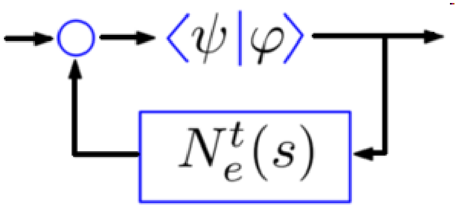


EE-565: Mobile Robotics

Non-Parametric Filters

Module 2, Lecture 5

Dr Abubakr Muhammad
Assistant Professor
Electrical Engineering, LUMS
Director, CYPHYNETS Lab
http://cyphynets.lums.edu.pk

Resources

Course material from

- Stanford CS-226 (Thrun) [slides]
- KAUST ME-410 (Abubakr, 2011)
- LUMS EE-662 (Abubakr, 2013)

http://cyphynets.lums.edu.pk/index.php/Teaching

Textbooks

- Probabilistic Robotics by Thrun et al.
- Principles of Robot Motion by Choset et al.

Part 1.

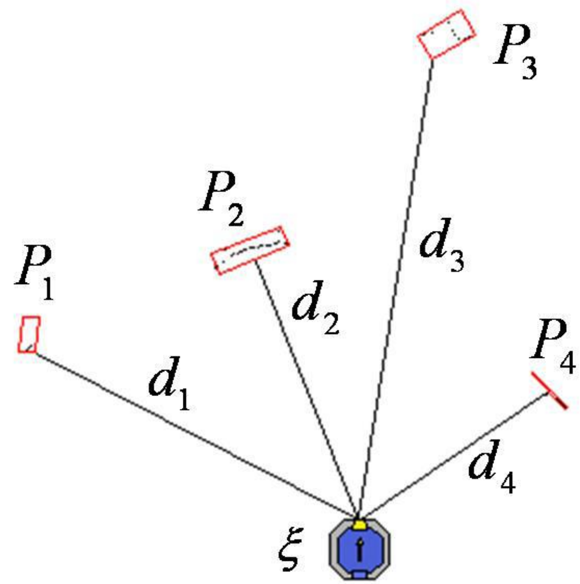
BAYESIAN PHILOSOPHY FOR STATE ESTIMATION

State Estimation Problems

- What is a state?
- Inferring "hidden states" from observations
- What if observations are noisy?
- More challenging, if state is also dynamic.
- Even more challenging, if the state dynamics are also noisy.

State Estimation Example: Localization

- Definition. Calculation of a mobile robot's position / orientation relative to an external reference system

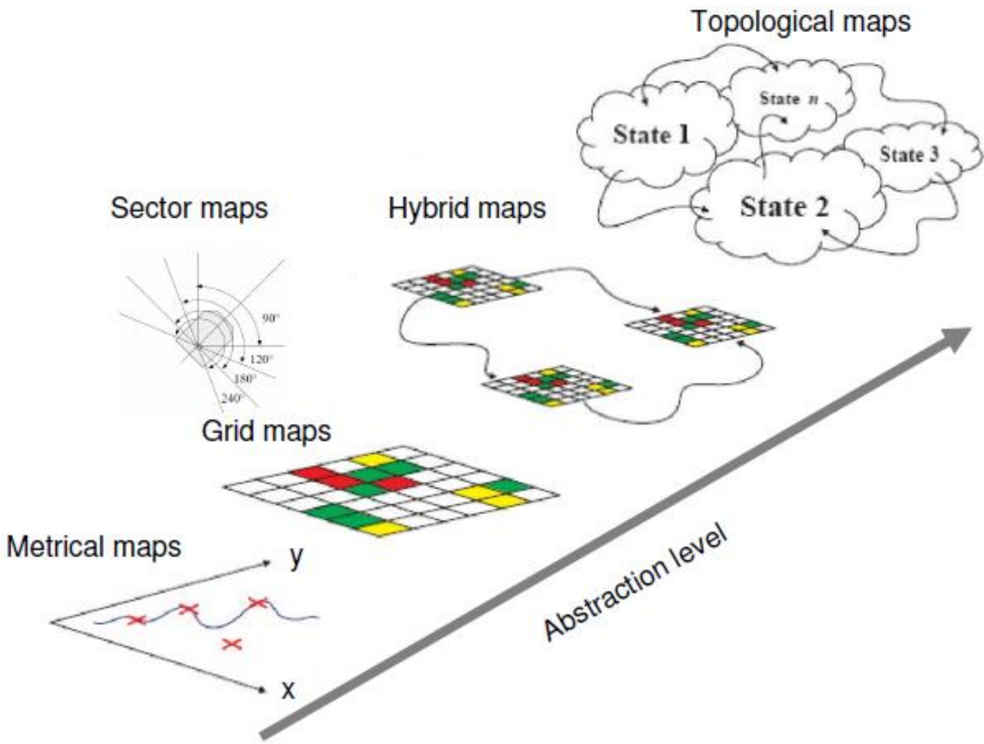


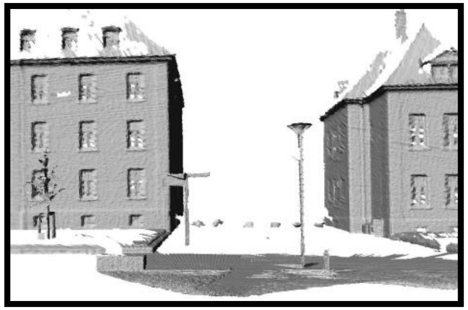
- Usually world coordinates serve as reference
- Basic requirement for several robot functions:
  - approach of target points, path following
  - avoidance of obstacles, dead-ends
  - autonomous environment mapping

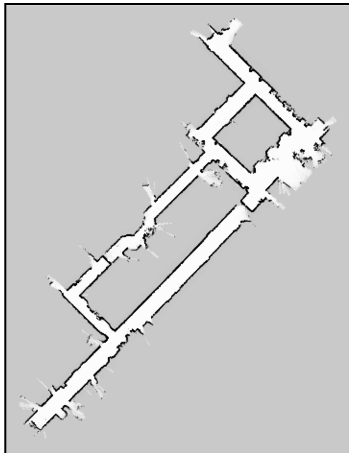
Requires accurate maps !!

State Estimation Example: Mapping

- Objective: Store information outside of sensory horizon
- Map provided a-priori or can be online
- Types
  - world-centric maps
    navigation, path planning
  - robot-centric maps
    pilot tasks (e. g. collision avoidance)
- Problem: inaccuracy due to sensor systems

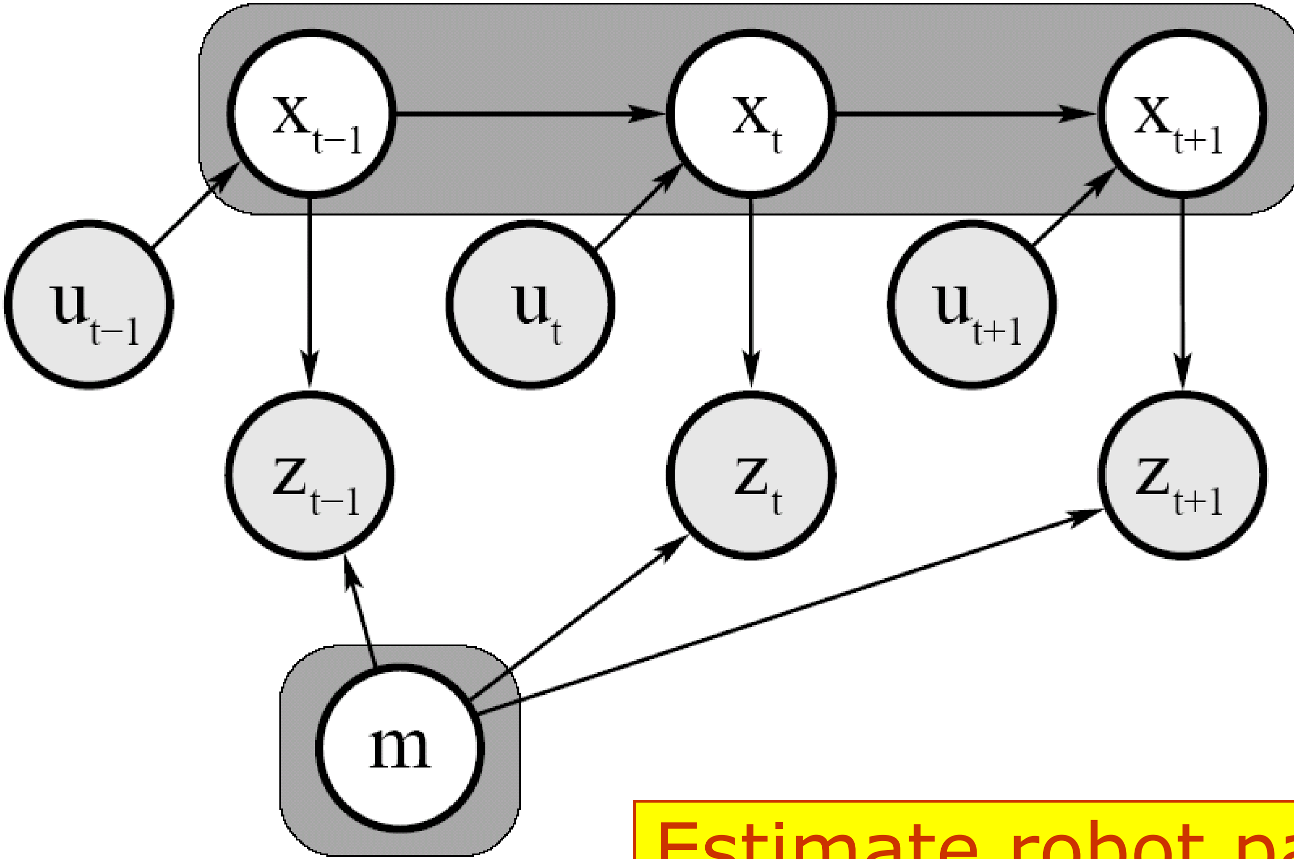






Requires accurate localization!!

Probabilistic Graphical Models

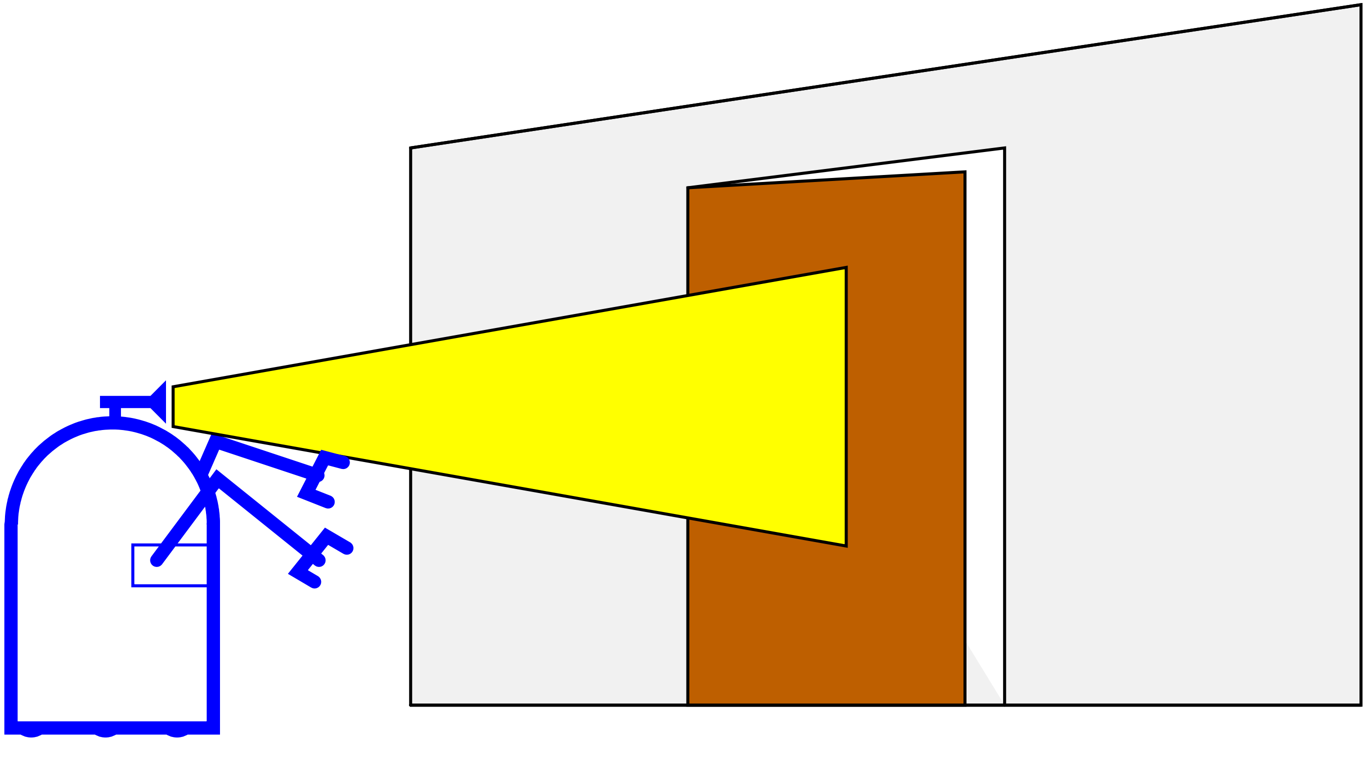


Estimate robot path and/or map!

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

Simple Example of State Estimation

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



Bayes Formula

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

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

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

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

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:

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

Example

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

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

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

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

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

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

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

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

$$= \eta_{1\dots n} \Big[\prod_{i=1\dots n} P(z_i \mid x)\Big] P(x)$$

Example: Second Measurement

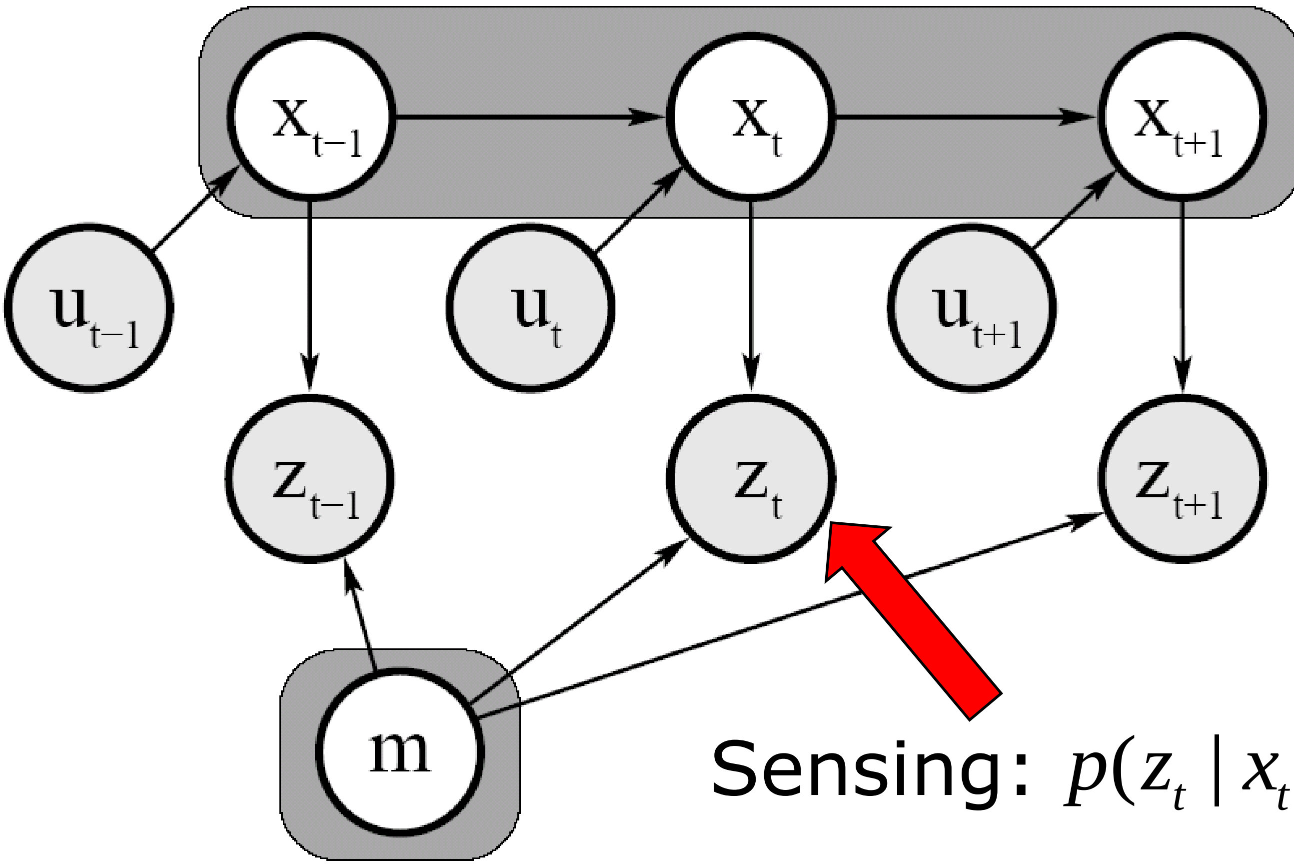
- $P(z_2|open) = 0.5$ $\qquad$ $P(z_2|\neg open) = 0.6$
- $P(open|z_1)=2/3$

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

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

z_2 lowers the probability that the door is open.

