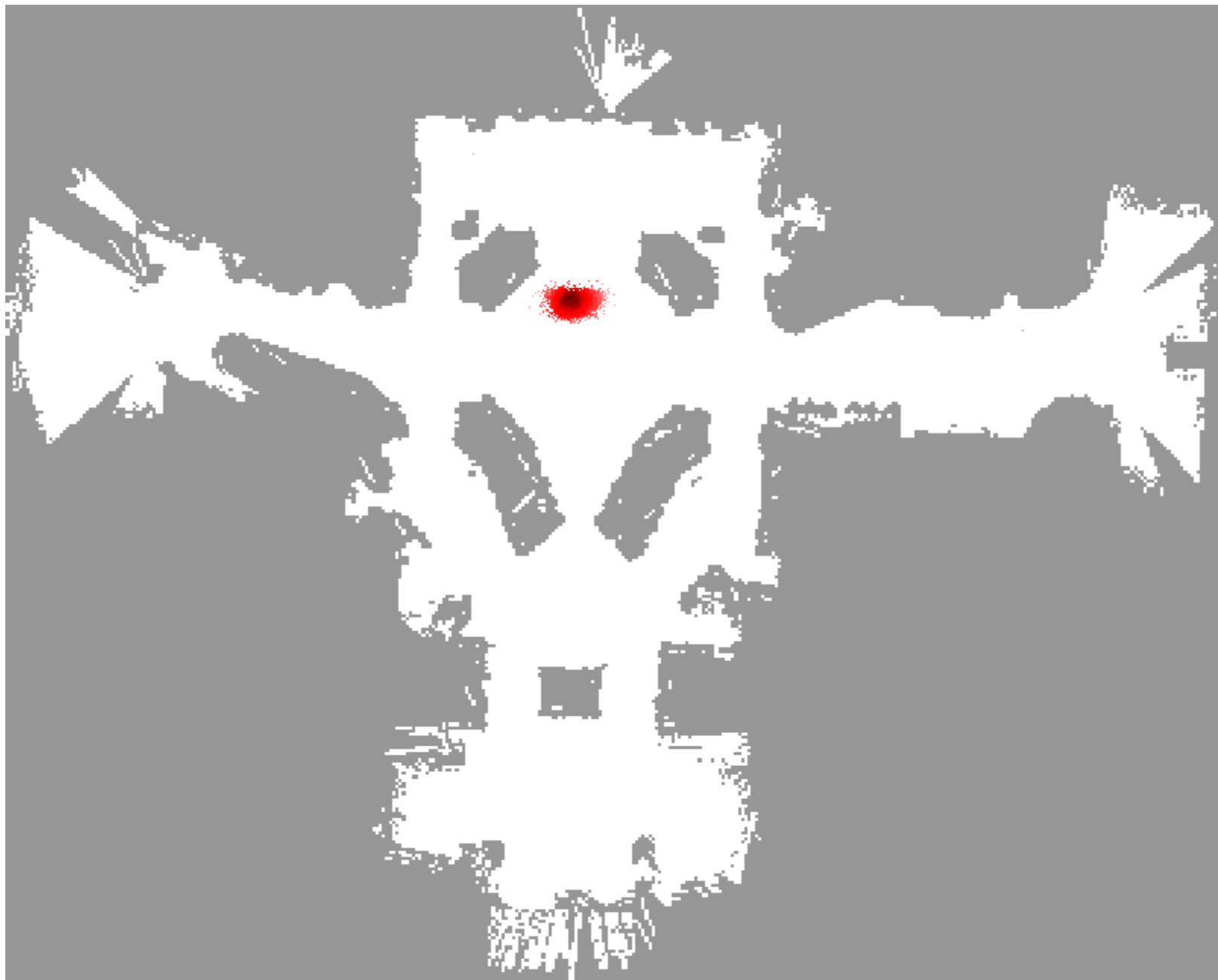


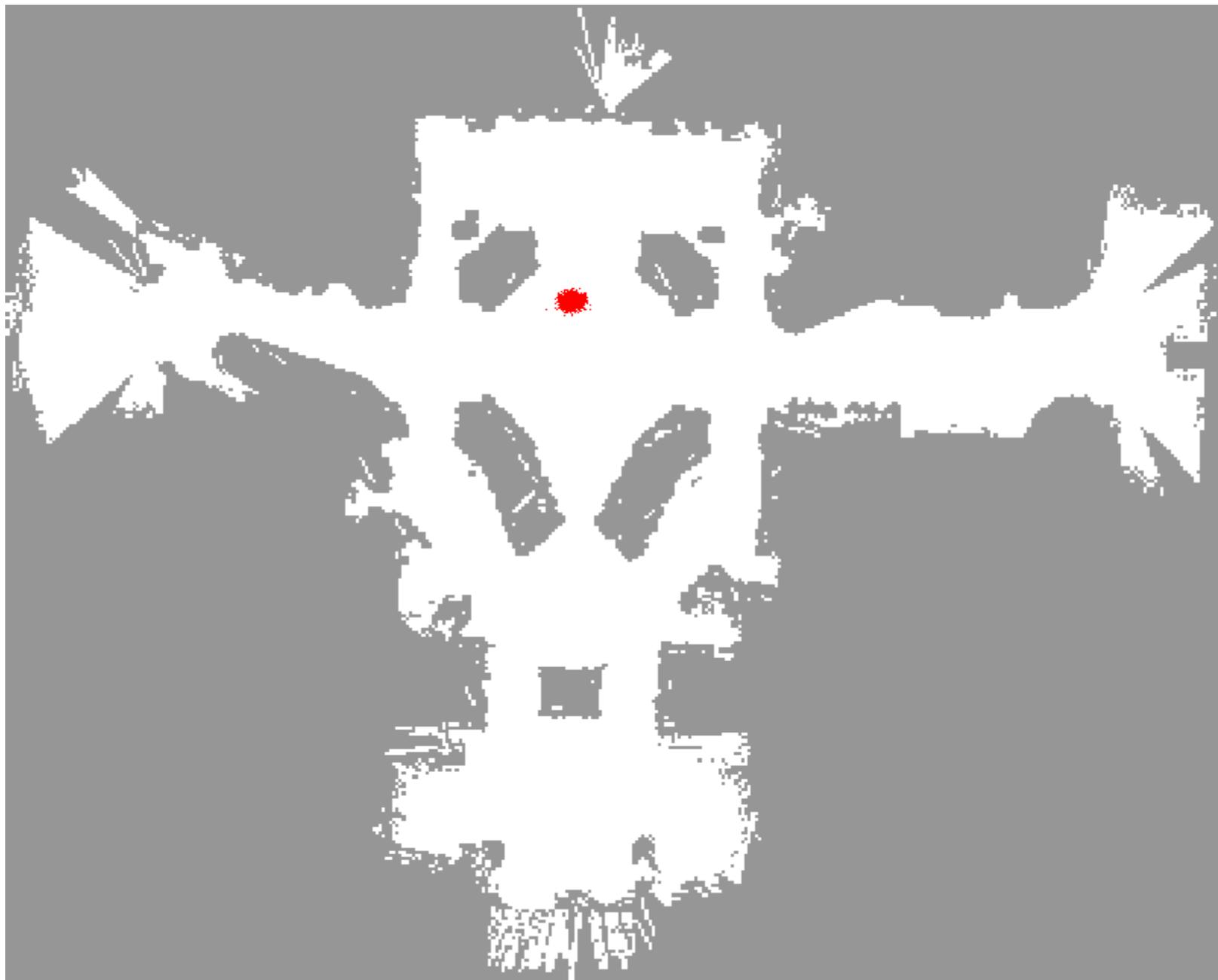


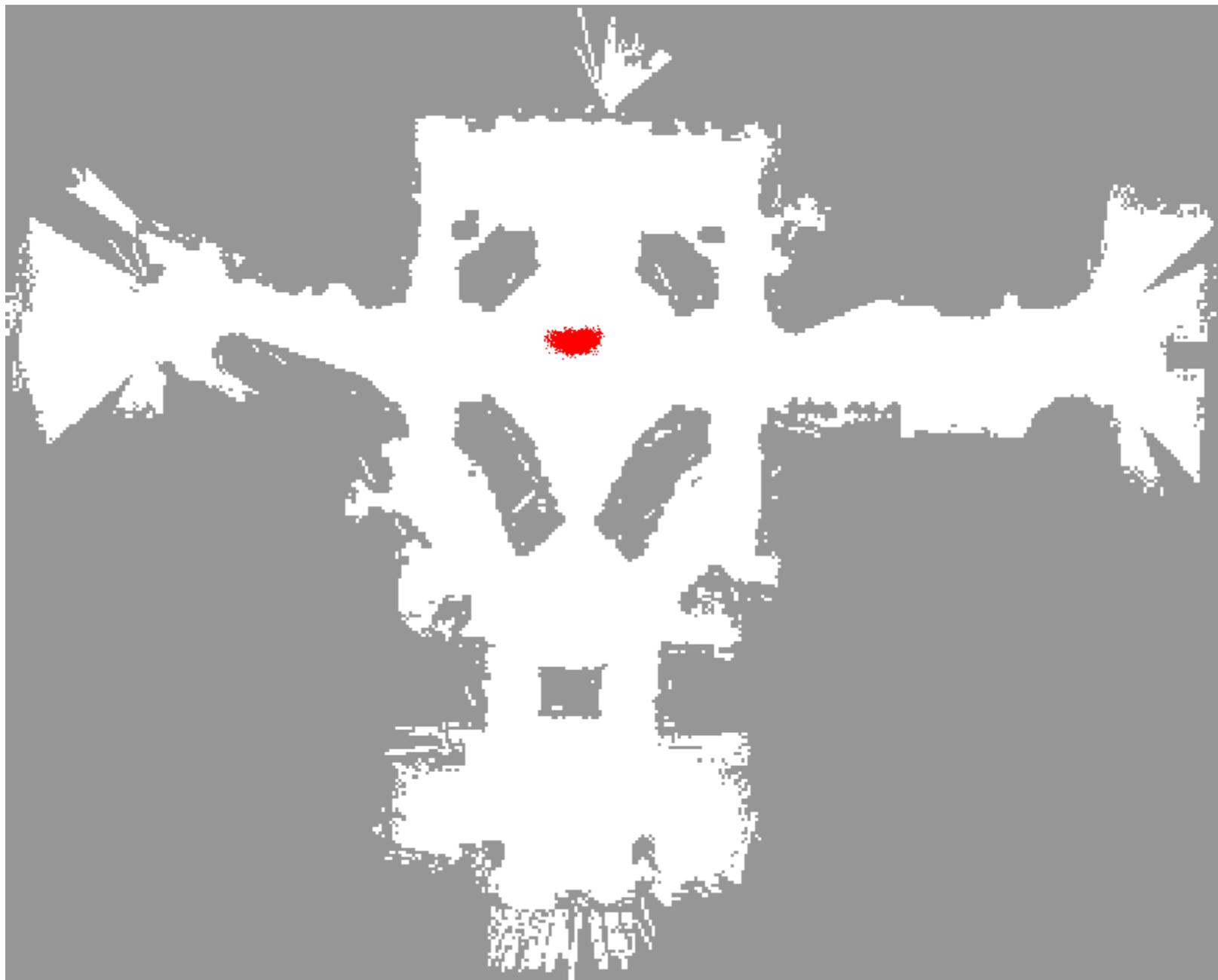


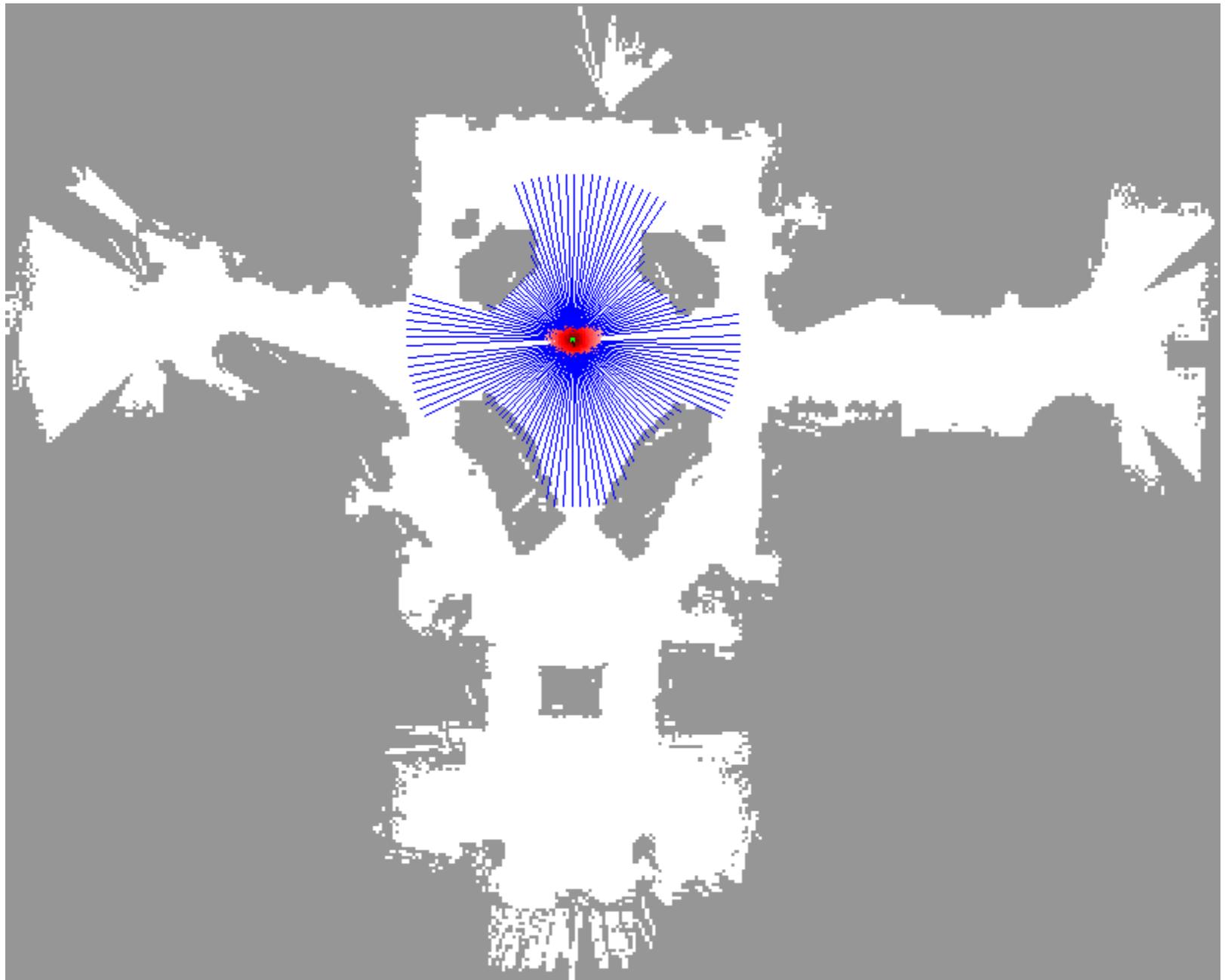


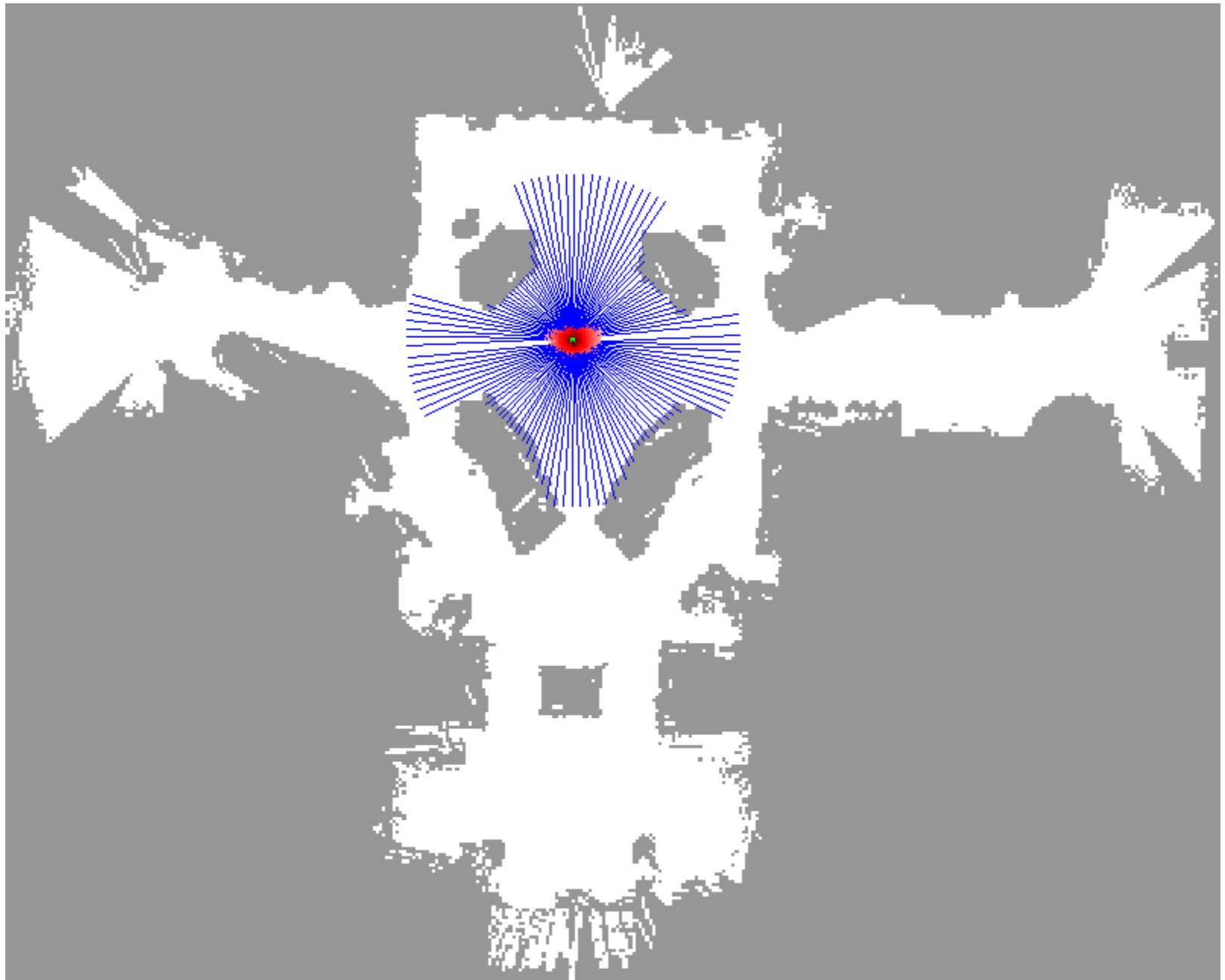












