Recursive Bayes Filtering

CS689
Robot Motion Planning
Amarda Shehu

Notes modified from Wolfram Burgard, University of Freiburg

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

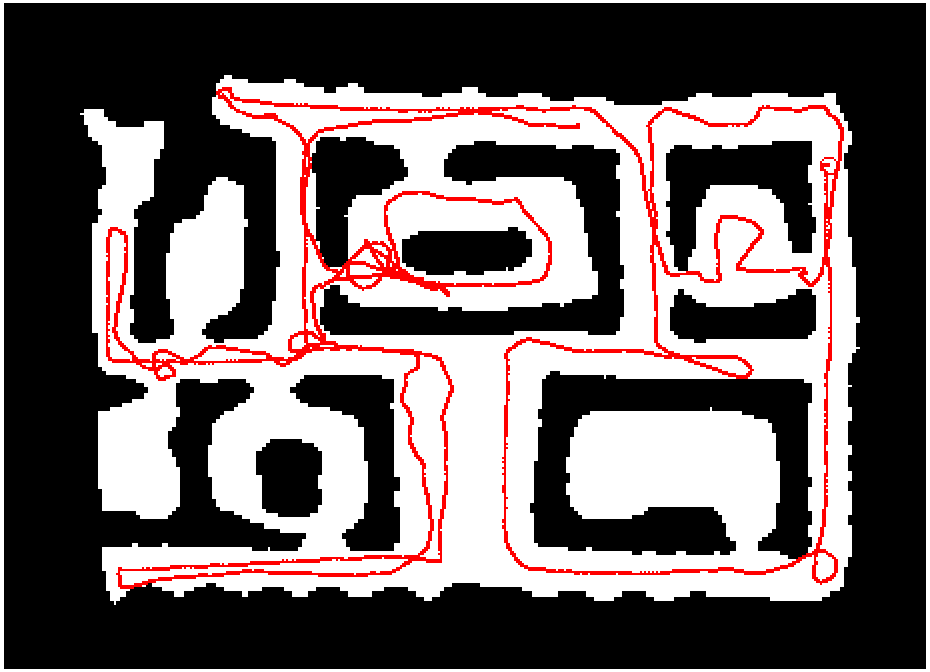
Rhino, 1997

Minerva, 1998

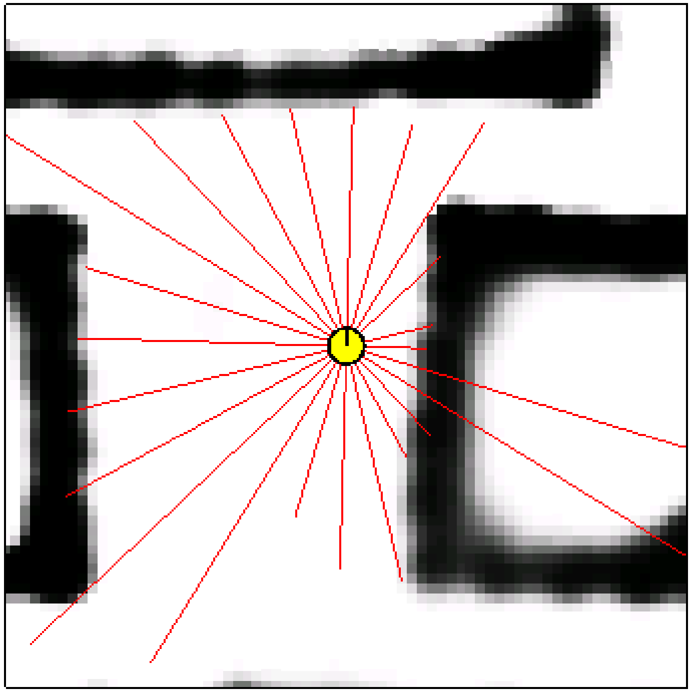
Technical Challenges

- 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

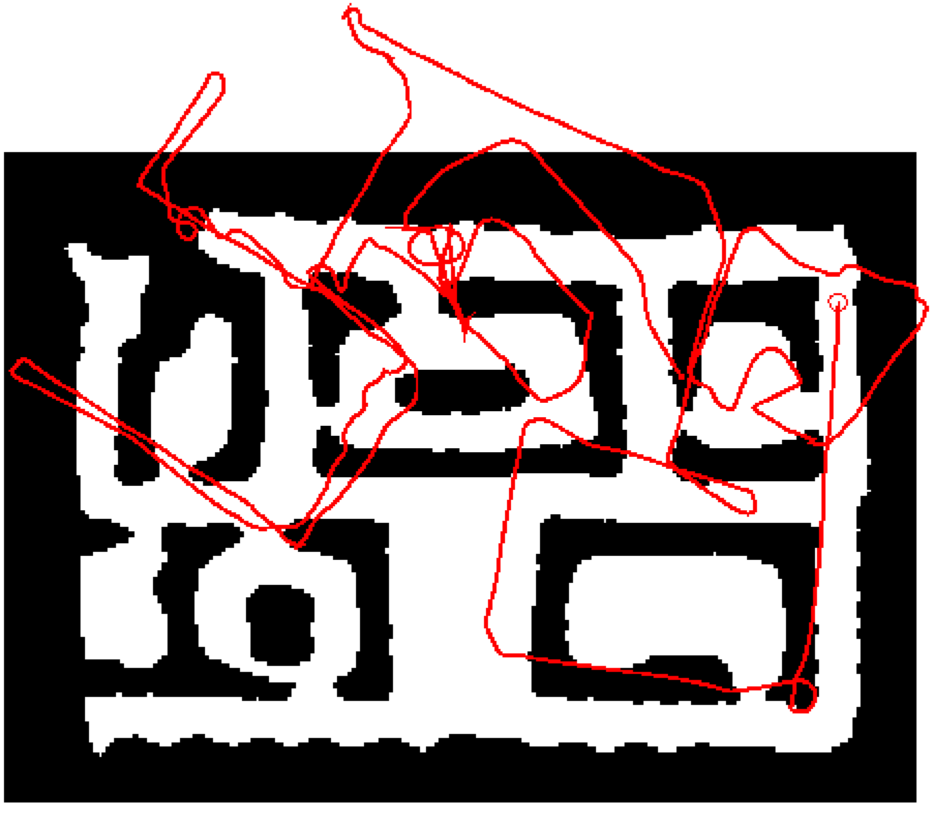


Odometry Data

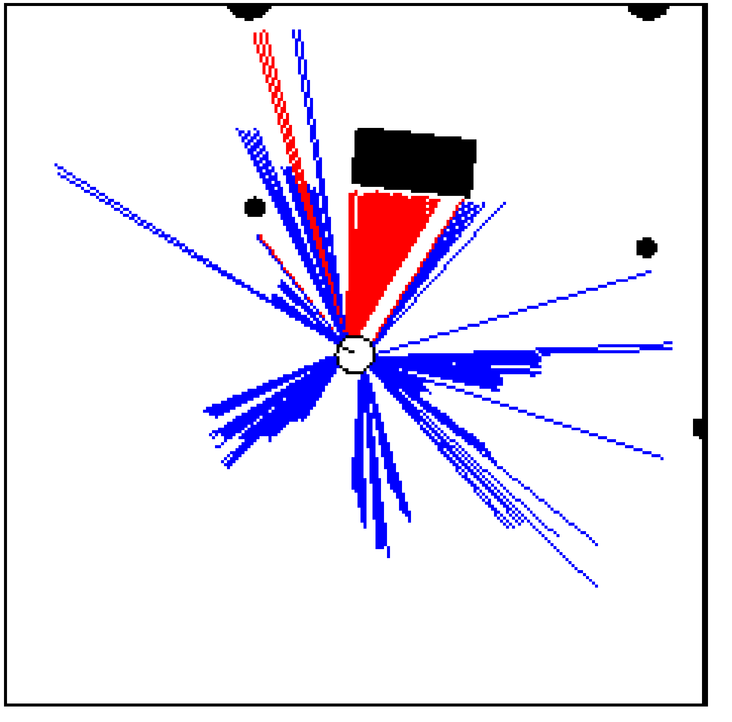


Range Data

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

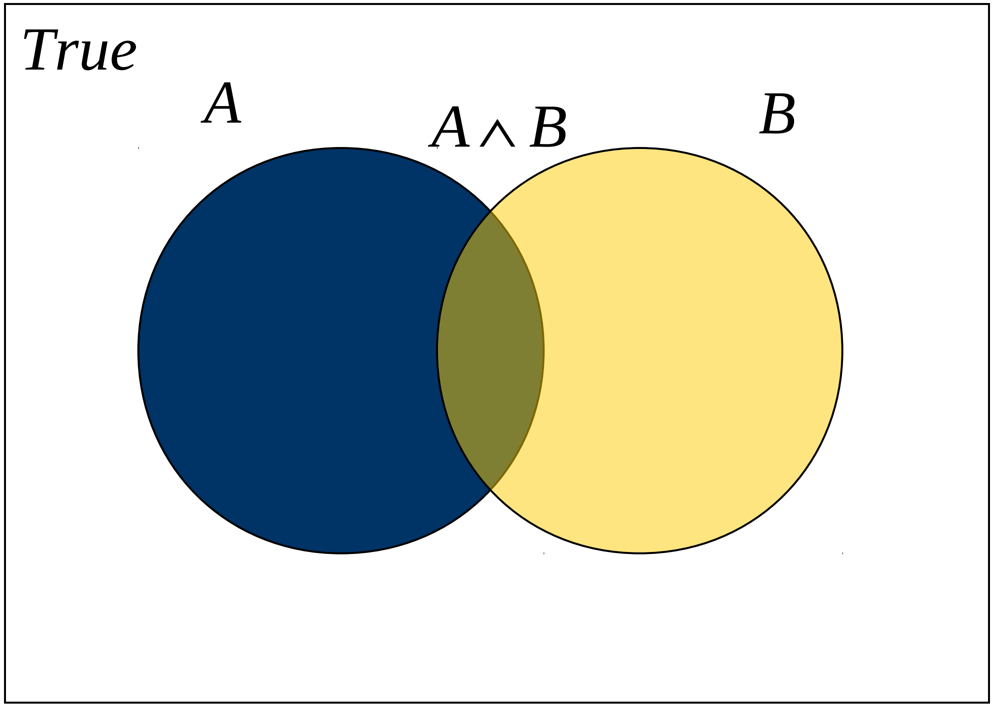
Axioms of Probability Theory

Pr*(A)* denotes probability that proposition *A* is true.

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

A Closer Look at Axiom 3

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



Using the Axioms

$$
\begin{aligned}
\Pr(A \vee \neg A) &= \Pr(A) + \Pr(\neg A) - \Pr(A \wedge \neg A) \\
\Pr(True) &= \Pr(A) + \Pr(\neg A) - \Pr(False) \\
1 &= \Pr(A) + \Pr(\neg A) - 0 \\
\Pr(\neg A) &= 1 - \Pr(A)
\end{aligned}
$$

Discrete Random Variables

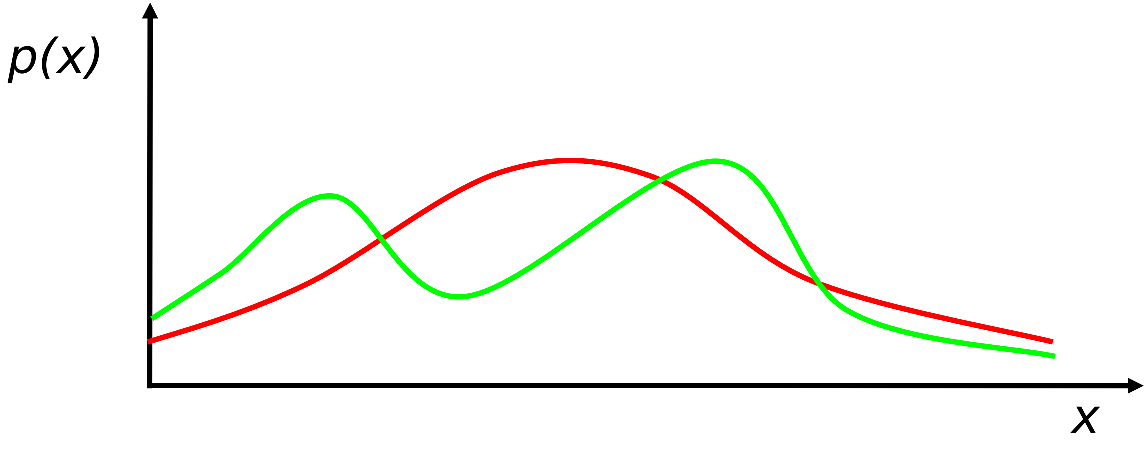
- X denotes a random variable.
- X can take on a finite number of values in $\{x_1, x_2, \ldots, x_n\}$.
- $P(X=x_i)$, or $P(x_i)$, is the probability that the random variable X takes on value x_i.

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

- $P(X{=}x$ and $Y{=}y) = P(x,y)$

- If X and Y are independent then

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

- $P(x \mid y)$ is the probability of x given y

  $$P(x \mid y) = P(x,y) / P(y)$$

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

- If X and Y are independent then

  $$P(x \mid y) = P(x)$$

Law of Total Probability, Marginals

Discrete case

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

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

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

Continuous case

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

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

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

Bayes Formula

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

$$\Rightarrow$$

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

Normalization

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

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

Algorithm:

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

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

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

Conditioning

- Total probability:

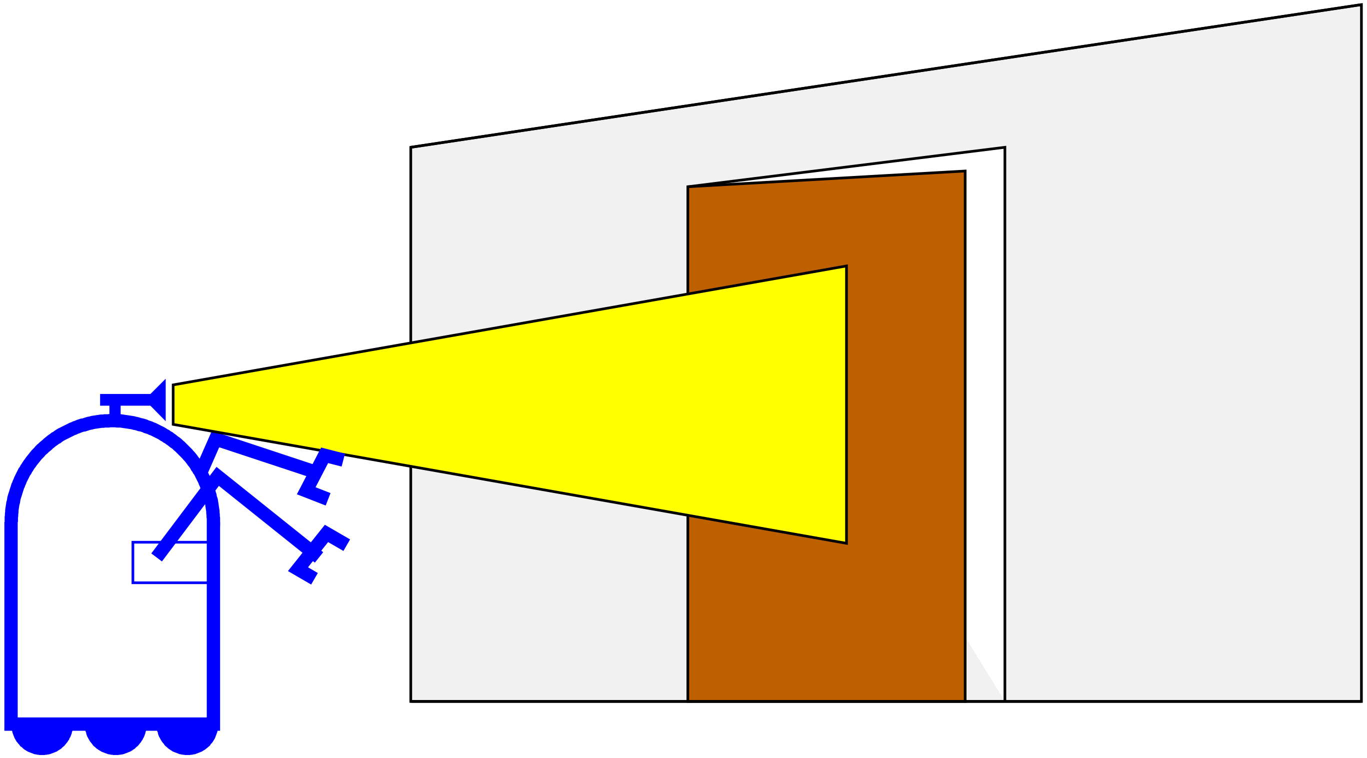
$$P(x|y) = \int P(x \mid y, z)\, P(z \mid y)\; dz$$

- Bayes rule and background knowledge:

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

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:

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

Causal vs. Diagnostic Reasoning

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

count frequencies!

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

Example

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

$$P(open \mid z) = ?$$

Example

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

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

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

- 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 $P(x| z_1...z_n)$?

Recursive Bayesian Updating

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

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.*

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

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

$P(open \mid z2, z1) = ?$

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** since
  - **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**?

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**.

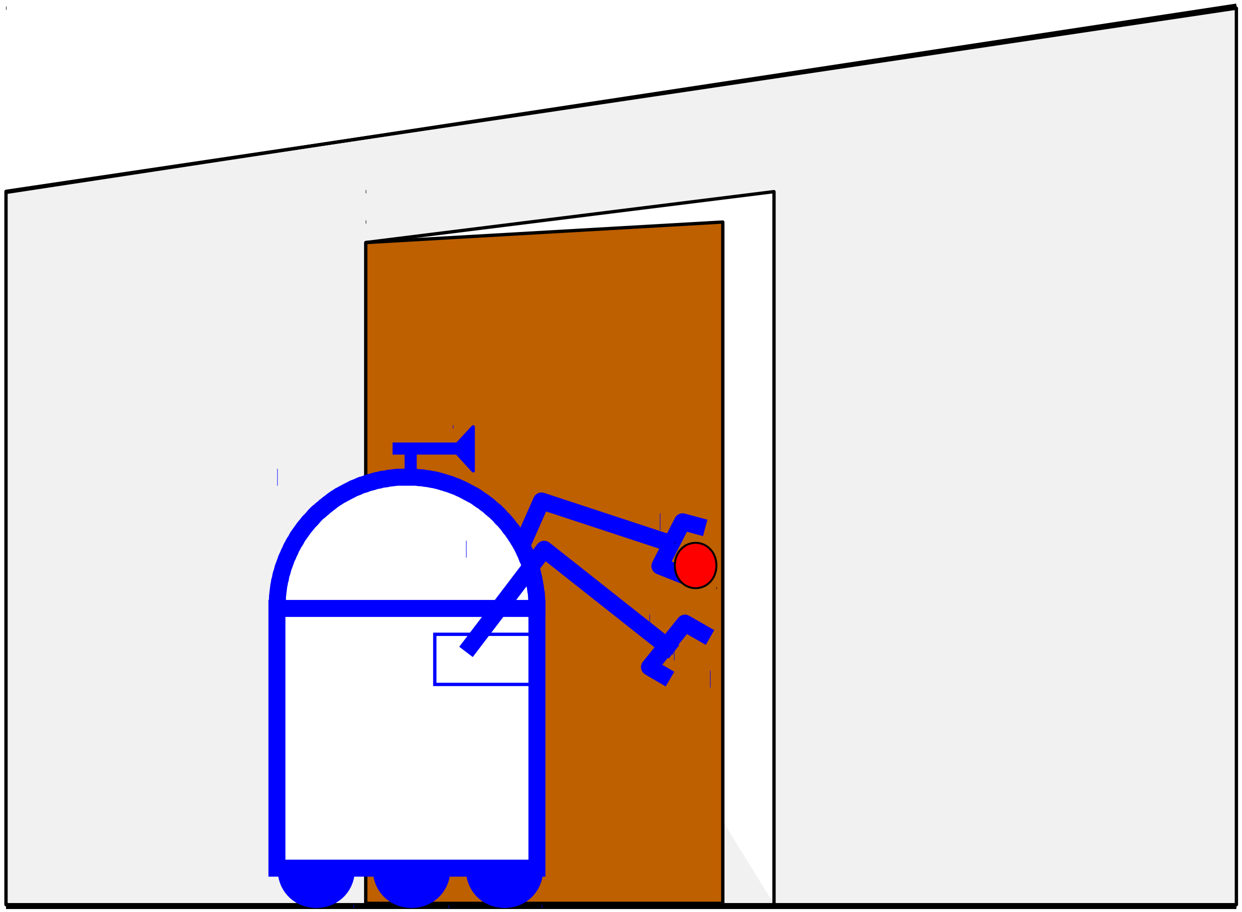
Modeling Actions

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

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

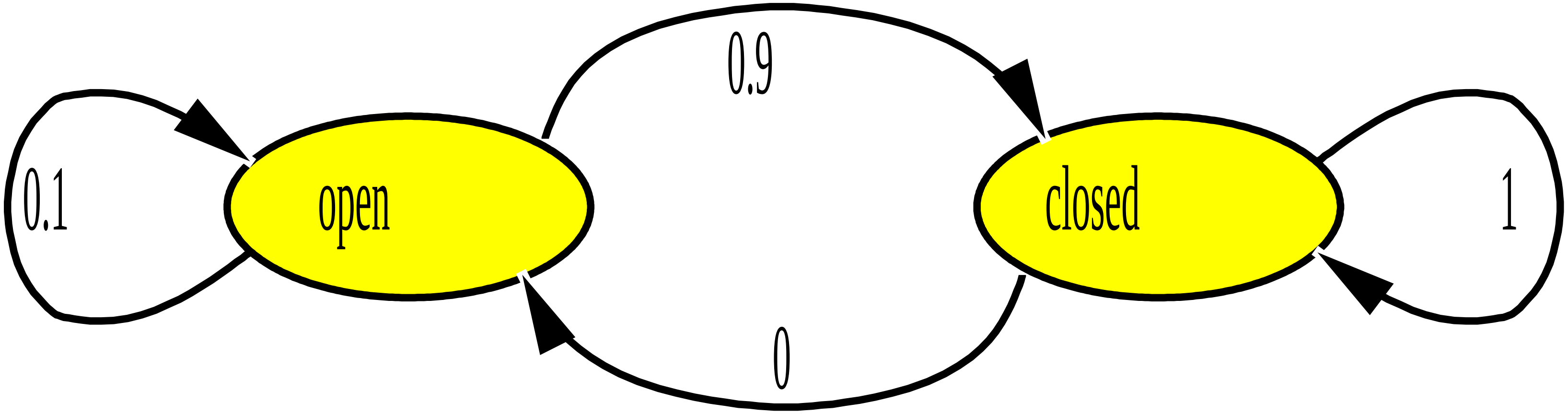
- This term specifies the pdf that **executing *u* changes the state from *x’ to x***.

Example: Closing the door



State Transitions

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

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

$$= P(closed \mid u, open)P(open)$$

$$+ P(closed \mid u, closed)P(closed)$$

$$= \frac{9}{10} * \frac{5}{8} + \frac{1}{1} * \frac{3}{8} = \frac{15}{16}$$

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

$$= P(open \mid u, open)P(open)$$

$$+ P(open \mid u, closed)P(closed)$$

$$= \frac{1}{10} * \frac{5}{8} + \frac{0}{1} * \frac{3}{8} = \frac{1}{16}$$

$$= 1 - P(closed \mid u)$$

Bayes Filters: Framework

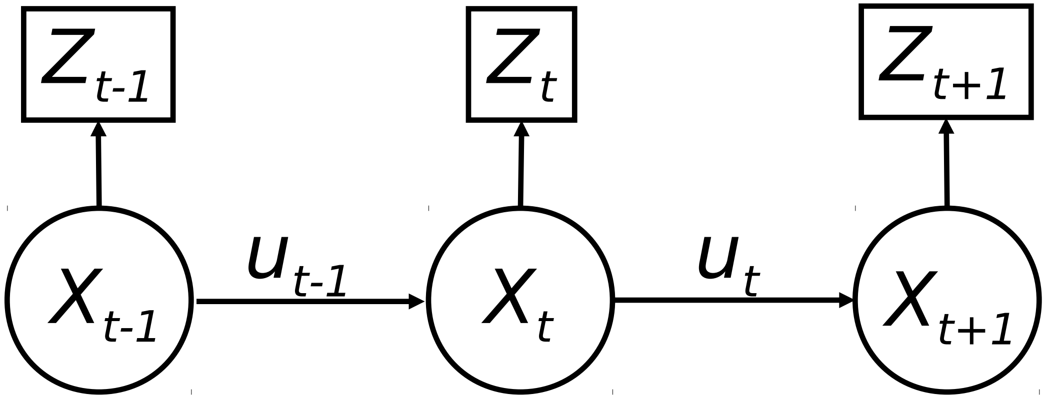
- **Given:**
  - Stream of observations *z* and action data *u:*

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

  - 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**:

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

Markov Assumption



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

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

$$p\left(x_t \middle| u_{t-1}, x_{t-1}, d_{t-2}, \ldots, d_0\right) = p\left(x_t \middle| u_{t-1}, x_{t-1}\right)$$

Underlying Assumptions

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

z = observation
u = action
x = state

Bayes Filters

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

Bayes Filters

z = observation
u = action
x = state

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

Bayes

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

Bayes Filters

z = observation
u = action
x = state

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

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

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

Bayes Filters

z = observation
u = action
x = state

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

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

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

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

Bayes Filters

z = observation
u = action
x = state

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

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

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

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

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

Bayes Filters

z = observation
u = action
x = state

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

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

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

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

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

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

Bayes Filters

z = observation
u = action
x = state

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

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

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

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

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

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

Bayes Filter Algorithm

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

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

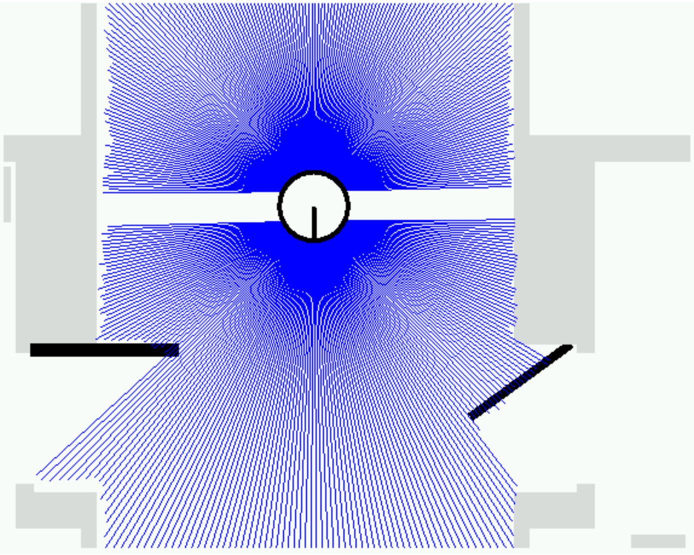
Bayes Filters are Familiar!