Probabilistic Graphical Models

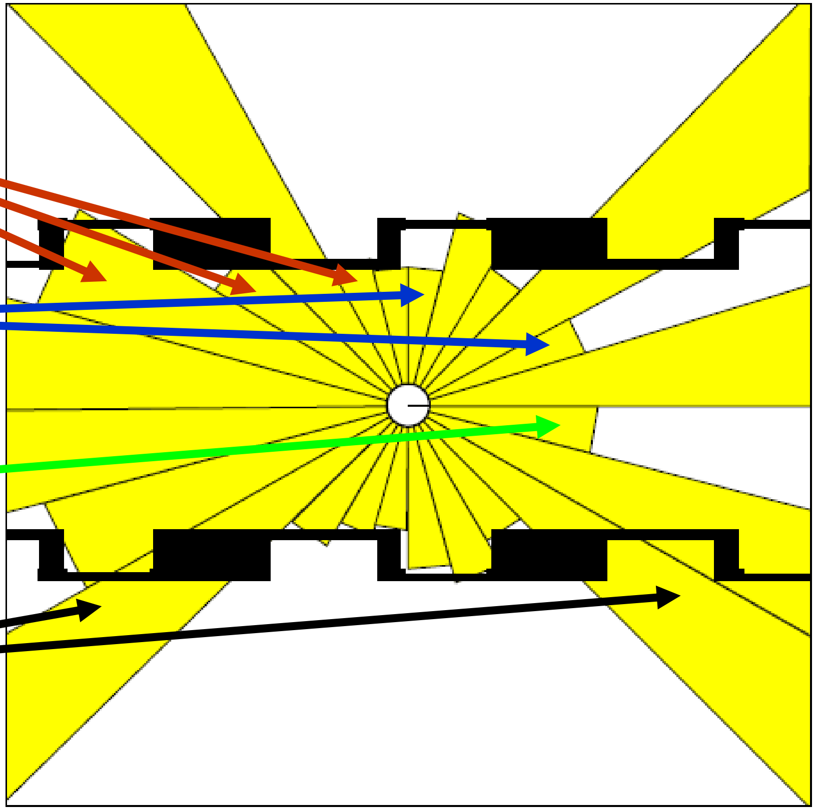


Sensing: $p(z_t \mid x_t, m)$

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

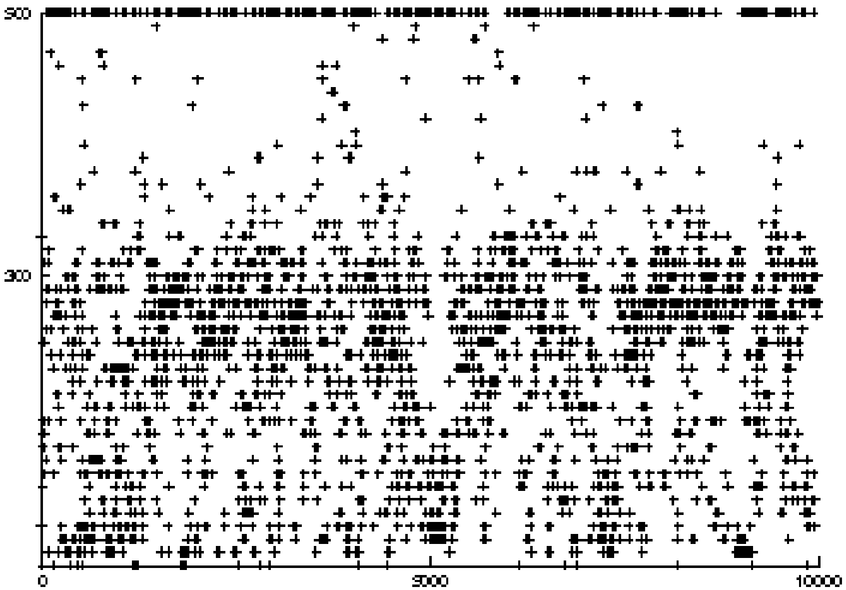
Typical Measurement Errors of an Range Measurements

1. Beams reflected by obstacles
2. Beams reflected by persons / caused by crosstalk
3. Random measurements
4. Maximum range measurements

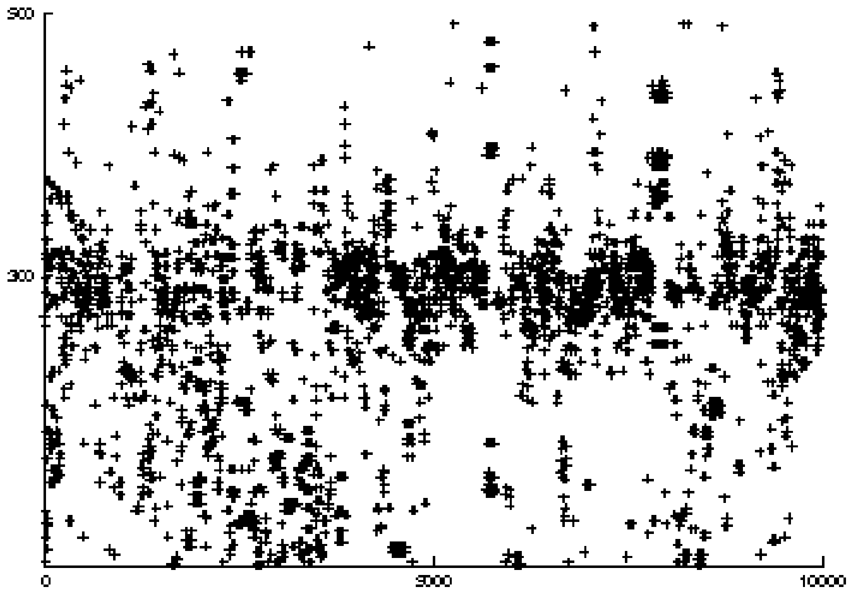


Raw Sensor Data

Measured distances for expected distance of 300 cm.

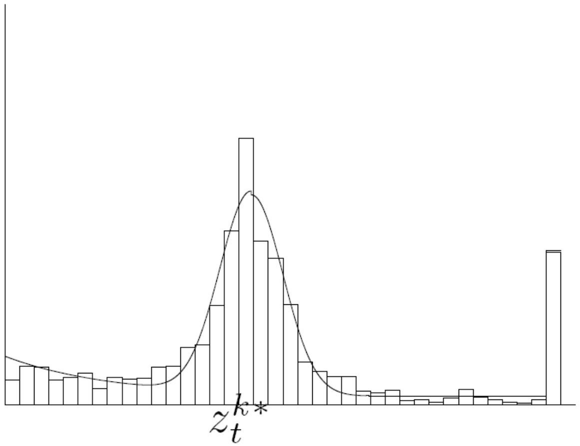


Sonar

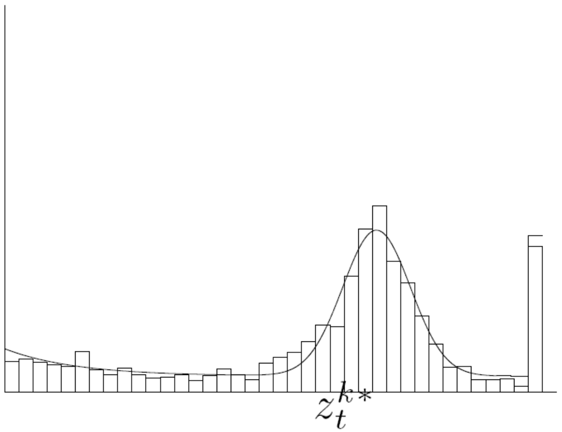


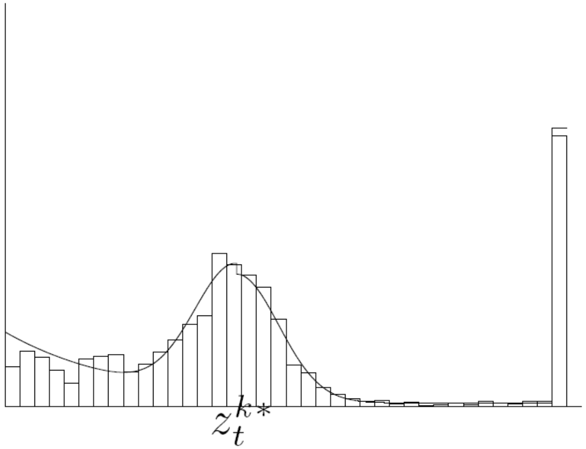
Laser

Approximation Results

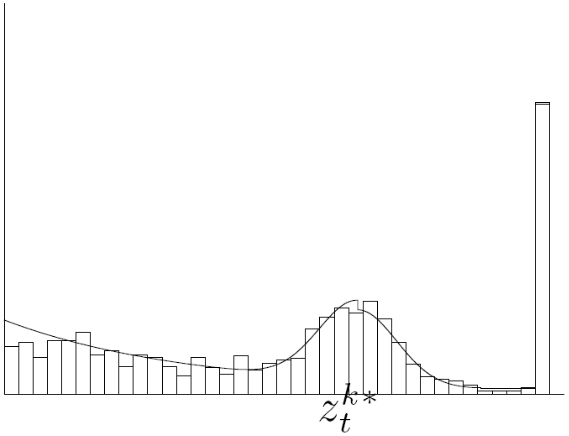


Laser





Sonar



300cm

400cm

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

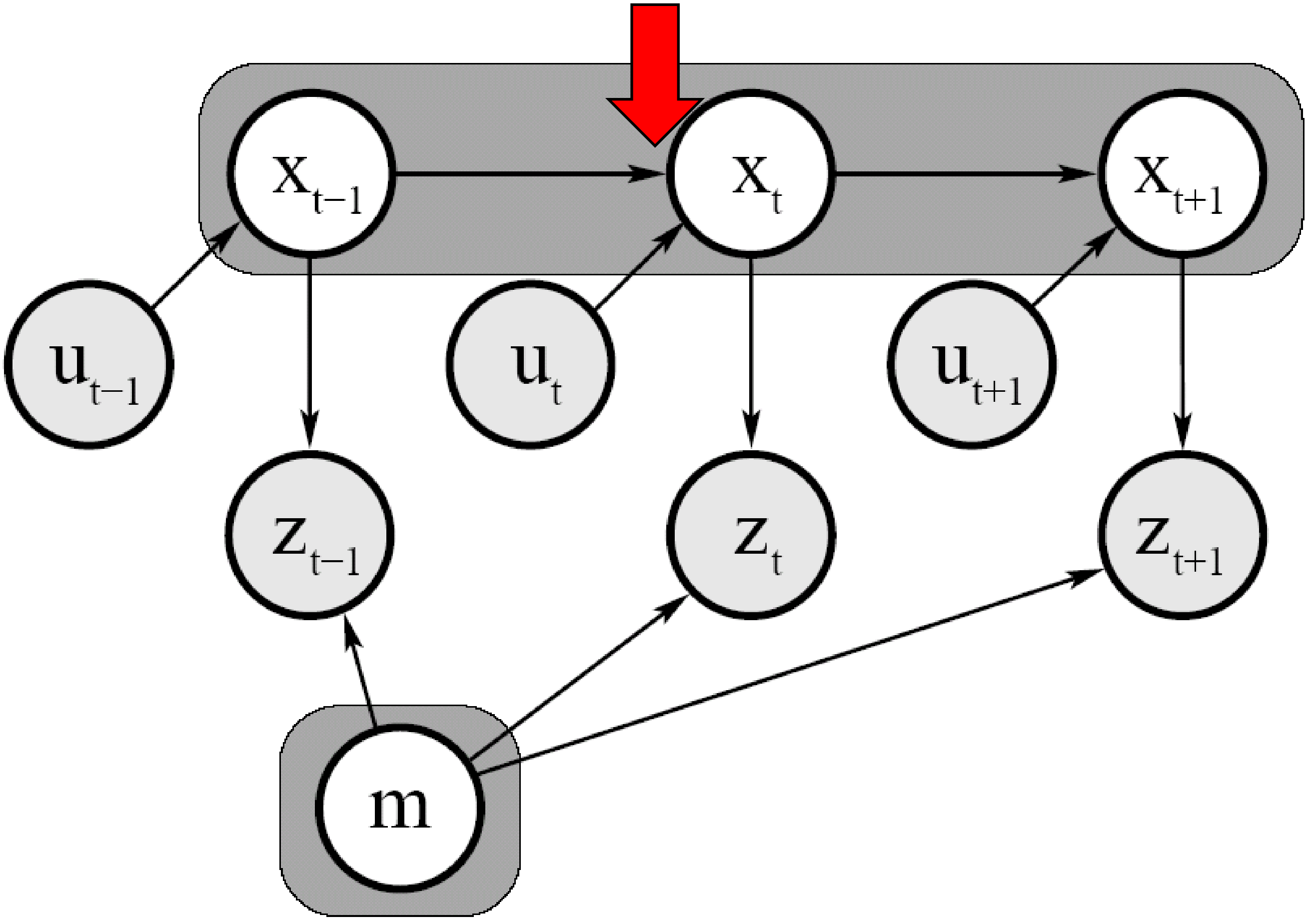
- To incorporate the outcome of an action *u* into the current "belief", we use the conditional pdf

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

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

Probabilistic Graphical Models

Action: $p(x_t \mid x_{t-1}, u_{t-1})$



$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

Odometry Model

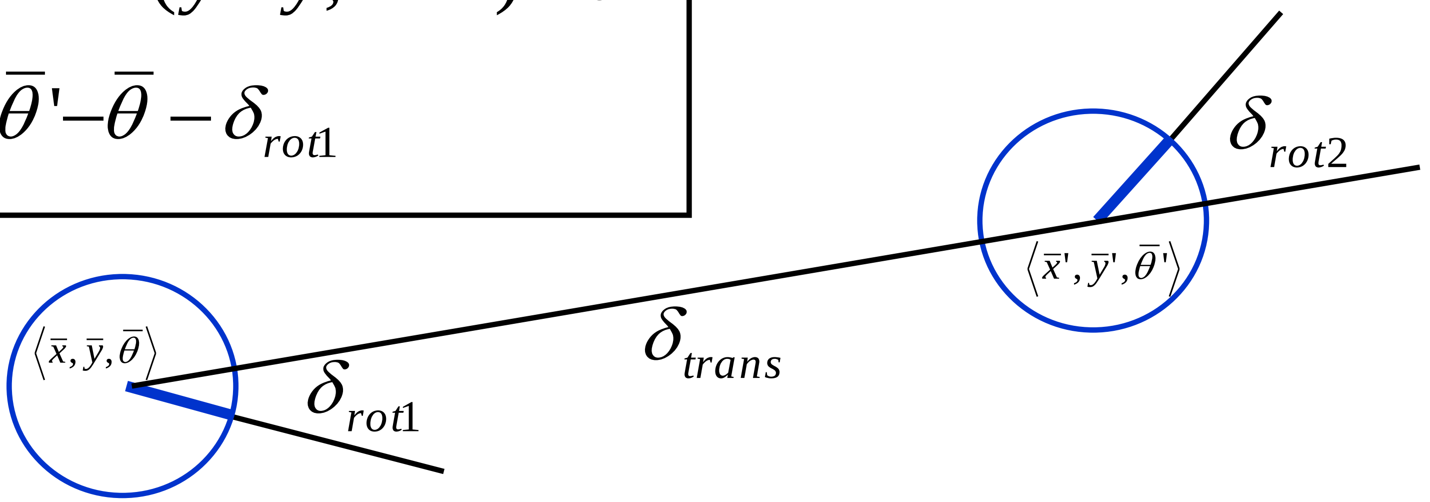
Robot moves from $\langle \bar{x}, \bar{y}, \bar{\theta} \rangle$ to $\langle \bar{x}', \bar{y}', \bar{\theta}' \rangle$.

Odometry information $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$

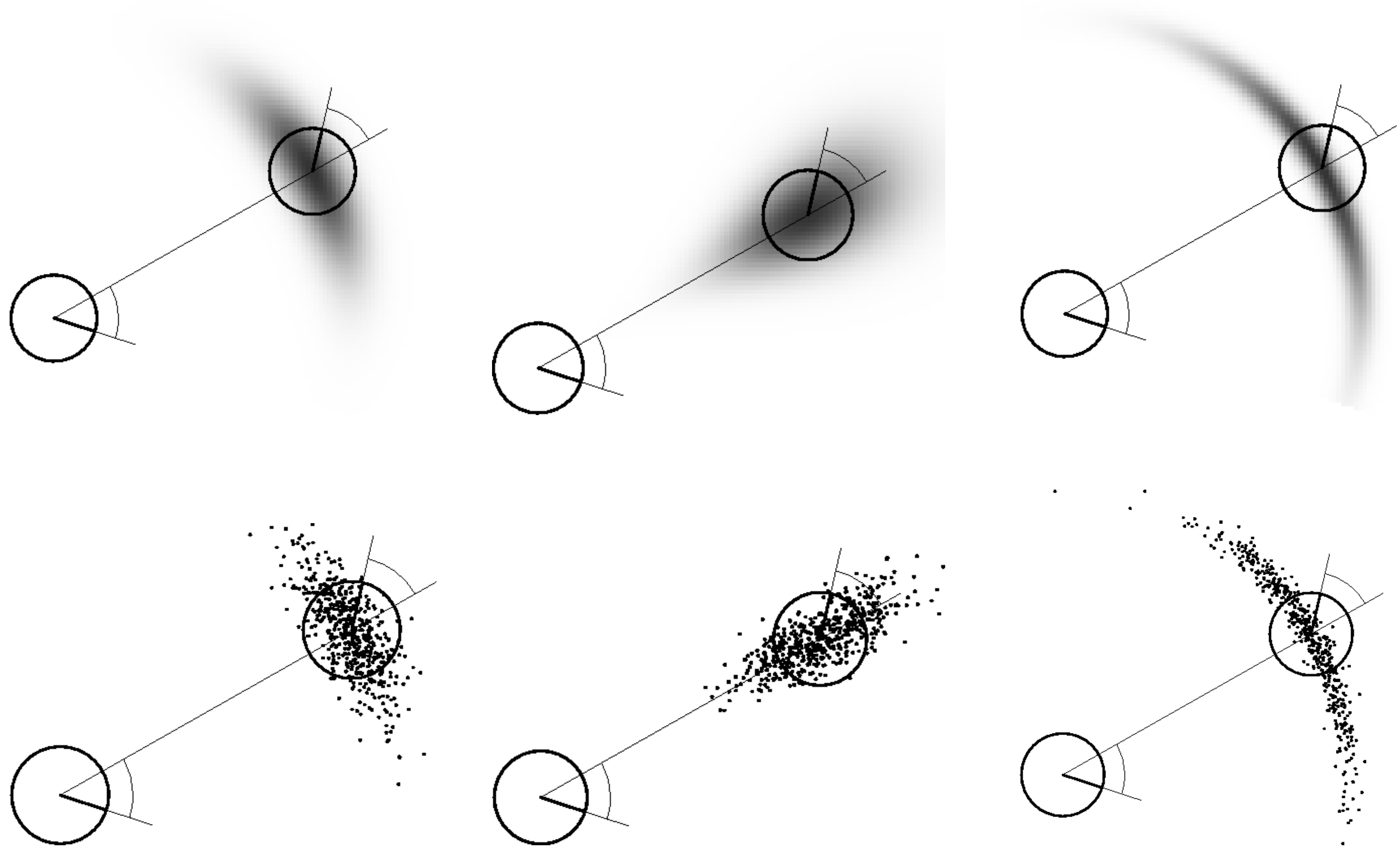
$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$