Recovery from Failure

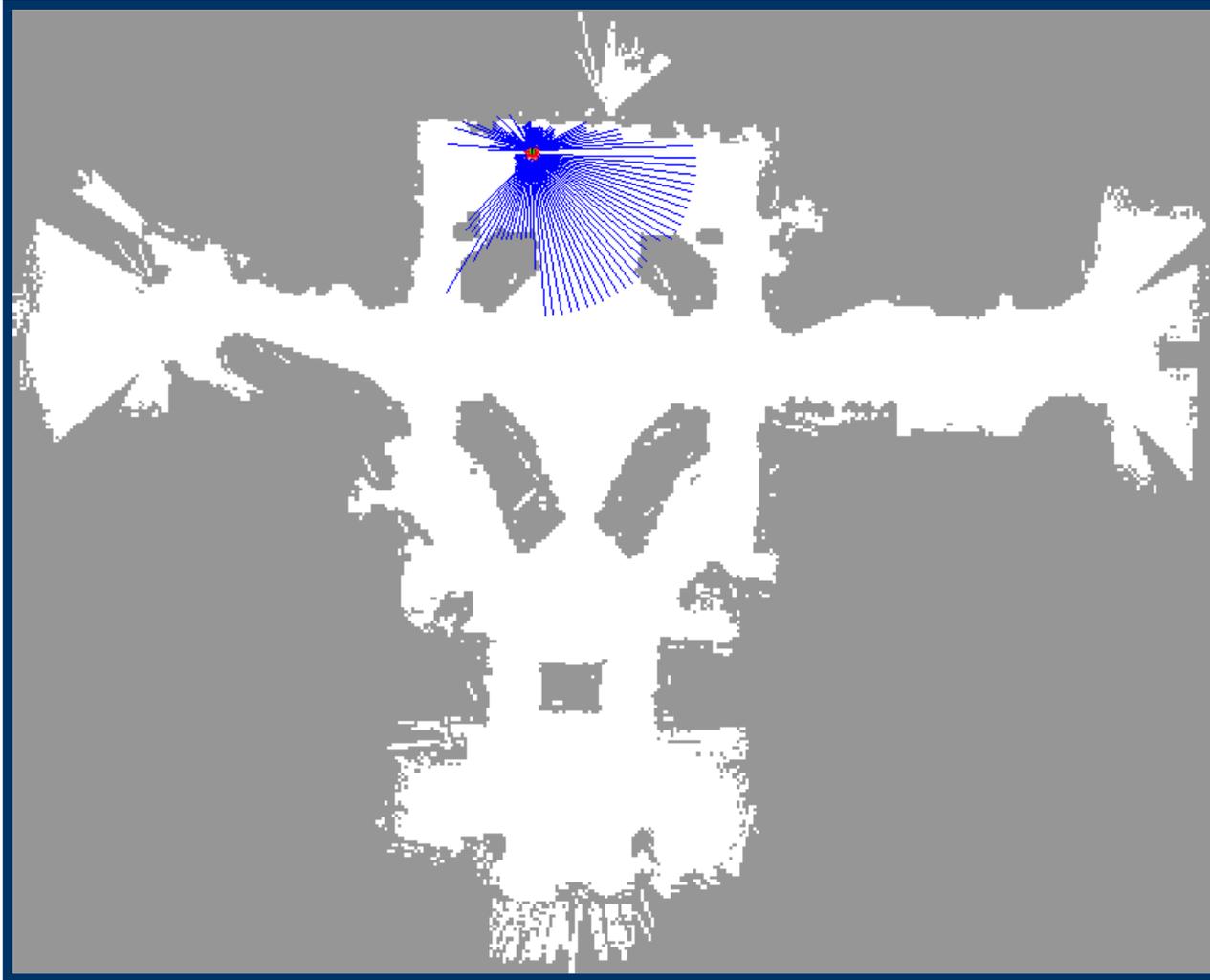
■ **Problem:**

- Samples are highly concentrated during tracking
- True location is not covered by samples if position gets lost