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

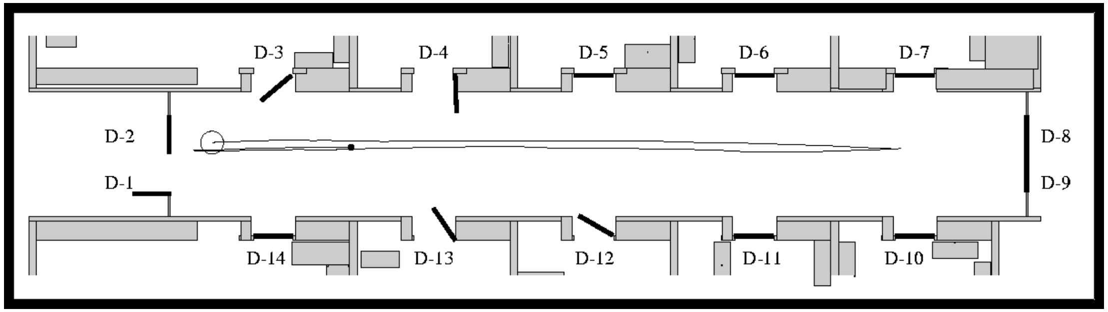
- Kalman filters
- Particle filters
- Hidden Markov models
- Dynamic Bayes networks
- Partially Observable Markov Decision Processes (POMDPs)

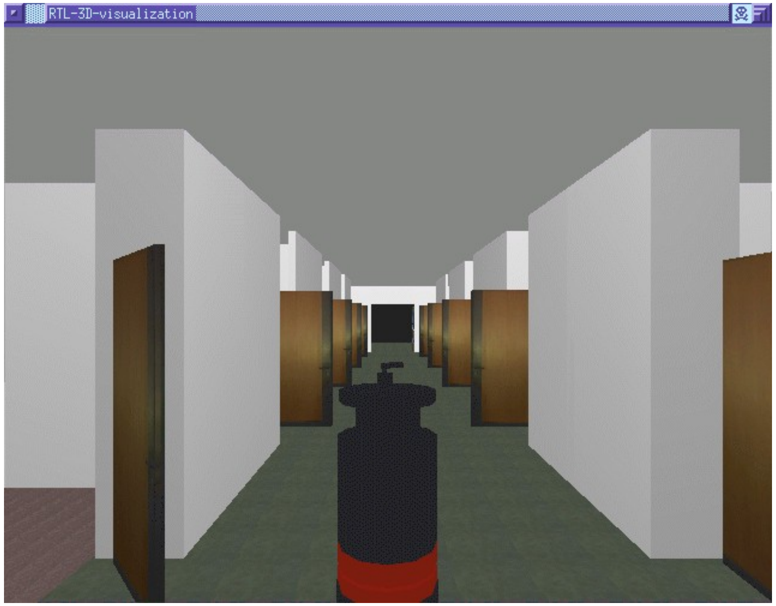
Application to Door State Estimation

- 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


D-3
D-4
D-5
D-6
D-7
D-2
D-8
D-1
D-9
D-14
D-13
D-12
D-11
D-10


RTL-3D-visualization


RTL-3D-visualization

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

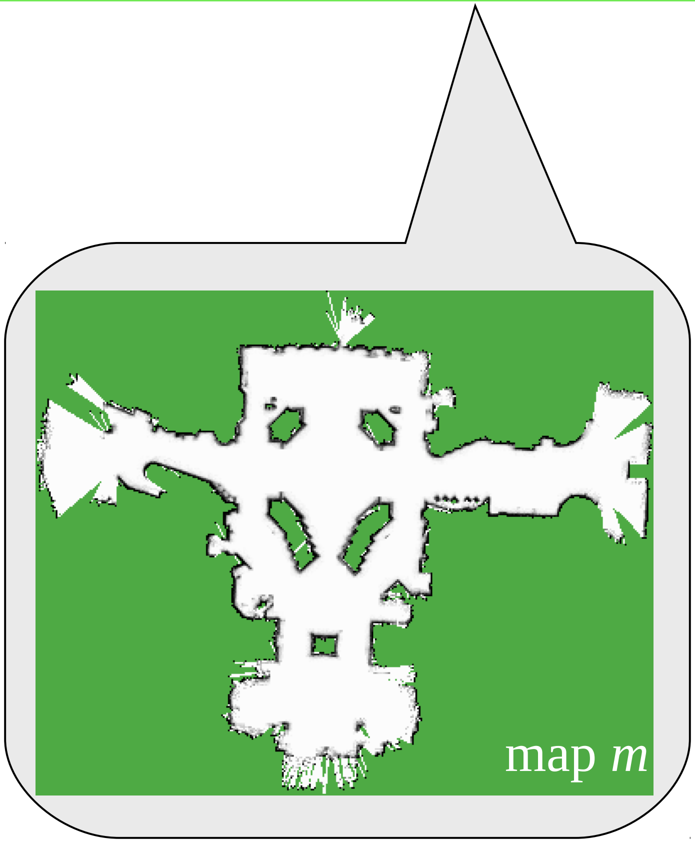
$$bel(x_t) = \eta \ p(s_t \mid x_t) \int p(x_t \mid x_{t-1}, a_{t-1}) \, bel(x_{t-1}) \, dx_{t-1}$$

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

Localization with Bayes Filters

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

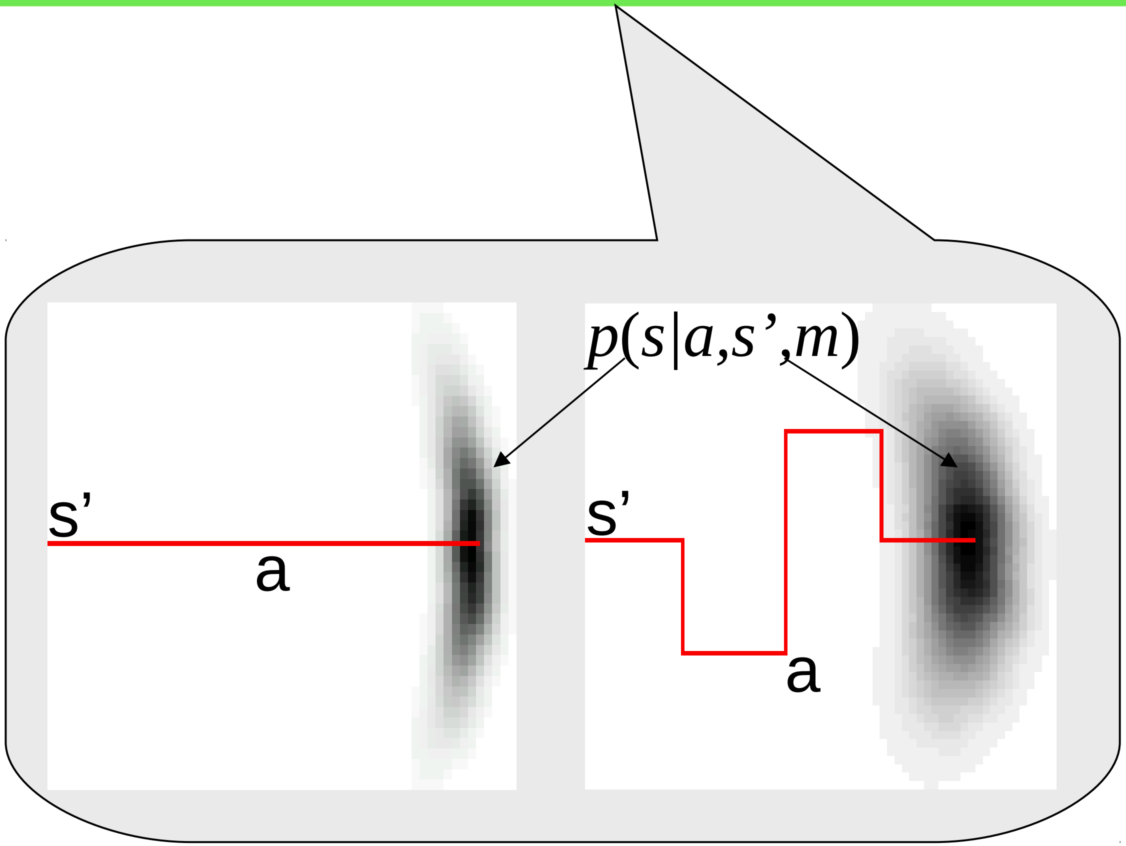
$$\Rightarrow \; bel(x_t \mid m) = \eta \; p(s_t \mid x_t, m) \int p(x_t \mid x_{t-1}, a_{t-1}, m) \, bel(x_{t-1} \mid m) \, dx_{t-1}$$



Localization with Bayes Filters

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

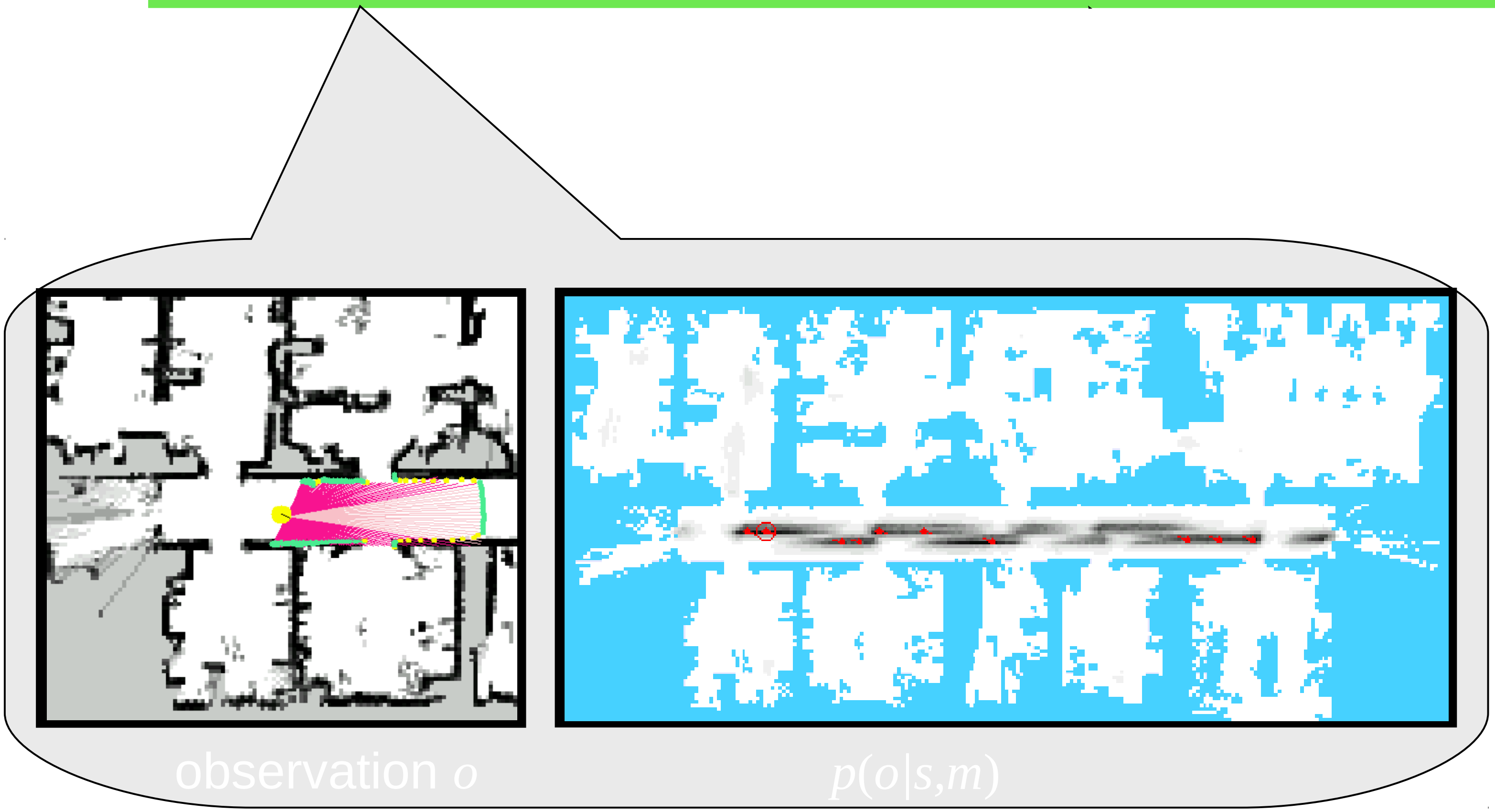
$$\Rightarrow \; bel(x_t \mid m) = \eta \; p(s_t \mid x_t, m) \int p(x_t \mid x_{t-1}, a_{t-1}, m) \, bel(x_{t-1} \mid m) \, dx_{t-1}$$



Localization with Bayes Filters

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

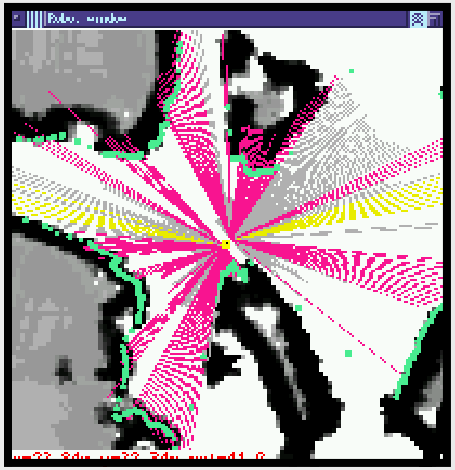
$$\Rightarrow \; bel(x_t \mid m) = \eta \; p(s_t \mid x_t, m) \int p(x_t \mid x_{t-1}, a_{t-1}, m) \, bel(x_{t-1} \mid m) \, dx_{t-1}$$



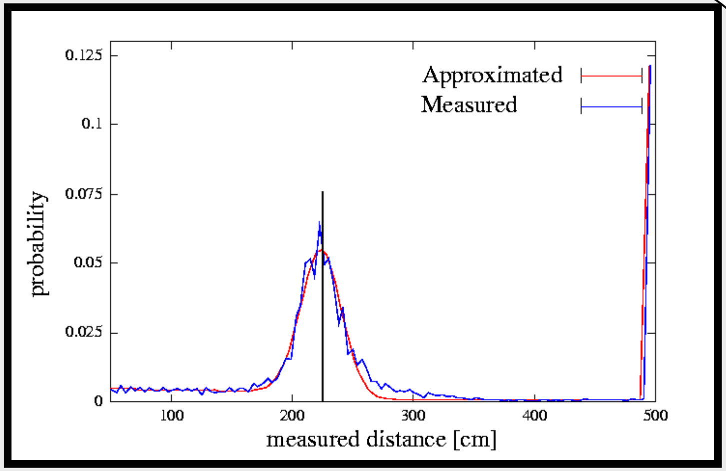
Localization with Bayes Filters

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

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



laser data



$p(o \mid s, m)$

Localization with Bayes Filters

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

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

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

Gaussians

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

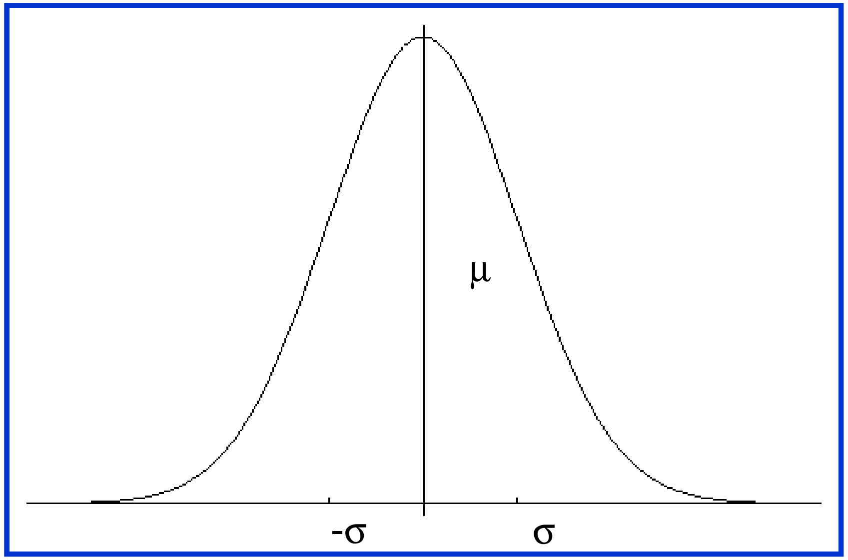
$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

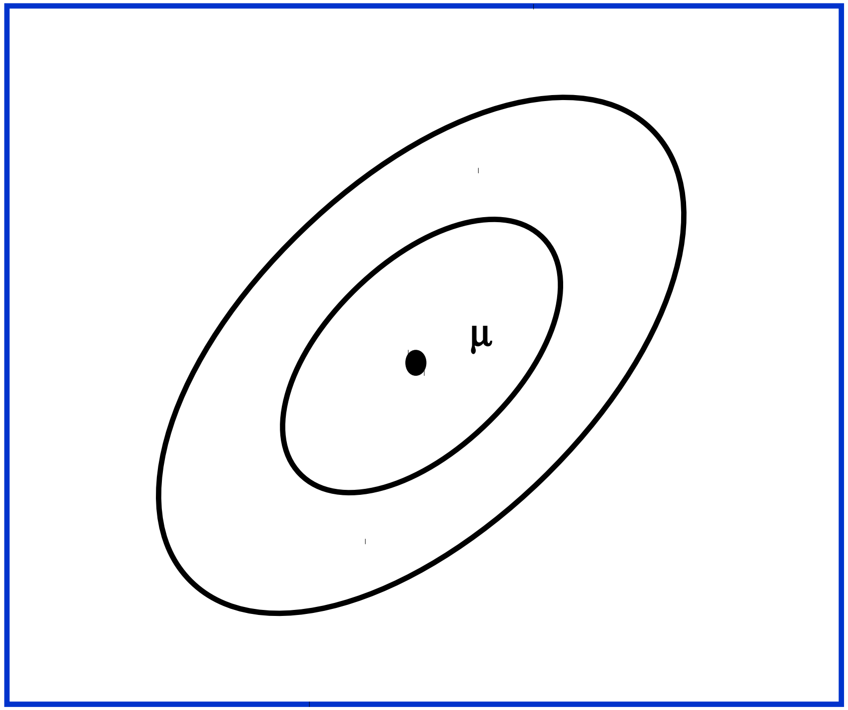
Univariate

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

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

Multivariate





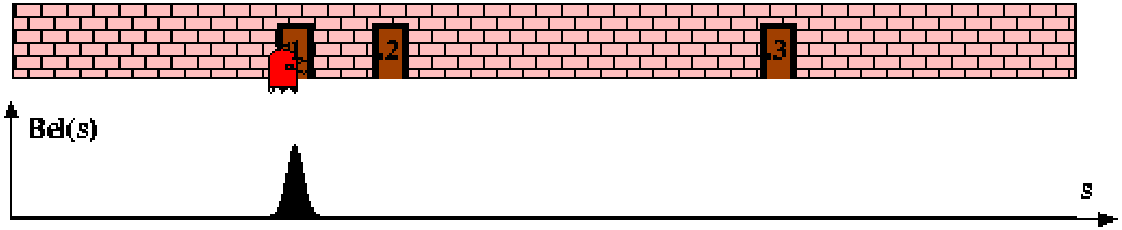
Kalman Filters

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

$$x_{t+1} = Ax_t + Bu_t + v_t$$
$$z_t = Cx_t + w_t$$

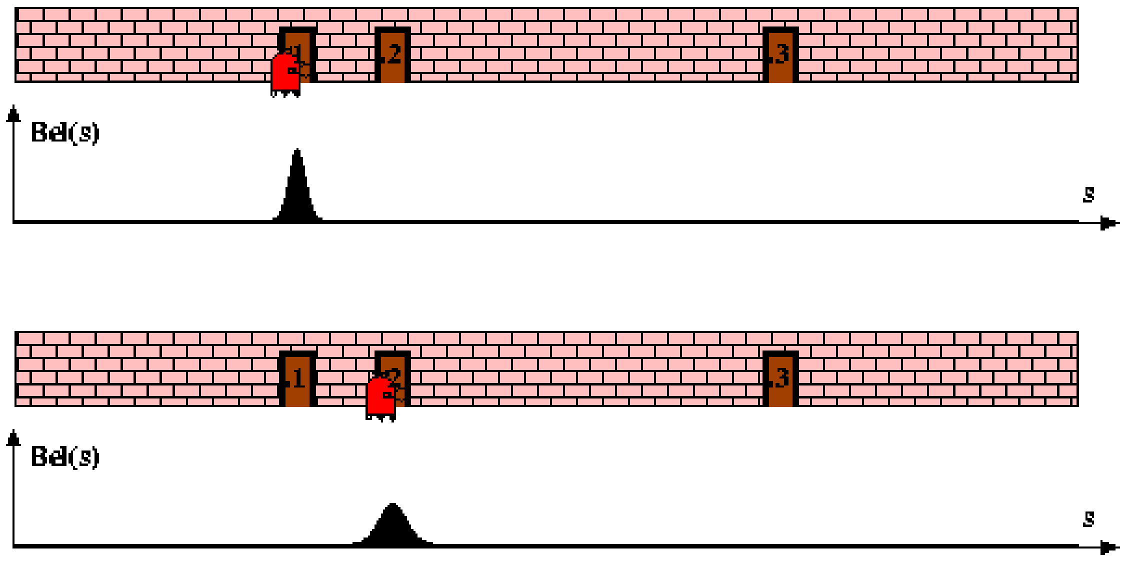
The random variables v_t and w_t represent the process measurement noise and are assumed to be independent, white and with normal probability distributions.

Kalman Filters



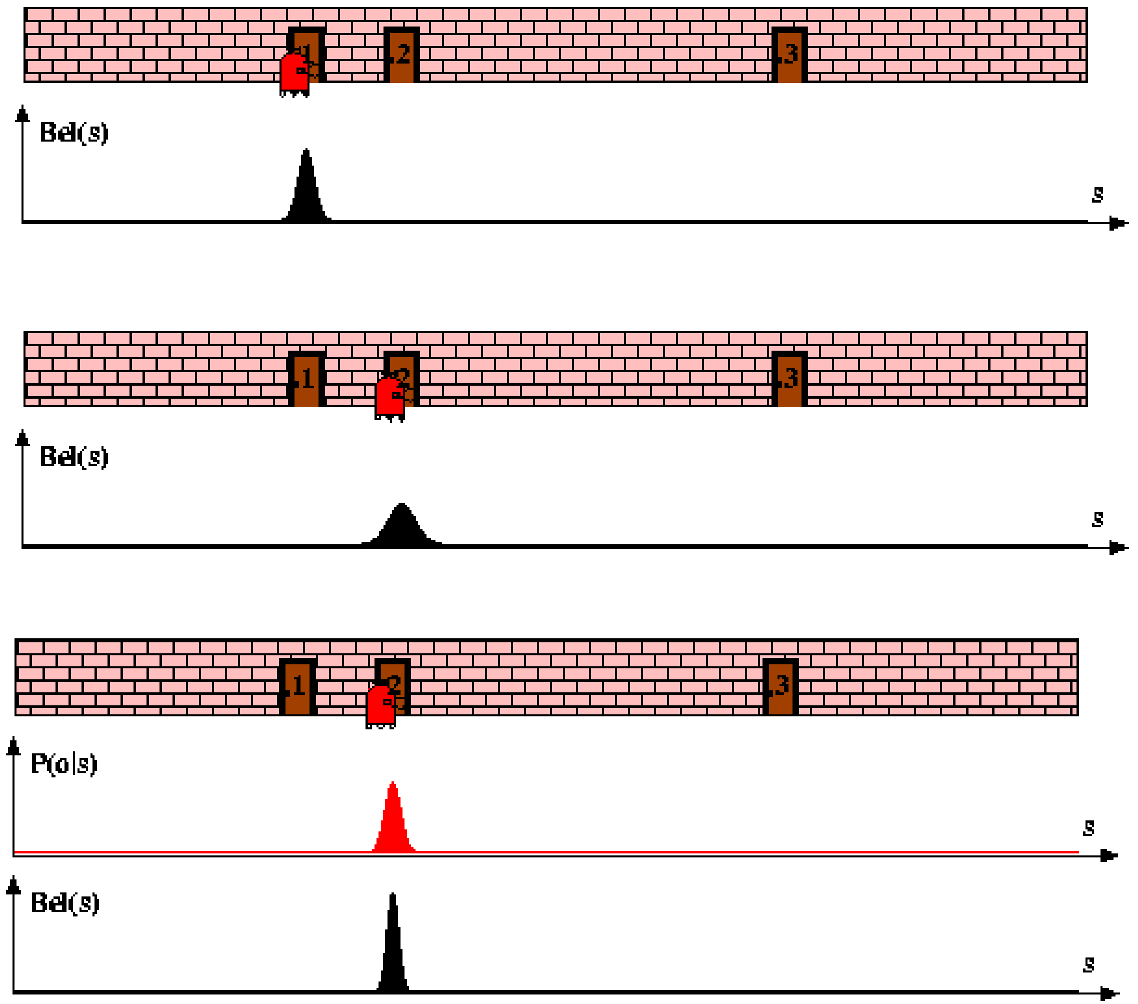
[Schiele et al. 94], [Weiß et al. 94], [Borenstein 96], [Gutmann et al. 96, 98], [Arras 98]

Kalman Filters



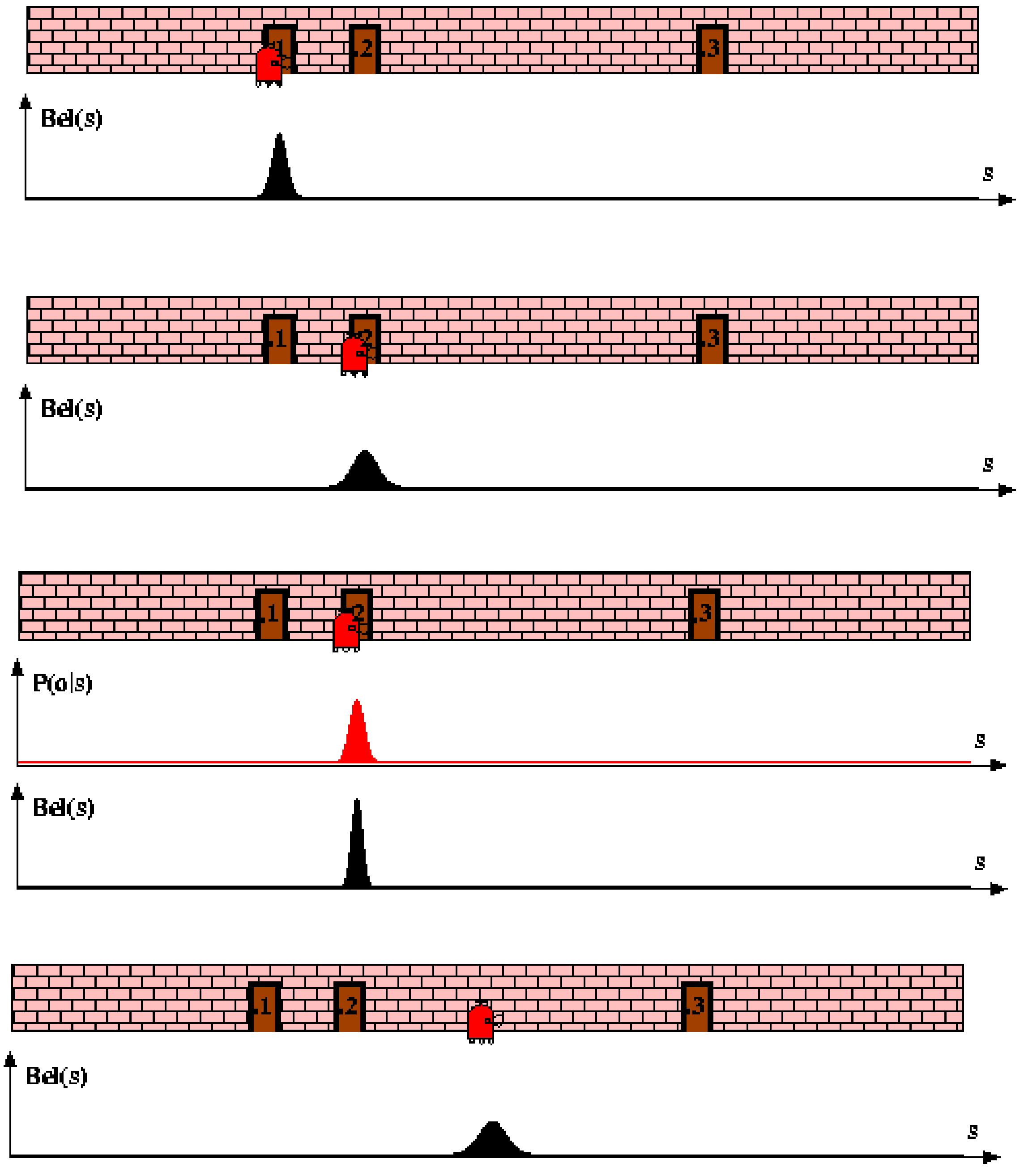
[Schiele et al. 94], [Weiß et al. 94], [Borenstein 96], [Gutmann et al. 96, 98], [Arras 98]

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**(<μ,Σ>, *d*):

- If *d* is a perceptual data item *z* then
- $K = \Sigma C^T \left(C \Sigma C^T + \Sigma_{obs} \right)^{-1}$
- $\mu = \mu + K(z - C\mu)$
- $\Sigma = (I - KC)\Sigma$

- Else if *d* is an action data item *u* then
- $\mu = A\mu + Bu$
- $\Sigma = A \Sigma A^T + \Sigma_{act}$

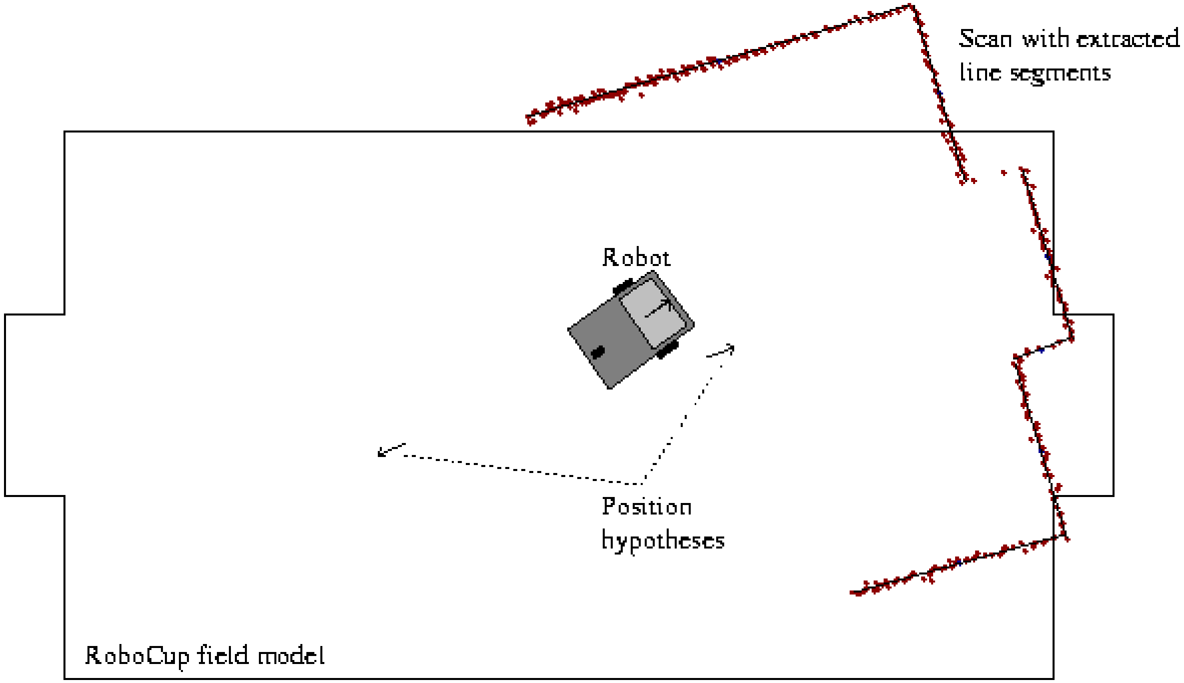
- Return <μ,Σ>

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.