$$\delta_{rot1} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

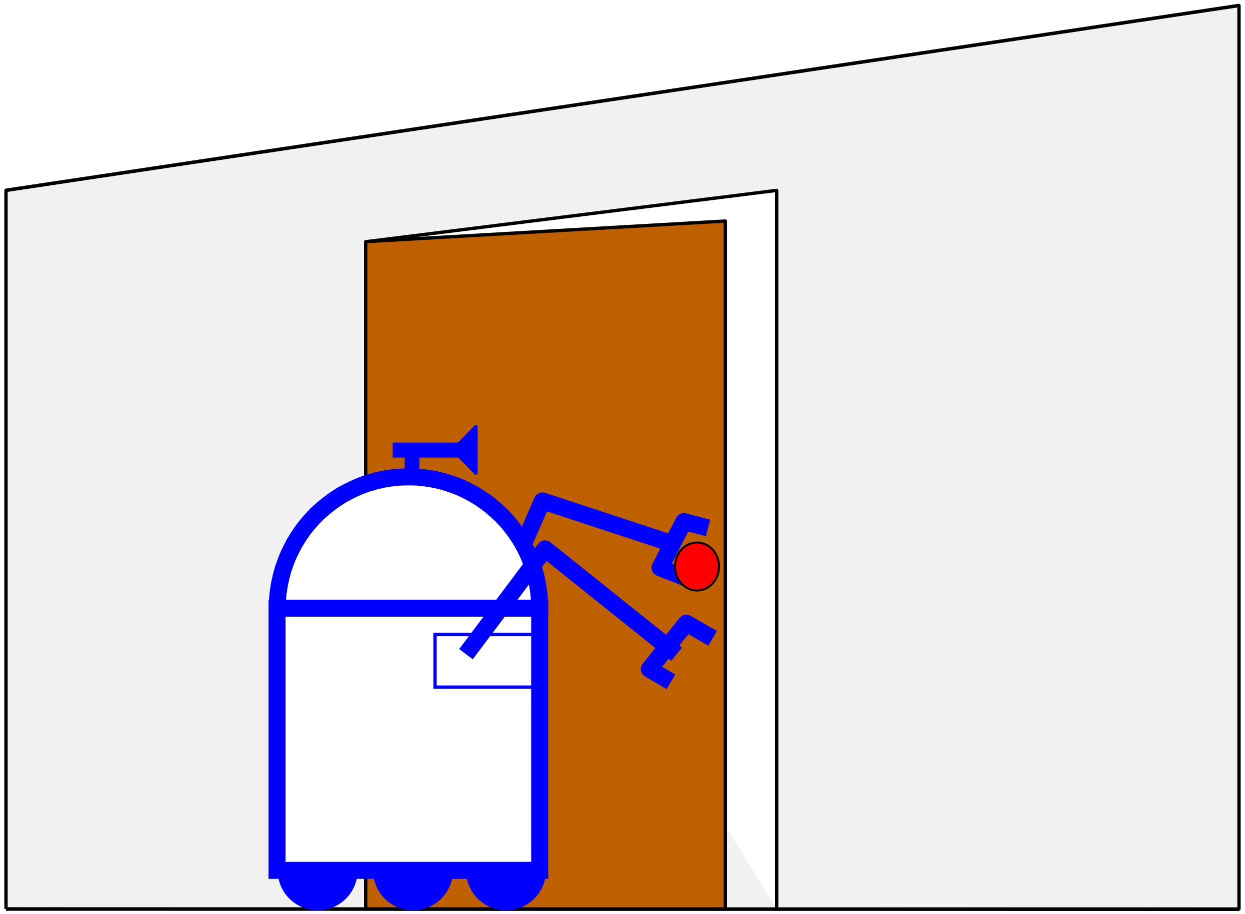
$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$



Effect of Distribution Type

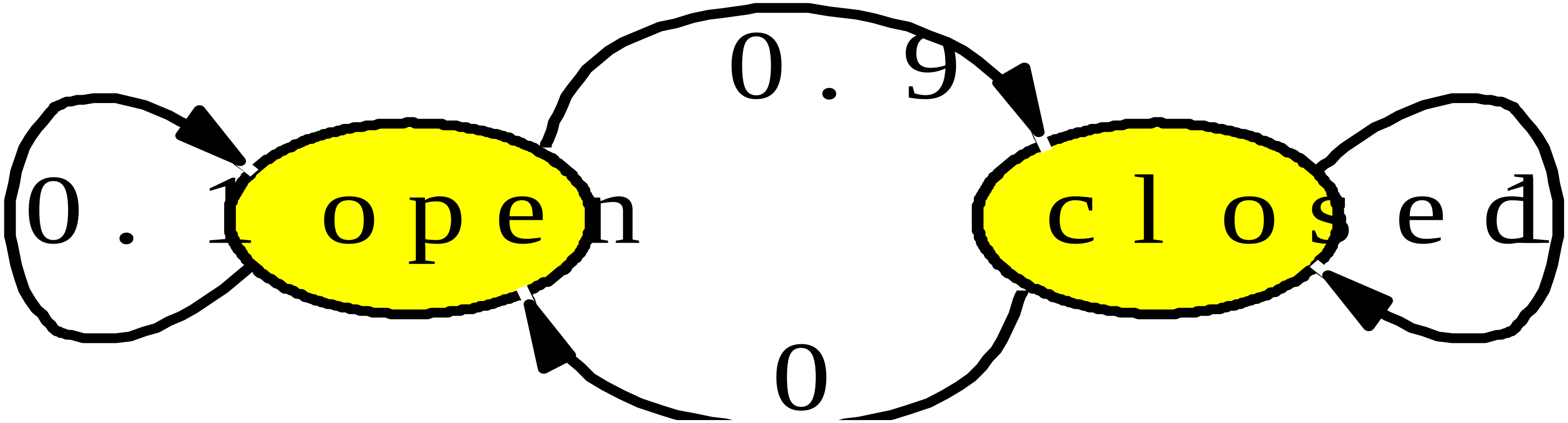


Example: Closing the door



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

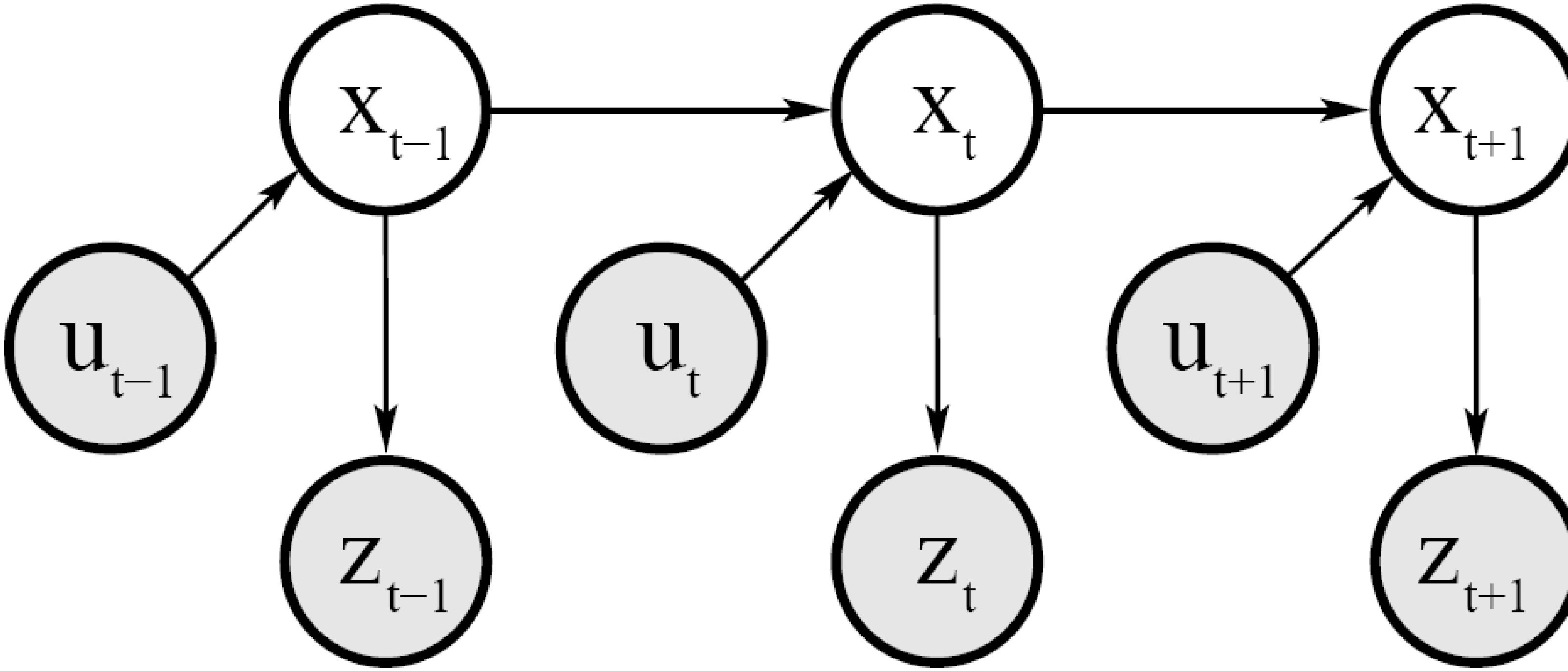
- **Given:**
  - Stream of observations *z* and action data *u:*

    $$\{u_1, z_1 \ \ldots, u_t, z_t\}$$

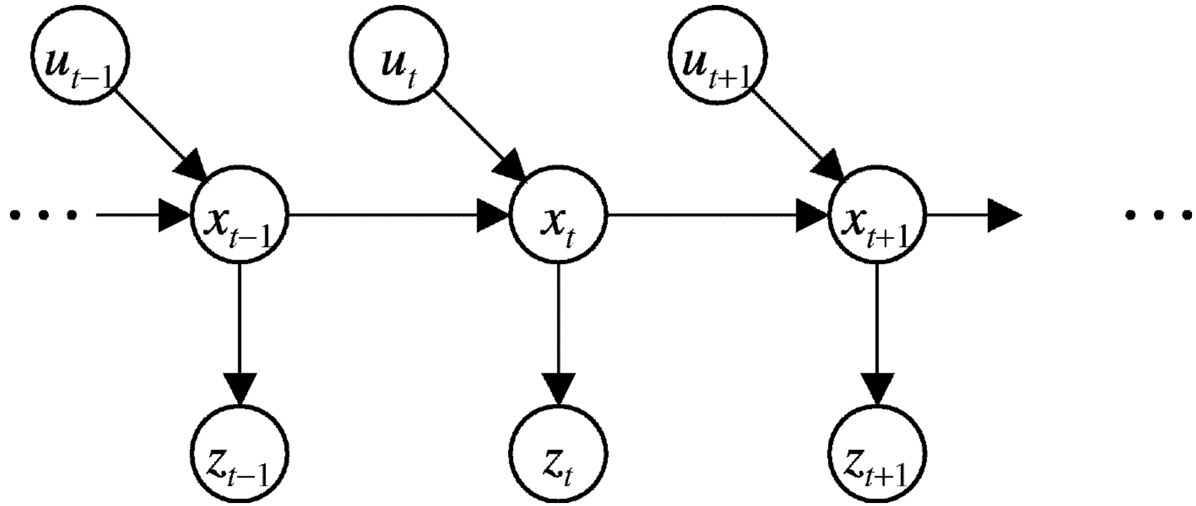
  - Sensor model *P(z|x).*
  - Action model *P(x|u,x')*.
  - Prior probability of the system state *P(x).*
- **Wanted:**
  - Estimate of the state *X* of a dynamical system.
  - The posterior of the state is also called **Belief**:

$$Bel(x_t) = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Dynamic Bayesian Network for Controls, States, and Sensations



Markov Assumption



$$p(z_t \mid x_{0:t}, z_{1:t}, u_{1:t}) = p(z_t \mid x_t)$$

$$p(x_t \mid x_{1:t-1}, z_{1:t}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

Underlying Assumptions

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

Bayes Filters

z = observation
u = action
x = state

$$Bel(x_t) = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Bayes $$= \eta \; P(z_t \mid x_t, u_1, z_1, \ldots, u_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

Markov $$= \eta \; P(z_t \mid x_t) \; P(x_t \mid u_1, z_1, \ldots, u_t)$$

Total prob. $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \; dx_{t-1}$$

Markov $$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, u_t) \; dx_{t-1}$$

Markov $$= \eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; P(x_{t-1} \mid u_1, z_1, \ldots, z_{t-1}) \; dx_{t-1}$$

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

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

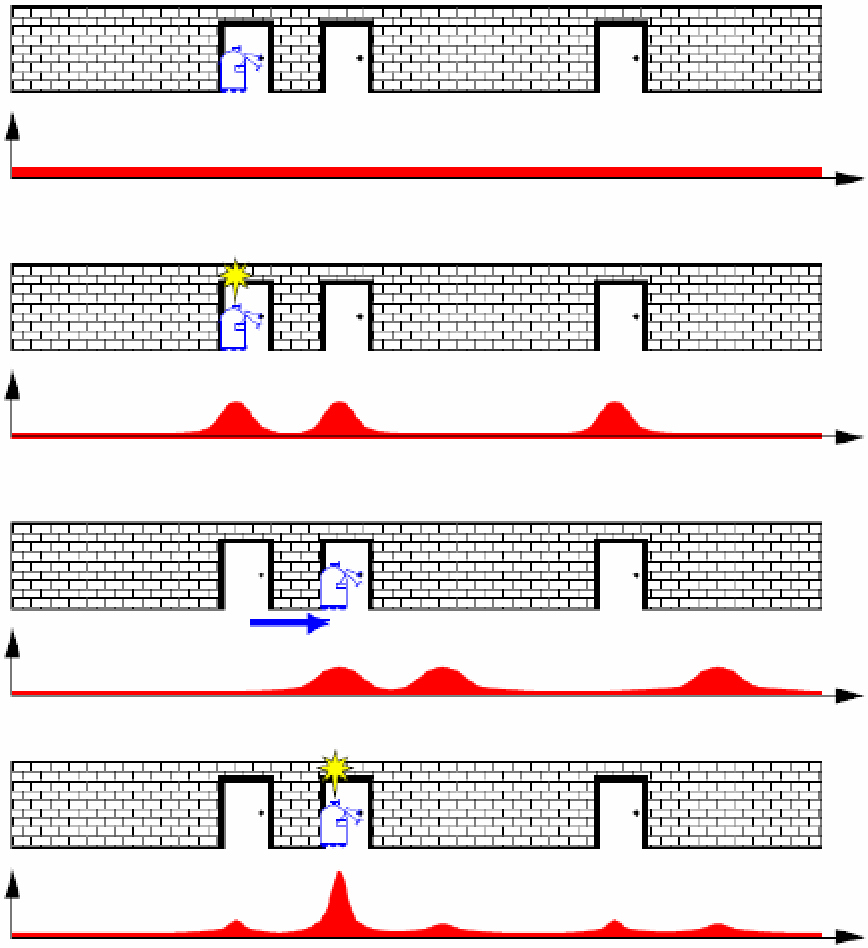
1. Algorithm **Bayes_filter**(*Bel(x),d*):
2. $\eta = 0$
3. If *d* is a perceptual data item *z* then
4. For all *x* do
5. $Bel'(x) = P(z \mid x) Bel(x)$
6. $\eta = \eta + Bel'(x)$
7. For all *x* do
8. $Bel'(x) = \eta^{-1} Bel'(x)$
9. Else if *d* is an action data item *u* then
10. For all *x* do
11.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

Bayes Filters are Familiar!

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

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

Bayes Filters in Localization



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

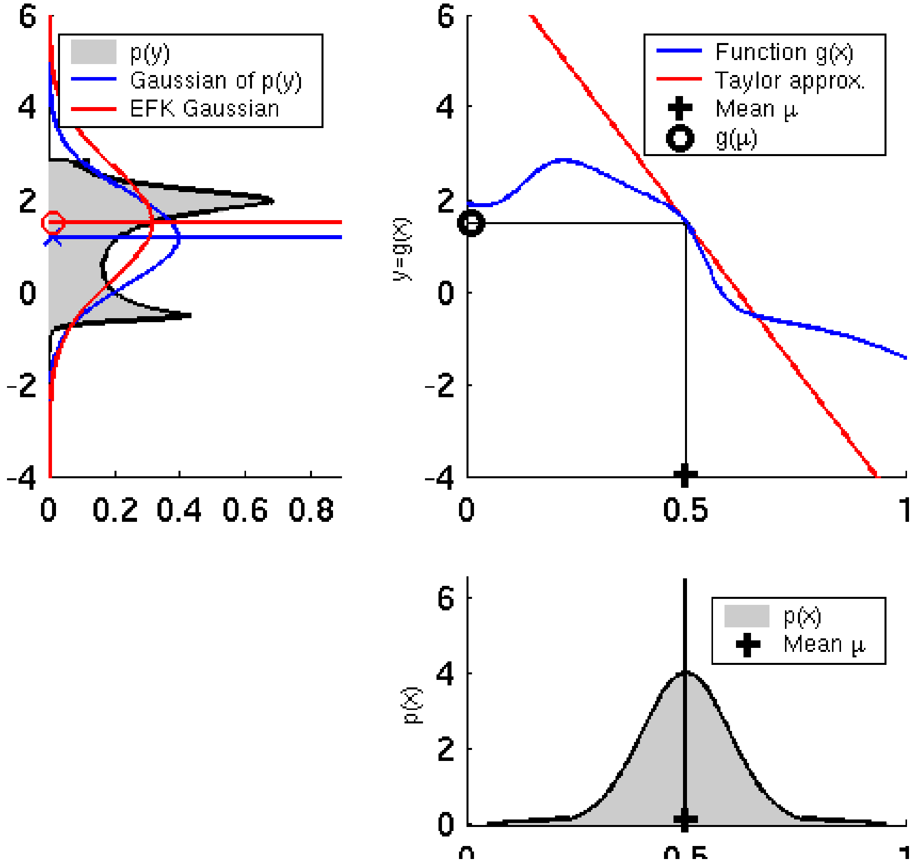
Summary so far ....