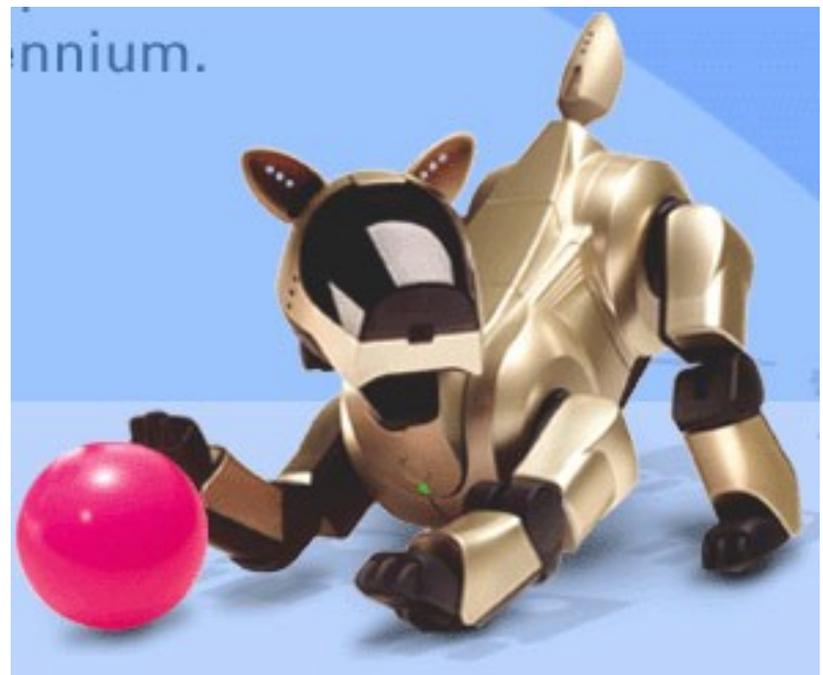
■ **Solutions:**

- Add uniformly distributed samples [Fox et al., 99]
- Draw samples according to observation density [Lenser et al., 00; Thrun et al., 00]