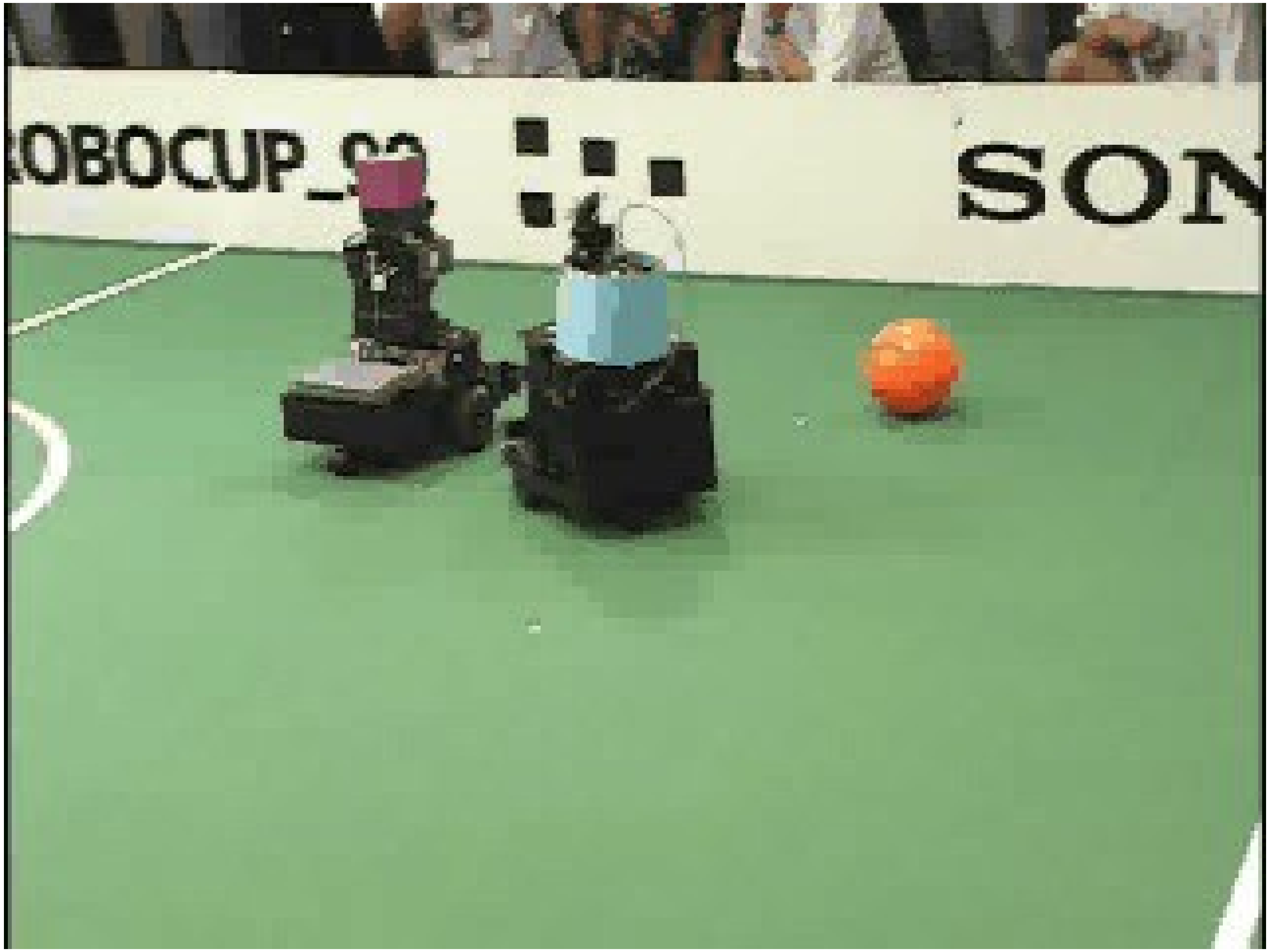
Kalman Filter-based Systems (1)

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



Courtesy of S. Gutmann

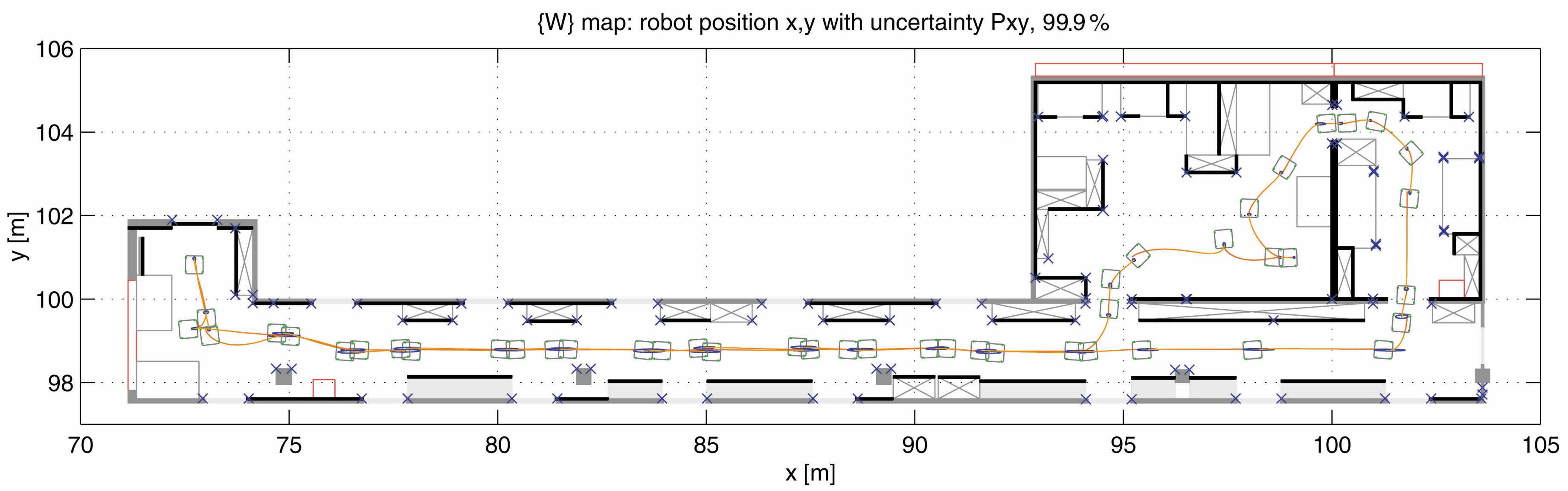
Kalman Filter-based Systems (2)



Courtesy of S. Gutmann

Kalman Filter-based Systems (3)

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

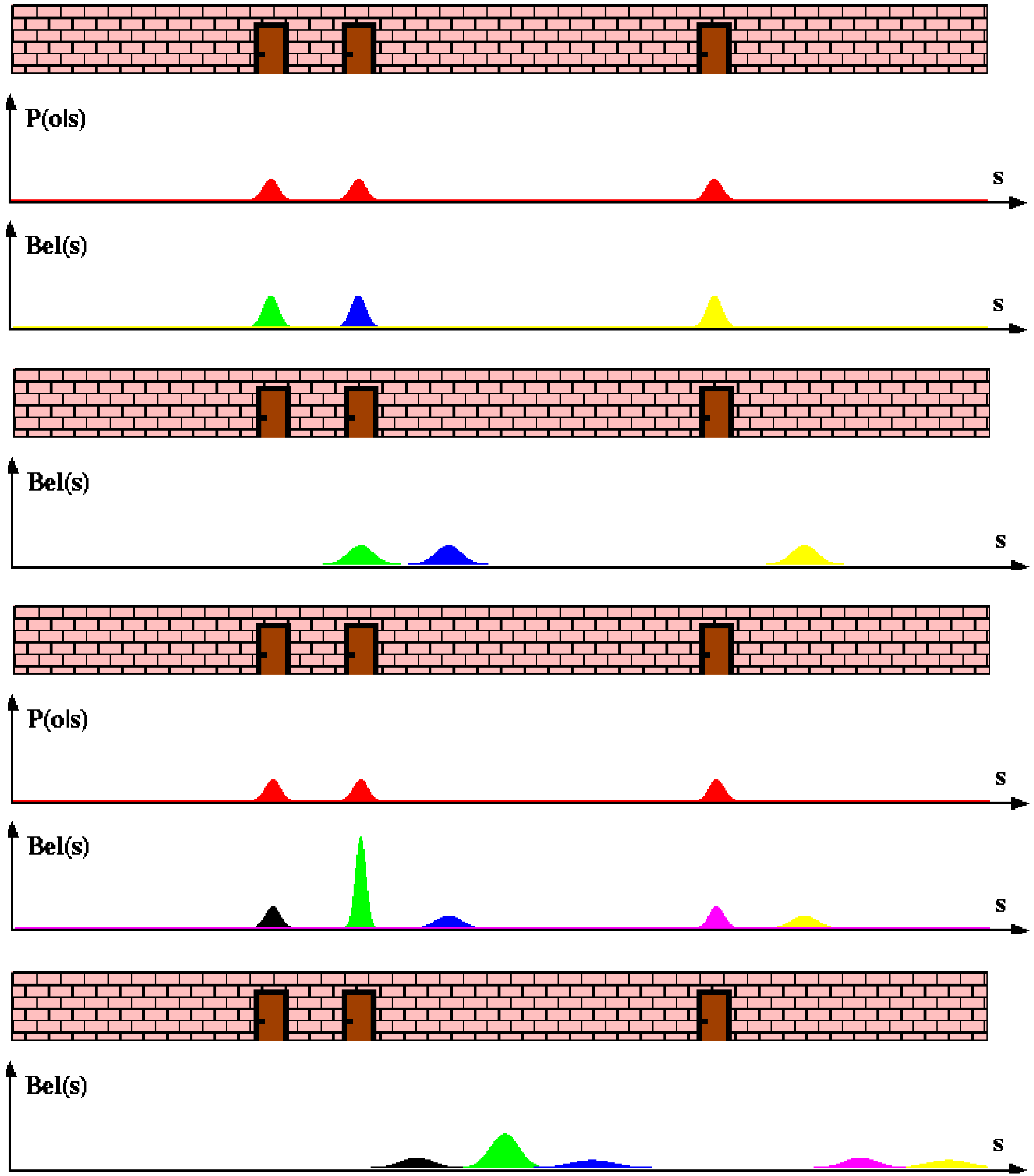


Courtesy of K. Arras

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.

See e.g. [Cox 93]

MHT: Implemented System (1)

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

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

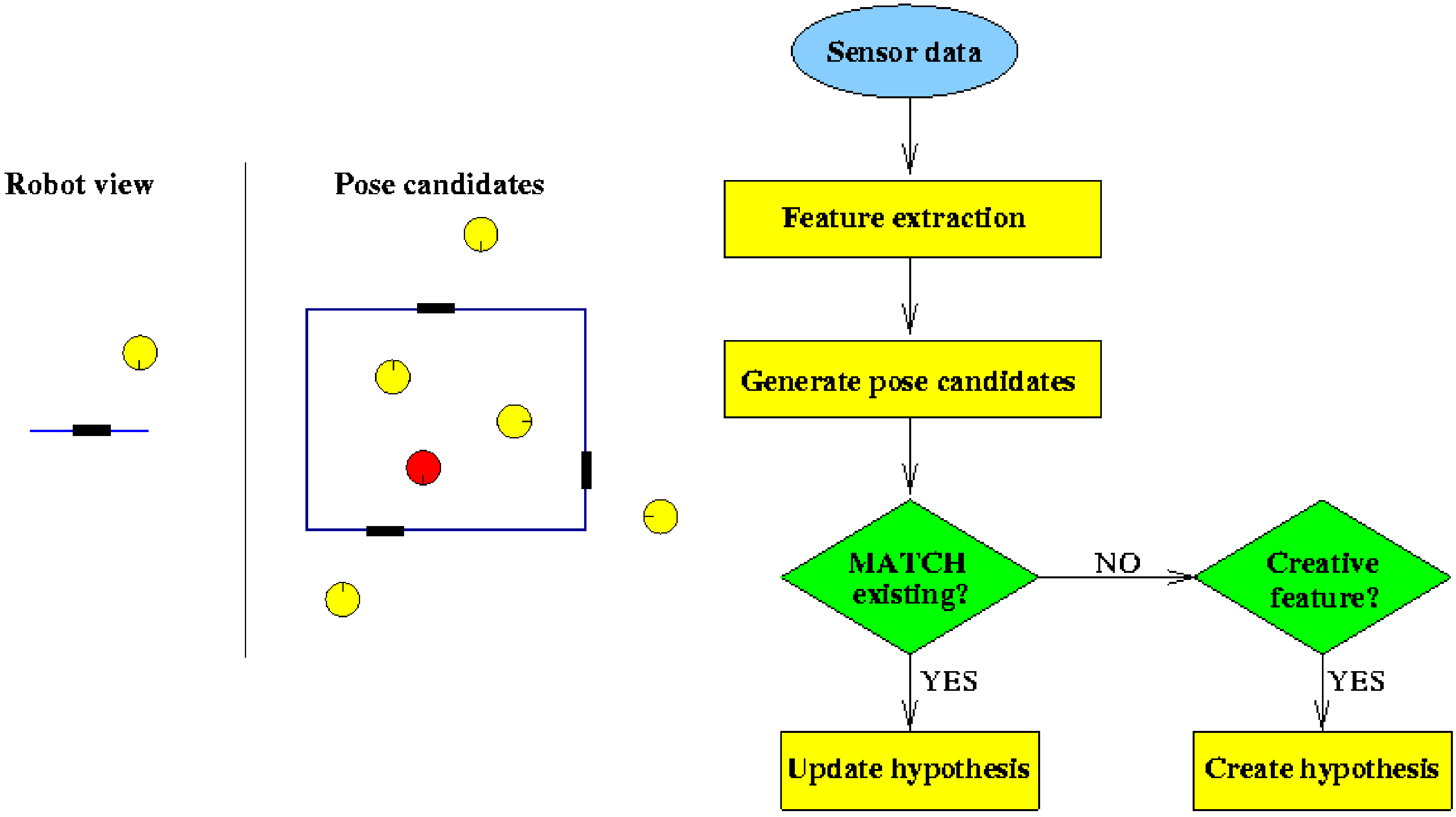
  - Hypothesis probability is computed using Bayes' rule

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

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

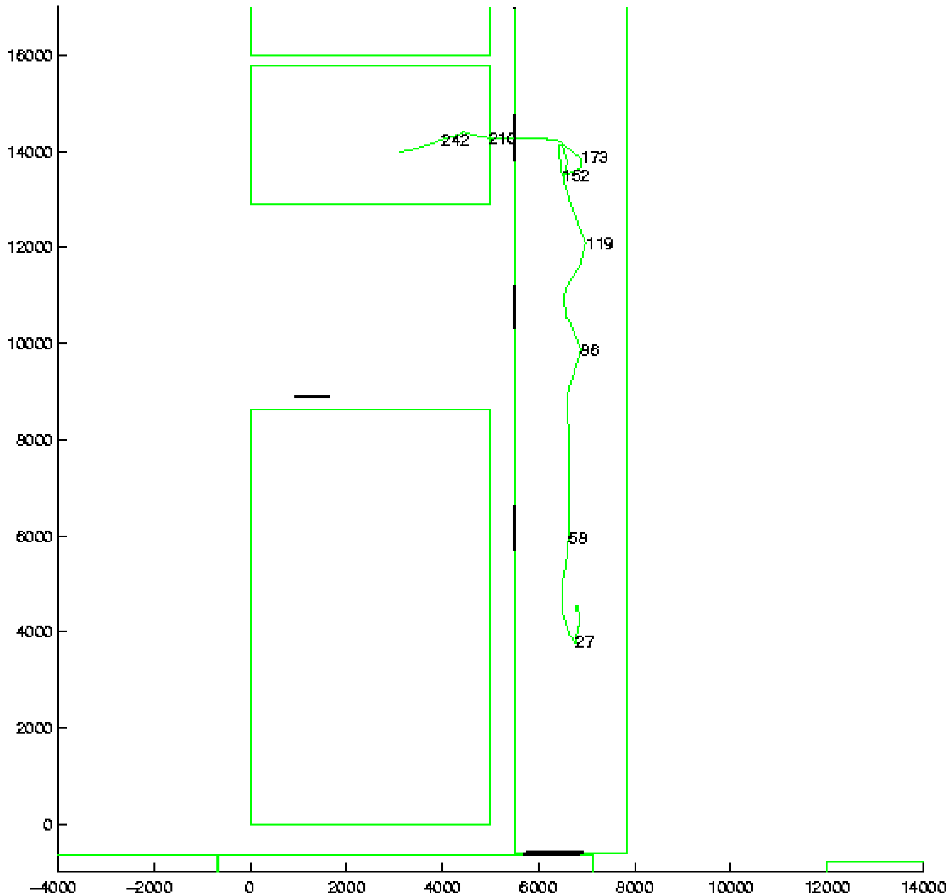
$$C_j = \{z_j, R_j\}$$

MHT: Implemented System (2)

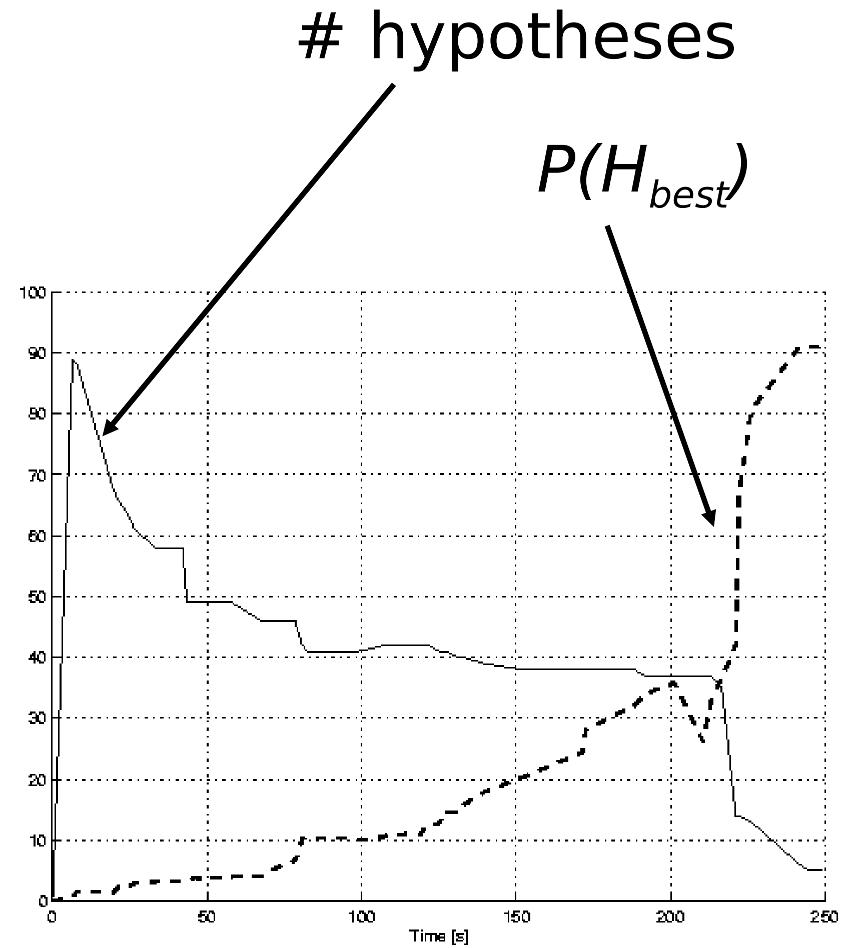


Courtesy of P. Jensfelt and S. Kristensen

MHT: Implemented System (3) Example run



Map and trajectory



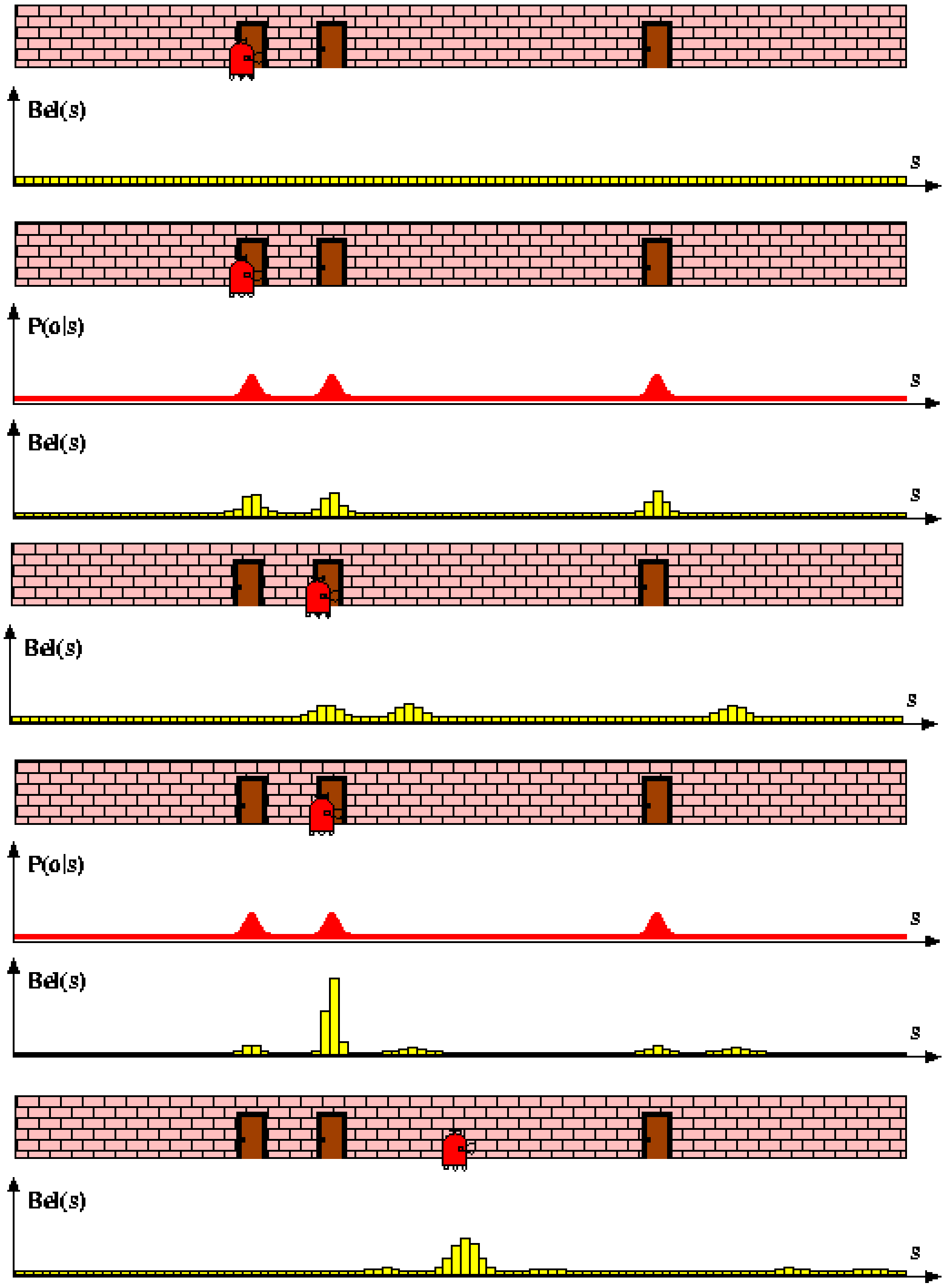
Hypotheses vs. time

Courtesy of P. Jensfelt and S. Kristensen

Localization Algorithms - Comparison

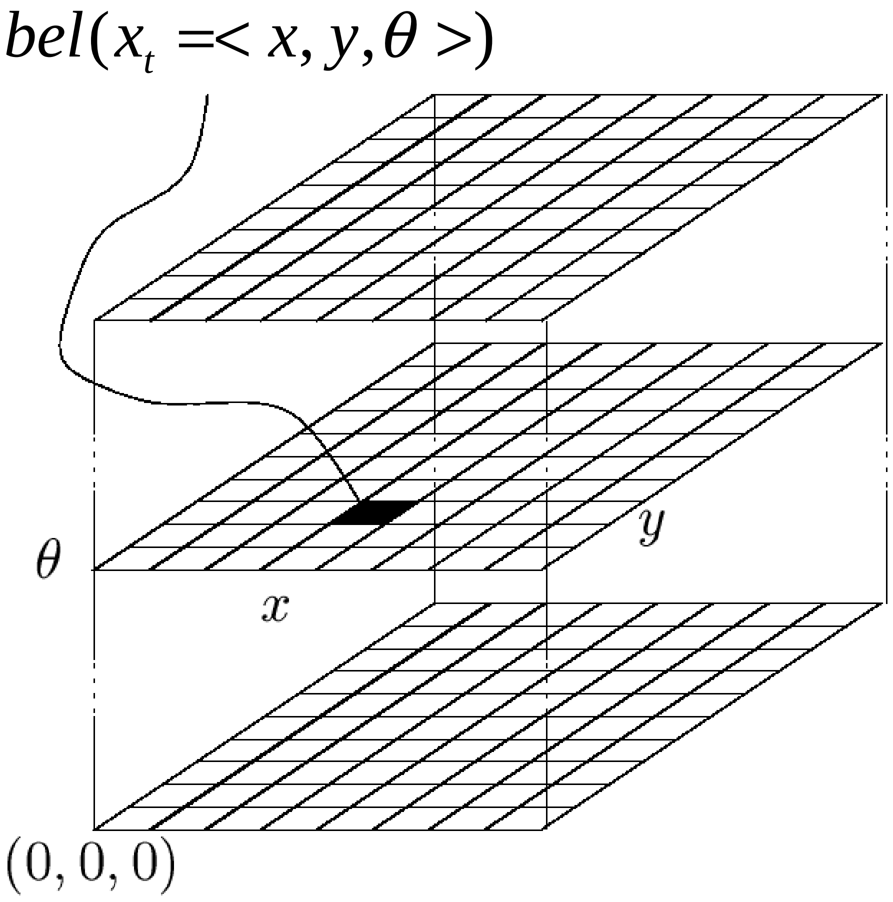
	Kalman filter	Multi-hypothesis tracking
Sensors	Gaussian	Gaussian
Posterior	Gaussian	Multi-modal
Efficiency (memory)	++	++
Efficiency (time)	++	++
Implementation	+	o
Accuracy	++	++
Robustness	-	+
Global localization	No	Yes

Piecewise Constant

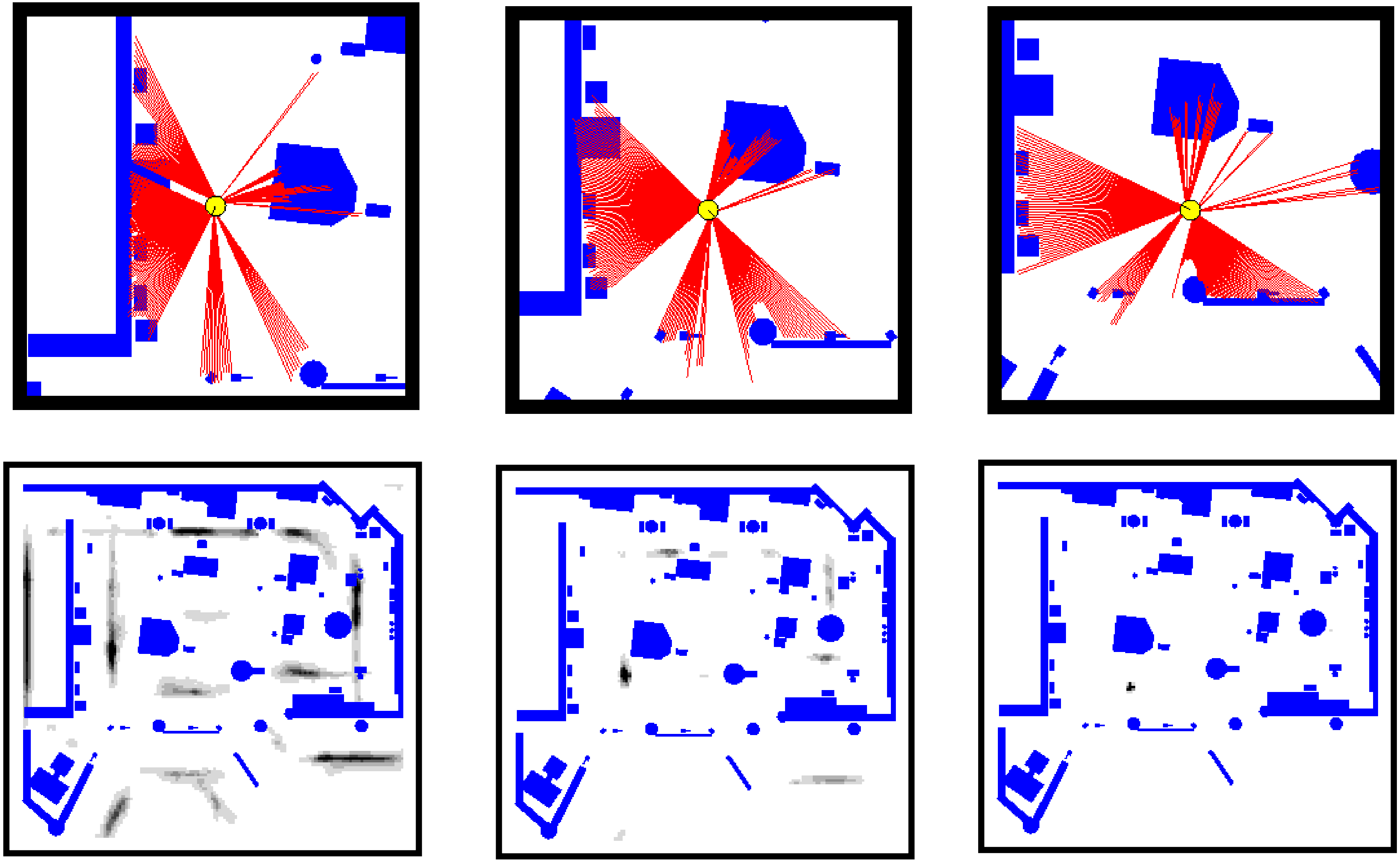


[Burgard et al. 96,98], [Fox et al. 99], [Konolige et al. 99]