- 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.

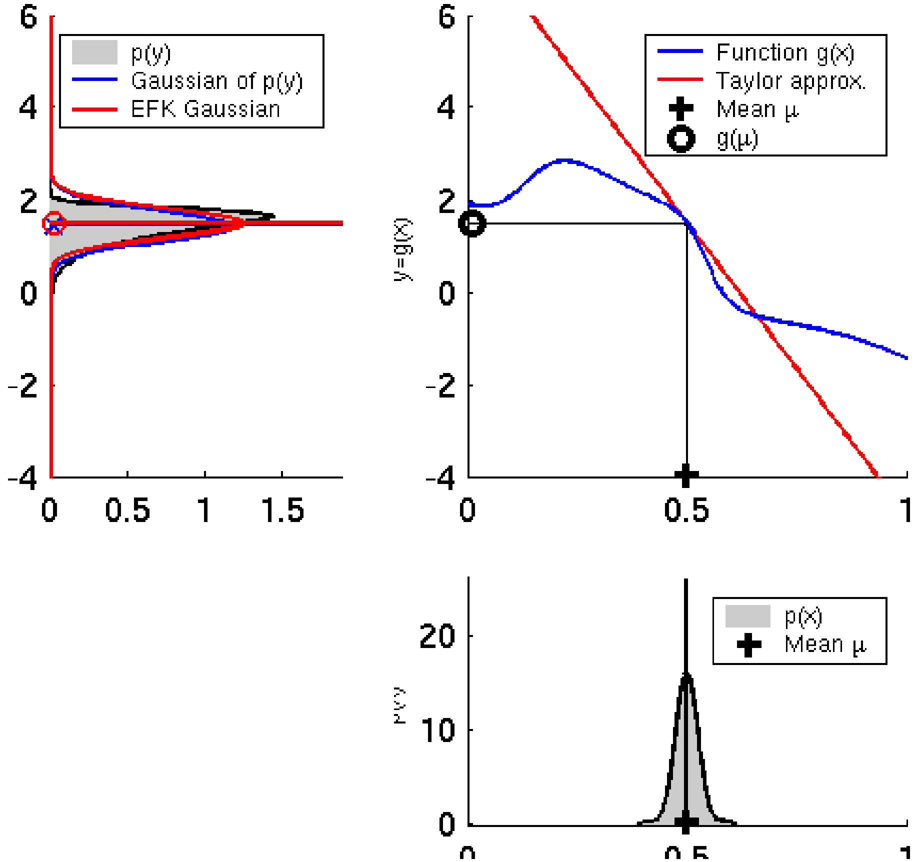
Parametric Vs. Non-parametric

- Representing distributions by using statistics or parameters (mean, variance)
- Non-parametric approach: Deal with distributions directly
- Remember:
  1. Gaussian distribution is completely parameterized by two numbers (mean, variance)
  2. Gaussian distribution remains Gaussian when mapped linearly.

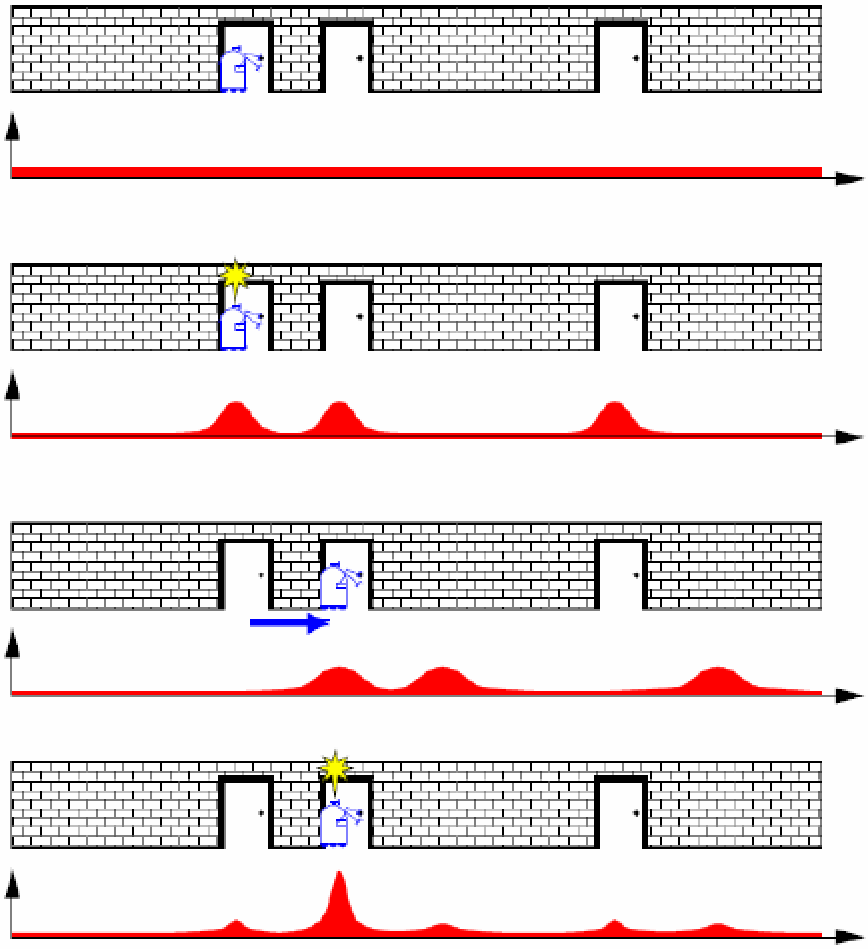
Linearization



Linearization (Cont.)

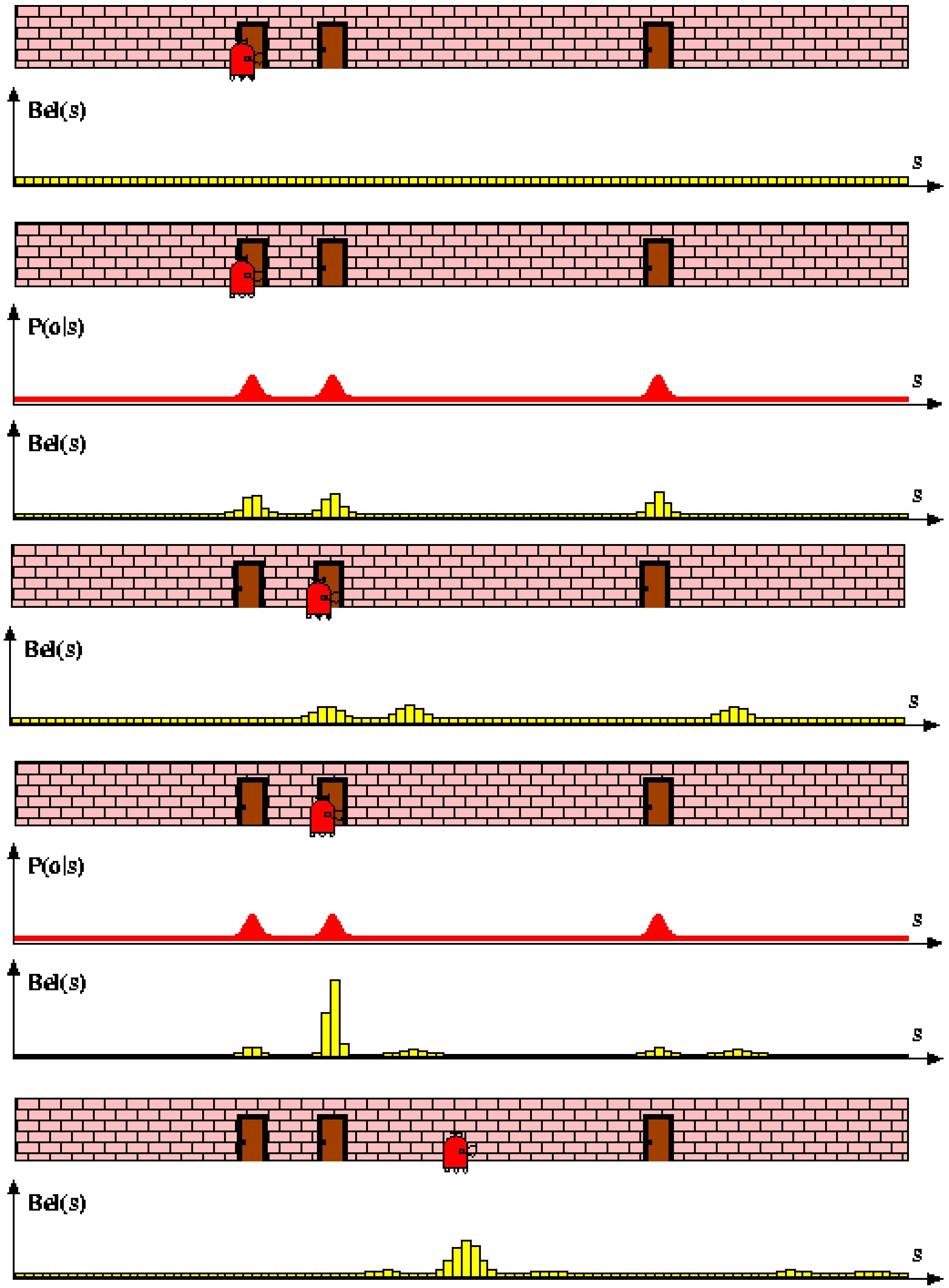


Bayes Filters in Localization

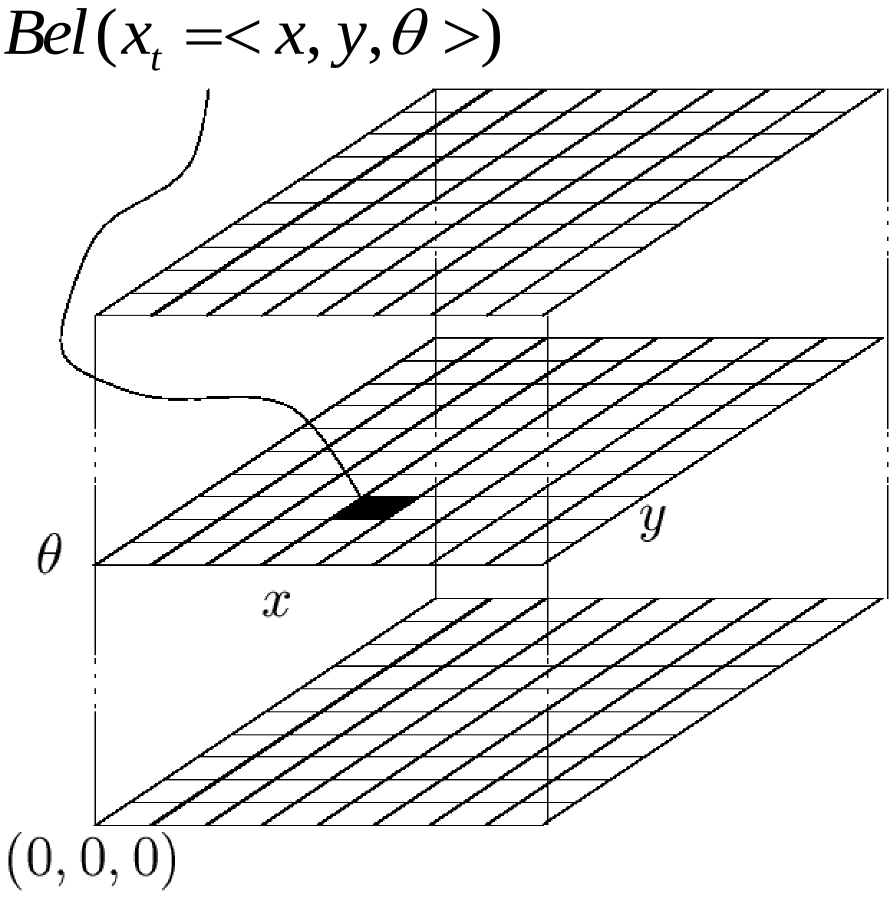


$$Bel(x_t) = \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}$$

Histogram = Piecewise Constant



Piecewise Constant Representation



Discrete Bayes Filter Algorithm

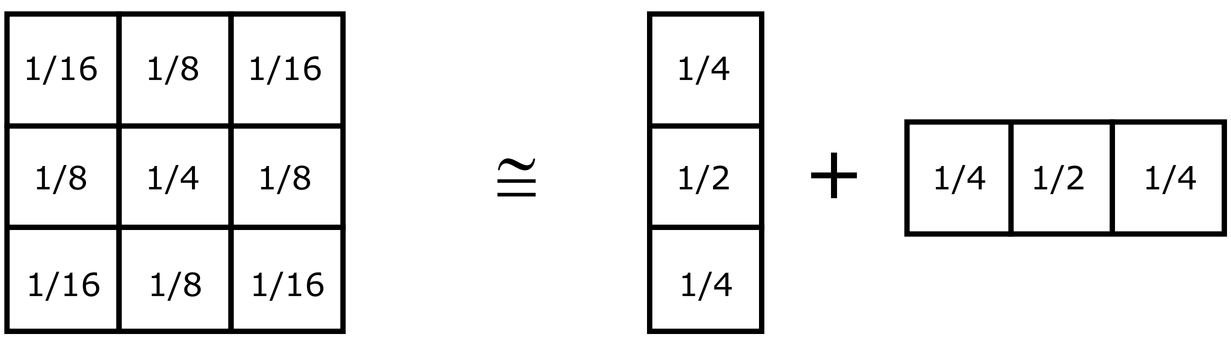
1. Algorithm **Discrete_Bayes_filter**(*Bel(x),d*):
2. $\eta=0$
3. If *d* is a perceptual data item *z* then
4. For all *x* do
5. $$Bel'(x) = P(z \mid x) Bel(x)$$
6. $$\eta = \eta + Bel'(x)$$
7. For all *x* do
8. $$Bel'(x) = \eta^{-1} Bel'(x)$$
9. Else if *d* is an action data item *u* then
10. For all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

Implementation (1)

- To update the belief upon sensory input and to carry out the normalization one has to iterate over all cells of the grid.
- Especially when the belief is peaked (which is generally the case during position tracking), one wants to avoid updating irrelevant aspects of the state space.
- One approach is not to update entire sub-spaces of the state space.
- This, however, requires to monitor whether the robot is de-localized or not.
- To achieve this, one can consider the likelihood of the observations given the active components of the state space.

Implementation (2)

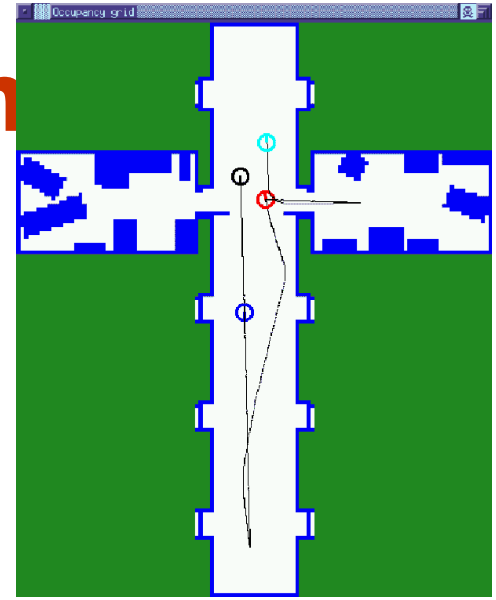
- To efficiently update the belief upon robot motions, one typically assumes a bounded Gaussian model for the motion uncertainty.
- This reduces the update cost from $O(n^2)$ to $O(n)$, where n is the number of states.
- The update can also be realized by shifting the data in the grid according to the measured motion.
- In a second step, the grid is then convolved using a separable Gaussian Kernel.
- Two-dimensional example:

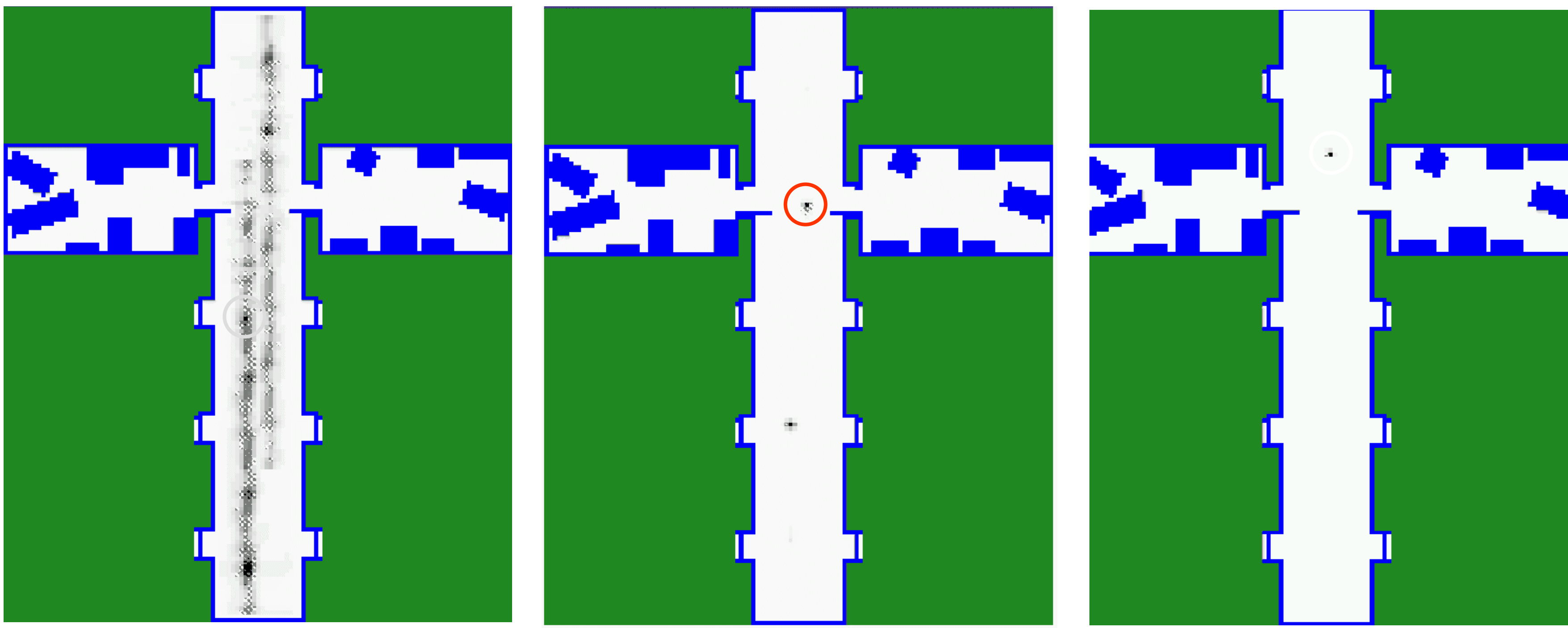


Fewer arithmetic operations

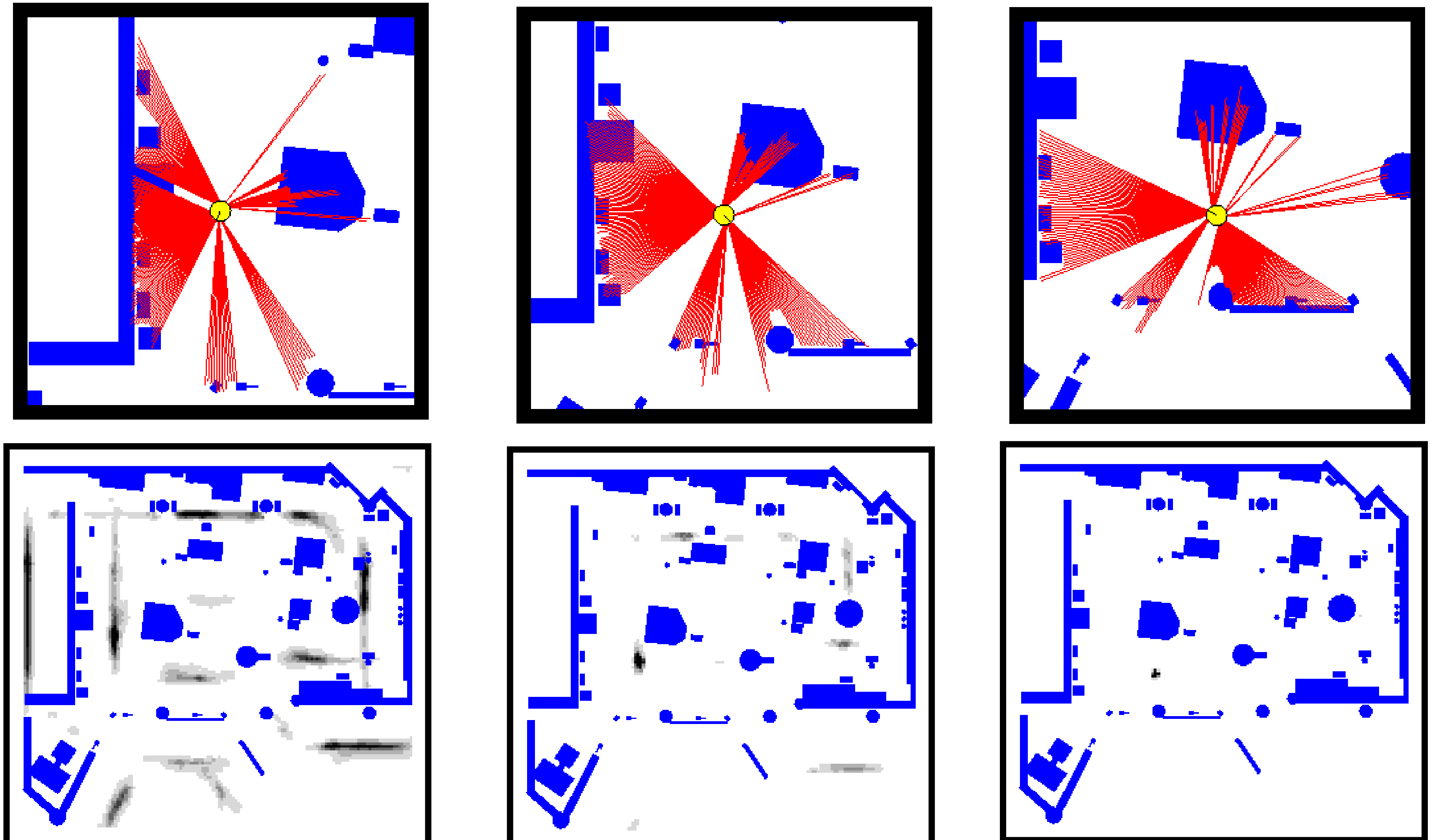
Easier to implement

Markov Localization in Grid Map





Grid-based Localization



Mathematical Description

- Set of weighted samples

$$S = \left\{ \left\langle s^{[i]}, w^{[i]} \right\rangle \mid i = 1, \ldots, N \right\}$$

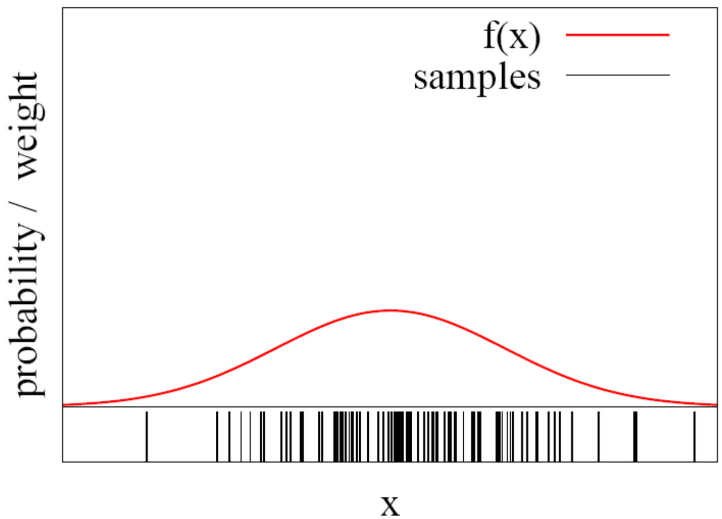
State hypothesis   Importance weight

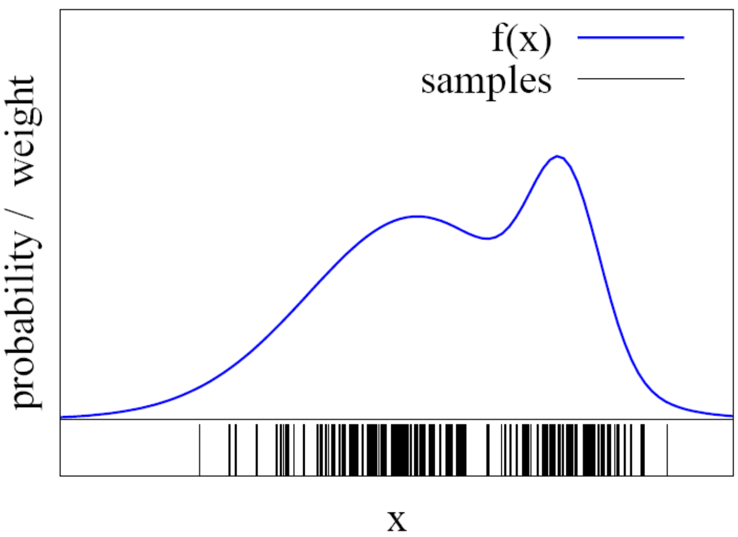
- The samples represent the posterior

$$p(x) = \sum_{i=1}^{N} w_i \cdot \delta_{s^{[i]}}(x)$$

Function Approximation

- Particle sets can be used to approximate functions



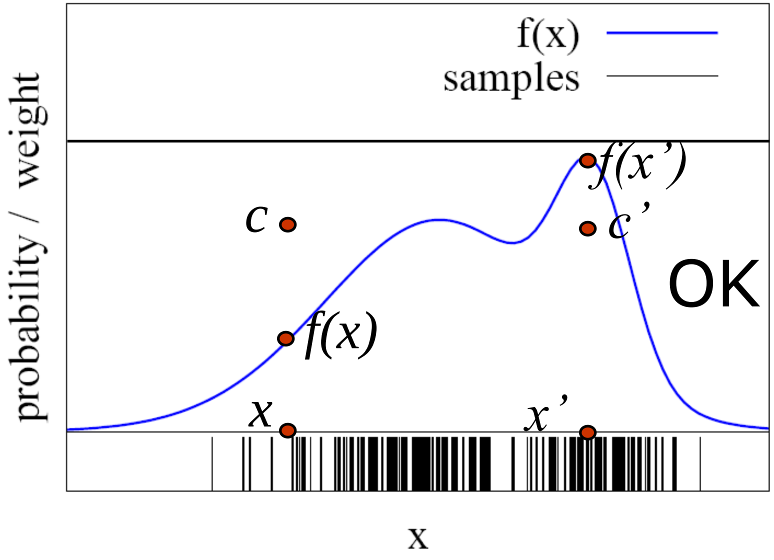


- The more particles fall into an interval, the higher the probability of that interval

- How to draw samples form a function/distribution?

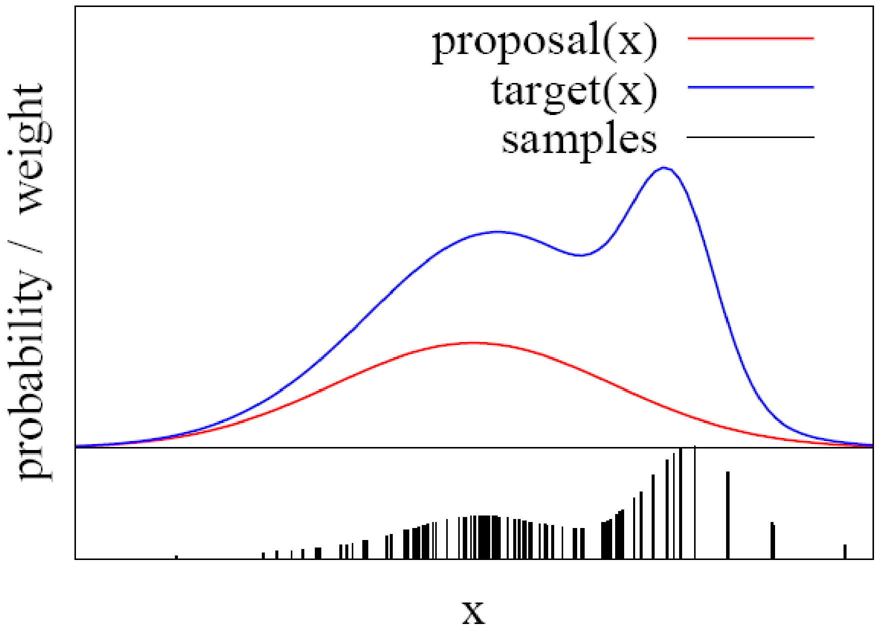
Rejection Sampling

- Let us assume that $f(x)<1$ for all x
- Sample x from a uniform distribution
- Sample c from [0,1]
- if $f(x) > c$ keep the sample
  otherwise reject the sampe



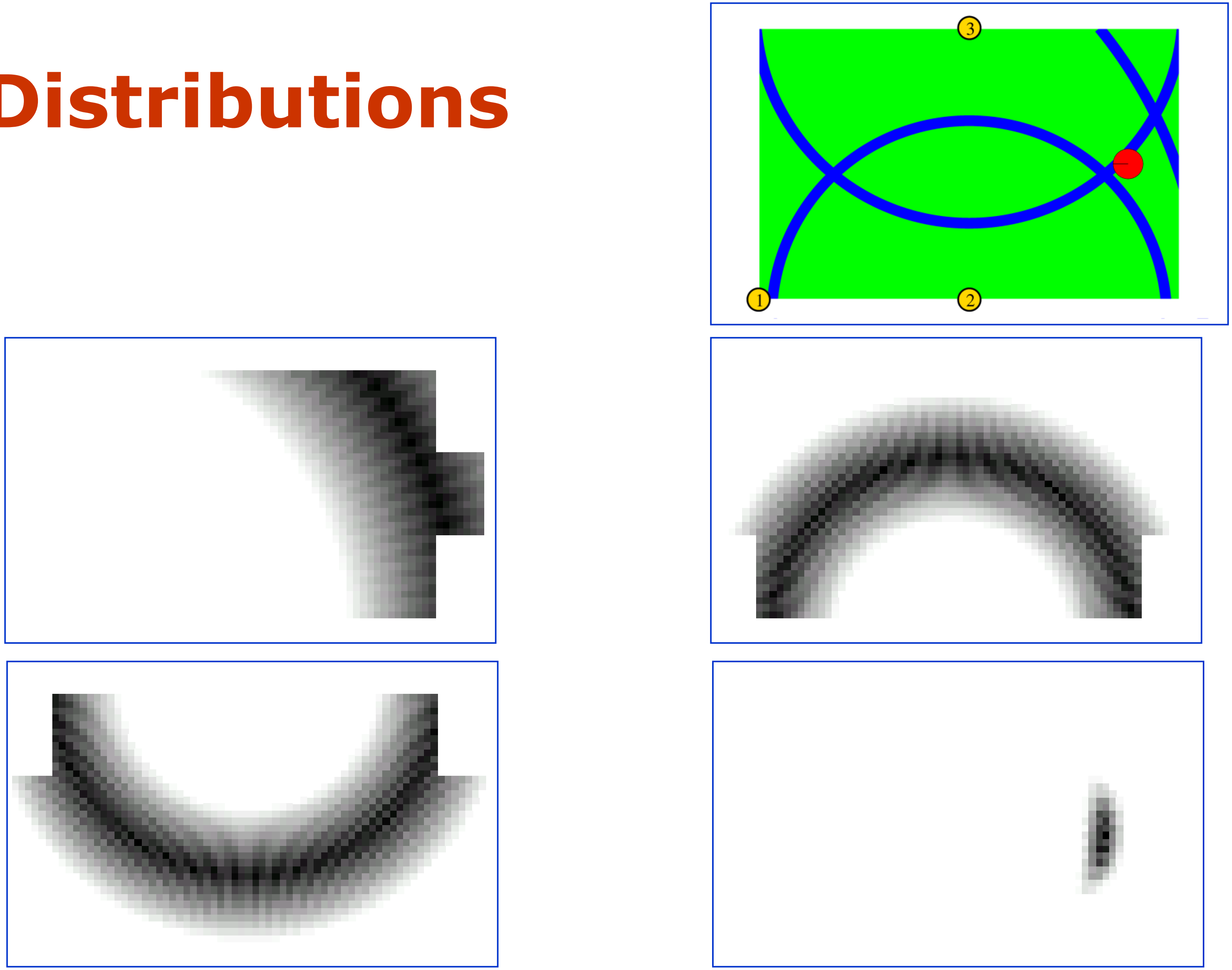
Importance Sampling Principle

- We can even use a different distribution g to generate samples from f
- By introducing an importance weight w, we can account for the "differences between g and f"
- $w = f / g$
- f is often called target
- g is often called proposal
- Pre-condition: $f(x)>0 \rightarrow g(x)>0$

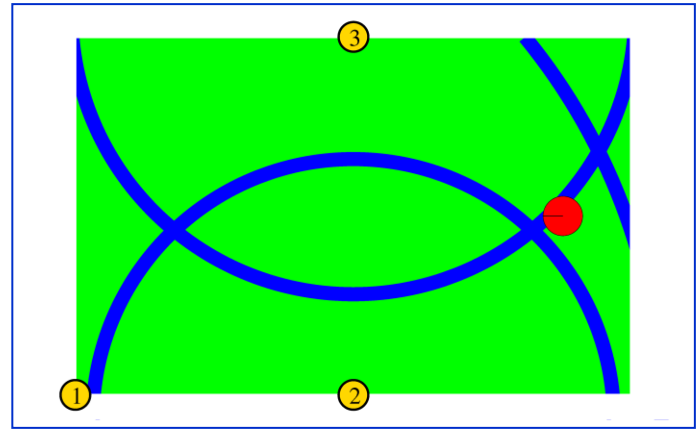


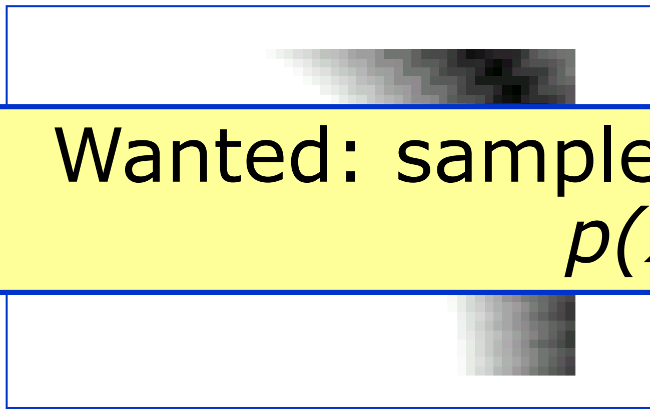
Importance Sampling with Resampling: Landmark Detection Example