MCL: Recovery from Failure



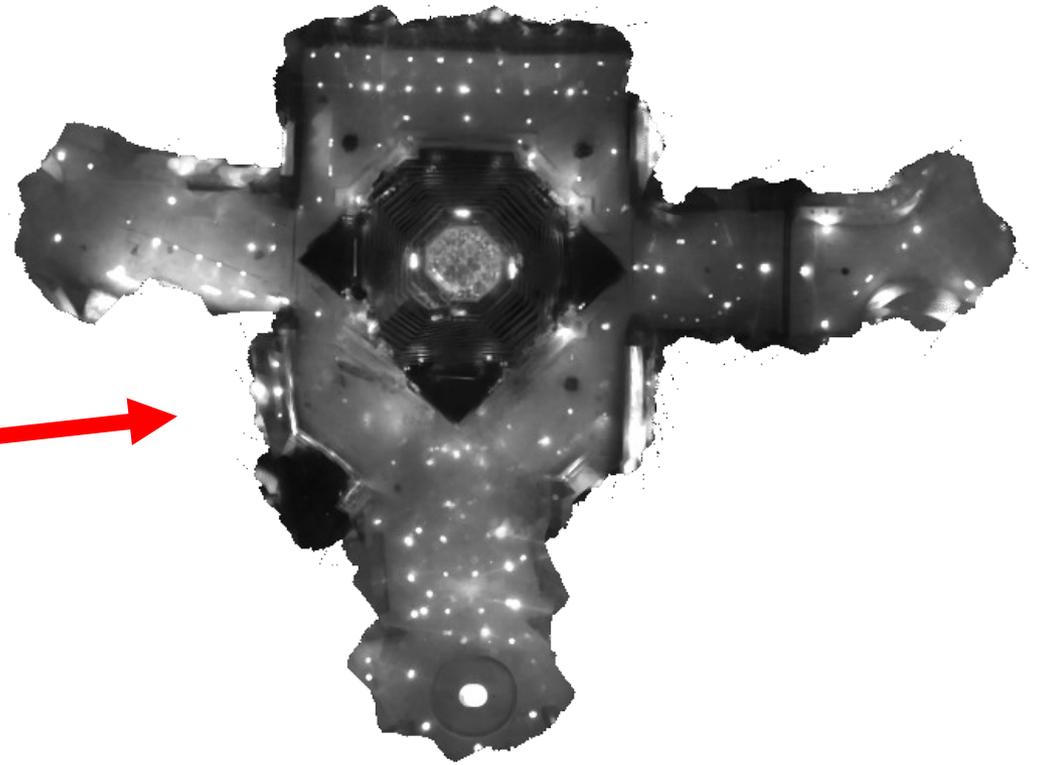
The RoboCup Challenge



- Dynamic, adversarial environments
- Limited computational power
- Multi-robot collaboration
- Particle filters allow efficient localization

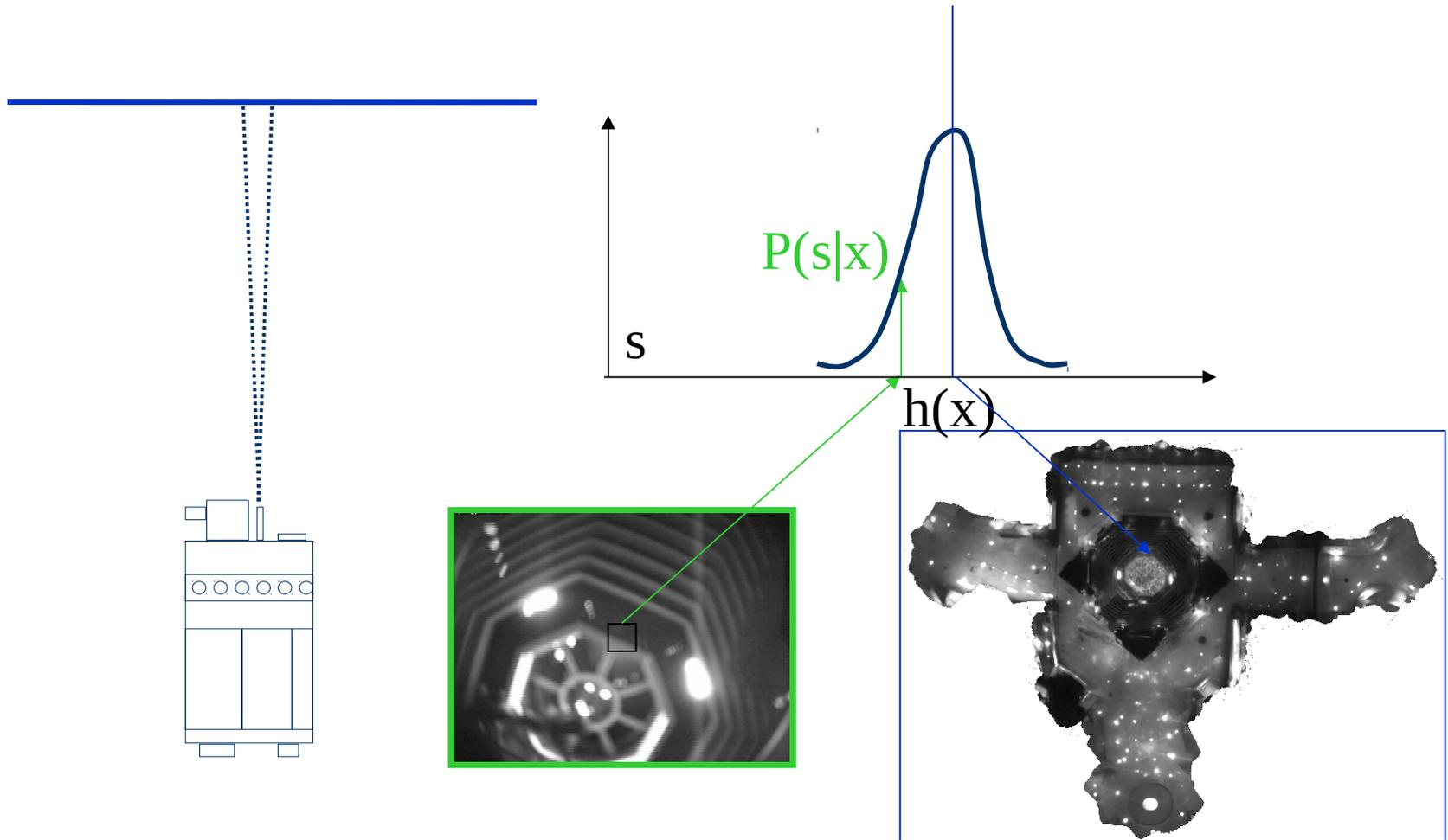
[Lenser et al. 00]

Using Ceiling Maps for Localization



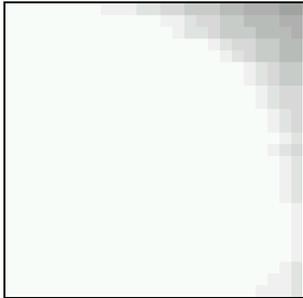
[Dellaert et al. 99]