Piecewise Constant Representation

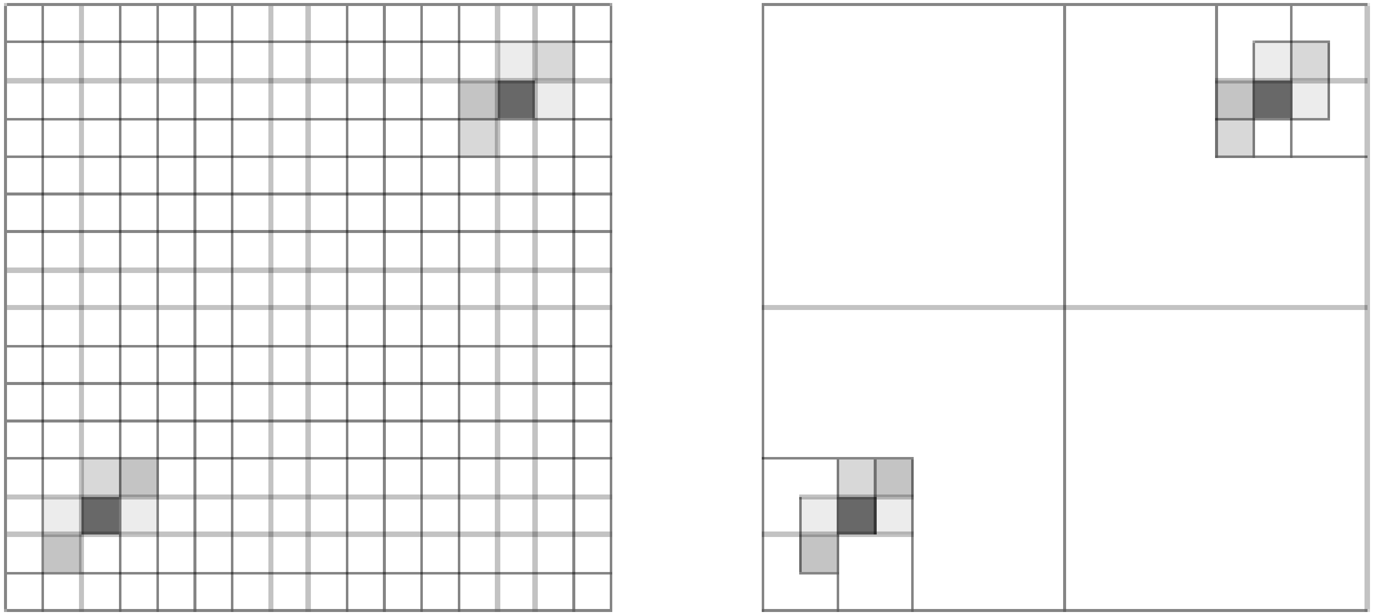


Grid-based Localization



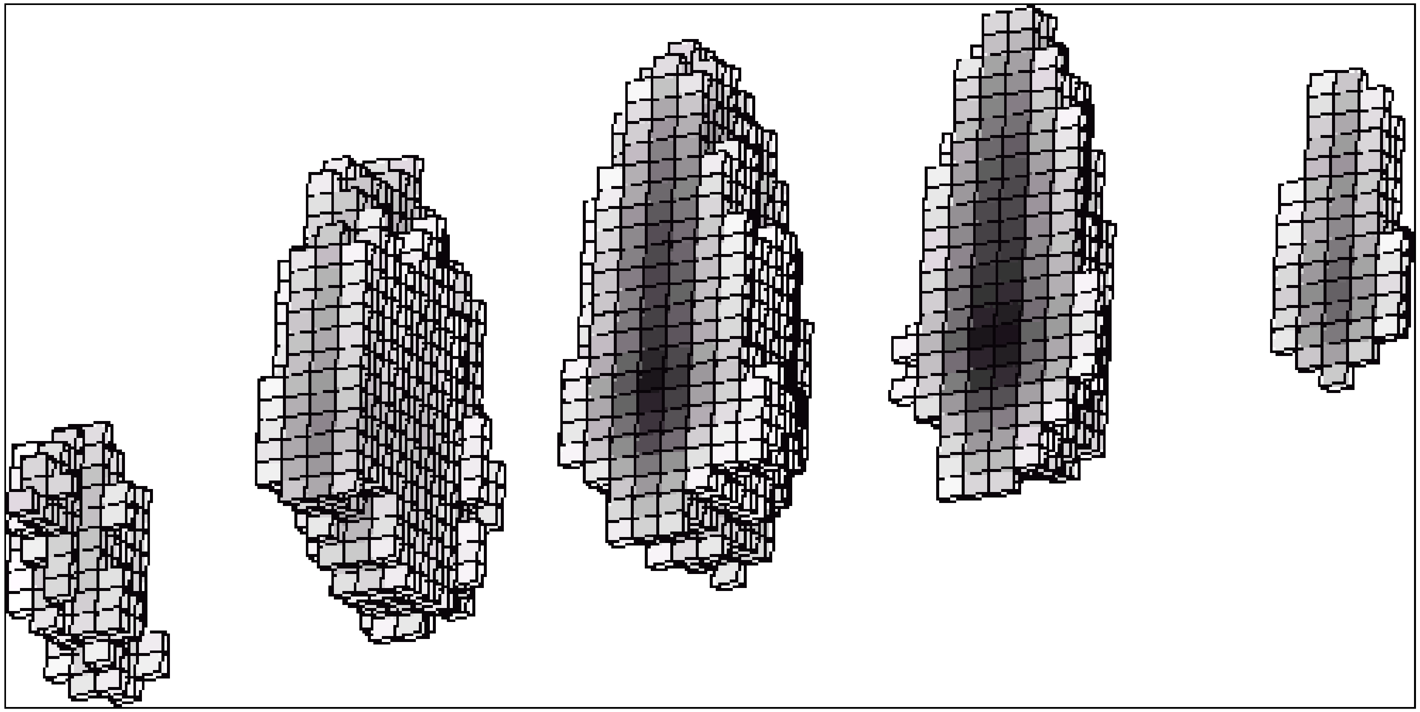
Tree-based Representations (1)

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-hypothesis 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-hypothesis 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

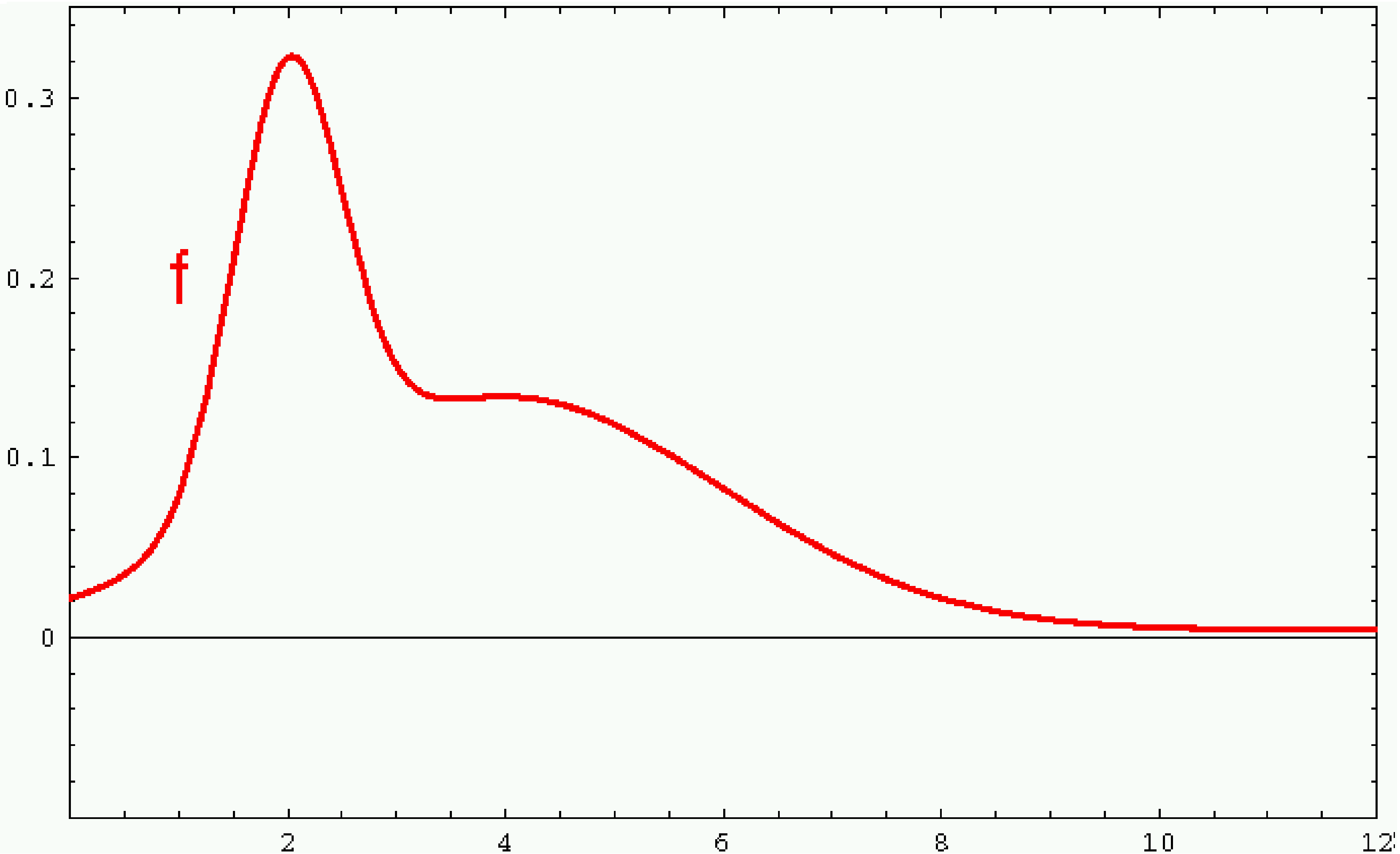
- Represent density by random samples
- Estimation of non-Gaussian, nonlinear processes

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

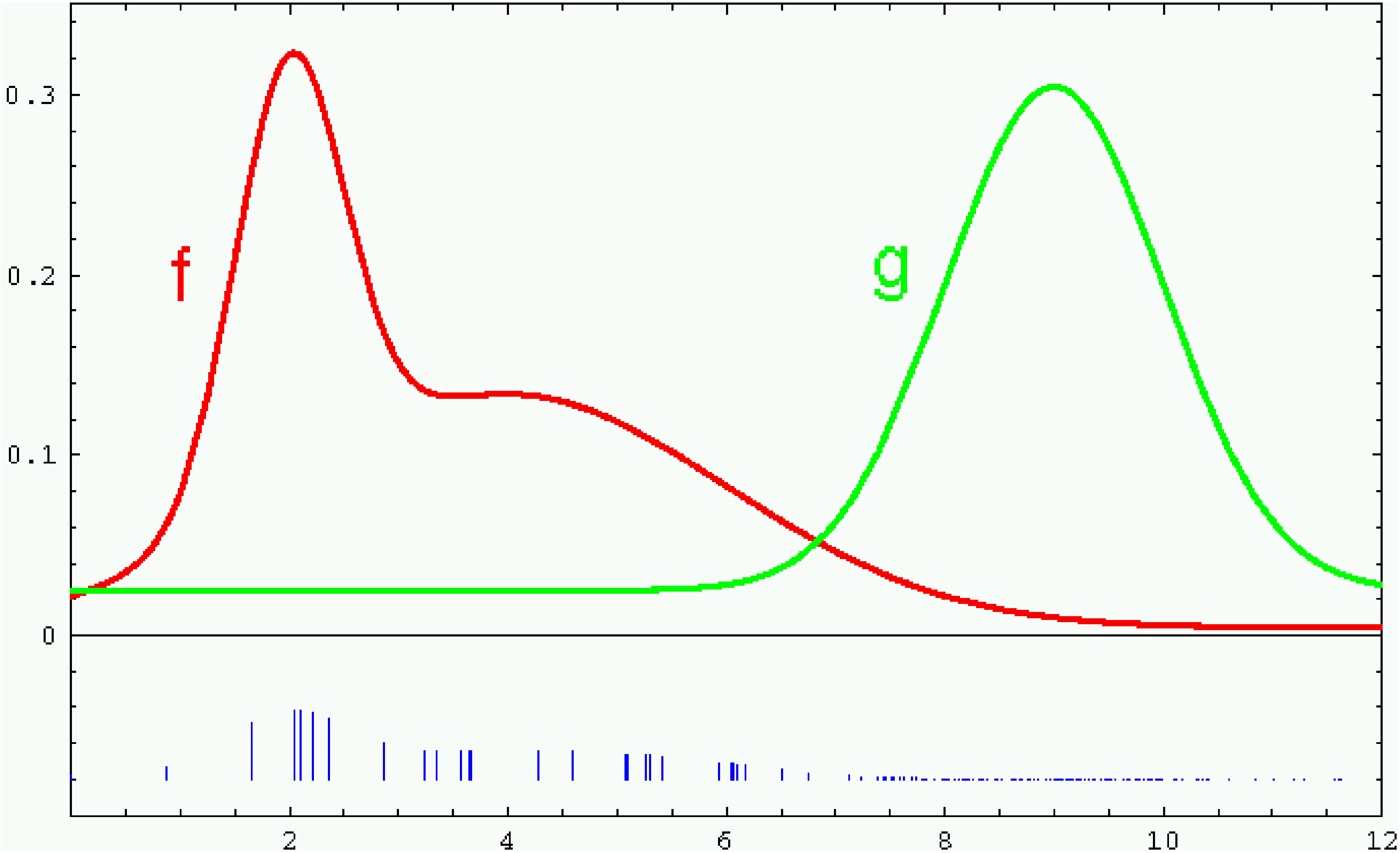
- Filtering: [Rubin, 88], [Gordon et al., 93], [Kitagawa 96]
- Computer vision: [Isard and Blake 96, 98]
- Dynamic Bayesian Networks: [Kanazawa et al., 95]

Monte Carlo Localization (MCL)

Represent Density Through Samples

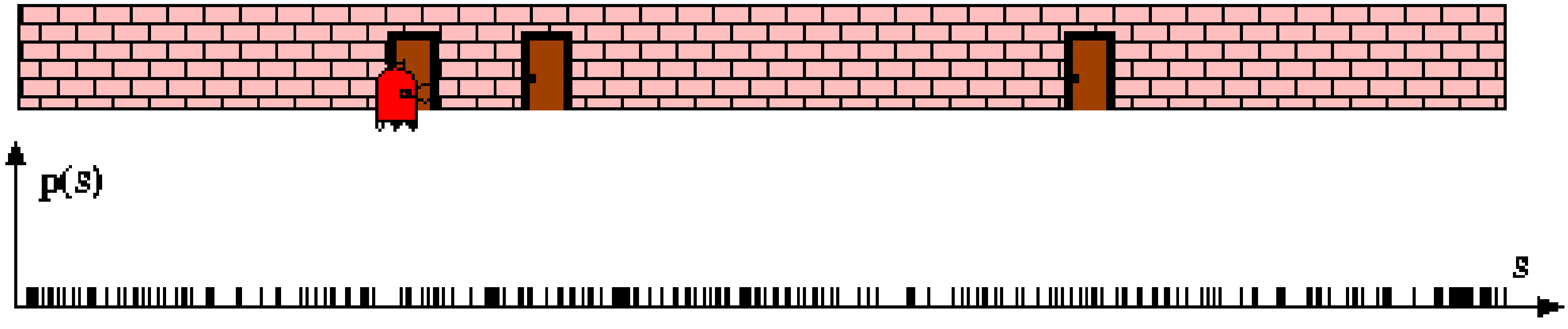


Importance Sampling



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

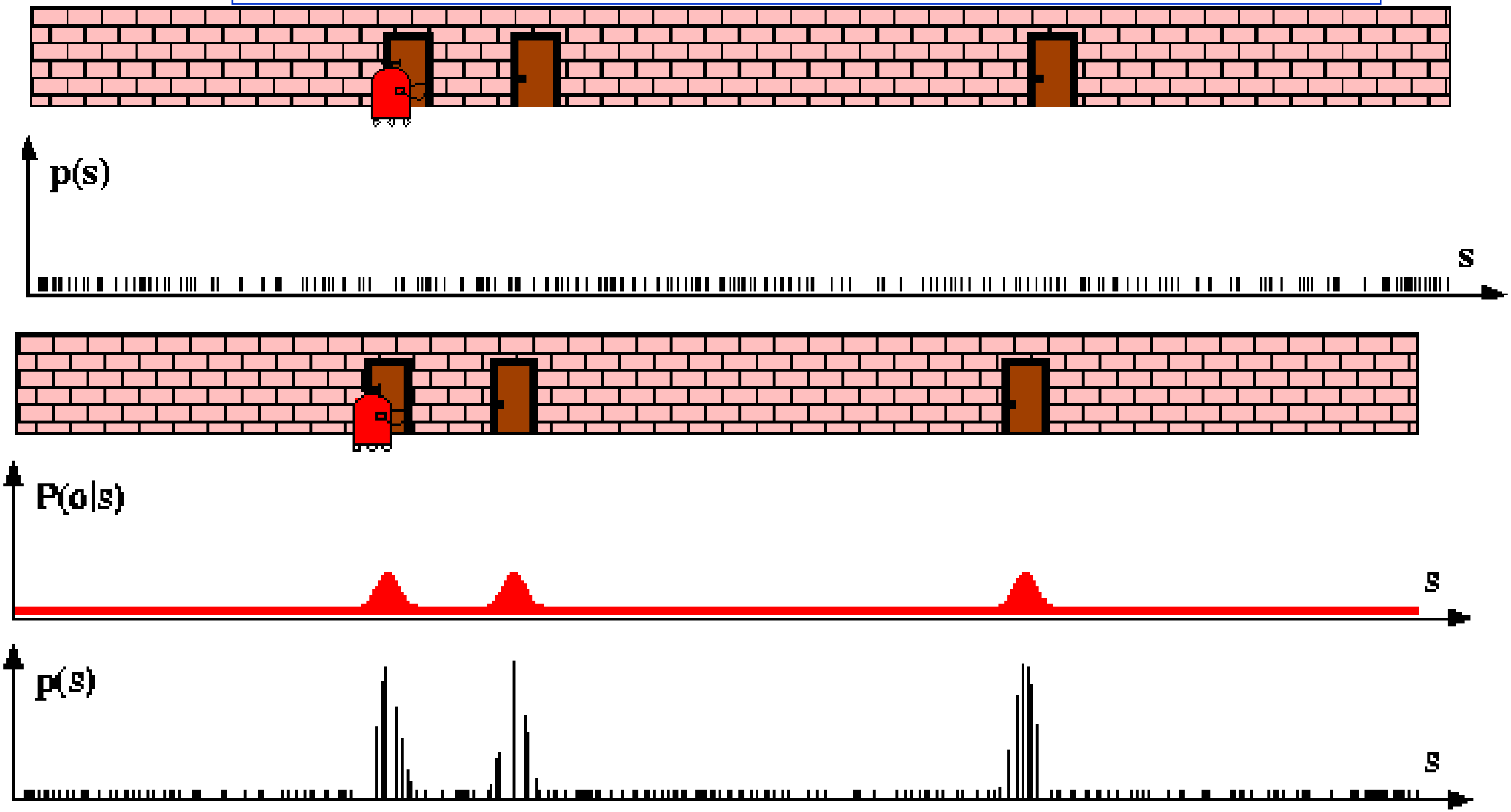
MCL: Global Localization



MCL: Sensor Update

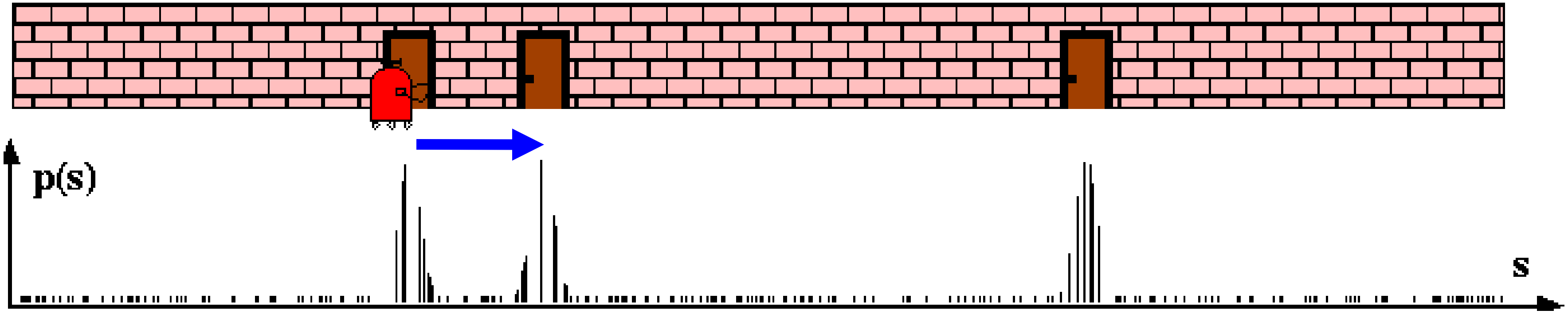
$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

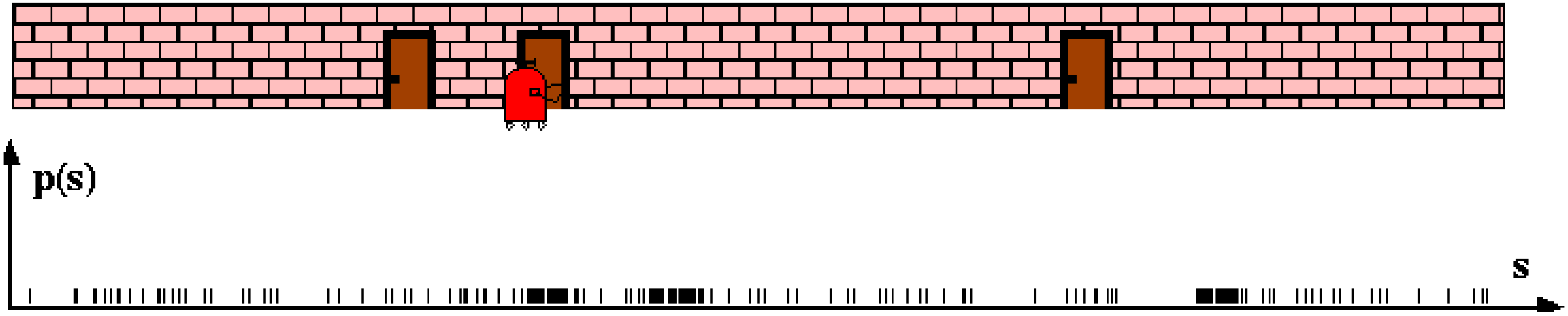
$$w \leftarrow \frac{\alpha\, p(z \mid x)\, Bel^{-}(x)}{Bel^{-}(x)} = \alpha\, p(z \mid x)$$



MCL: Robot Motion

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

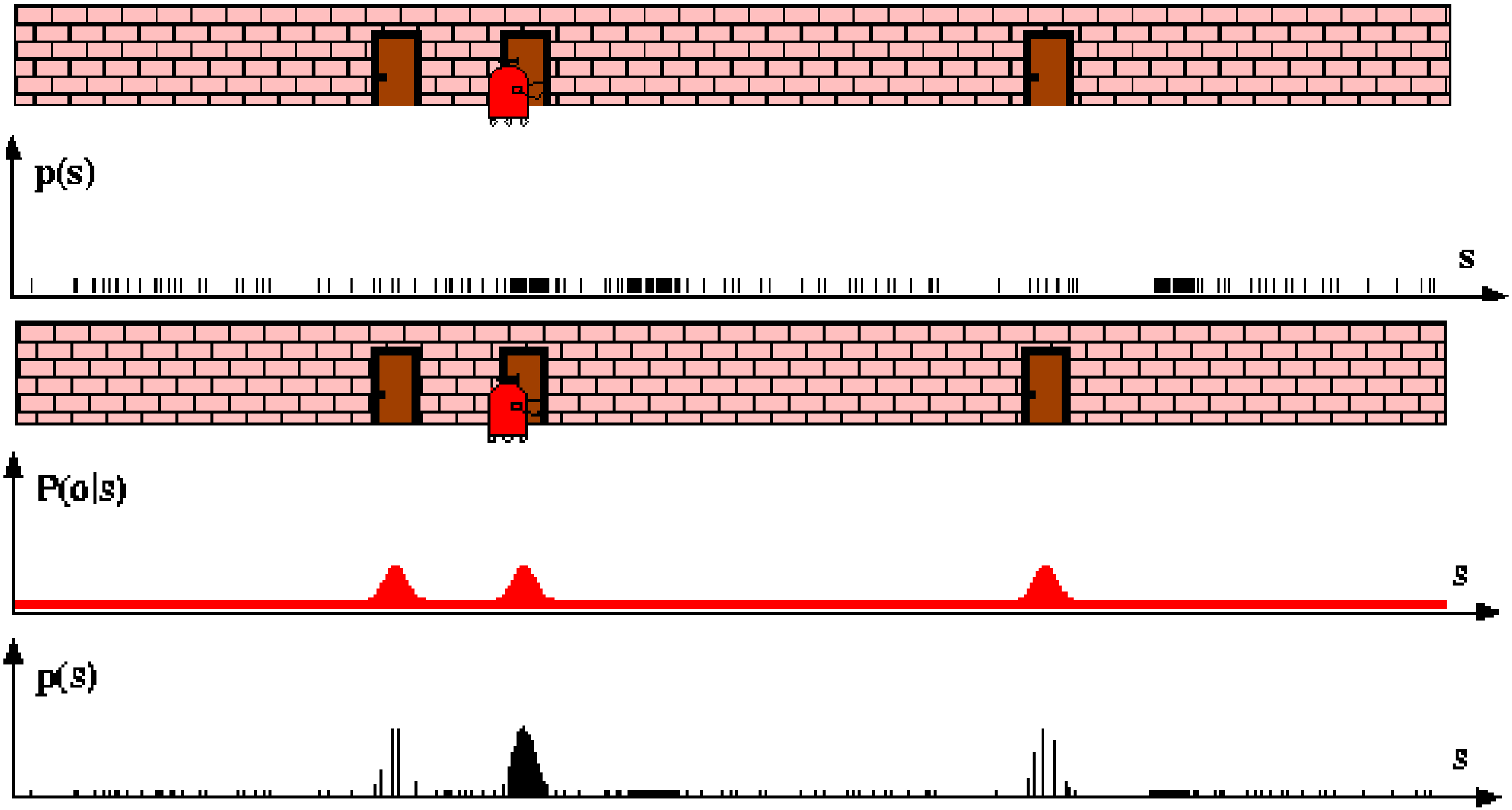




MCL: Sensor Update

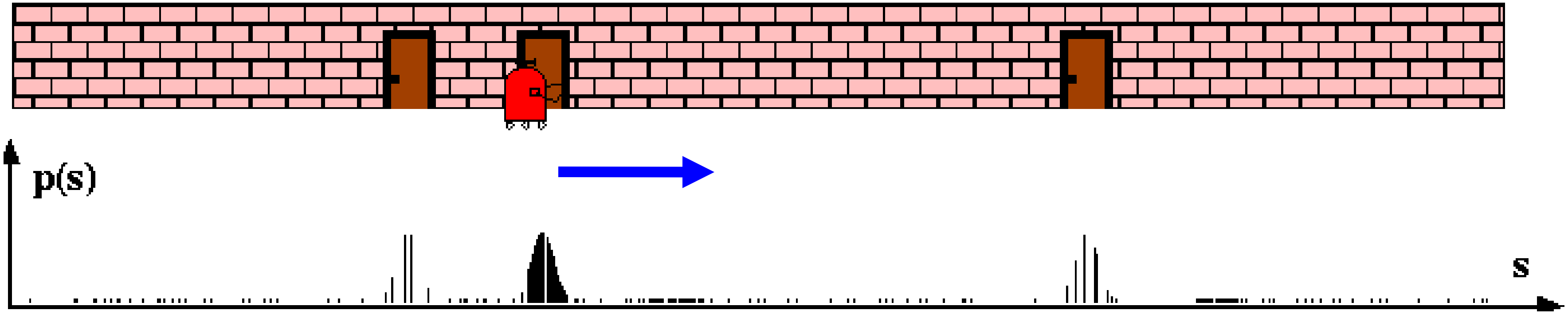
$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

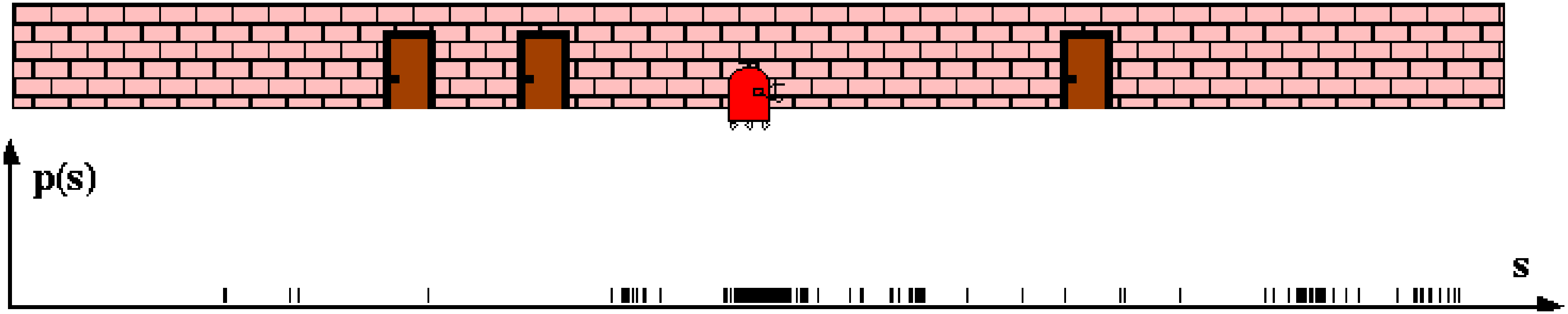
$$w \leftarrow \frac{\alpha\, p(z \mid x)\, Bel^{-}(x)}{Bel^{-}(x)} = \alpha\, p(z \mid x)$$



MCL: Robot Motion

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





Particle Filter Algorithm

1. Algorithm **particle_filter**(S_{t-1}, u_{t-1} z_t):
2. $S_t = \emptyset, \quad \eta = 0$
3. **For** $i = 1 \ldots n$ ***Generate new samples***

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

1. Sample x_t^i from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x_{t-1}^{j(i)}$ and u_{t-1}
2. $w_t^i = p(z_t \mid x_t^i)$ ***Compute importance weight***
3. $\eta = \eta + w_t^i$ ***Update normalization factor***
4. $S_t = S_t \cup \{< x_t^i, w_t^i >\}$ ***Insert***
5. **For** $i = 1 \ldots n$
6. $w_t^i = w_t^i / \eta$ ***Normalize weights***

Particle Filter Algorithm

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

draw x^i_{t-1} from $Bel(x_{t-1})$

draw x^i_t from $p(x_t \mid x^i_{t-1}, u_{t-1})$

Importance factor for x^i_t:

$$w^i_t = \frac{\text{target distribution}}{\text{proposal distribution}}$$

$$= \frac{\eta \; p(z_t \mid x_t) \; p(x_t \mid x_{t-1}, u_{t-1}) \; Bel(x_{t-1})}{p(x_t \mid x_{t-1}, u_{t-1}) \; Bel(x_{t-1})}$$

$$\propto p(z_t \mid x_t)$$

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** $i = 2 \ldots n$ ***Generate cdf***
3. $\quad c_i = c_{i-1} + w^i$
4. $u_1 \sim U[0, n^{-1}], i = 1$ ***Initialize threshold***