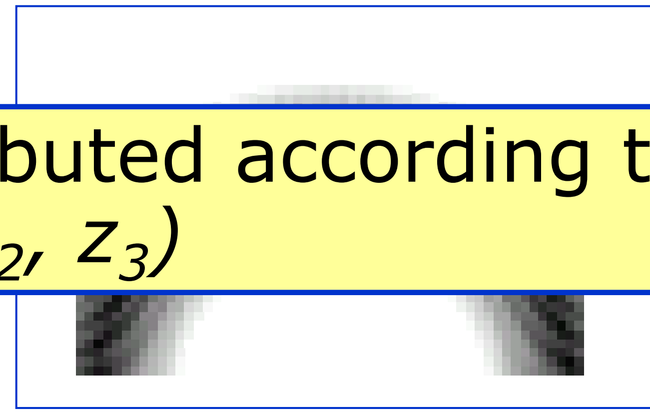
Distributions



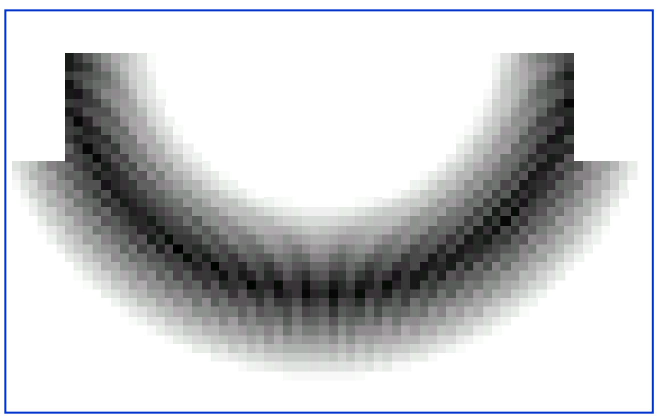
Distributions

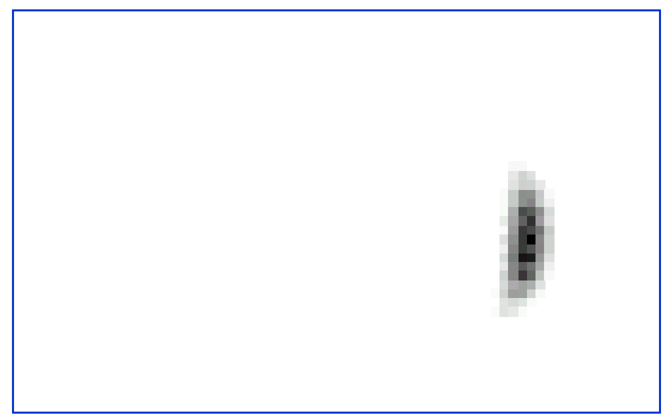






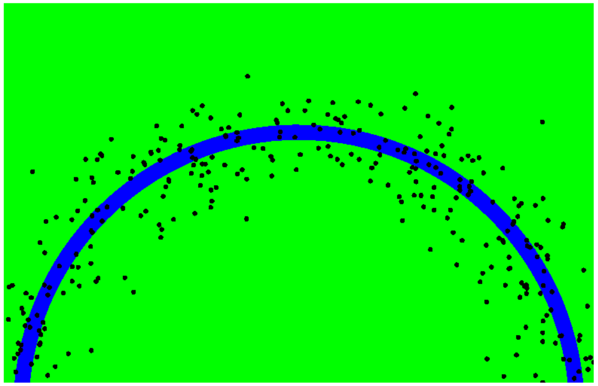
Wanted: samples distributed according to
$p(x|\ z_1, z_2, z_3)$

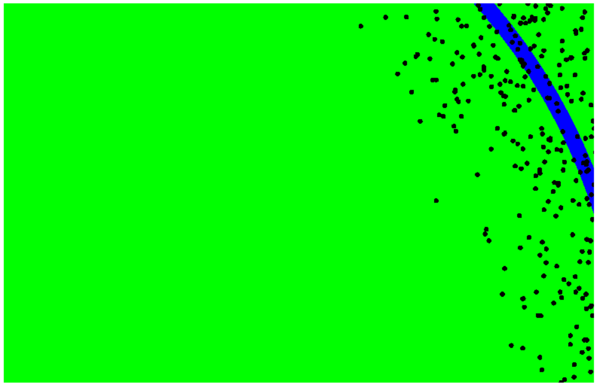


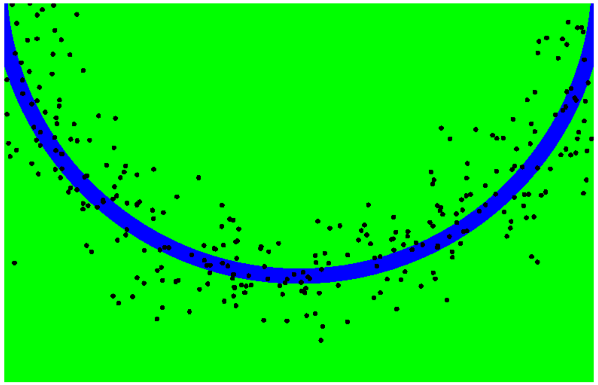


This is Easy!

We can draw samples from $p(x|z_l)$ by adding noise to the detection parameters.







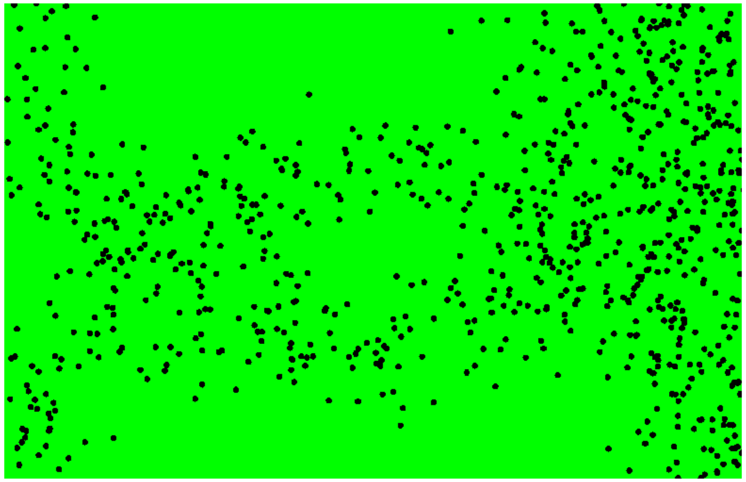
Importance Sampling

$$\text{Target distribution f}: p(x \mid z_1, z_2, ..., z_n) = \frac{\prod_k p(z_k \mid x) \quad p(x)}{p(z_1, z_2, ..., z_n)}$$

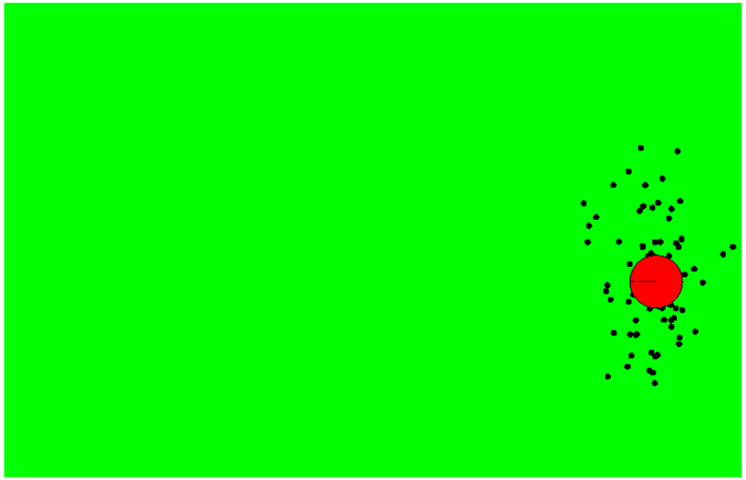
$$\text{Sampling distribution g}: p(x \mid z_l) = \frac{p(z_l \mid x)\, p(x)}{p(z_l)}$$

$$\text{Importance weights w}: \frac{f}{g} = \frac{p(x \mid z_1, z_2, ..., z_n)}{p(x \mid z_l)} = \frac{p(z_l) \prod_{k \neq l} p(z_k \mid x)}{p(z_1, z_2, ..., z_n)}$$

Importance Sampling with Resampling

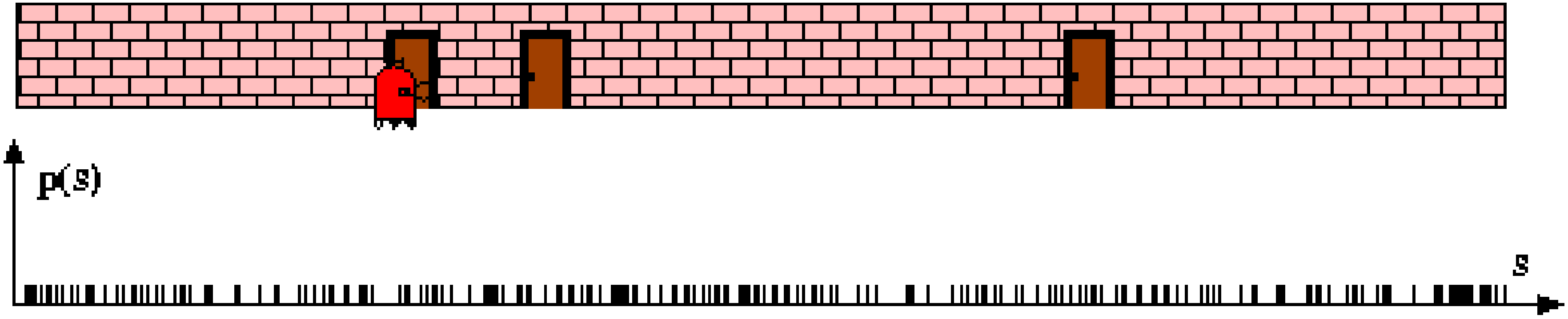


Weighted samples



After resampling

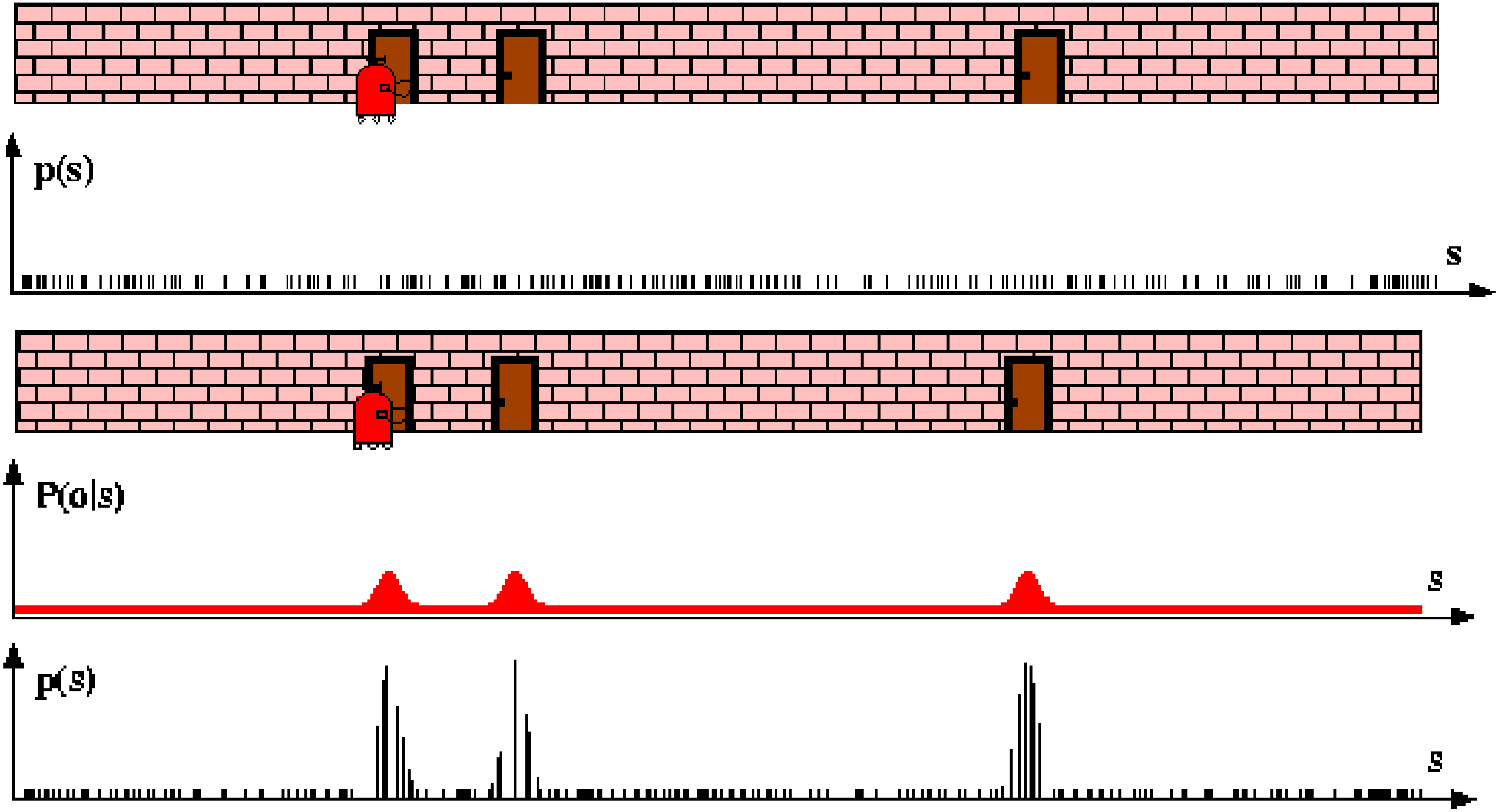
Particle Filters



Sensor Information: Importance Sampling

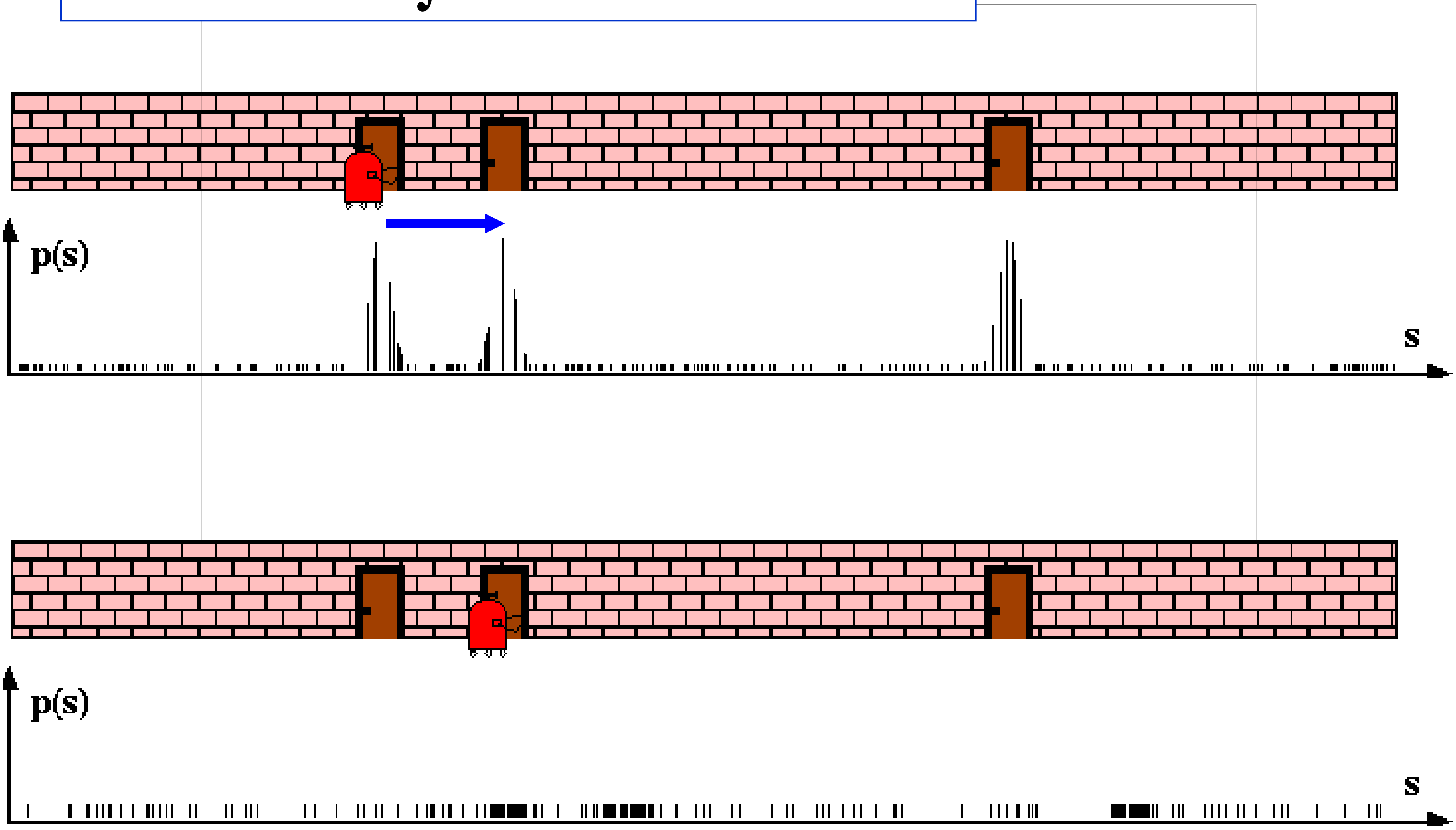
$$Bel(x) \leftarrow \alpha\, p(z\,|\,x)\, Bel^{-}(x)$$

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



Robot Motion

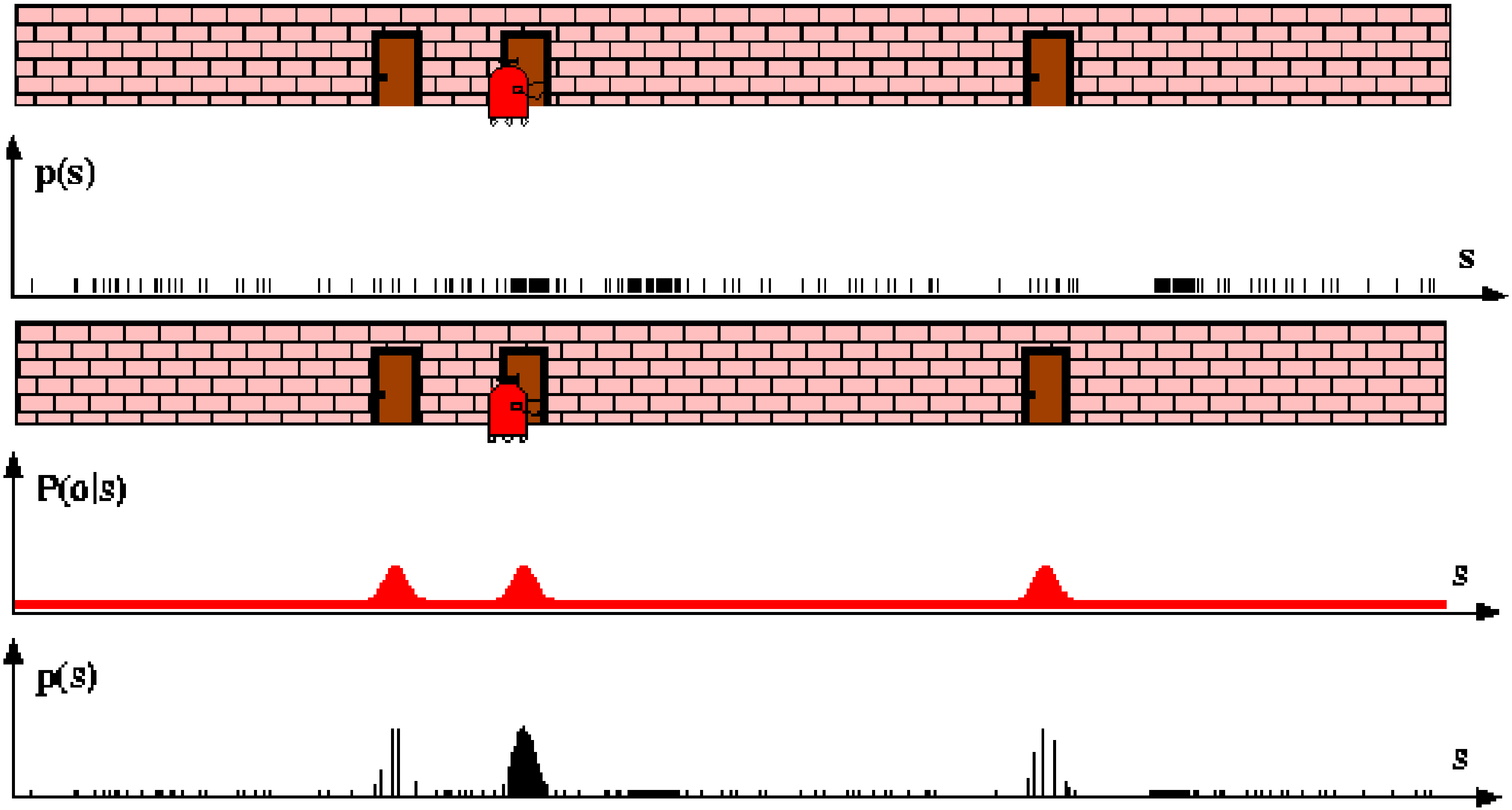
$$Bel^{-}(x) \leftarrow \int p(x \mid u, x')\, Bel(x')\, \mathrm{d}x'$$