Vision-based Localization

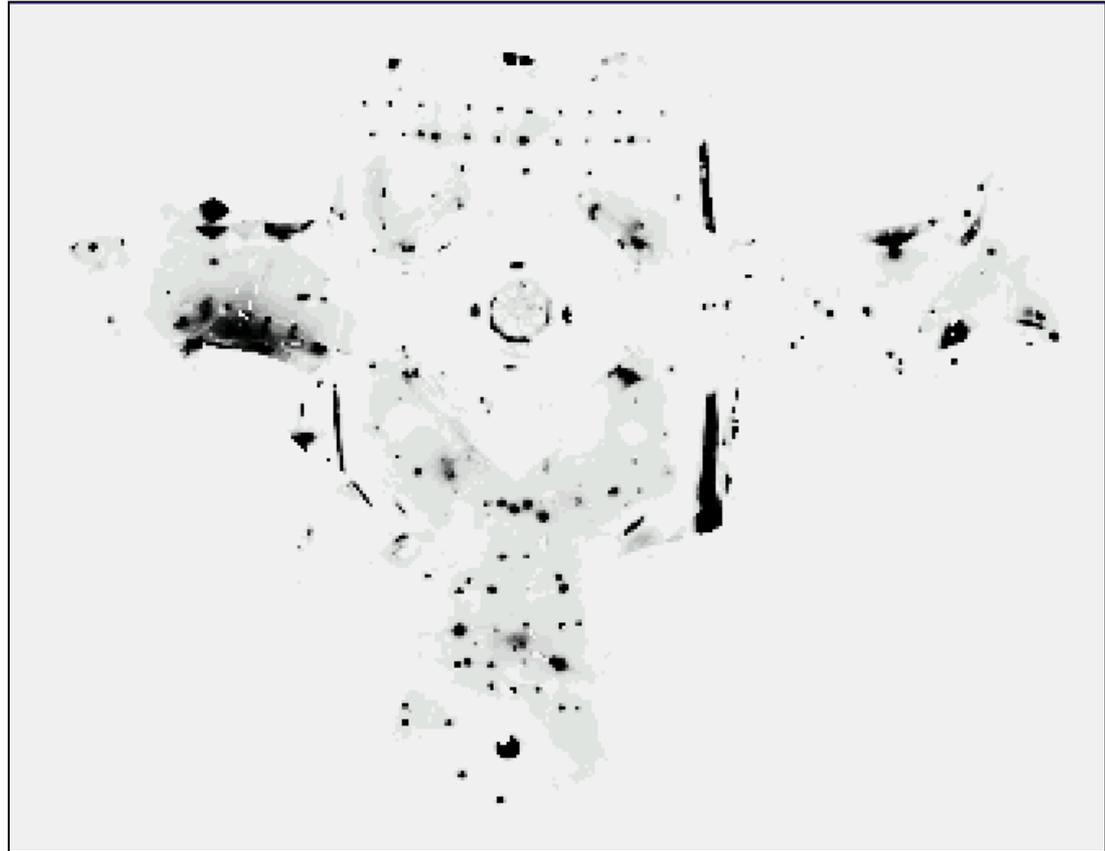


Under a Light

Measurement:

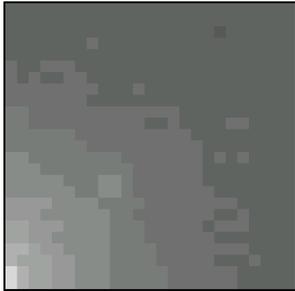


Resulting density:

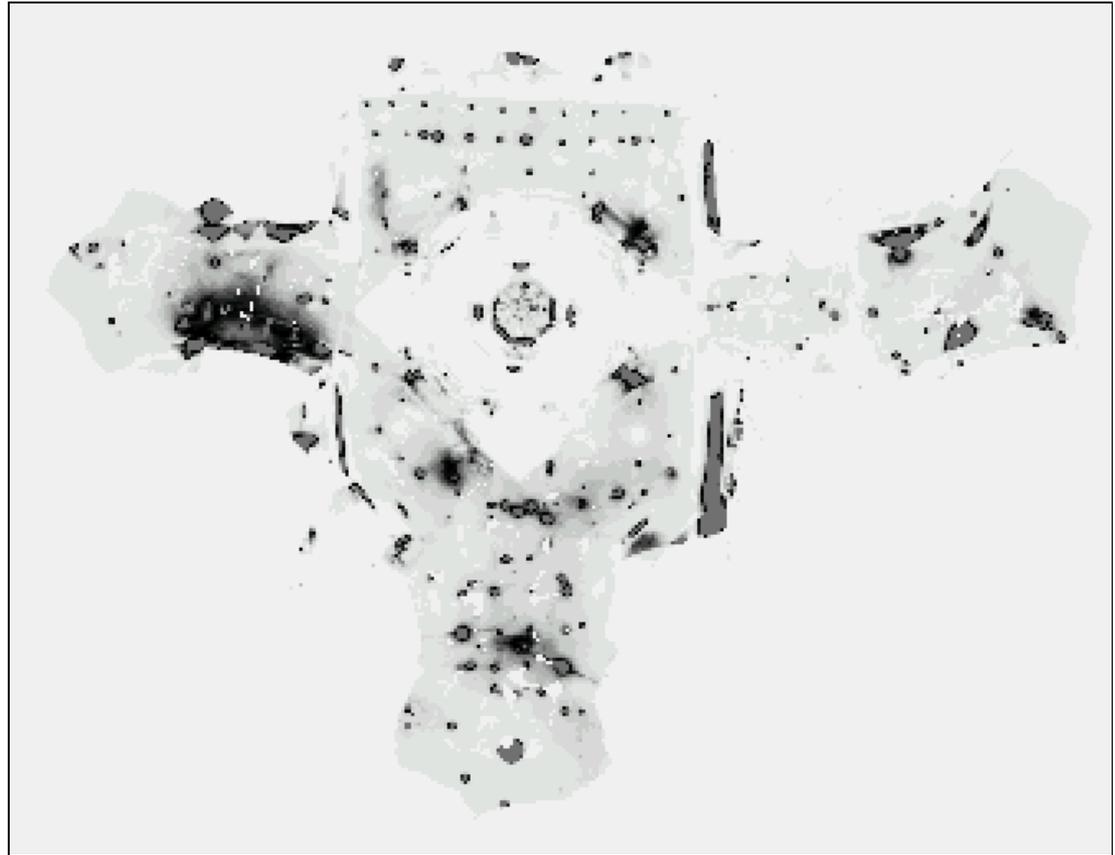


Next to a Light

Measurement:



Resulting density:

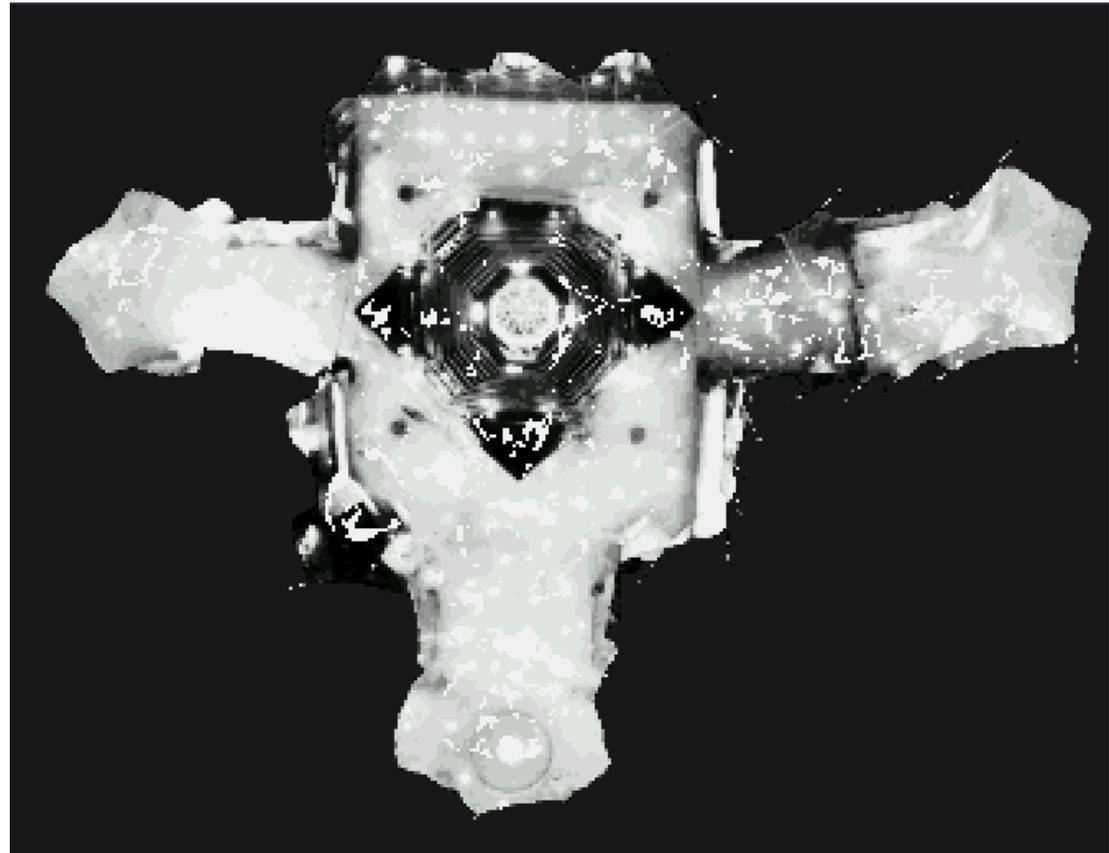


Elsewhere

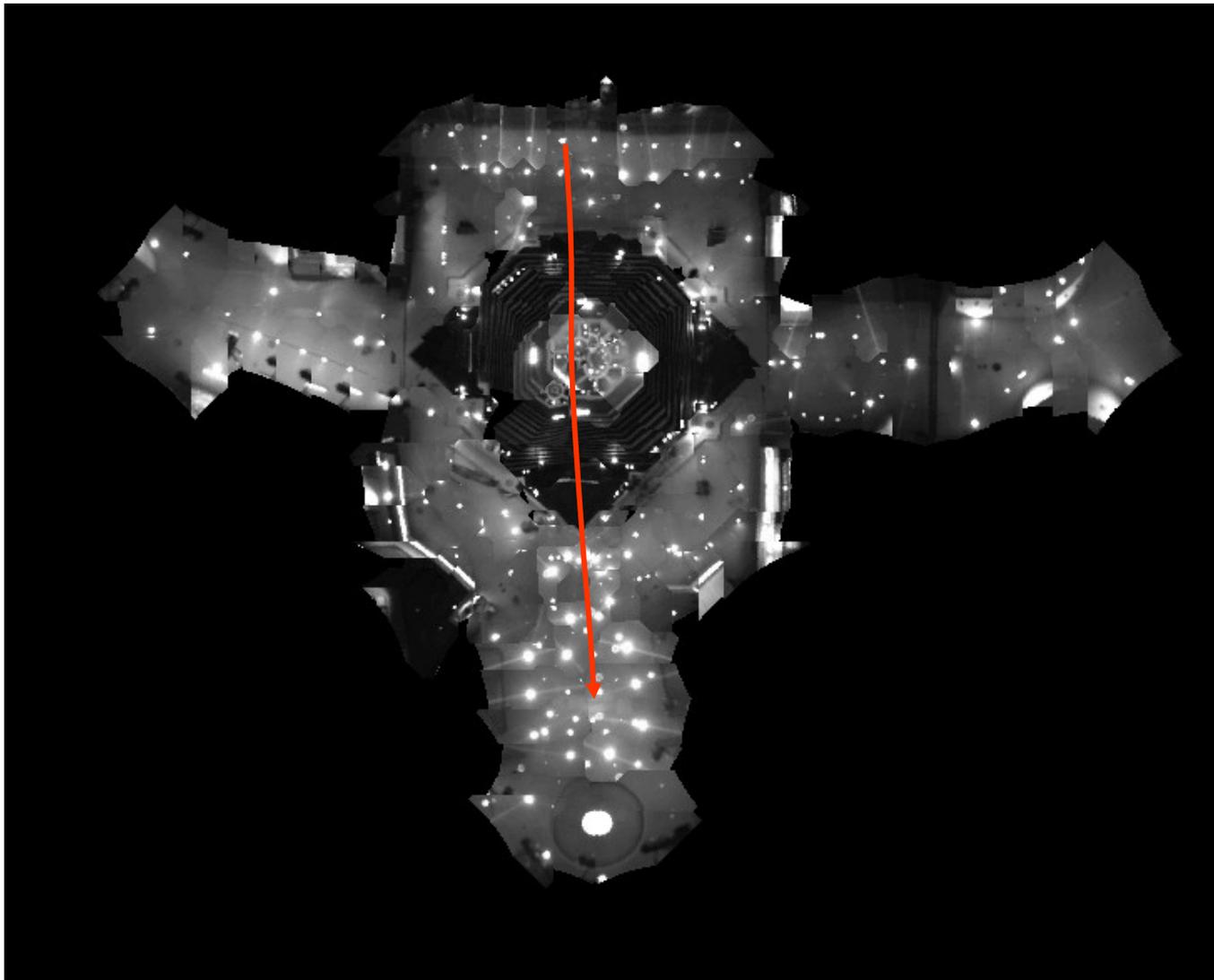
Measurement:



Resulting density:

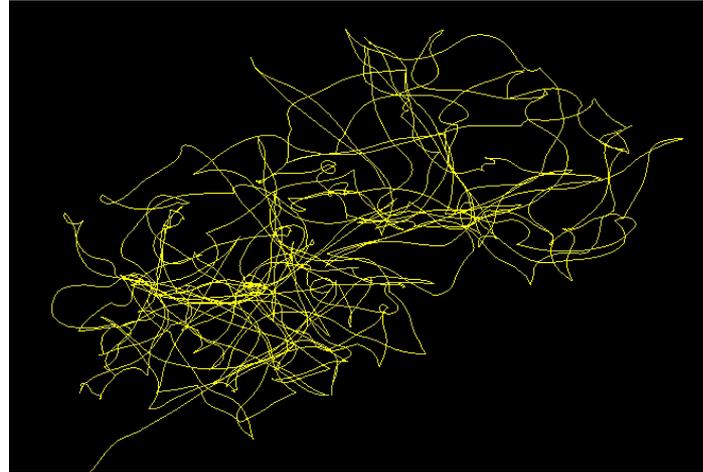


MCL: Global Localization Using Vision

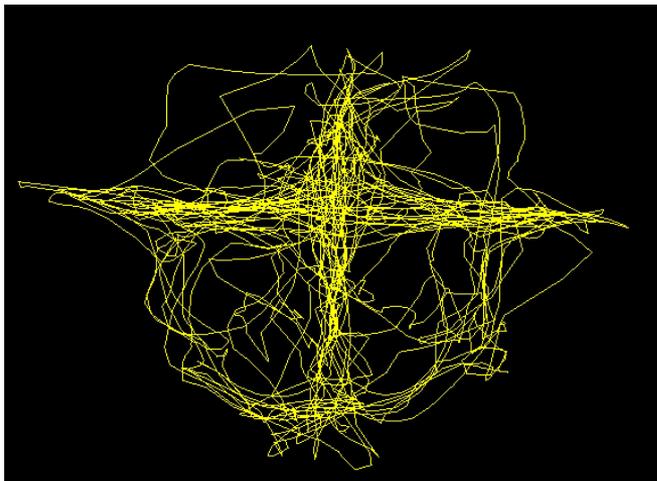


Vision-based Localization

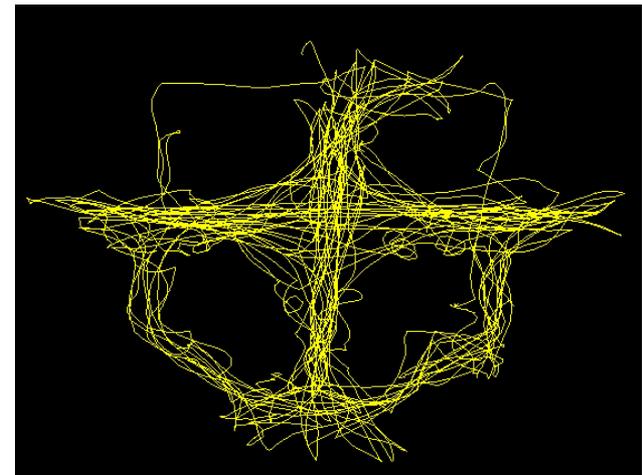
Odometry only:



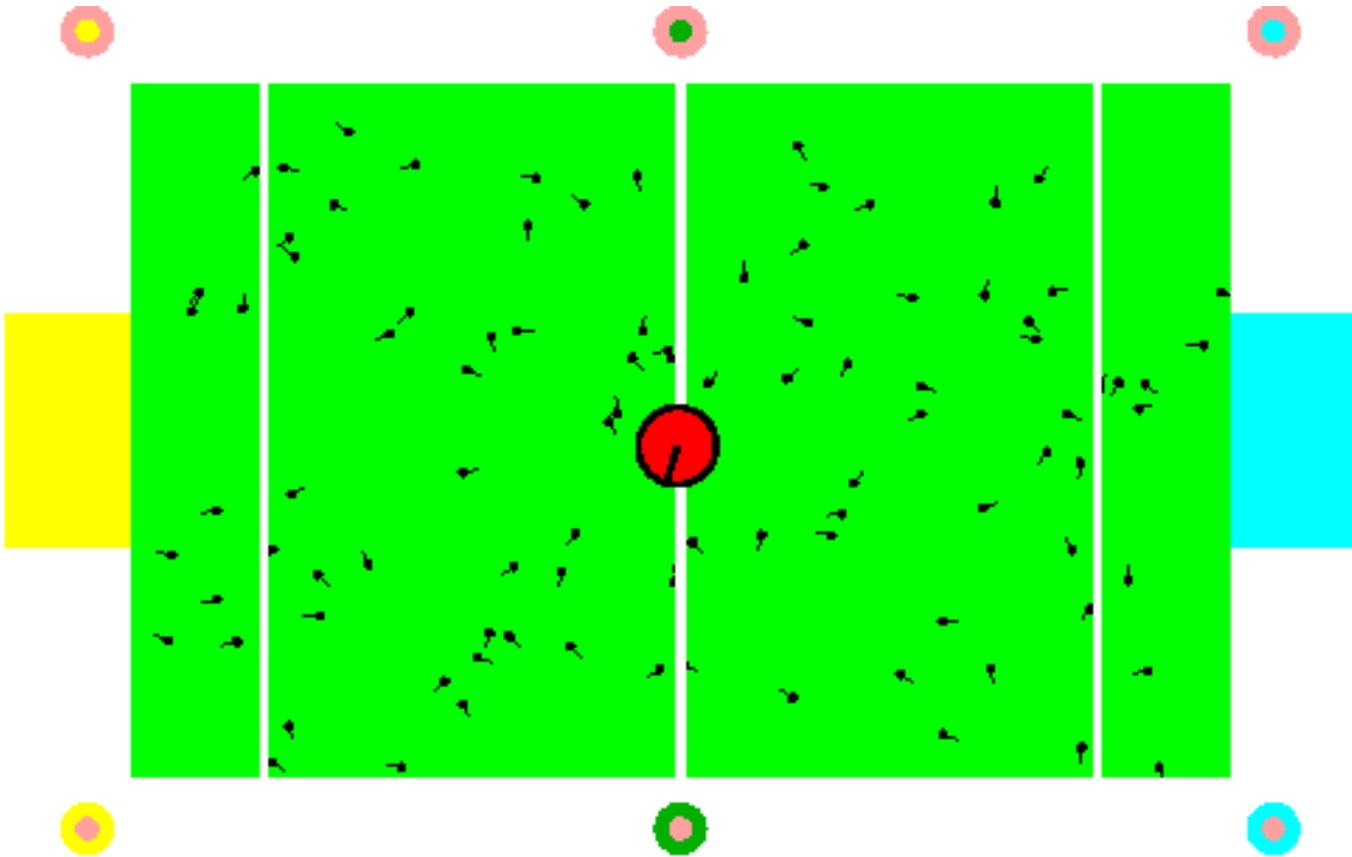
Vision:



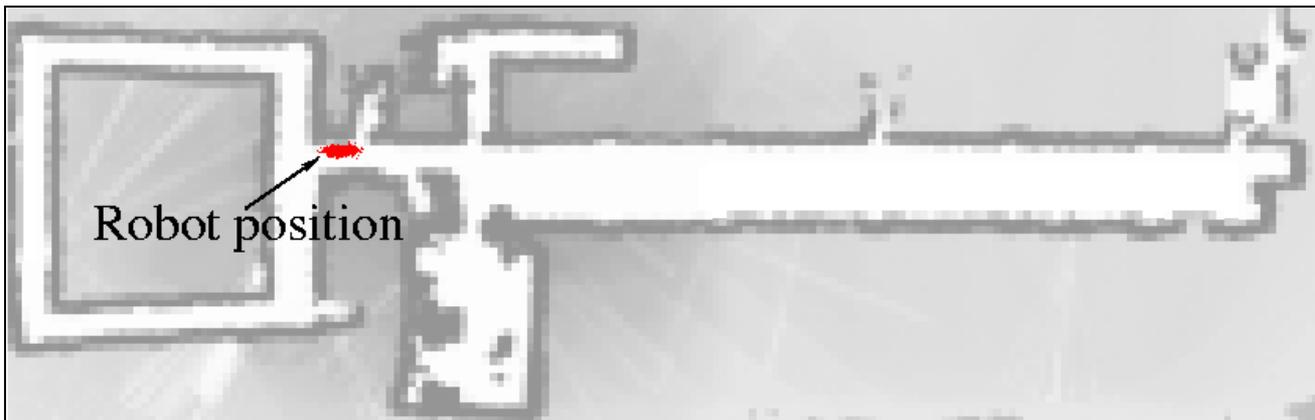
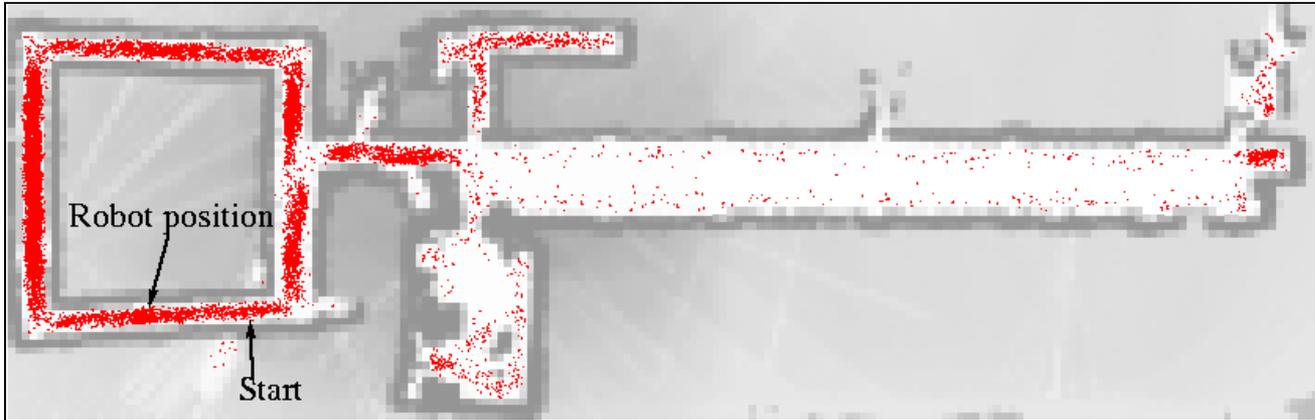
Laser:



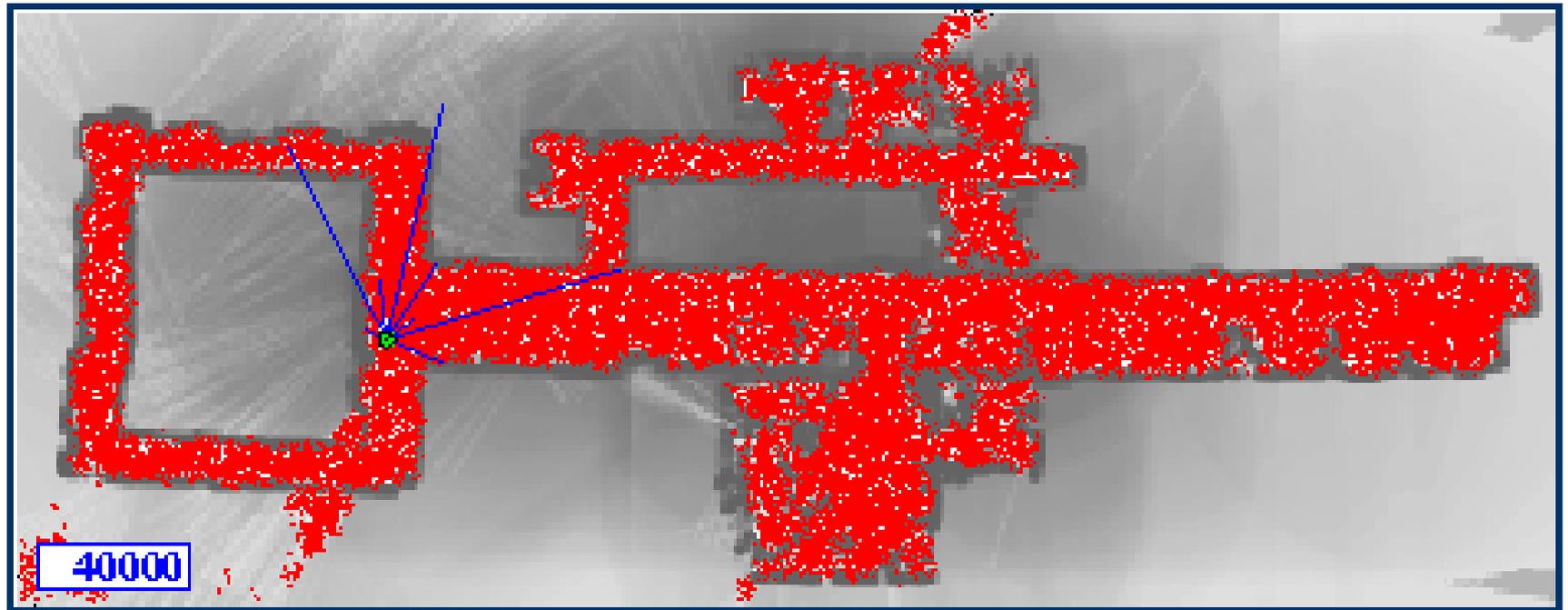
Localization for AIBO robots



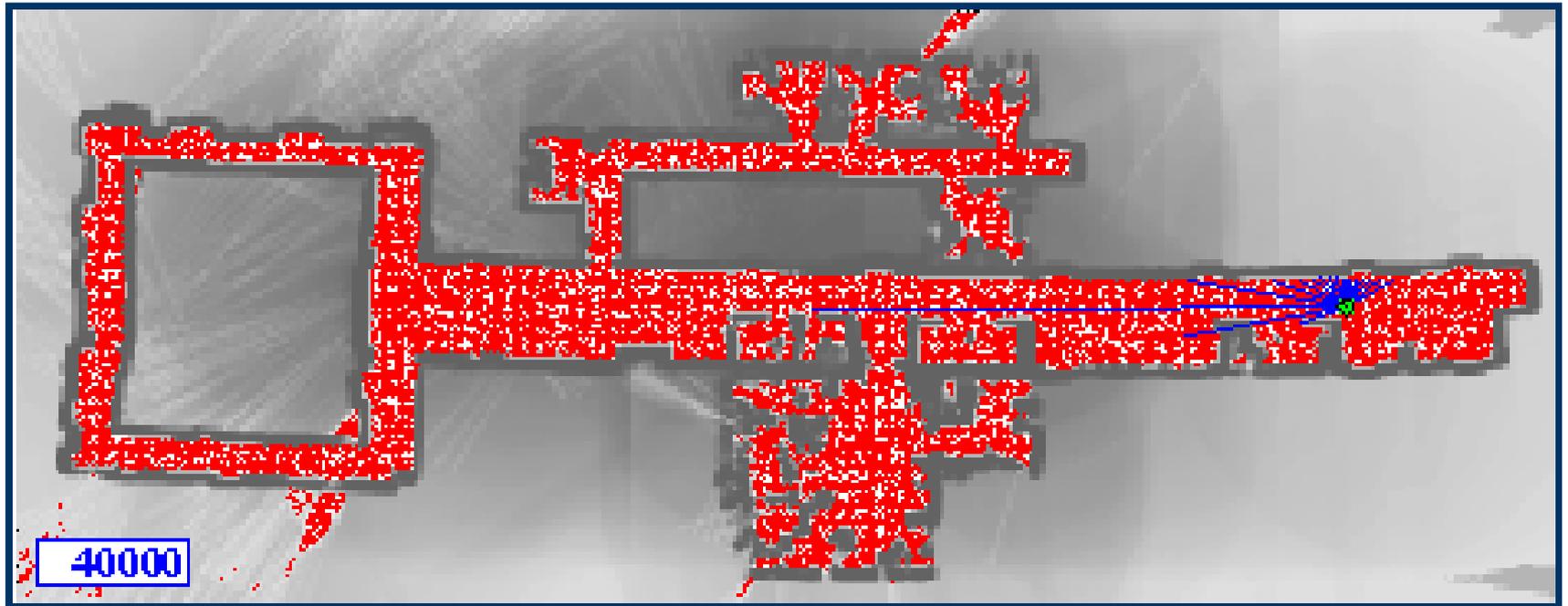
Adaptive Sampling



MCL: Adaptive Sampling (Sonar)

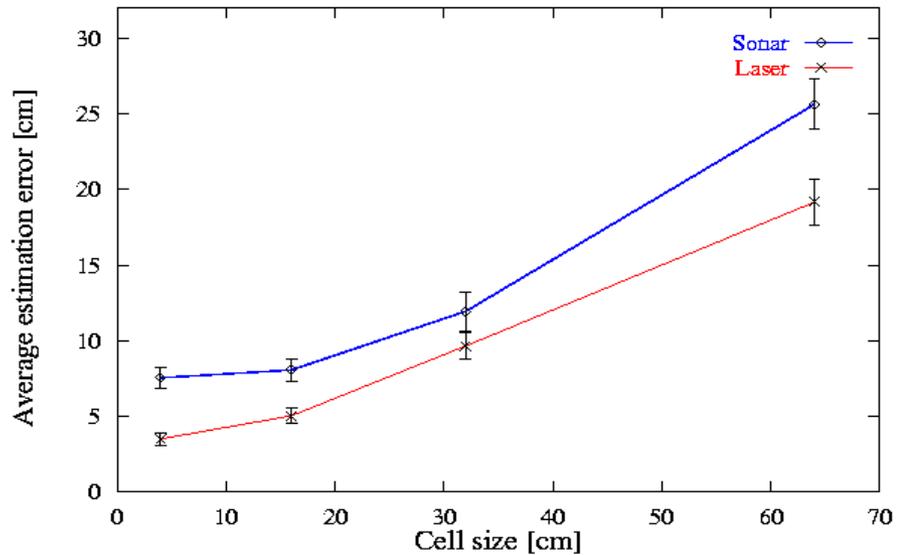


MCL: Adaptive Sampling (Laser)

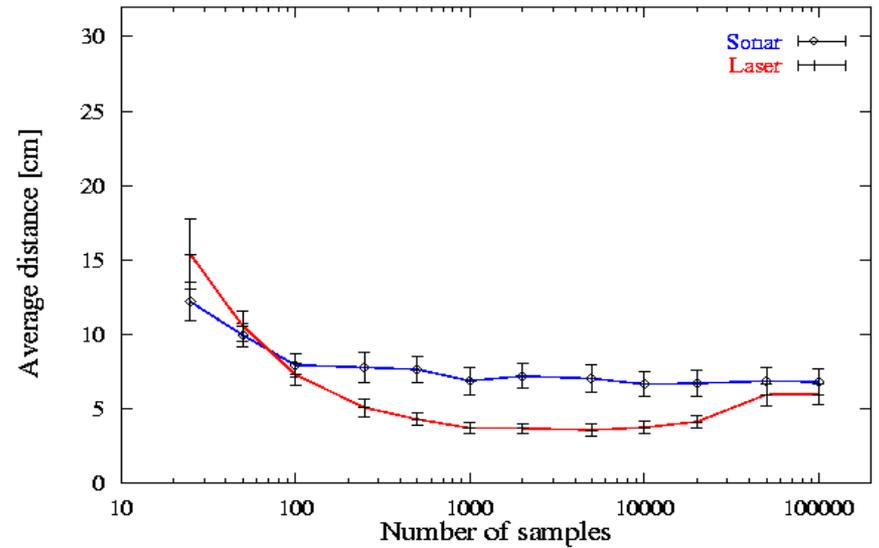


Performance Comparison

Grid-based localization



Monte Carlo localization



Particle Filters for Robot Localization (Summary)

- Approximate Bayes Estimation/Filtering
 - Full posterior estimation
 - Converges in $O(1/\sqrt{\#\text{samples}})$ [Tanner'93]
 - Robust: multiple hypotheses with degree of belief
 - Efficient in low-dimensional spaces: focuses computation where needed
 - Any-time: by varying number of samples
 - Easy to implement

Localization Algorithms - Comparison

	Kalman filter	Multi-hypot hesis tracking	Topological maps	Grid-based (fixed/variable)	Particle filter
Sensors	Gaussian		Features	Non-Gaussian	Non-Gaussian
Posterior	Gaussian	Multi-modal	Piecewise constant	Piecewise constant	Samples
Efficiency (memory)	++	++	++	-/+	+/>++
Efficiency (time)	++	++	++	o/+	+/>++
Implementation	+	o	+	+/o	++
Accuracy	++	++	-	+/>++	++
Robustness	-	+	+	++	+/>++
Global localization	No		Yes	Yes	Yes

Bayes Filtering: Lessons Learned

- General algorithm for recursively estimating the state of dynamic systems.
- Variants:
 - Hidden Markov Models
 - (Extended) Kalman Filters
 - Discrete Filters
 - Particle Filters