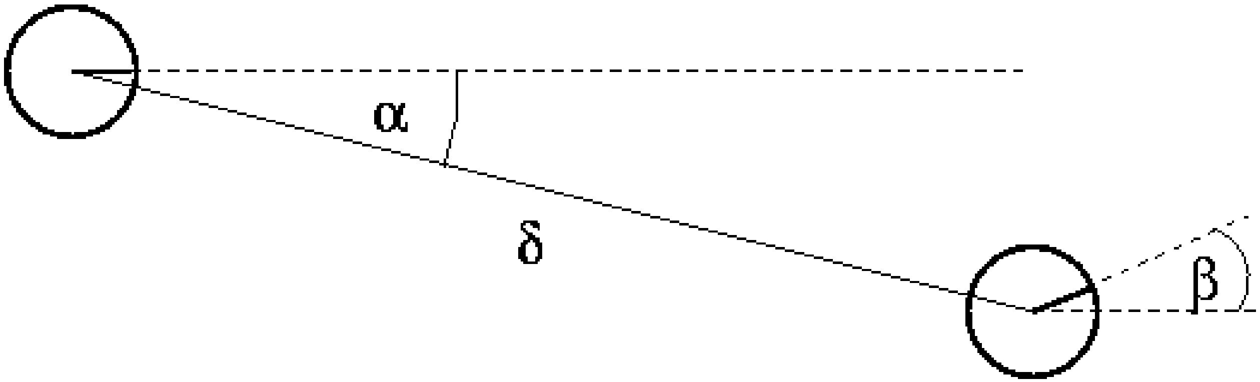
1. **For** $j = 1 \ldots n$ ***Draw samples …***
2. $\quad u_j = u_1 + n^{-1} \cdot (j-1)$ ***Advance threshold***
3. $\quad$ **While (** $u_j > c_i$**)** ***Skip until next threshold reached***
4. $\quad\quad i = i + 1$
5. $\quad S' = S' \cup \{ < x^i, n^{-1} > \}$ ***Insert***

1. **Return** S'

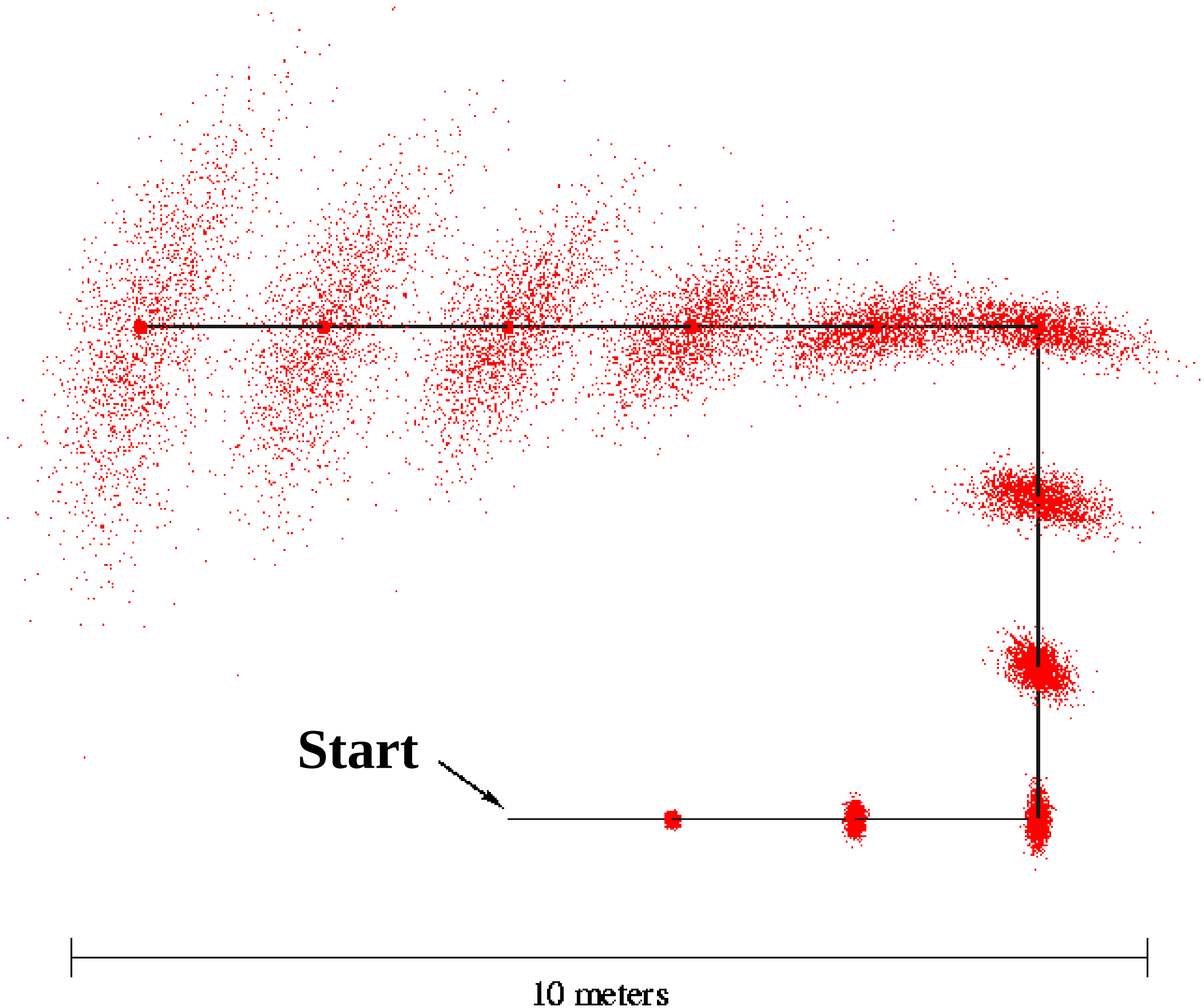
Also called **stochastic universal sampling**

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

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



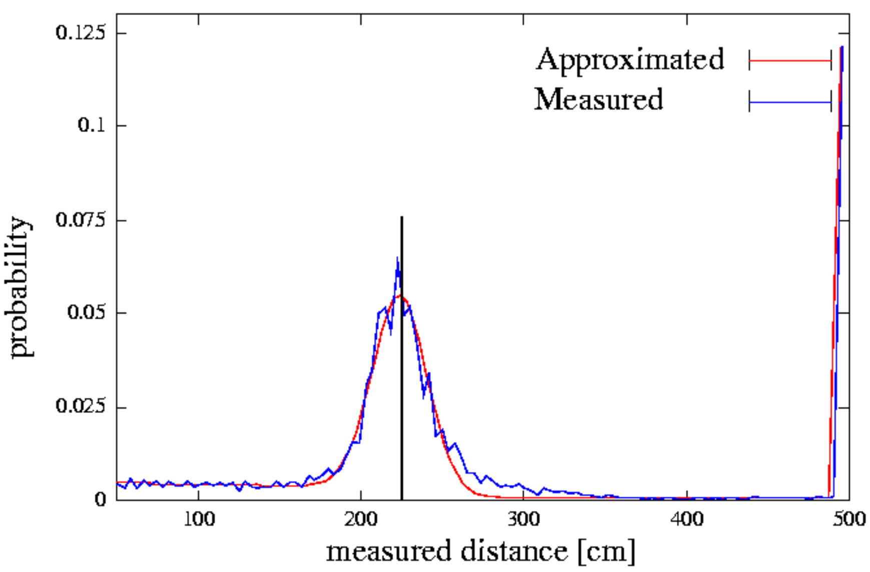
Motion Model $p(x_t \mid a_{t-1}, x_{t-1})$



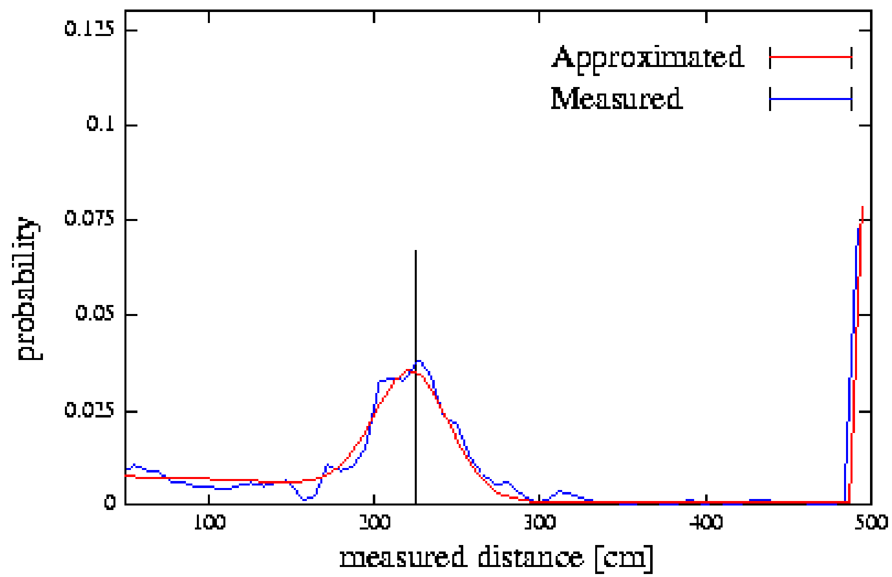
Model for Proximity Sensors

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

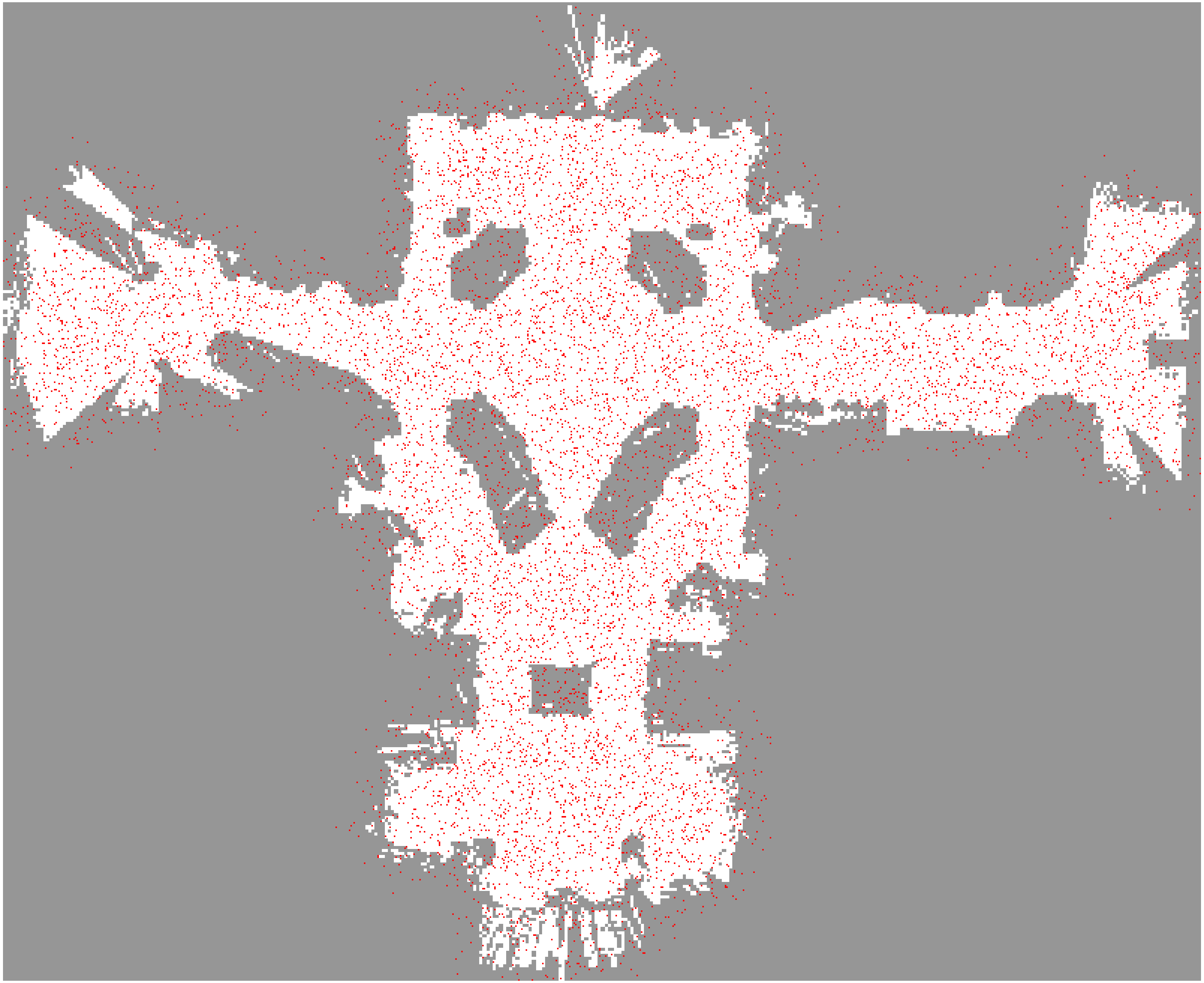
$$P(d_i \mid l) \;=\; 1-(1-(1-\sum_{j<i} P_u(d_j))\, c_d\, P_m(d_i \mid l)))\cdot(1-(1-\sum_{j<i} P(d_j))\, c_r)$$