Sensor Information: Importance Sampling

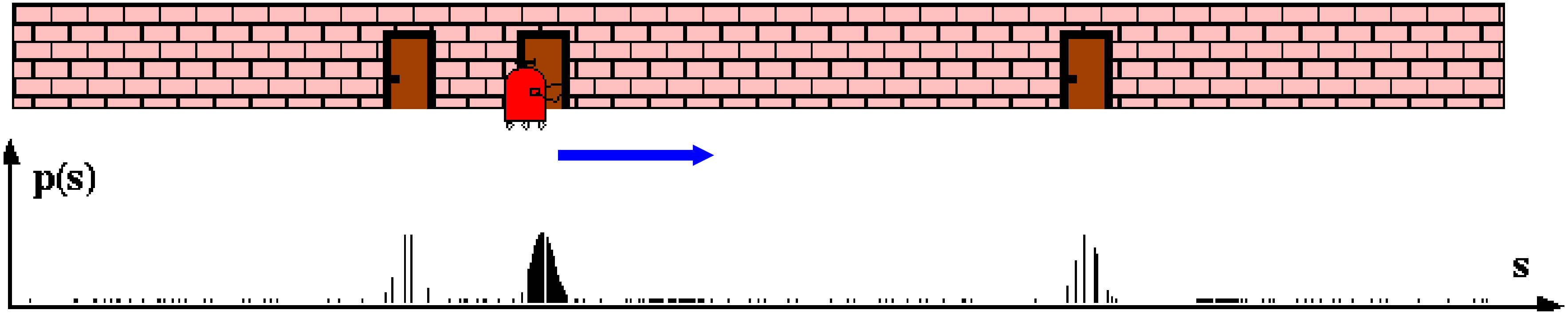
$$Bel(x) \leftarrow \alpha\ p(z\,|\,x)\ Bel^{-}(x)$$

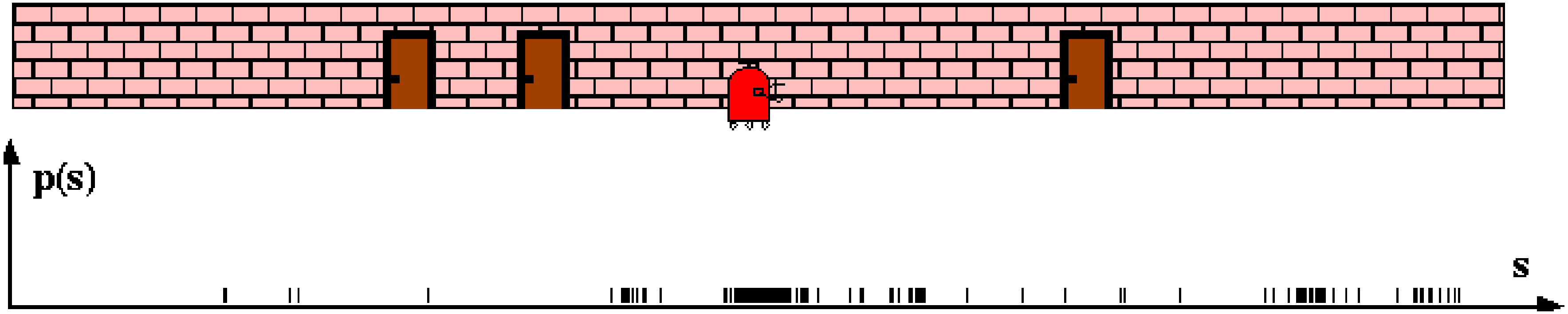
$$w \leftarrow \frac{\alpha\ p(z\,|\,x)\ Bel^{-}(x)}{Bel^{-}(x)} = \alpha\ p(z\,|\,x)$$



Robot Motion

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





Particle Filter Algorithm

- Sample the next generation for particles using the proposal distribution

- Compute the importance weights :

  $$weight = target\ distribution\ /\ proposal\ distribution$$

- Resampling: “Replace unlikely samples by more likely ones”

Particle Filter Algorithm

Algorithm **particle_filter**(M_{t-1}, u_{t-1} y_t):

$M_t = \varnothing, \quad \eta = 0$

For $i = 1 \ldots n$ — ***Generate new samples***

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

  $w_t^i = p(y_t \mid x_t^i)$ — ***Compute importance weight***

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

  $M_t = M_t \cup \{< x_t^i, w_t^i >\}$ — ***Insert***

For $i = 1 \ldots n$

  $w_t^i = w_t^i / \eta$ — ***Normalize weights***

RESAMPLE!!!

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:

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

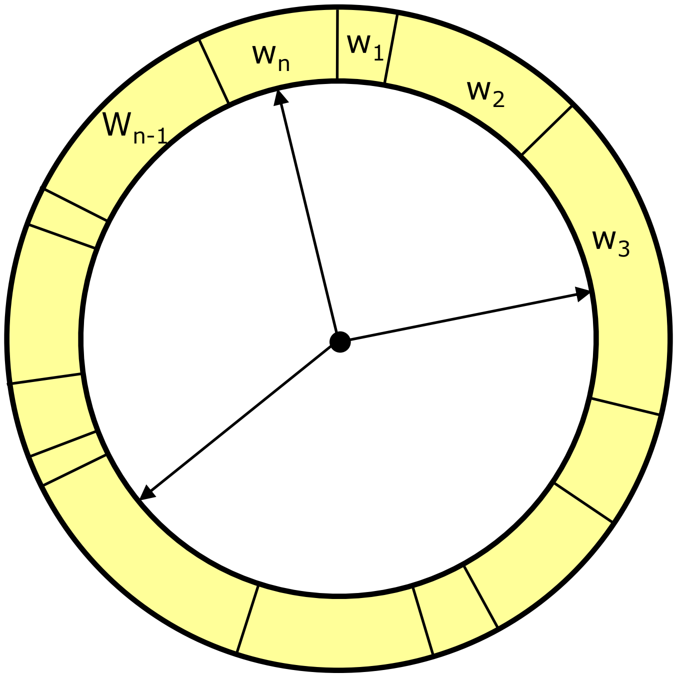
$$= \frac{\eta\, p(z_t \mid x_t)\, p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1})}{p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1})}$$

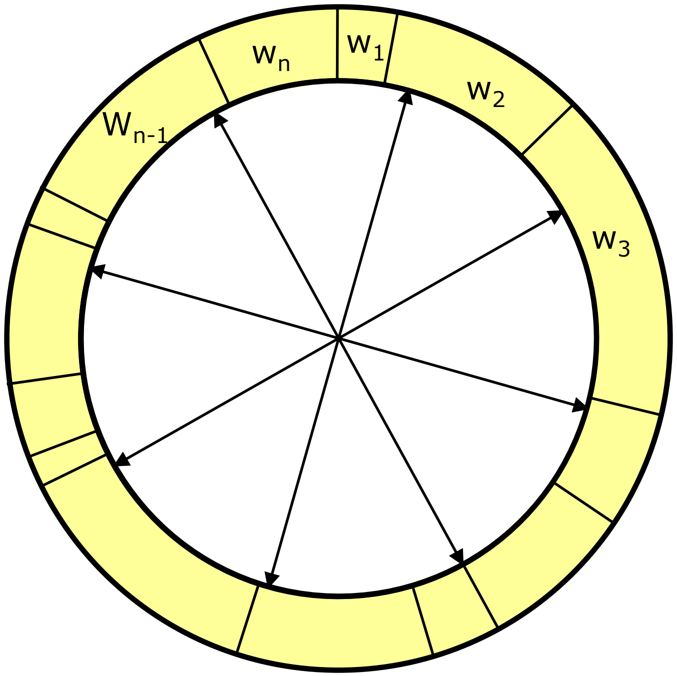
$$\propto p(z_t \mid x_t)$$

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





Roulette wheel

Binary search, n log n

Stochastic universal sampling

Systematic resampling

Linear time complexity

Easy to implement, low variance

Resampling Algorithm

1. Algorithm systematic_resampling(S,n):

2. $S' = \emptyset, c_1 = w^1$
3. For $i = 2 \ldots n$ — *Generate cdf*
4. $\quad c_i = c_{i-1} + w^i$
5. $u_1 \sim U]0, n^{-1}], i = 1$ — *Initialize threshold*

6. For $j = 1 \ldots n$ — *Draw samples …*
7. $\quad$ While ($u_j > c_i$) — *Skip until next threshold reached*
8. $\qquad i = i + 1$
9. $\quad S' = S' \cup \left\{ < x^i, n^{-1} > \right\}$ — *Insert*
10. $\quad u_{j+1} = u_j + n^{-1}$ — *Increment threshold*

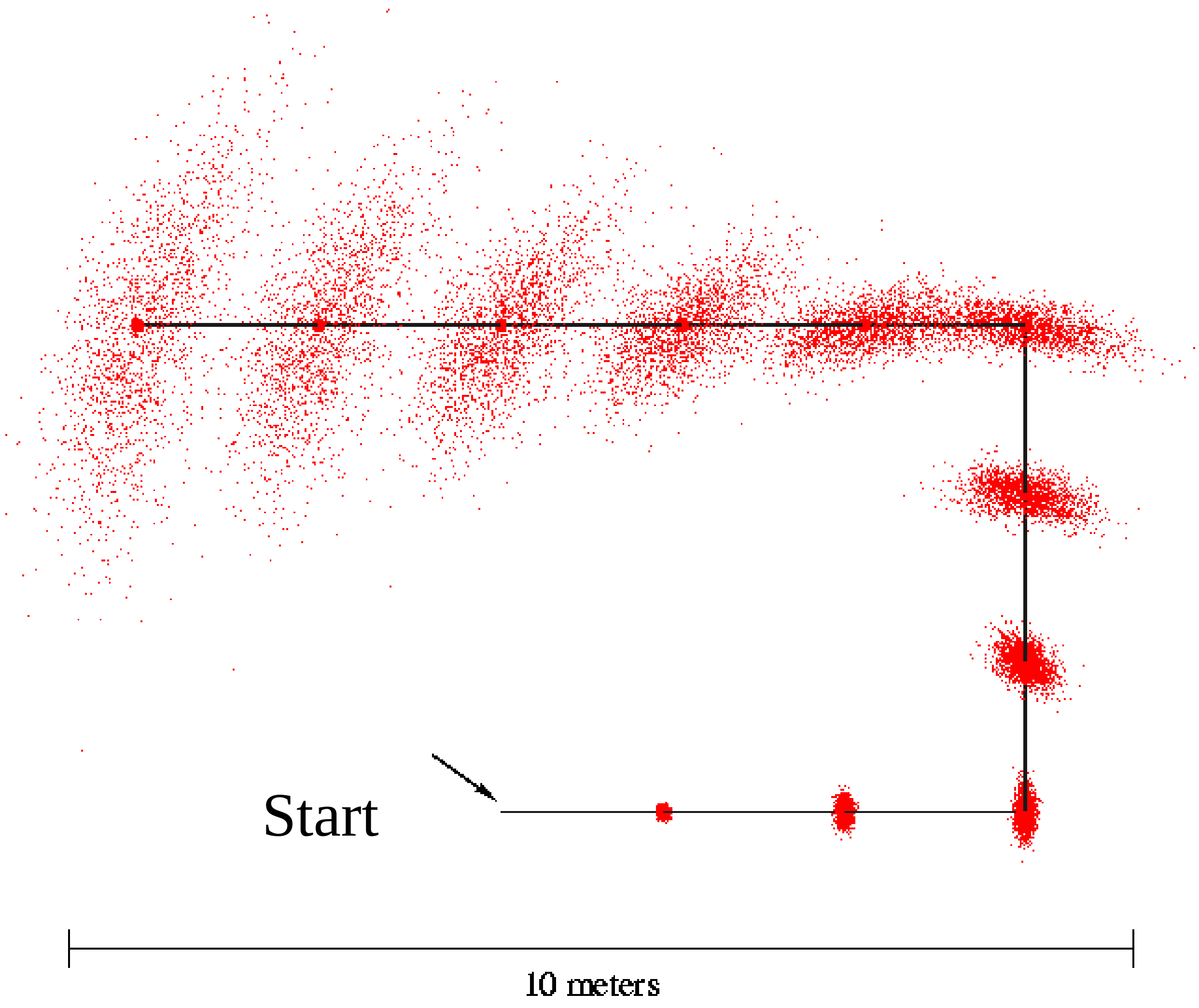
11. Return S'

Also called stochastic universal sampling

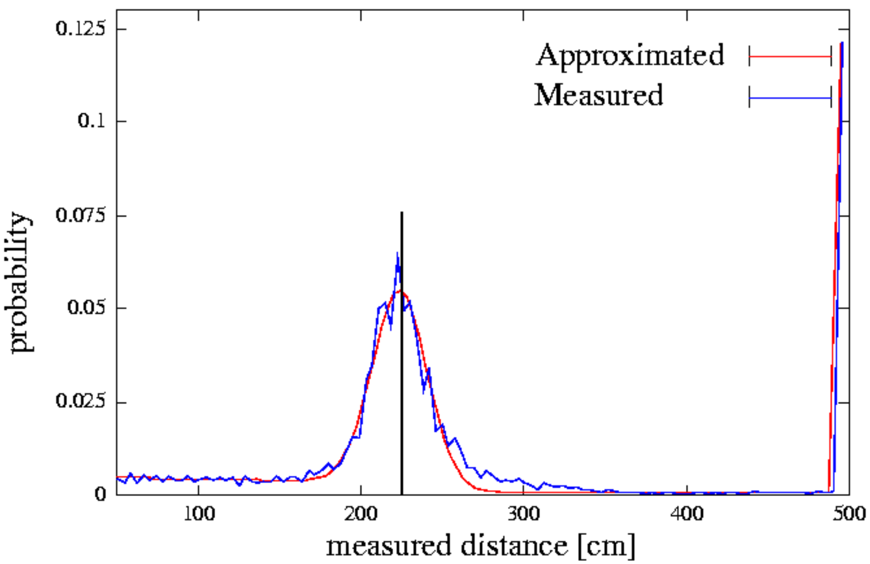
Mobile Robot Localization

- Each particle is a potential pose of the robot
- Proposal distribution is the motion model of the robot (prediction step)
- The observation model is used to compute the importance weight (correction step)