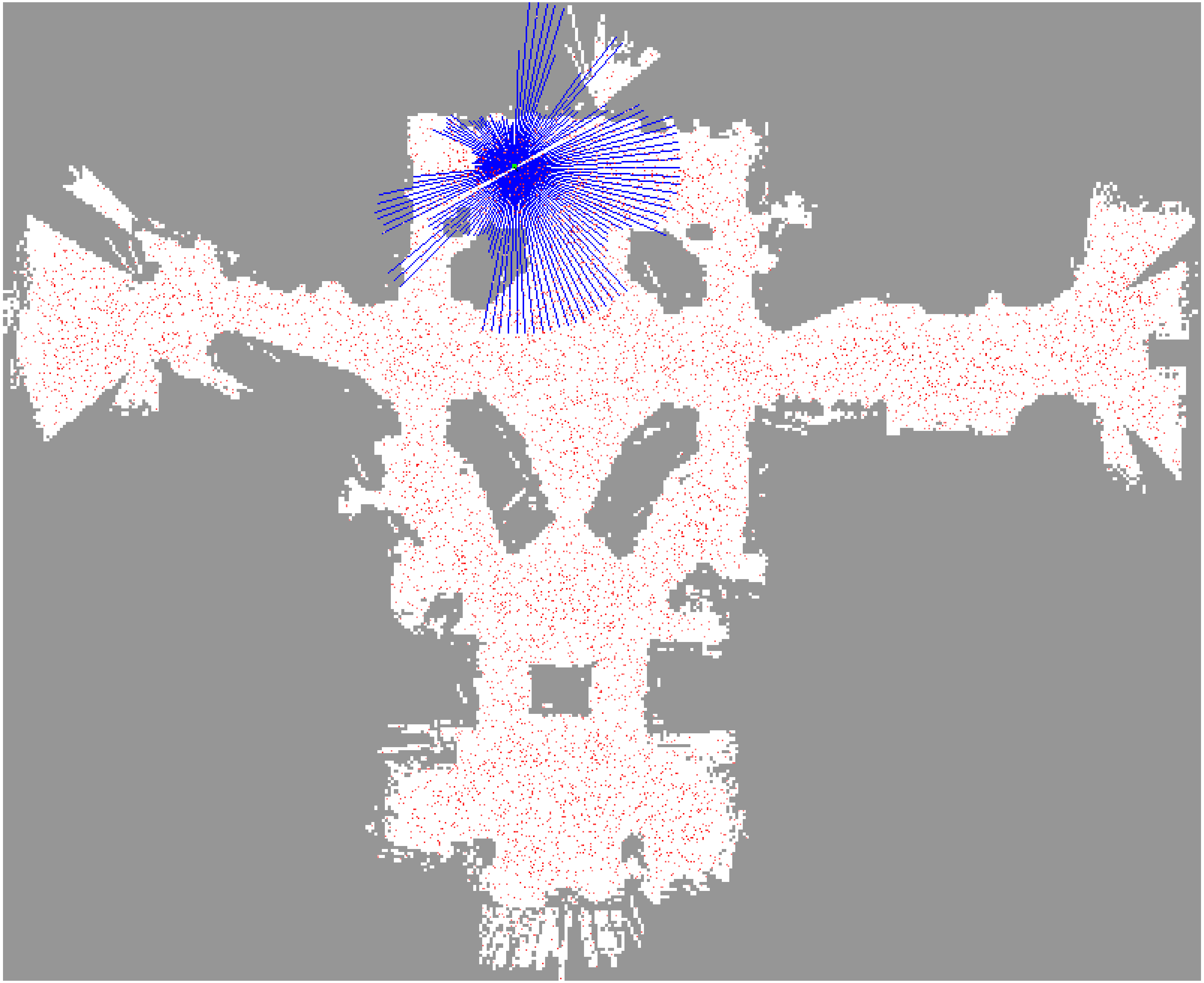


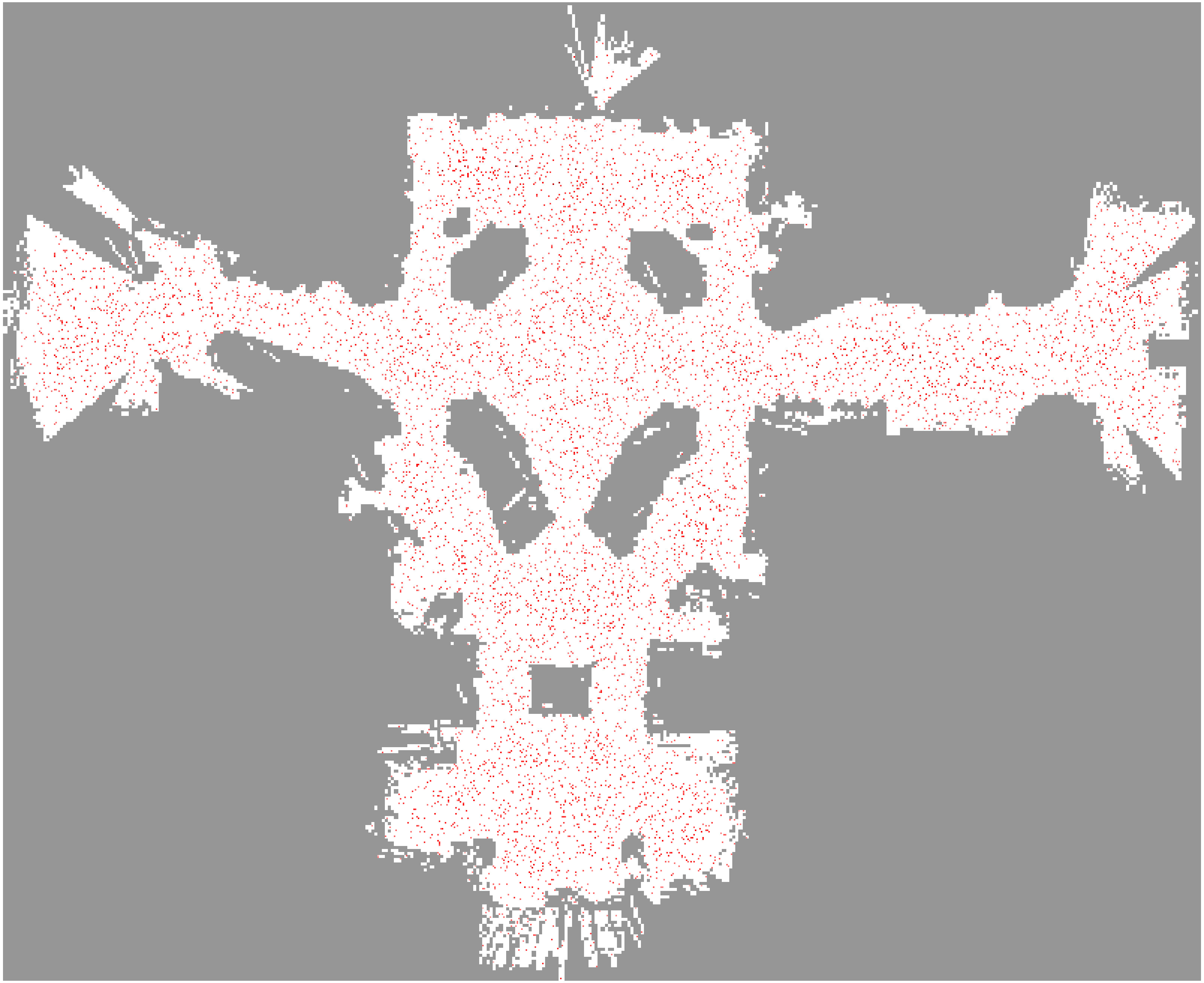
Laser sensor

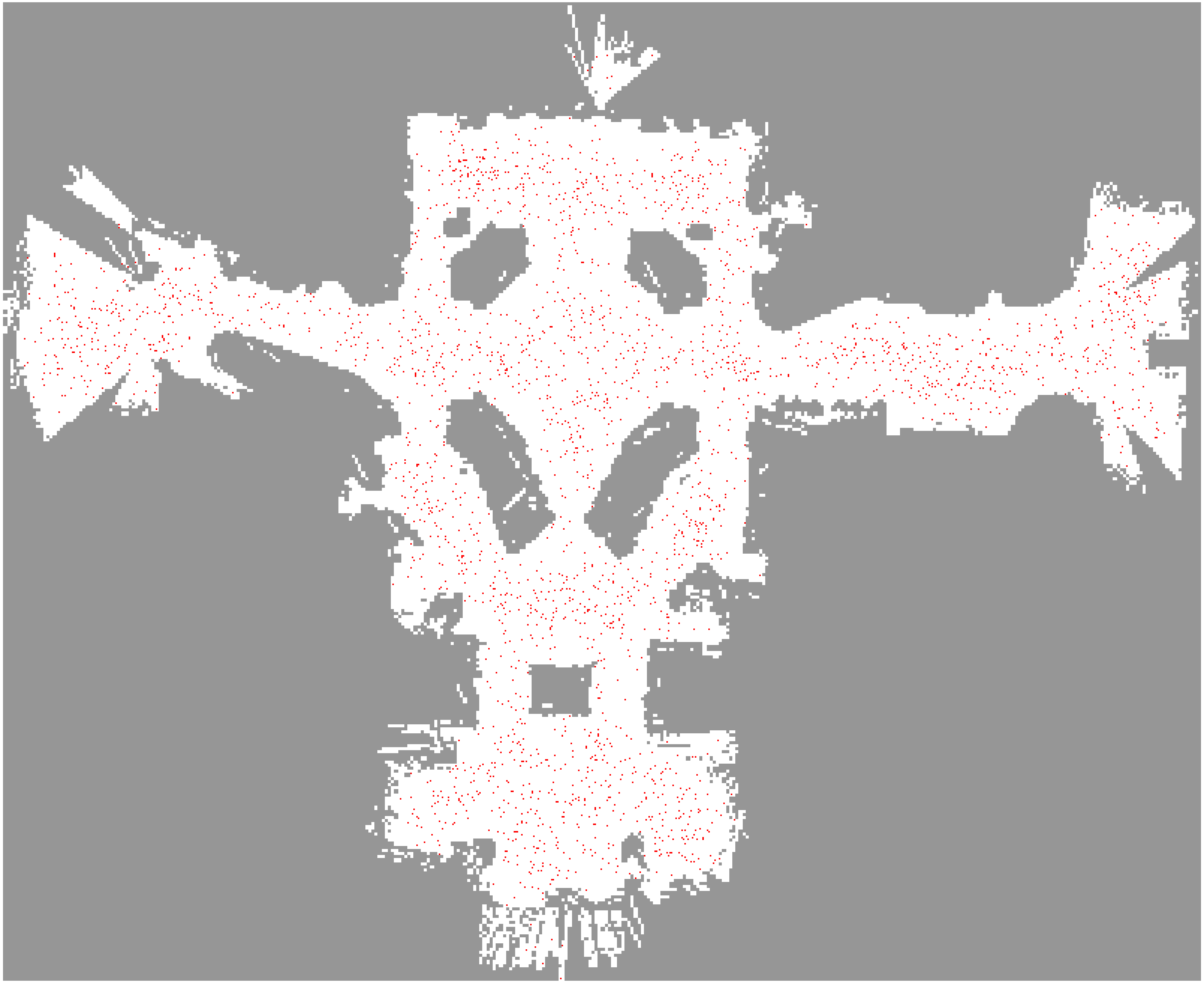


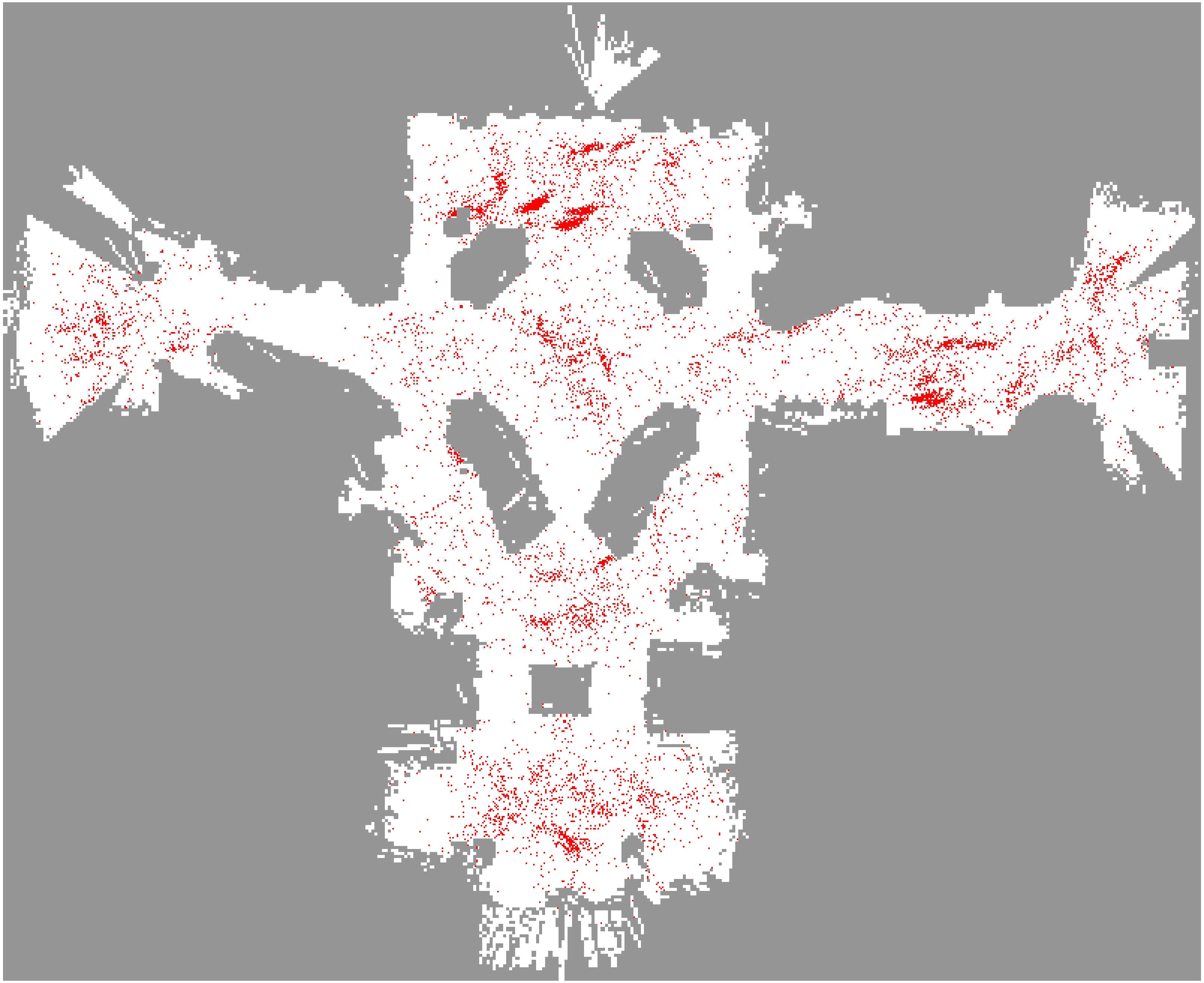
Sonar sensor

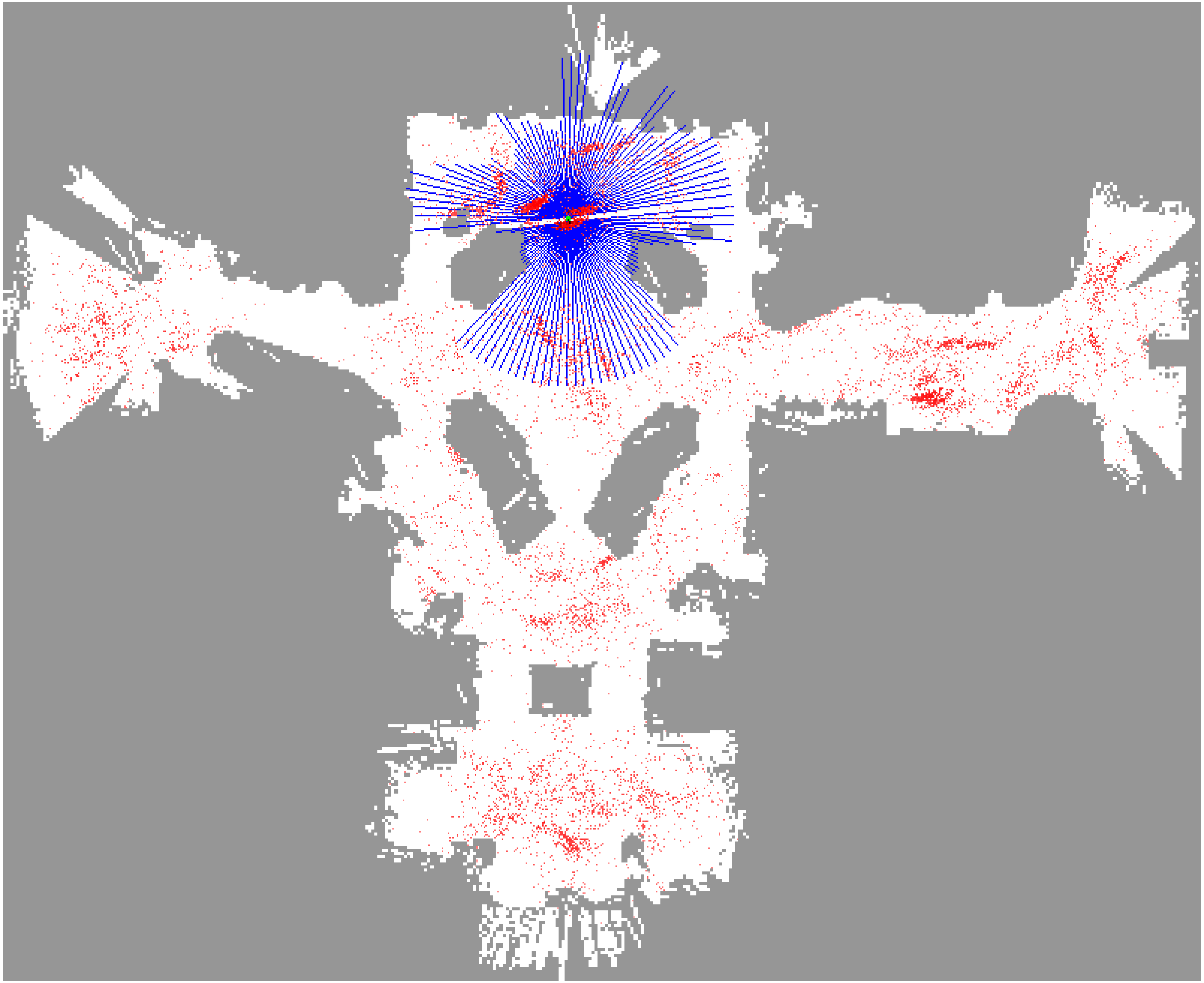


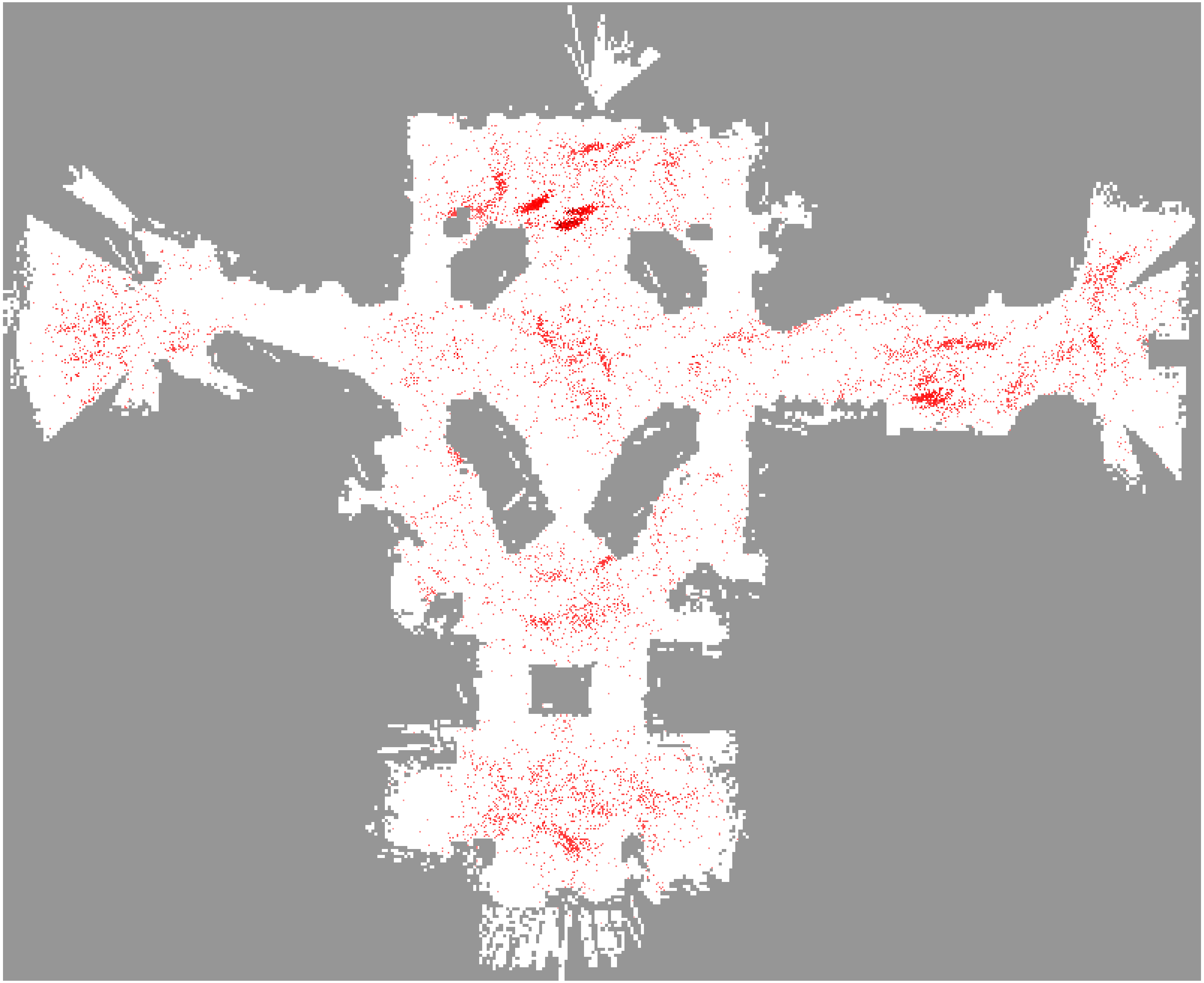


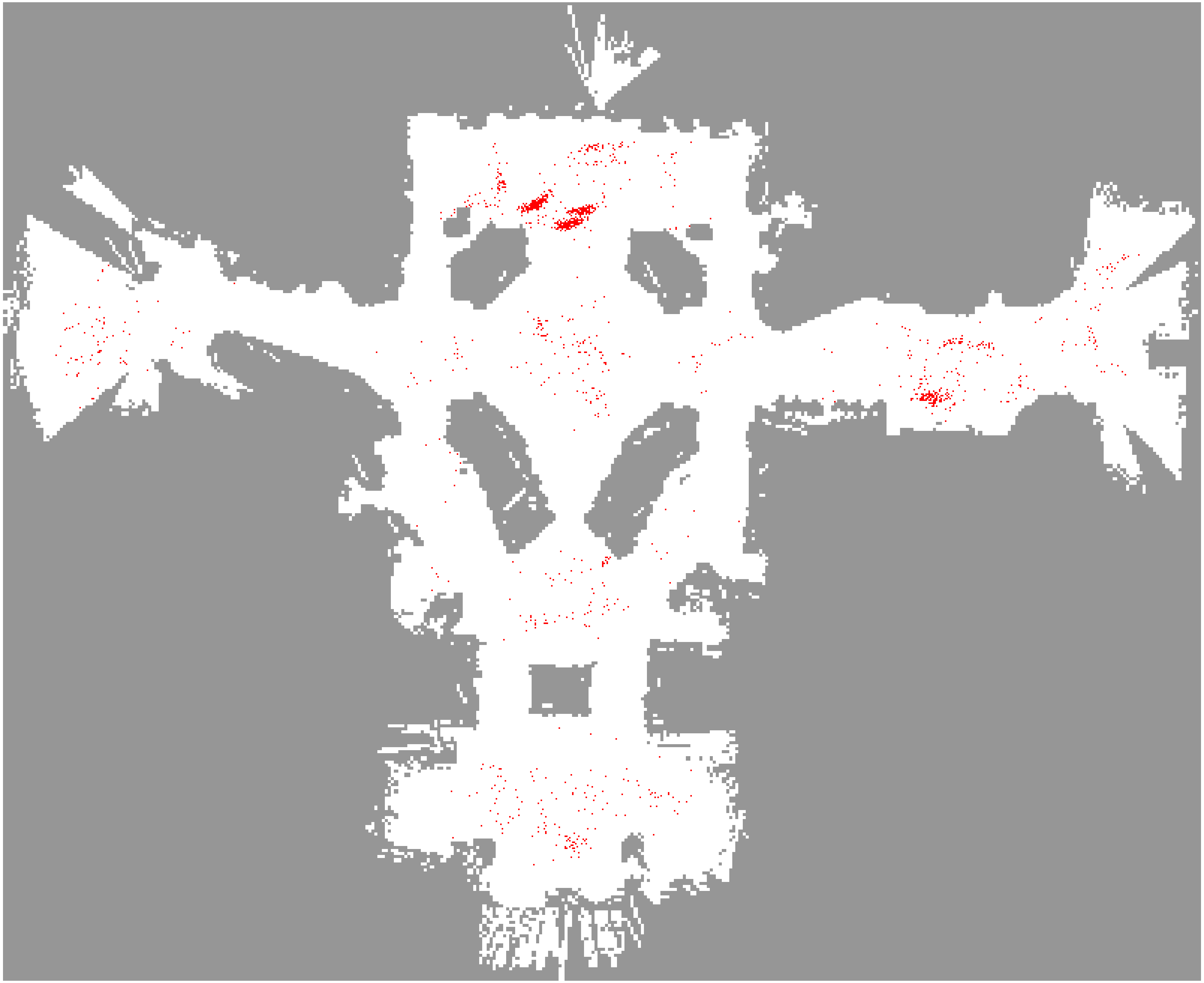


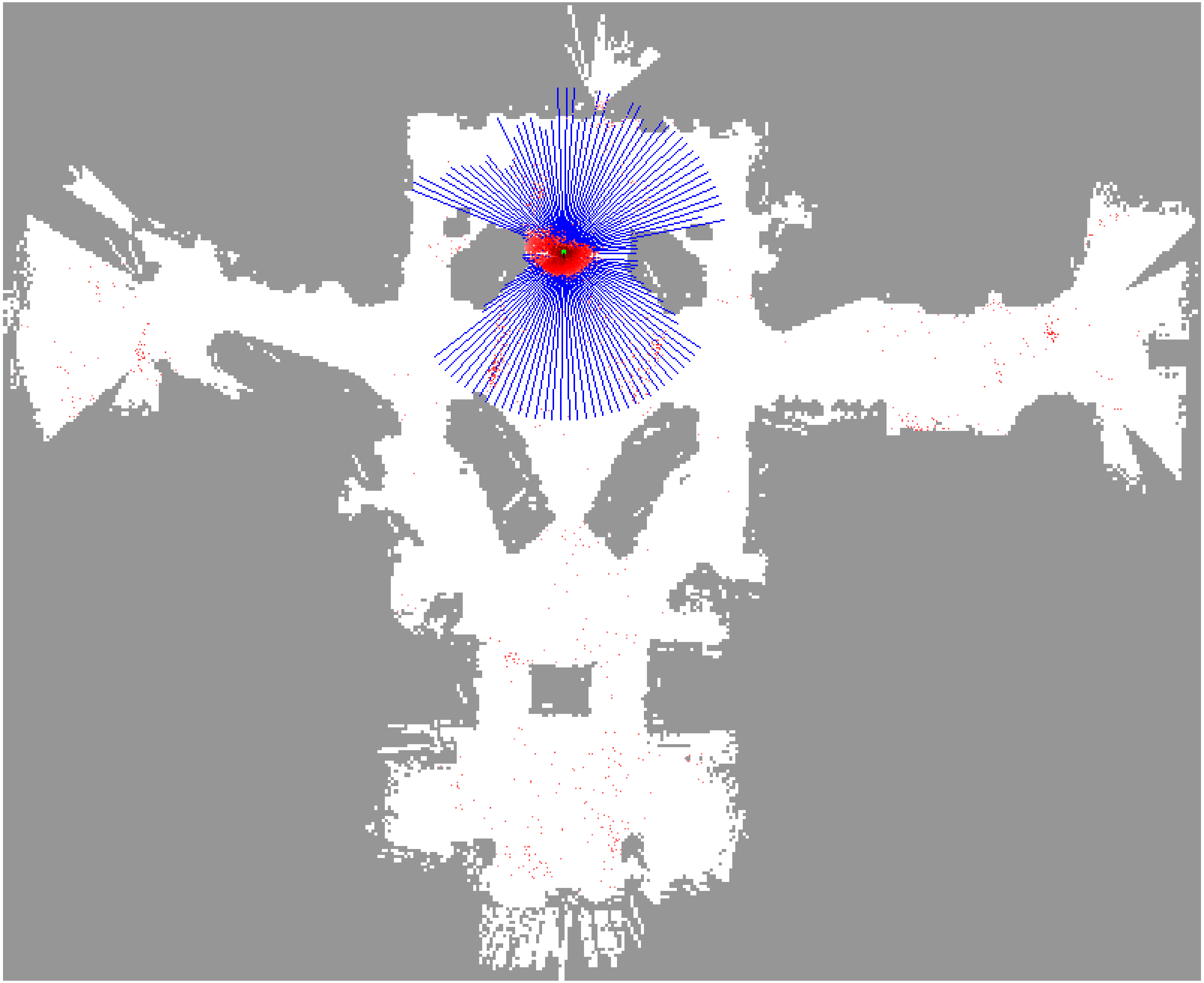


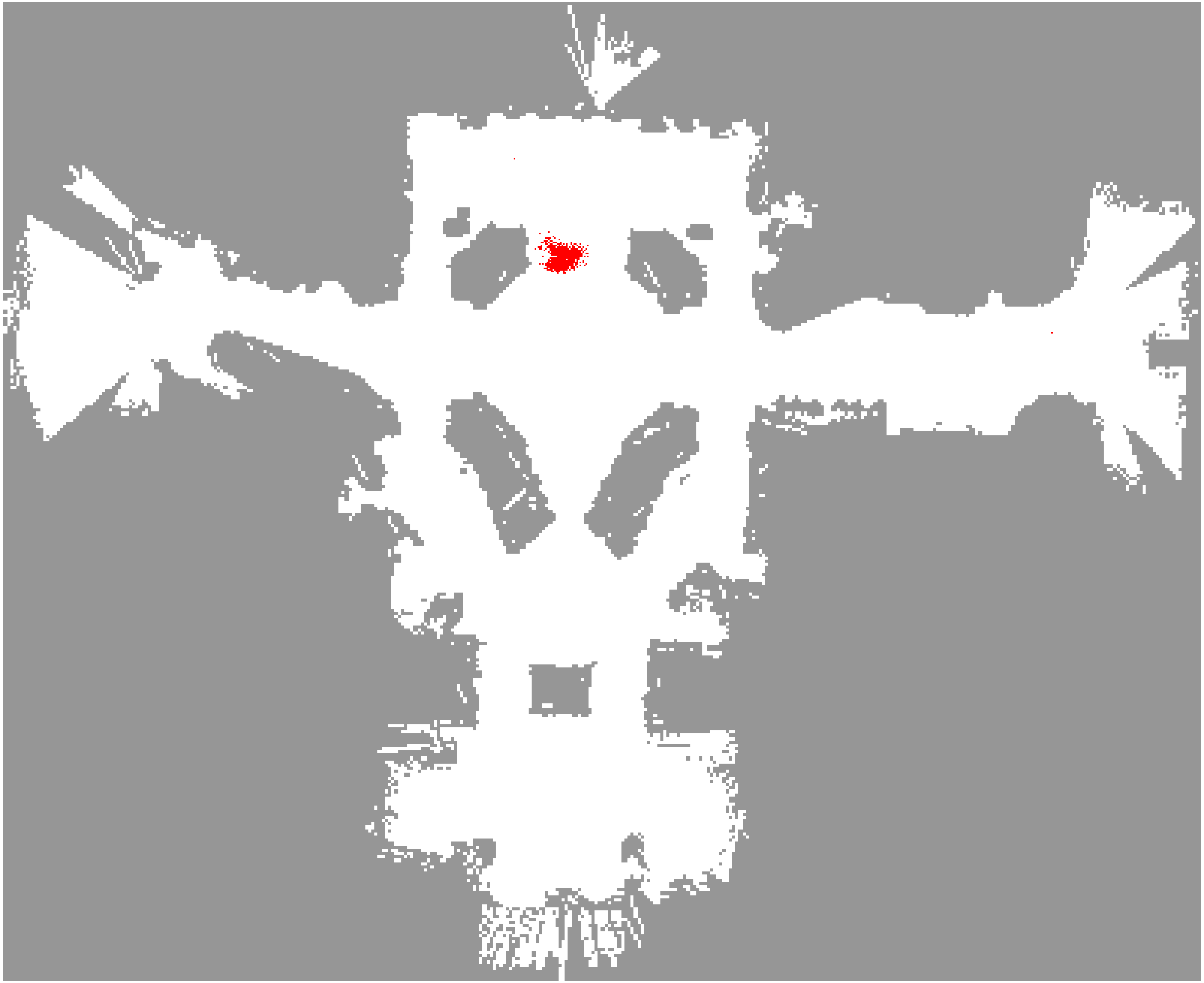


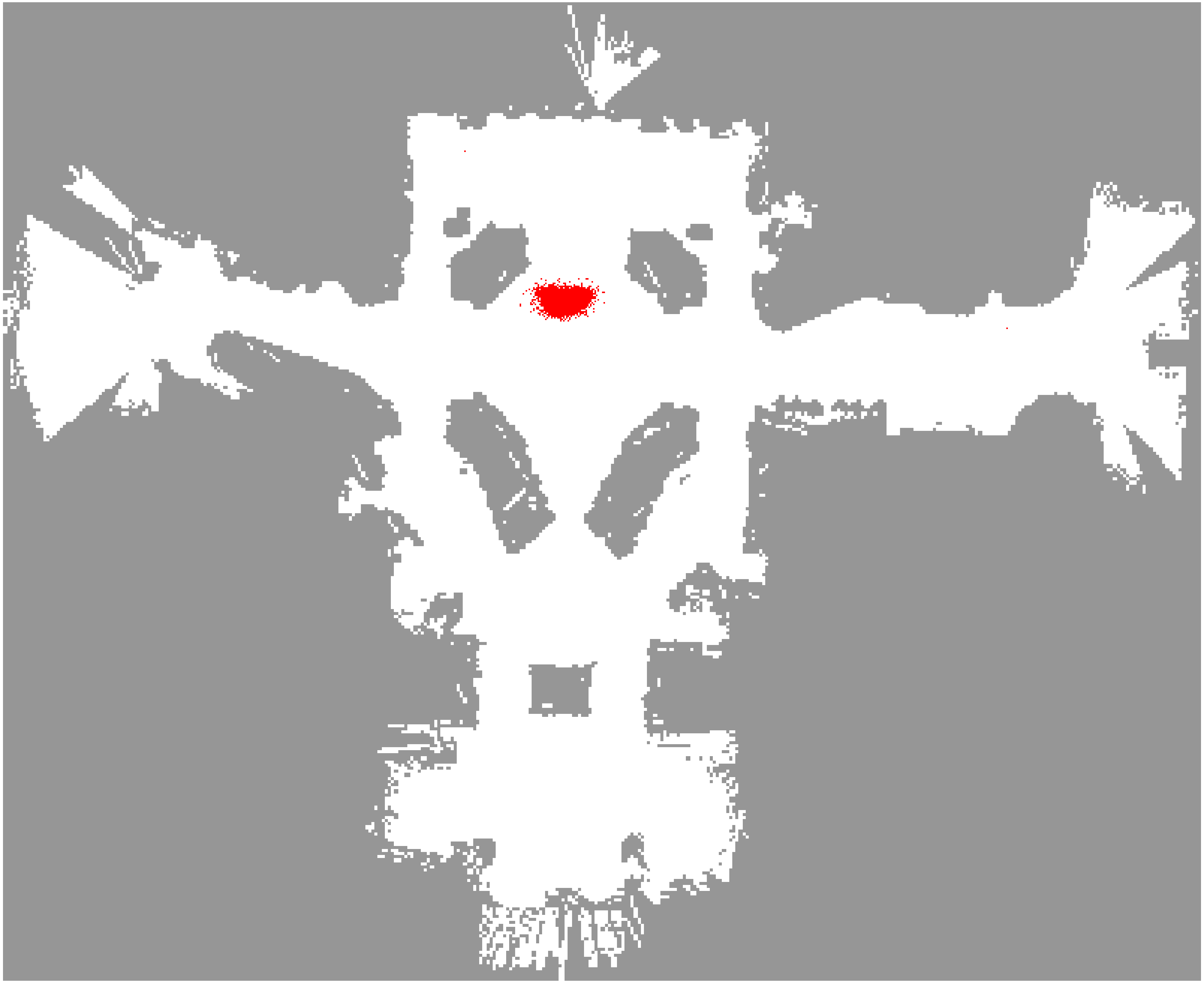


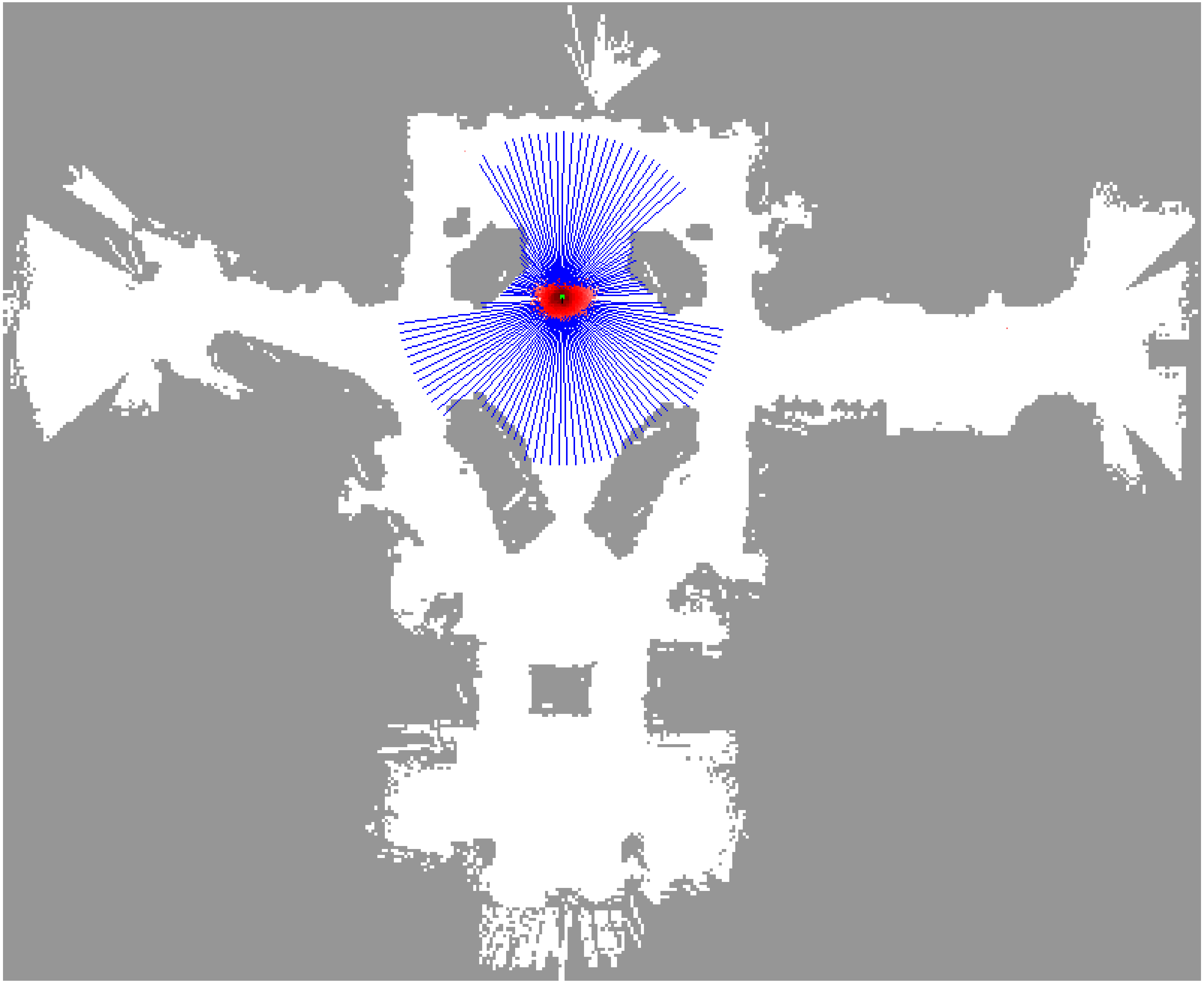


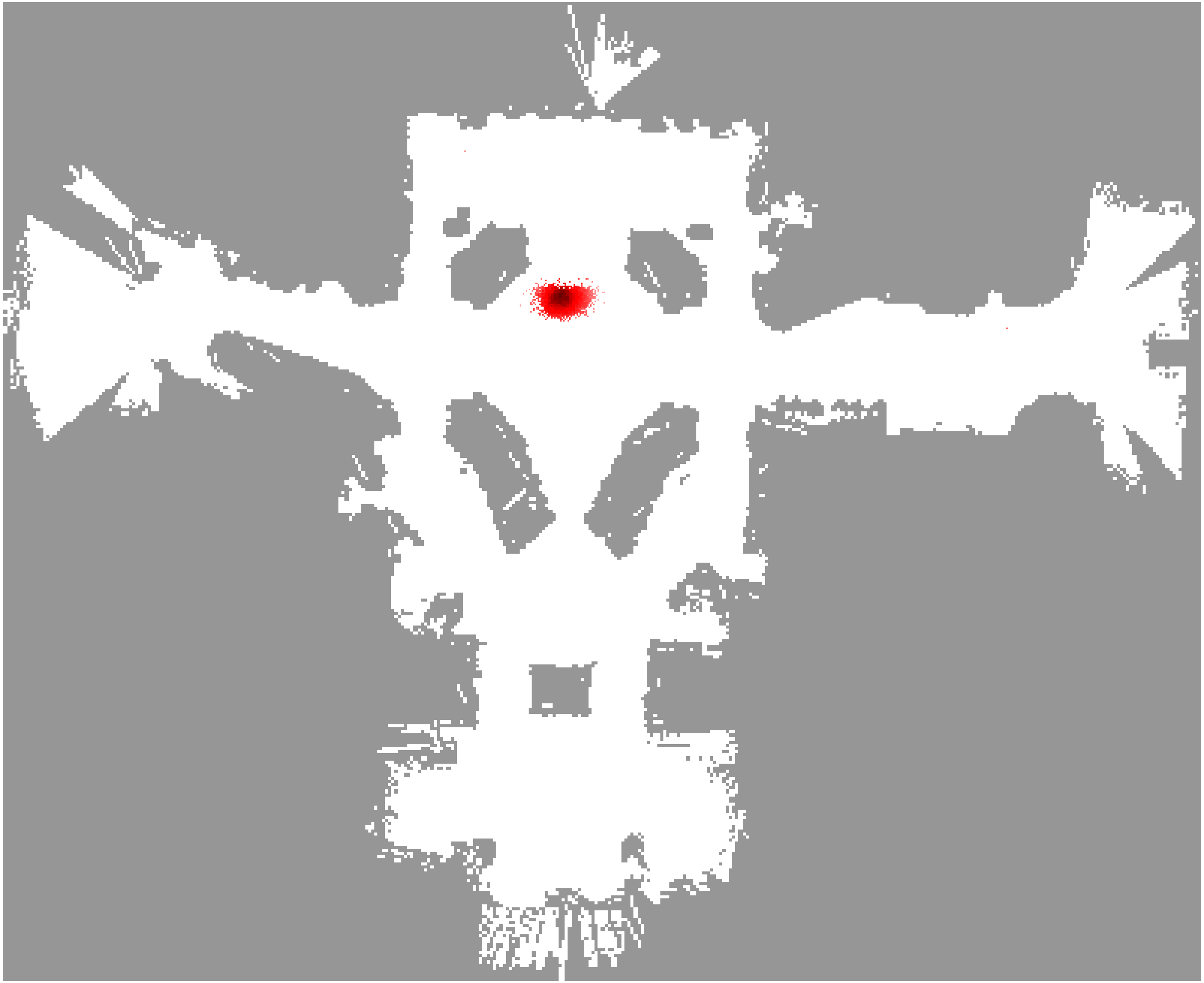


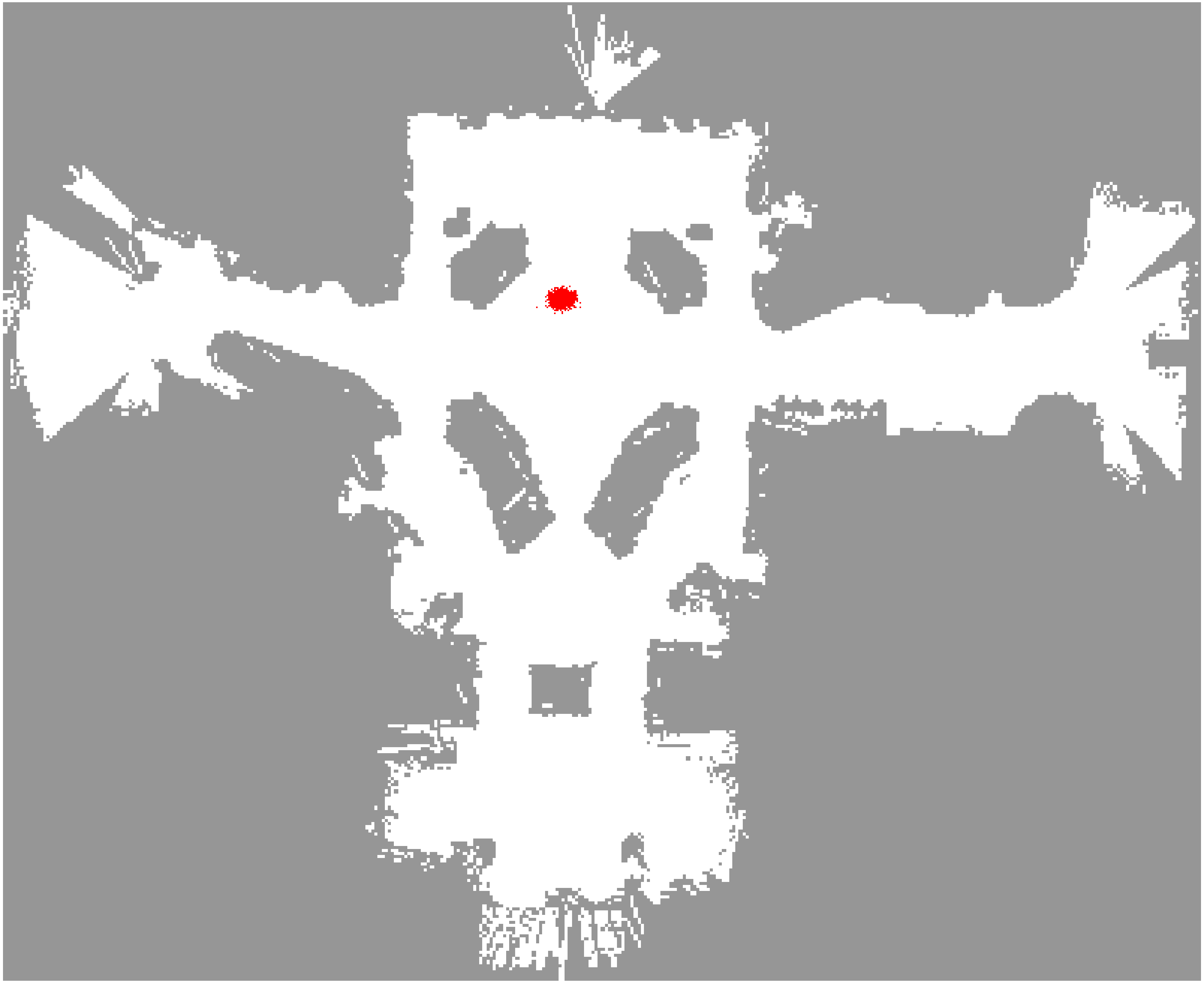


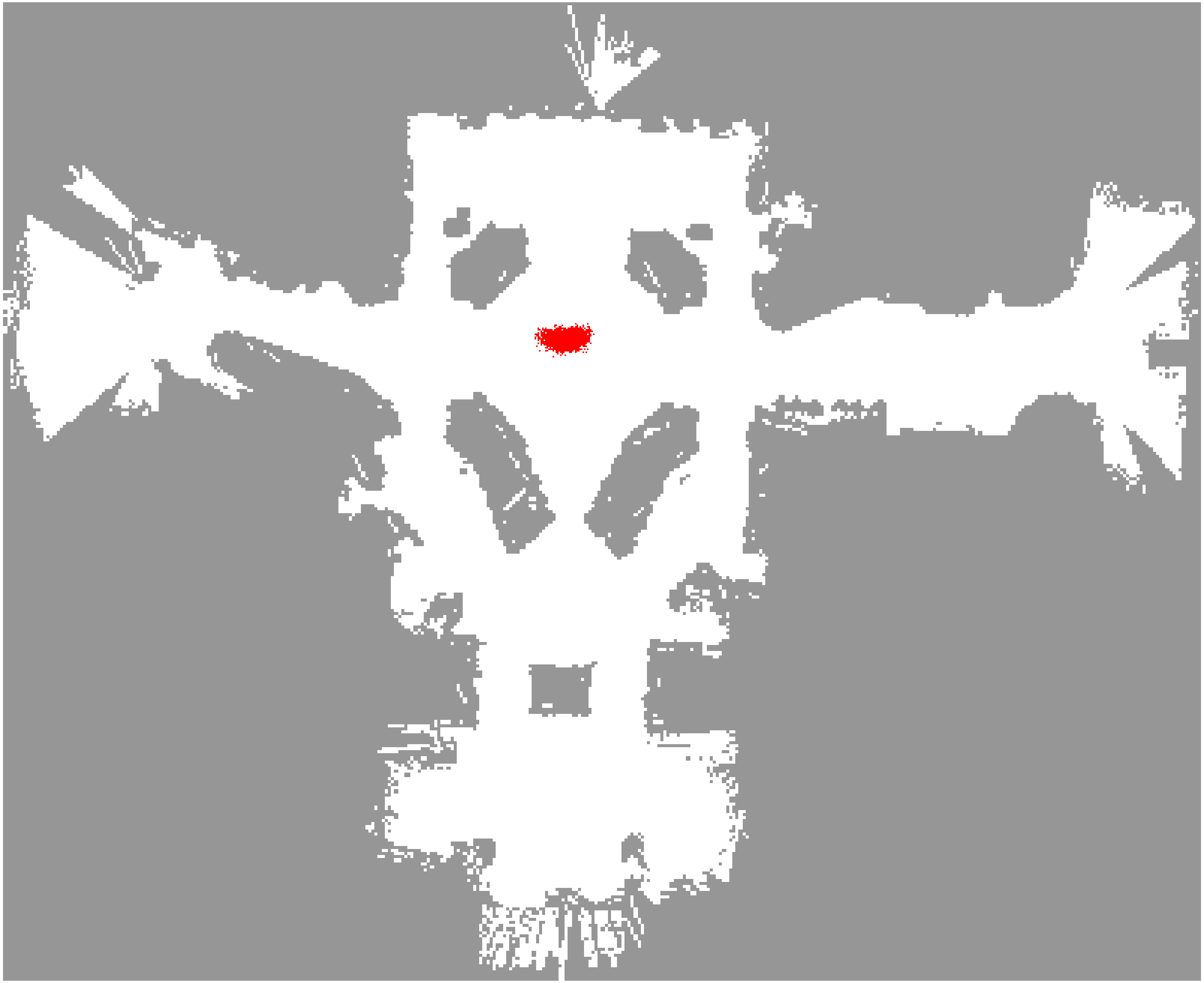


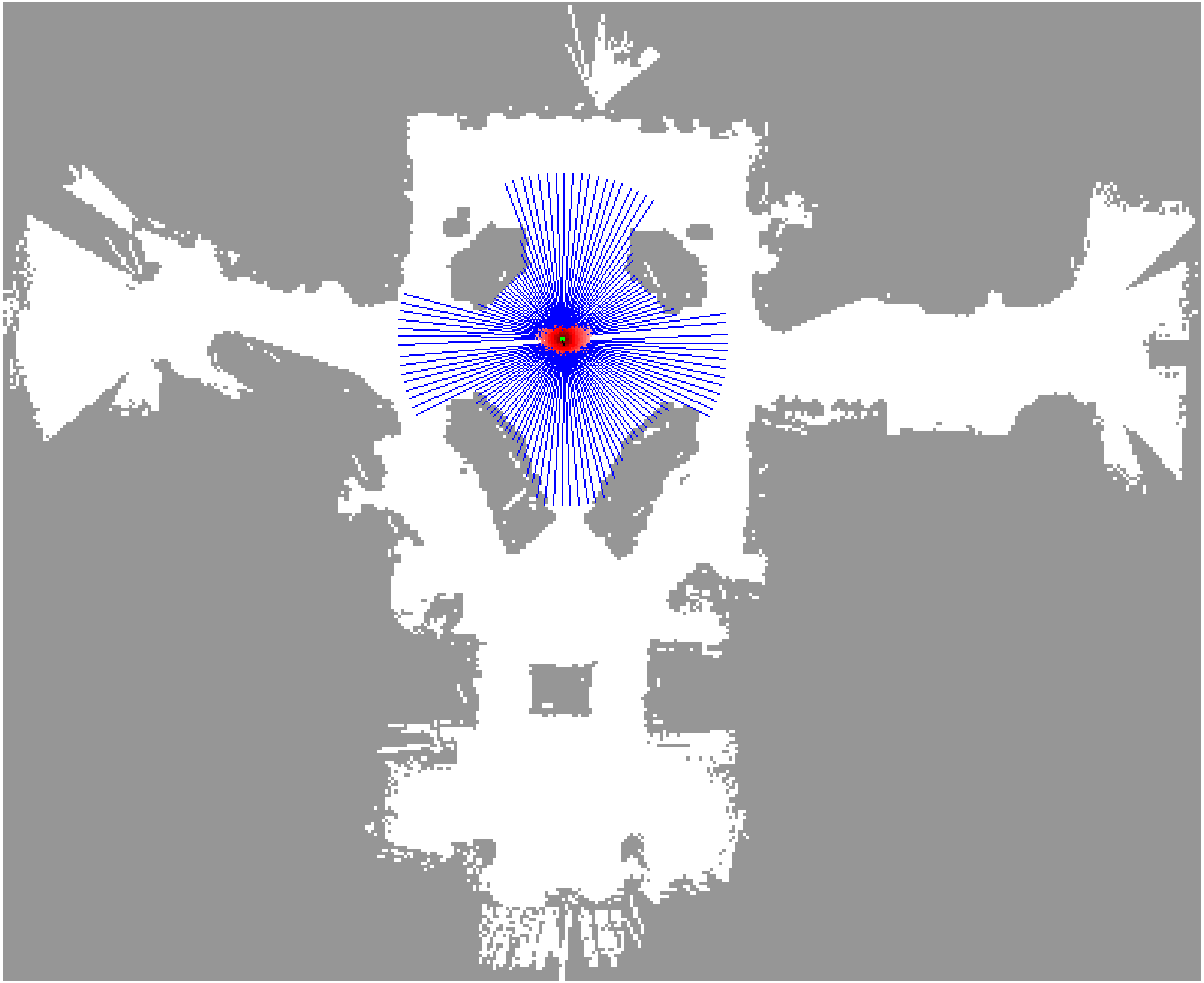


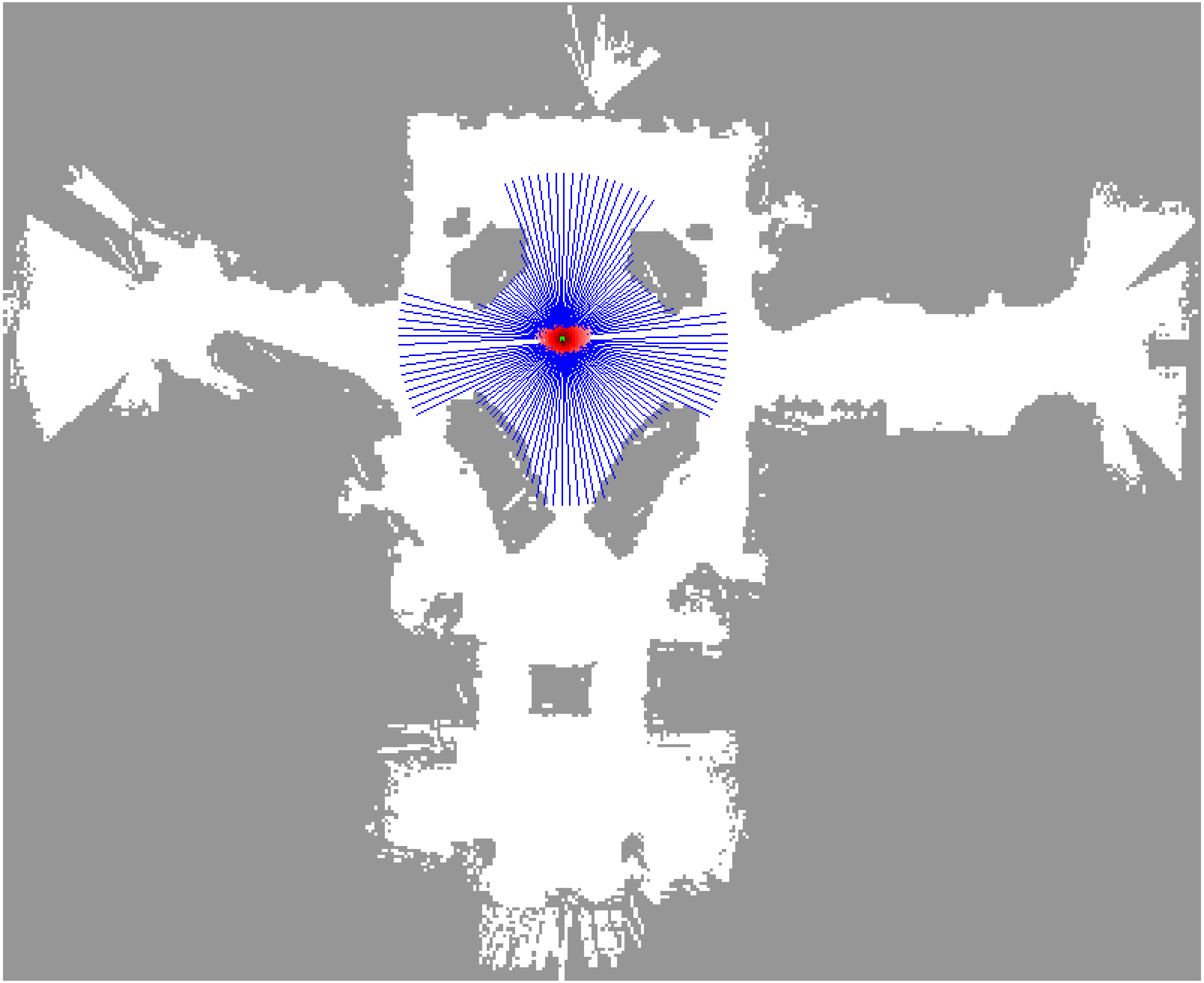








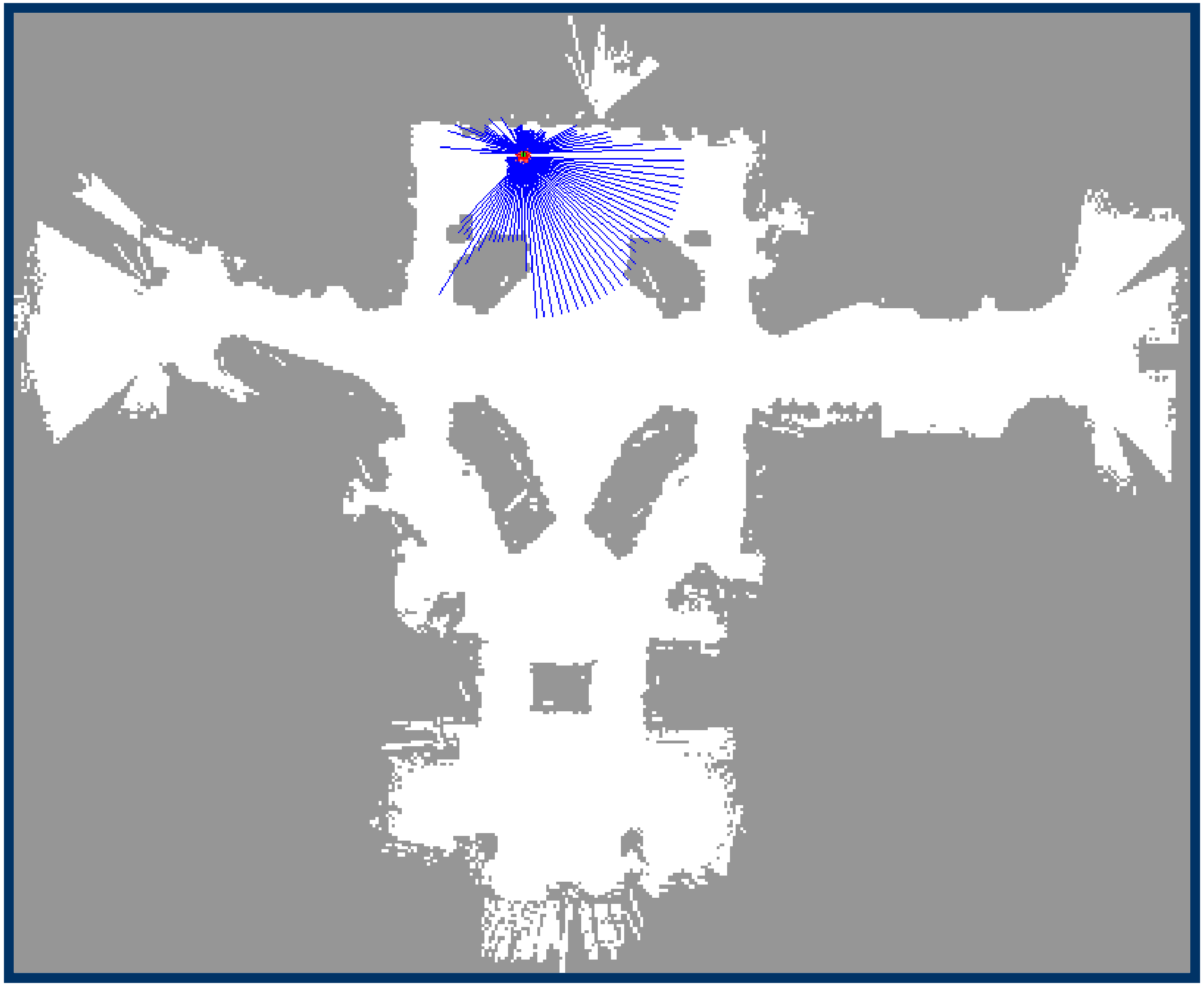




Recovery from Failure

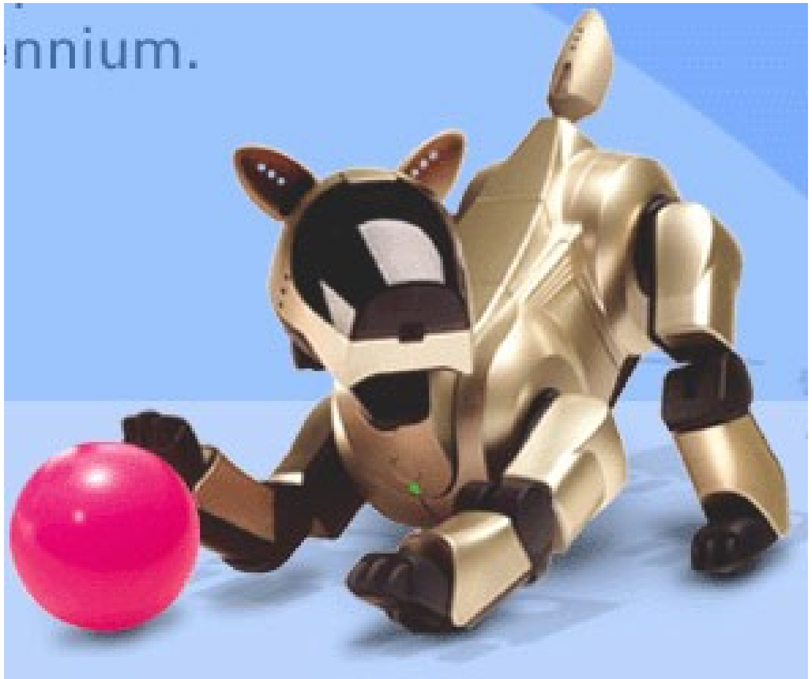
- **Problem:**
  - Samples are highly concentrated during tracking
  - True location is not covered by samples if position gets lost
- **Solutions:**
  - Add uniformly distributed samples [Fox et al., 99]
  - Draw samples according to observation density [Lenser et al.,00; Thrun et al., 00]

MCL: Recovery from Failure



[Fox et al. 99]

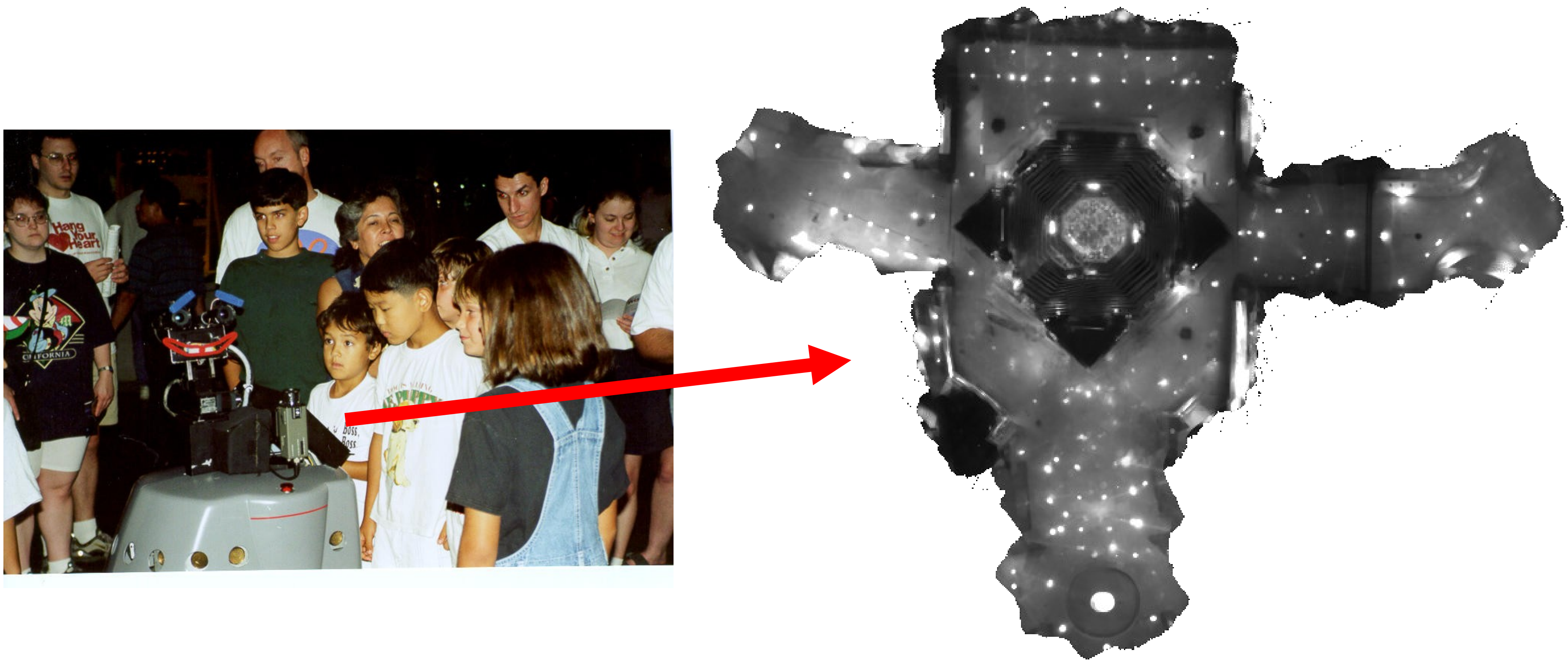
The RoboCup Challenge



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

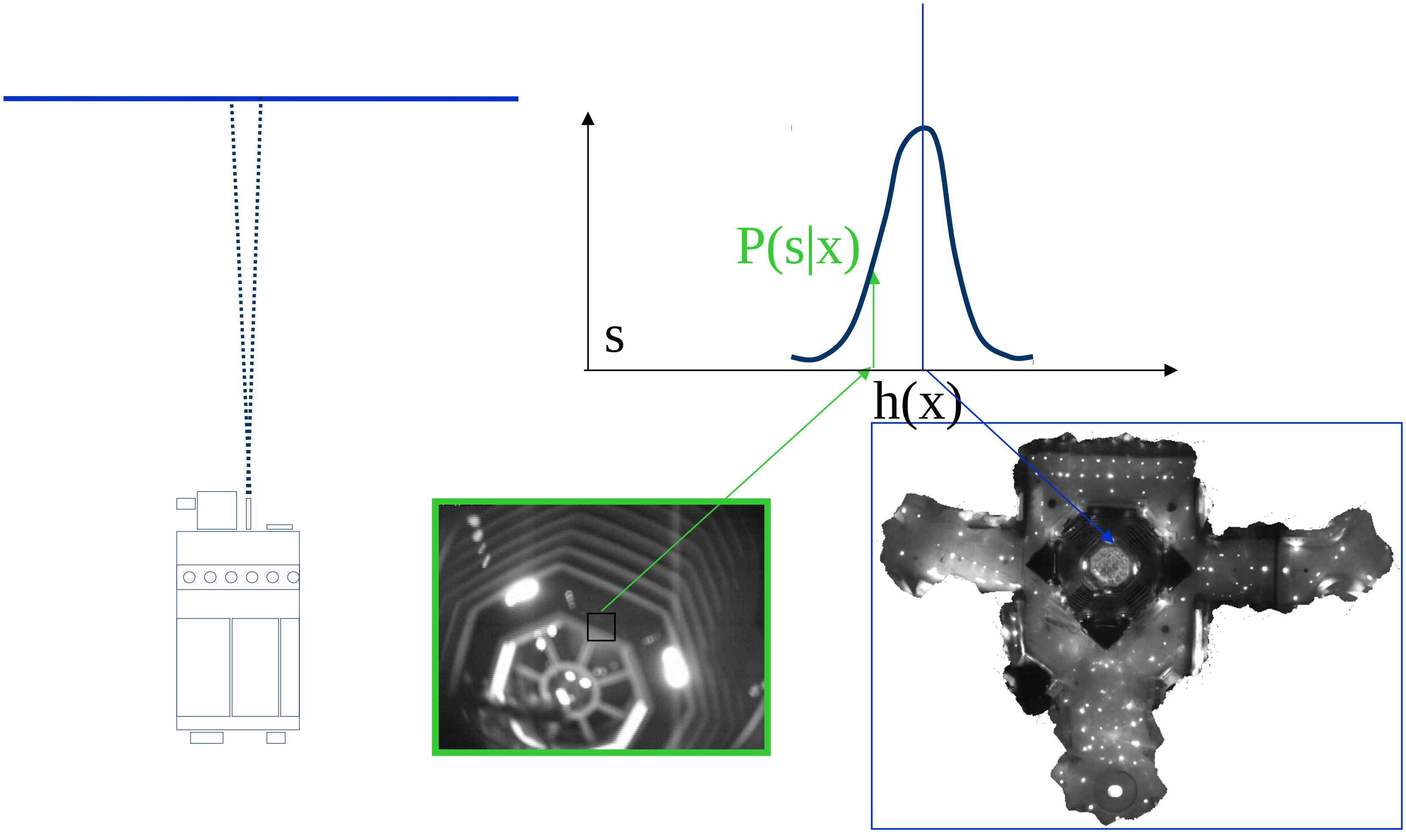
  [Lenser et al. 00]

Using Ceiling Maps for Localization



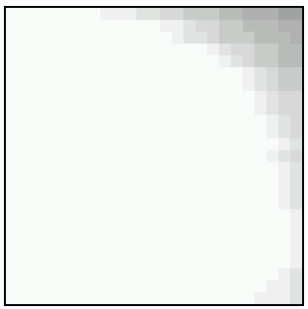
[Dellaert et al. 99]

Vision-based Localization

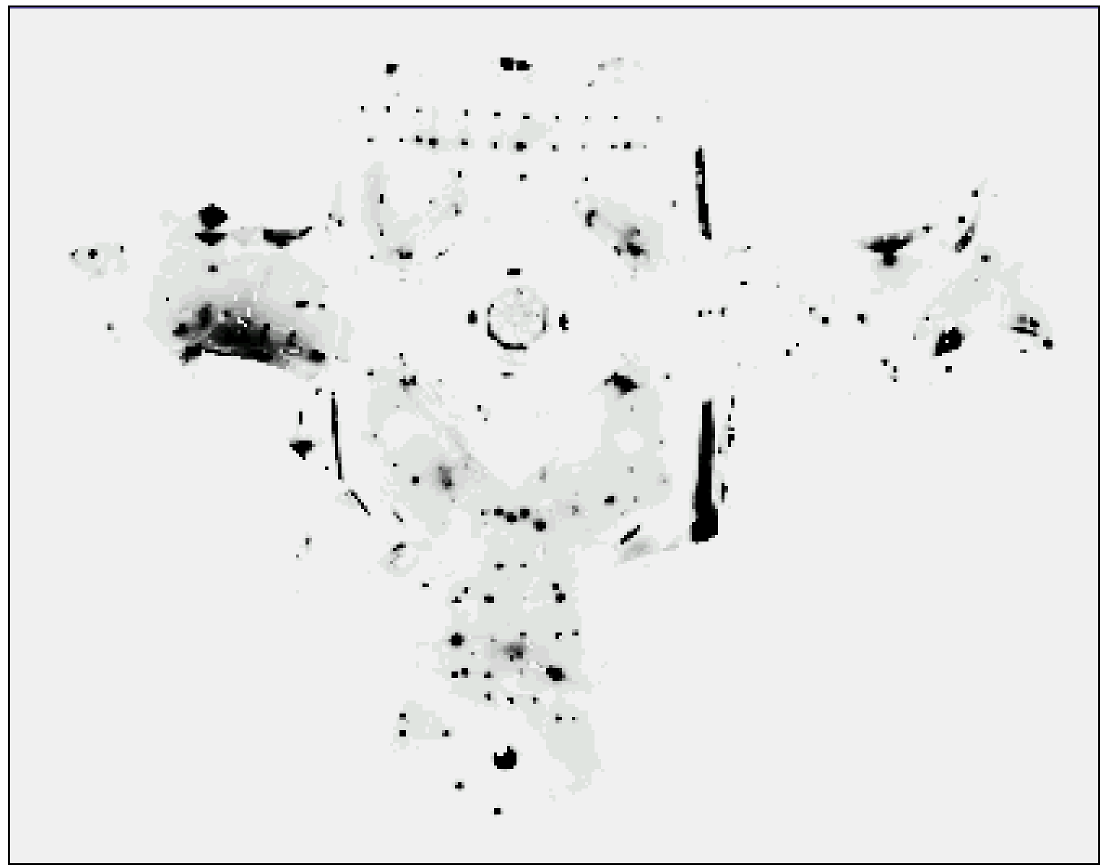


Under a Light

Measurement:

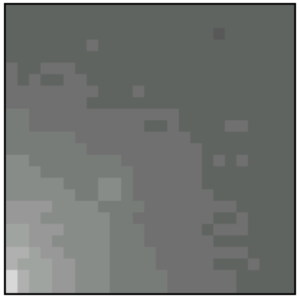


Resulting density:

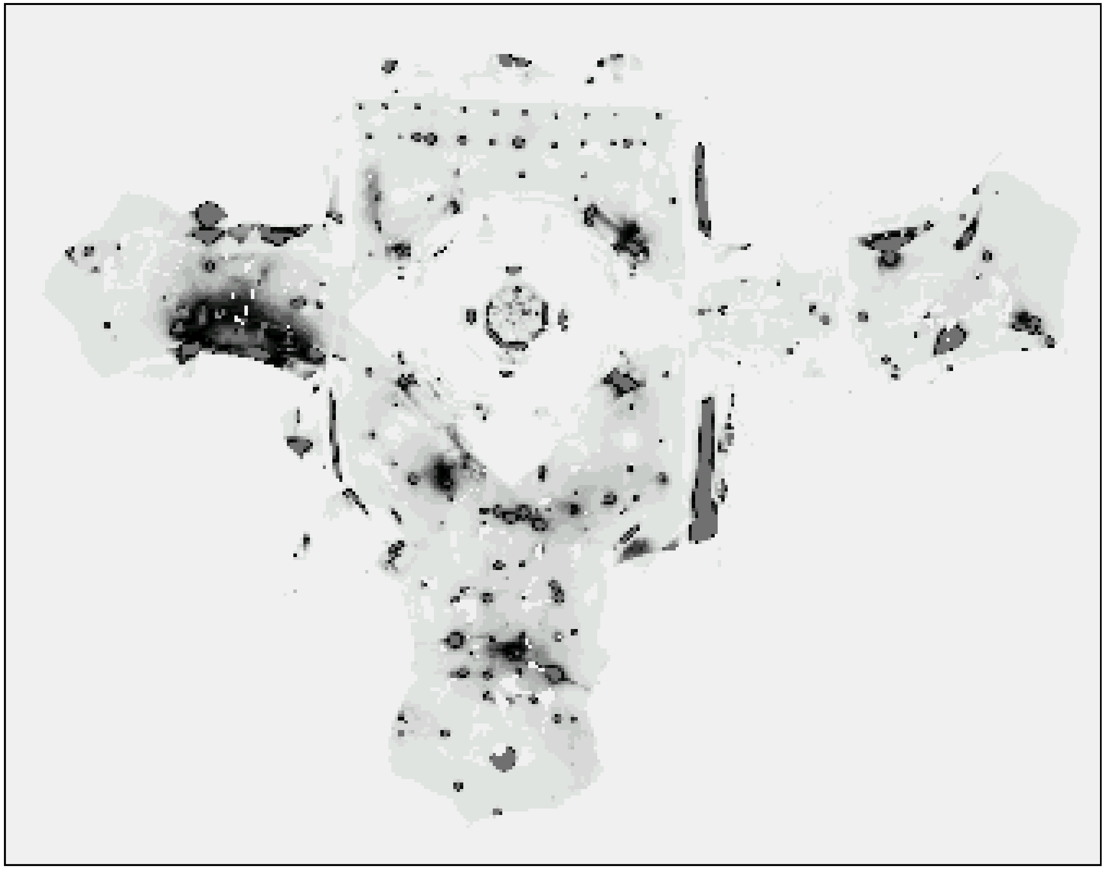


Next to a Light

Measurement:



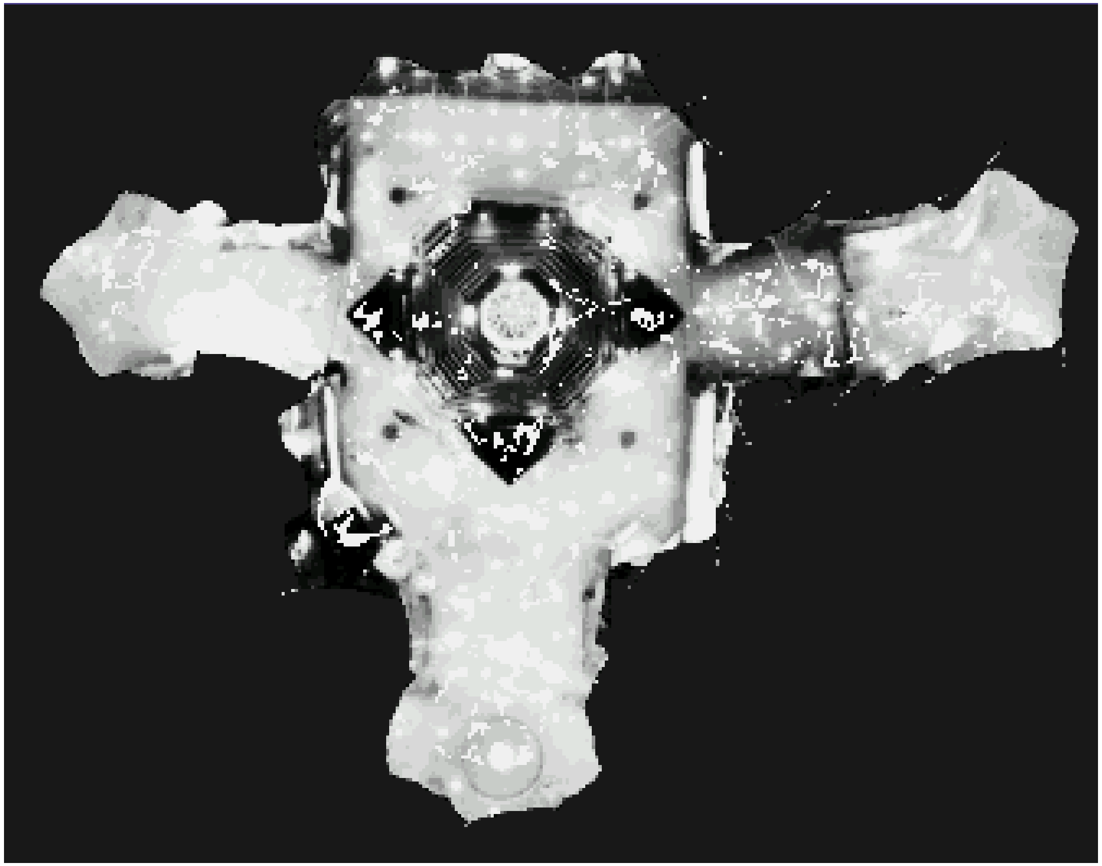
Resulting density:



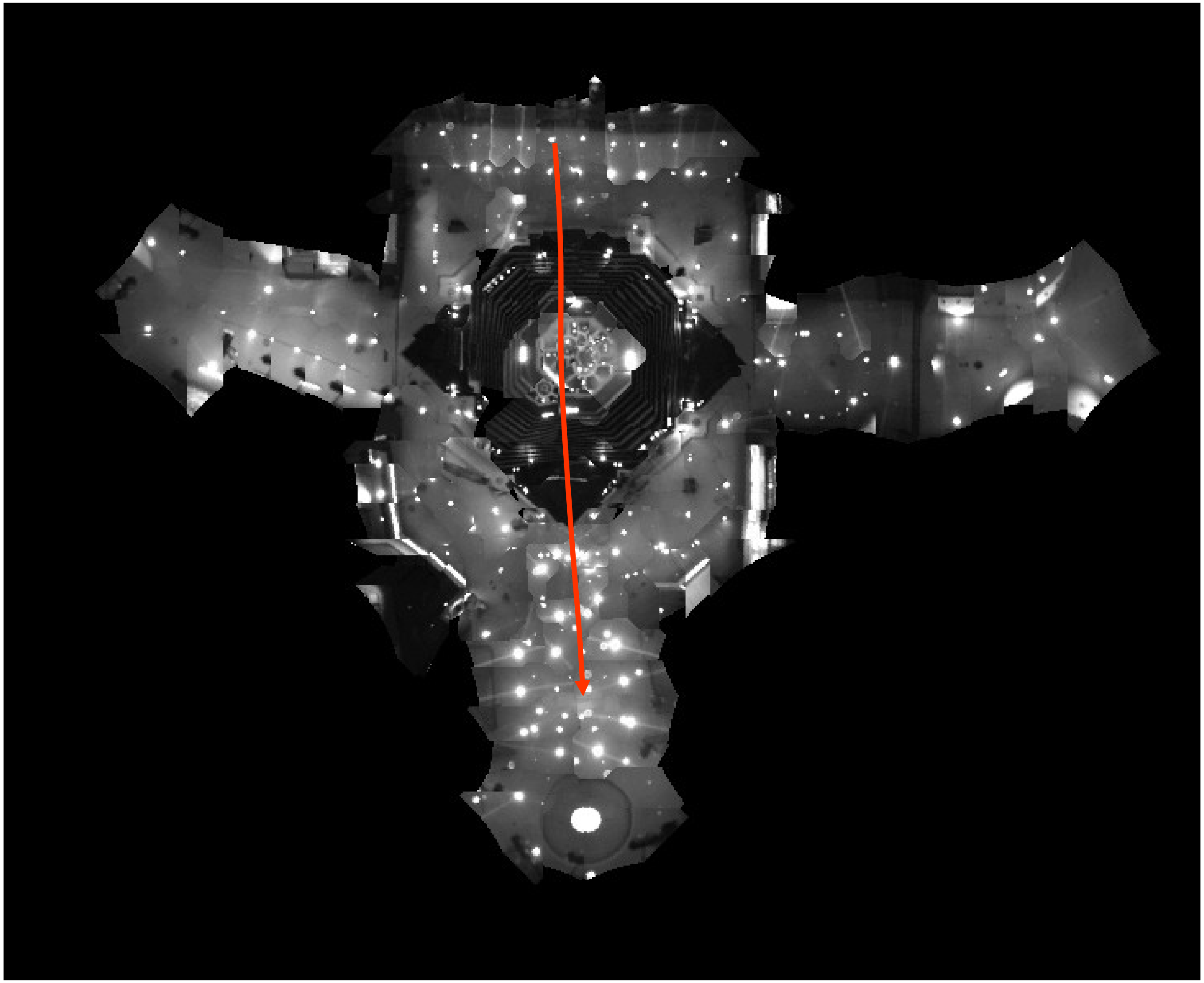
Elsewhere

Measurement:

Resulting density:

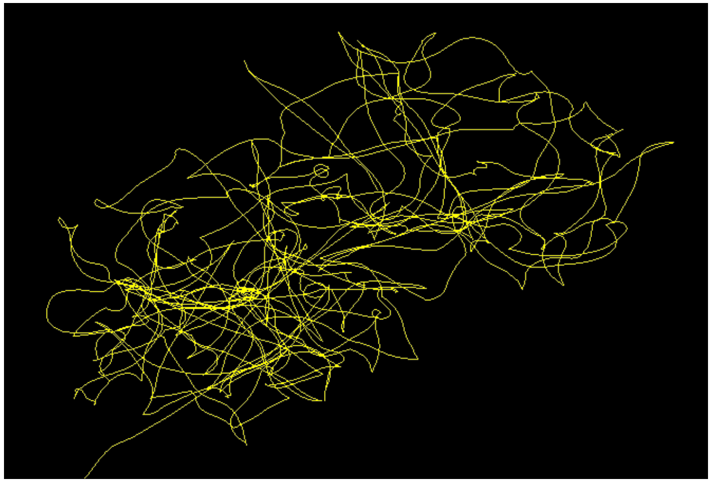


MCL: Global Localization Using Vision

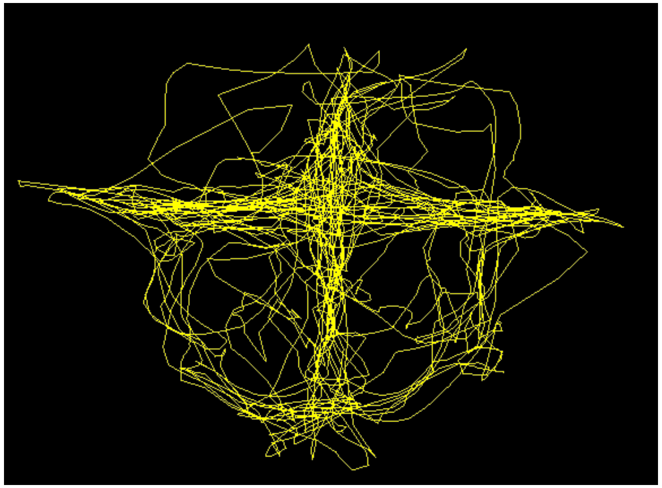


Vision-based Localization

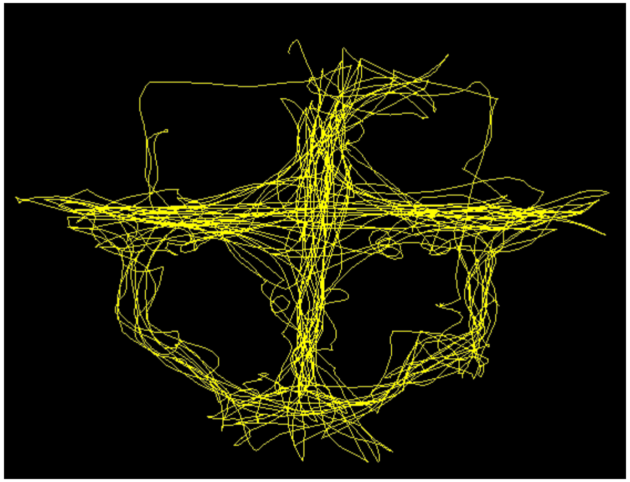
Odometry only:



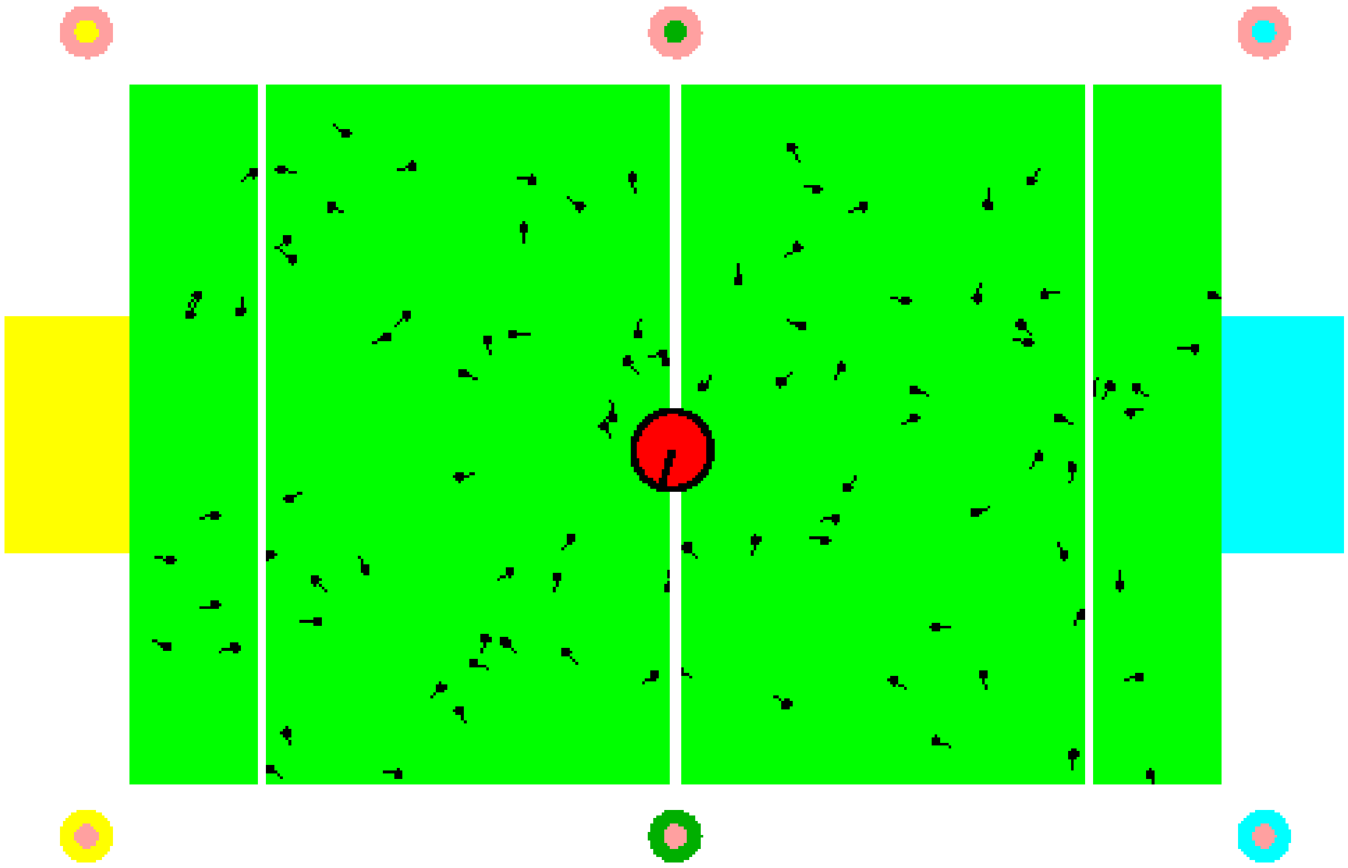
Vision:



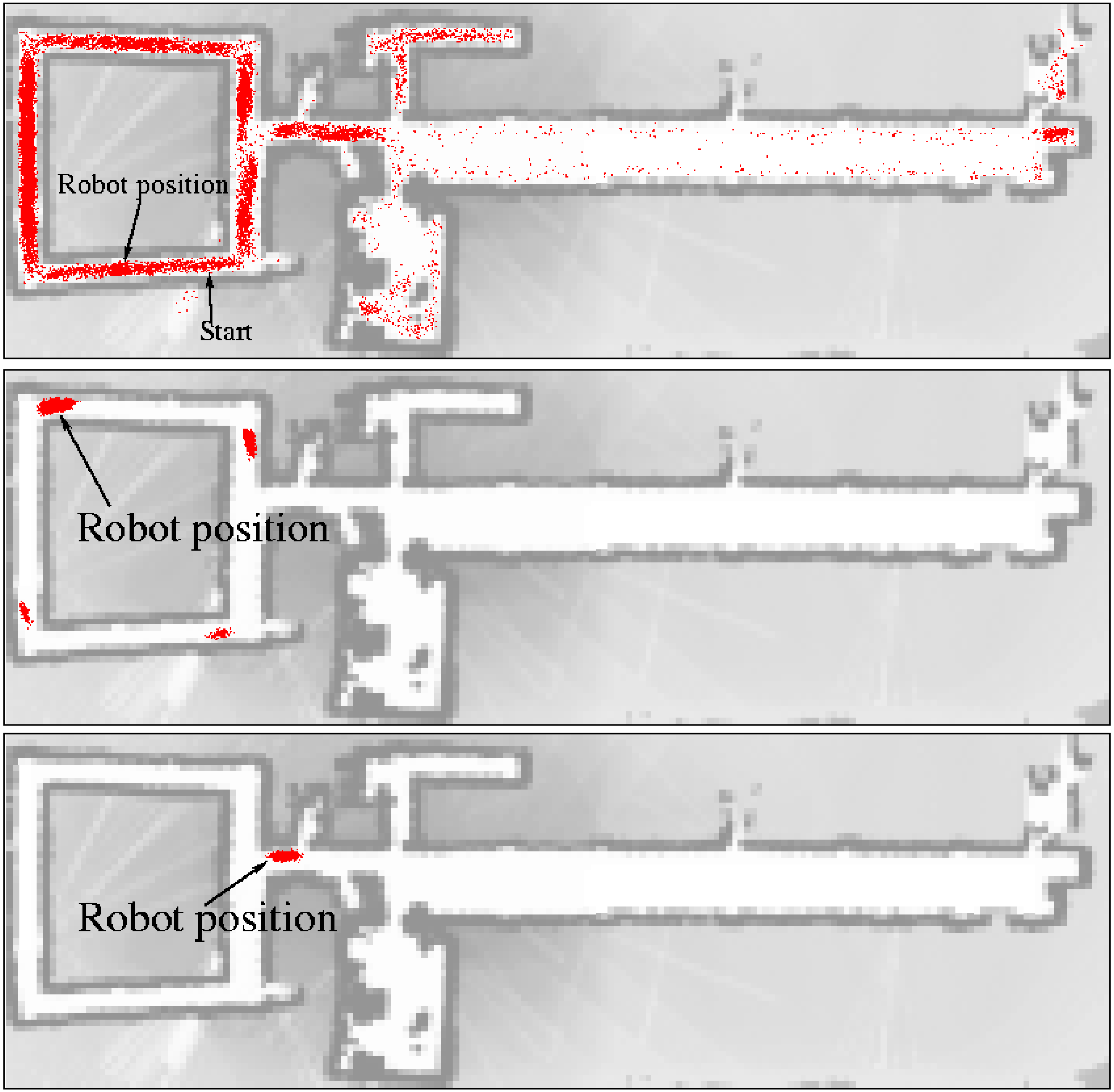
Laser:



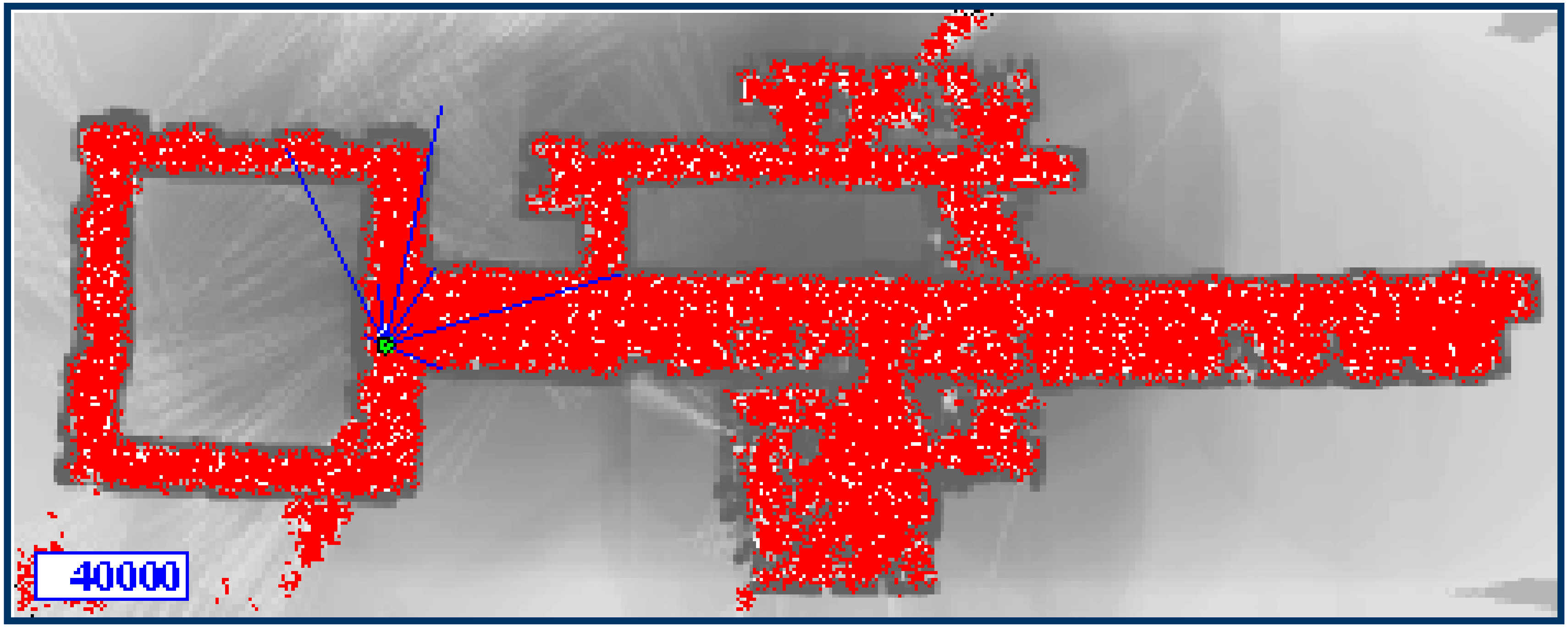
Localization for AIBO robots



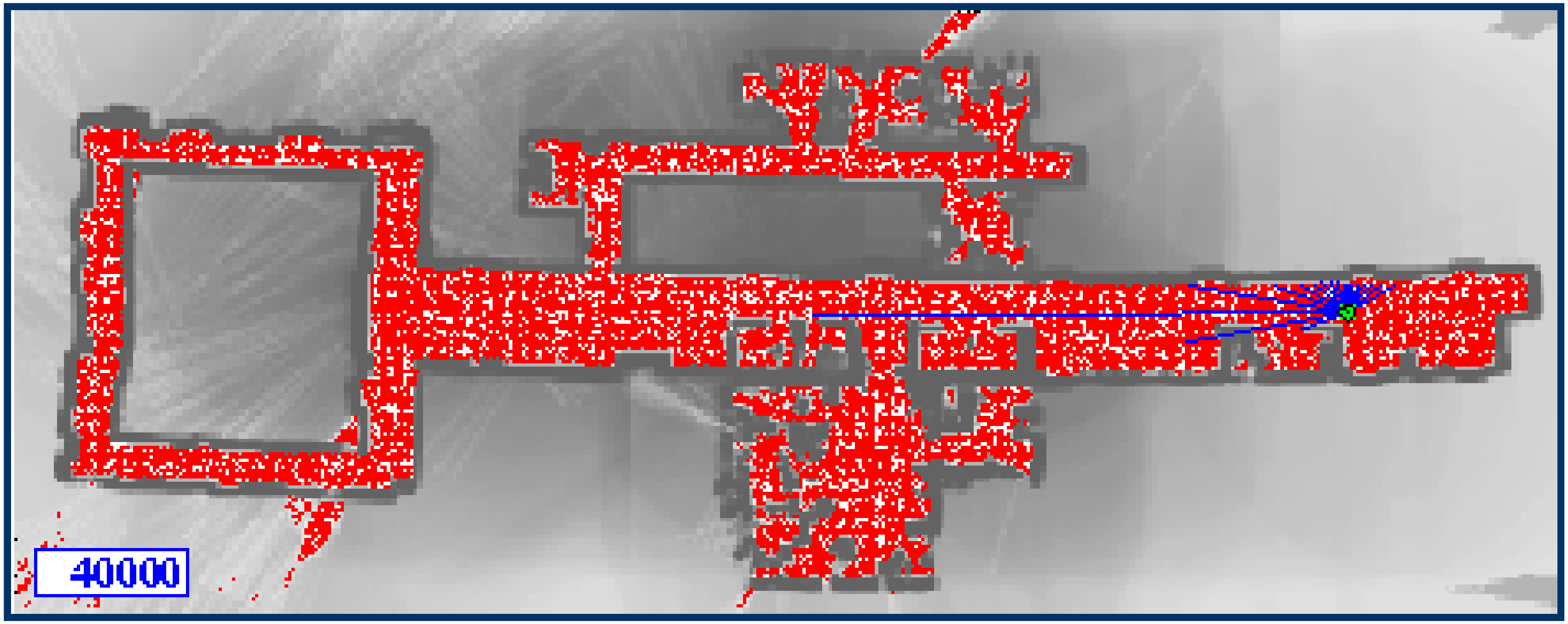
Adaptive Sampling



MCL: Adaptive Sampling (Sonar)

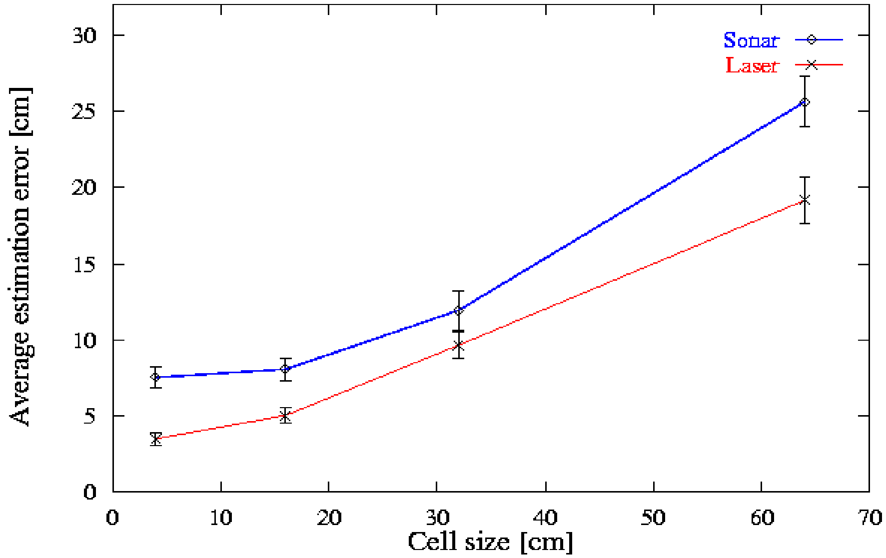


MCL: Adaptive Sampling (Laser)

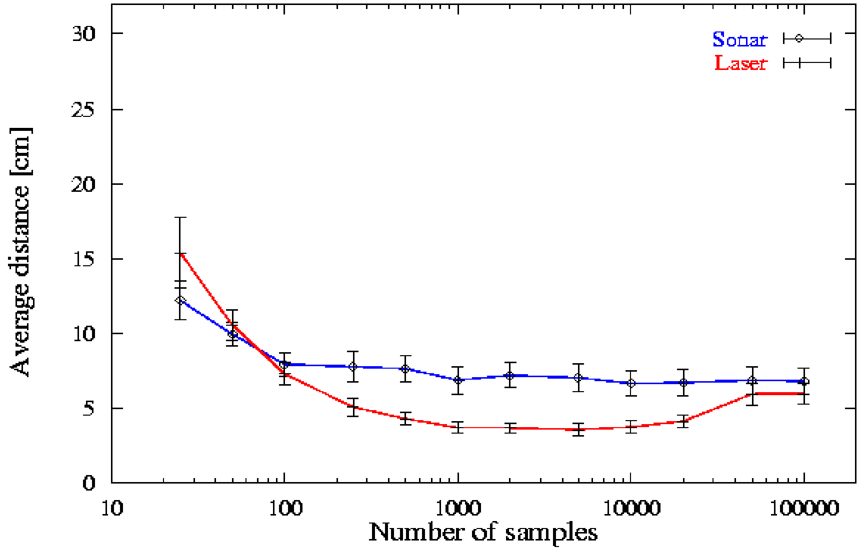


Performance Comparison

Grid-based localization



Monte Carlo localization



Particle Filters for Robot Localization (Summary)

- Approximate Bayes Estimation/Filtering
  - Full posterior estimation
  - Converges in $O(1/\sqrt{\#samples})$ [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-hypothesis 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	-	+	+	++	+/++

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