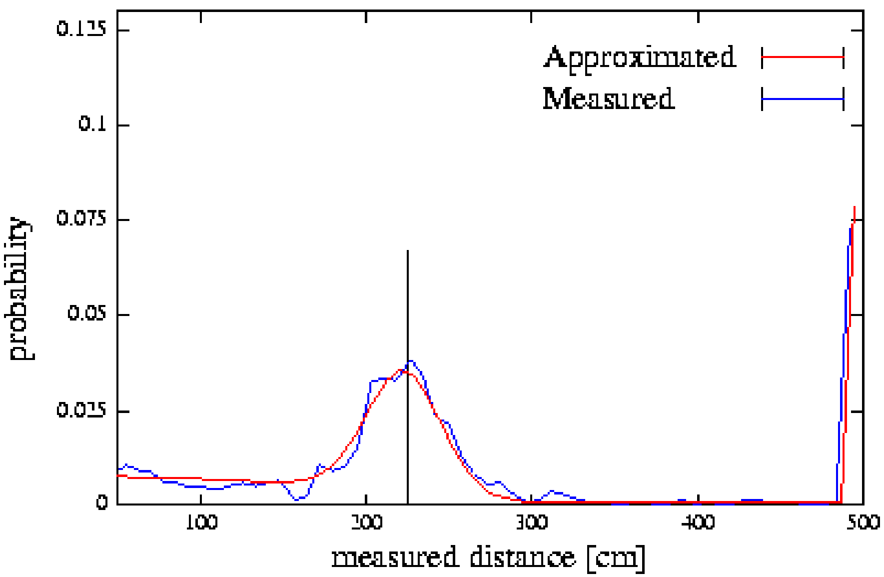
Motion Model



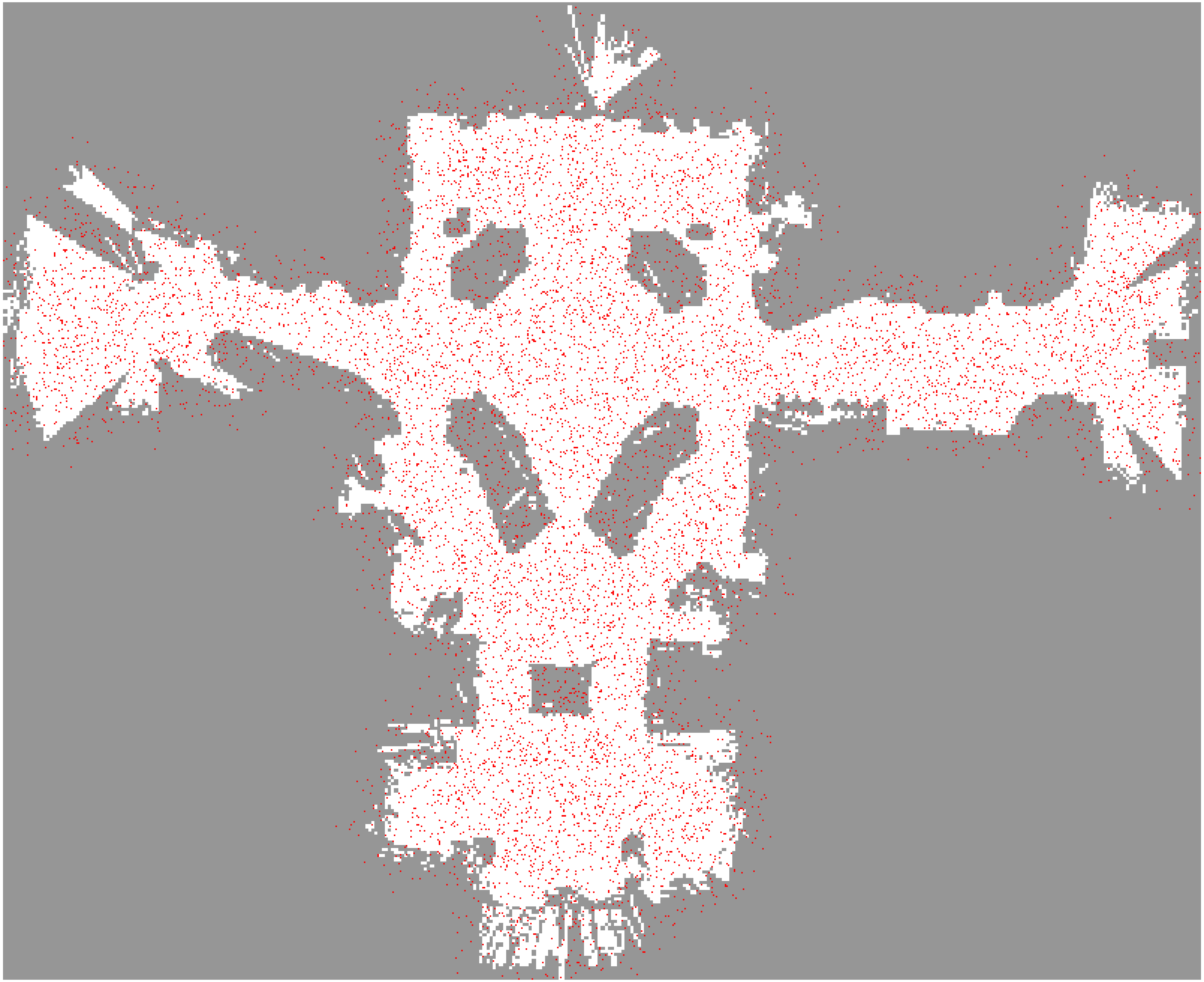
Proximity Sensor Model

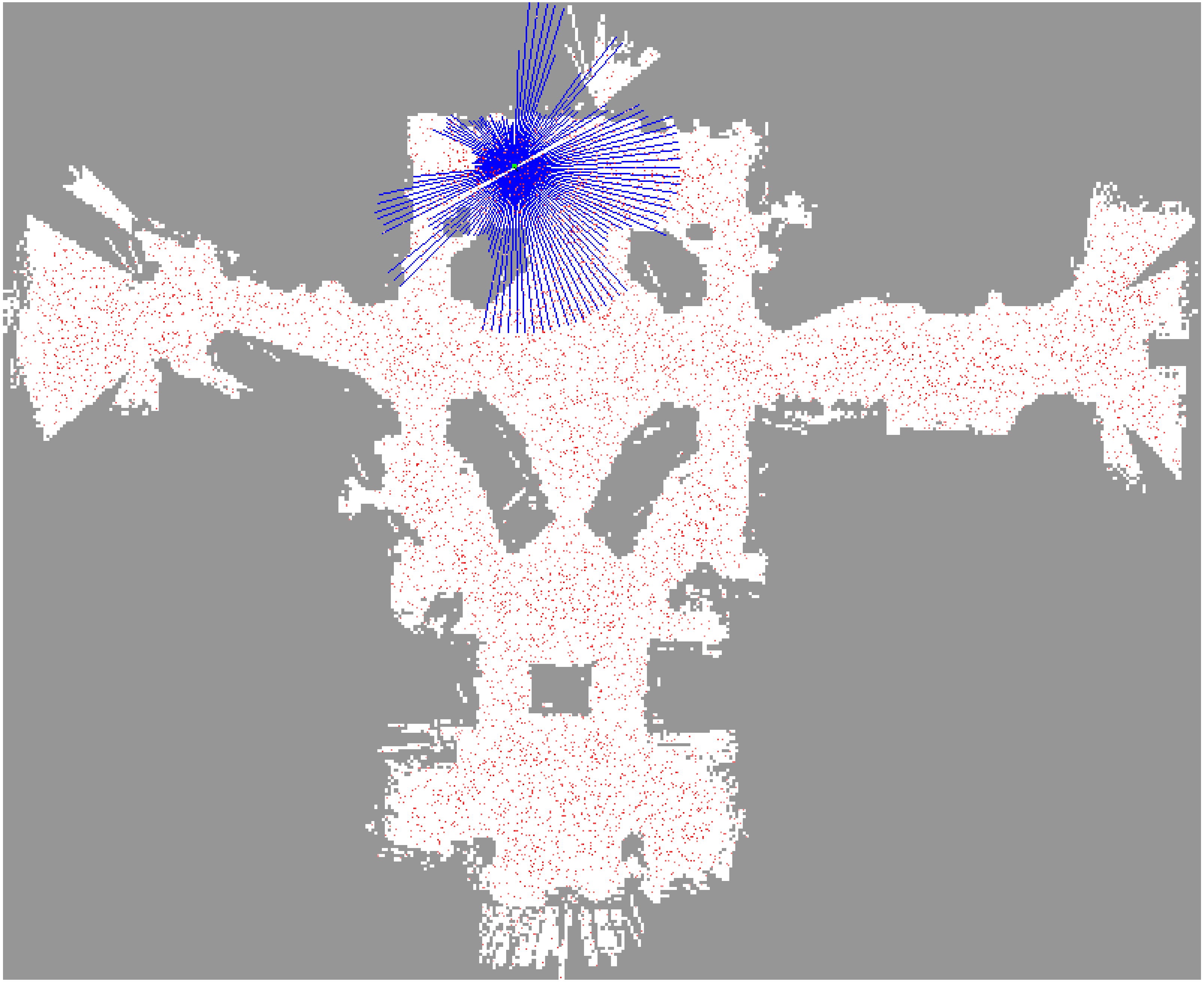


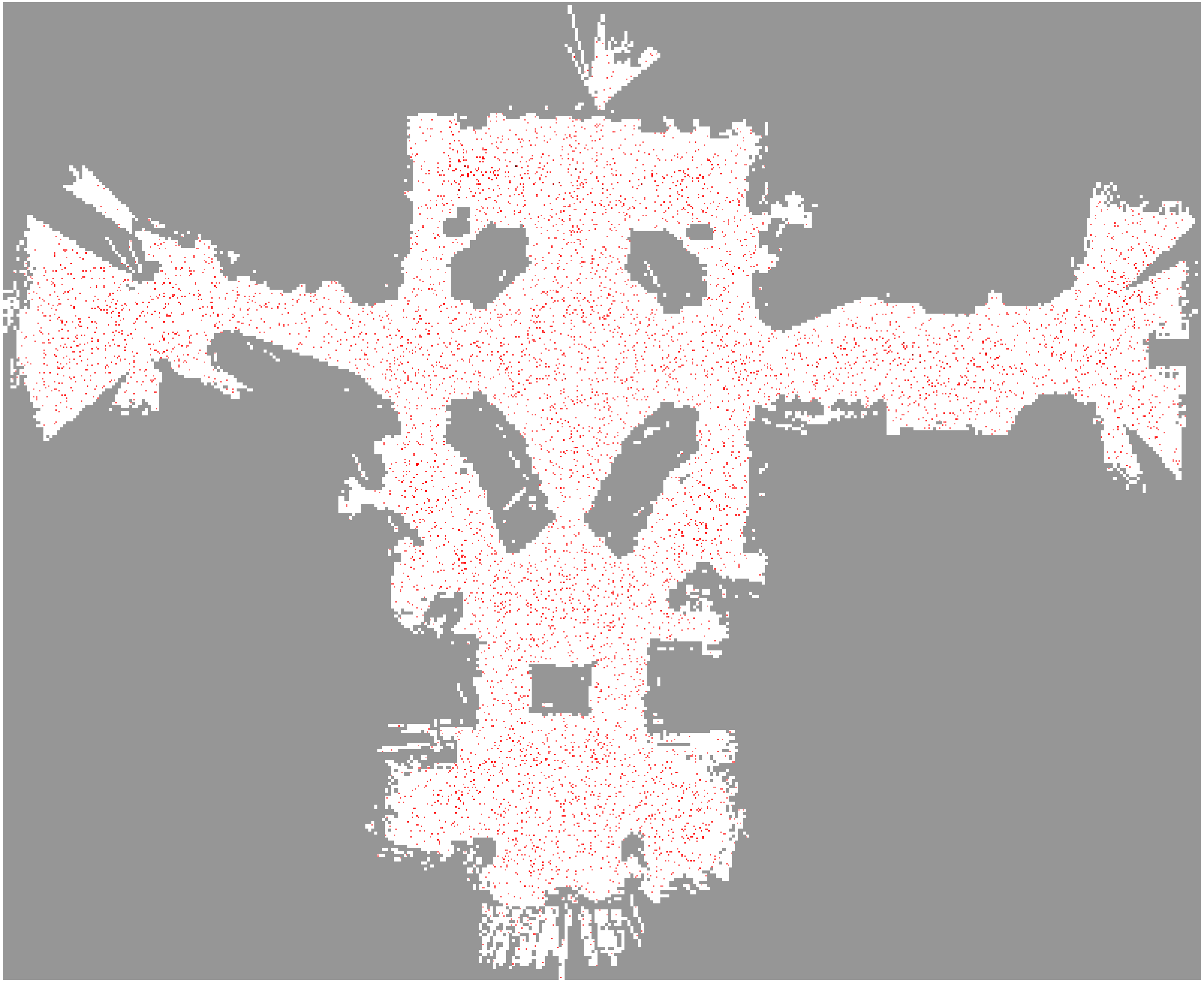
Laser sensor

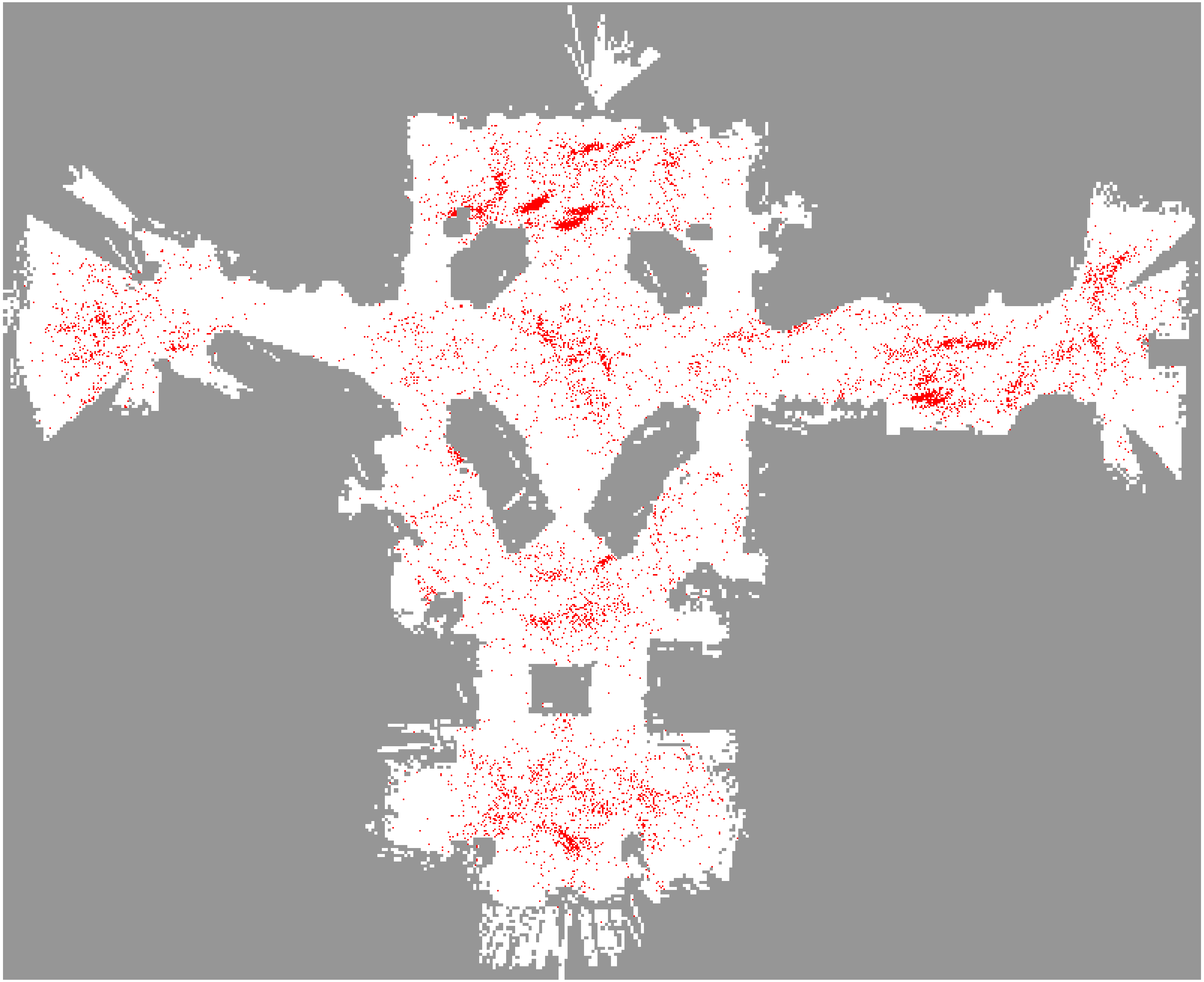


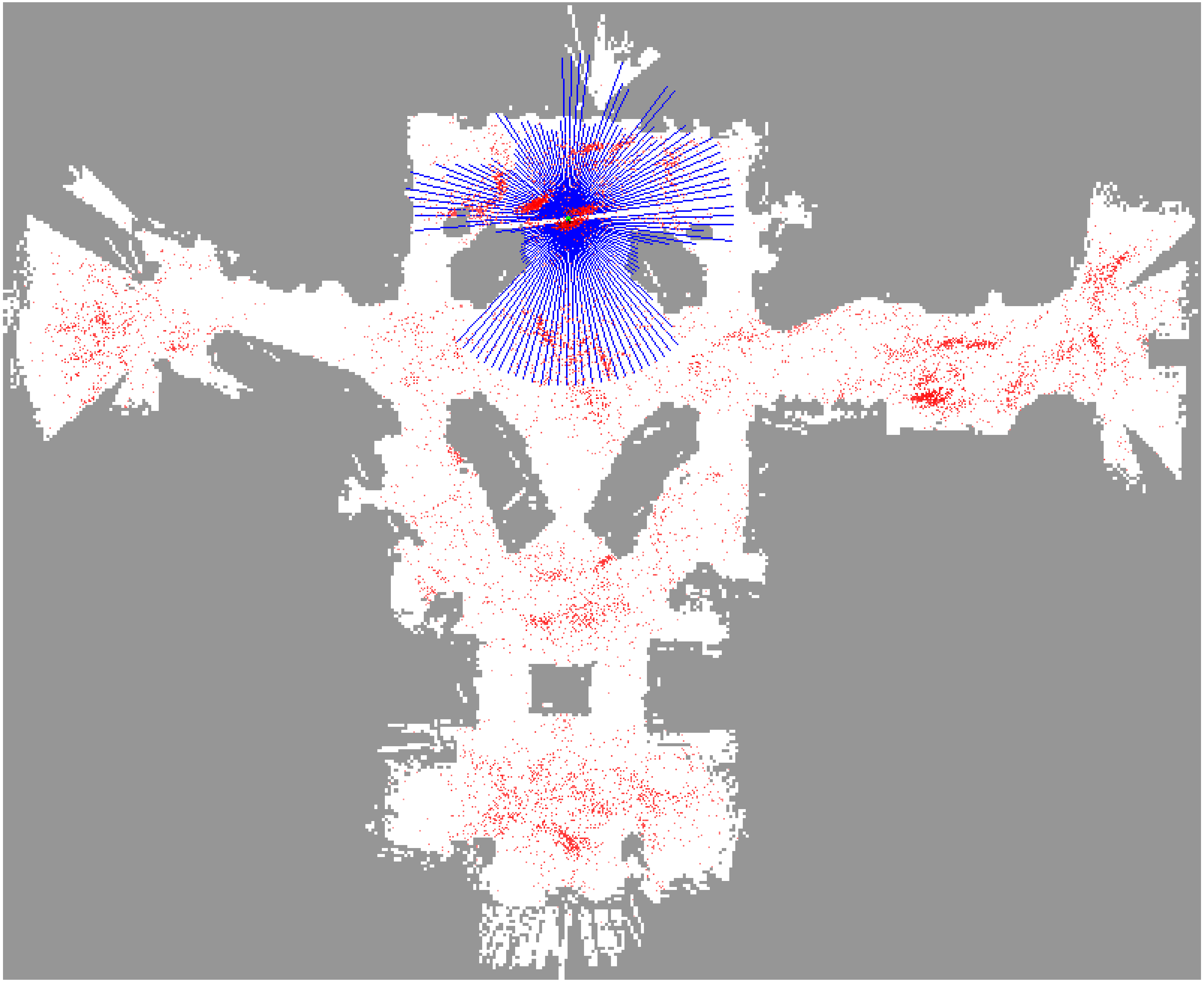
Sonar sensor

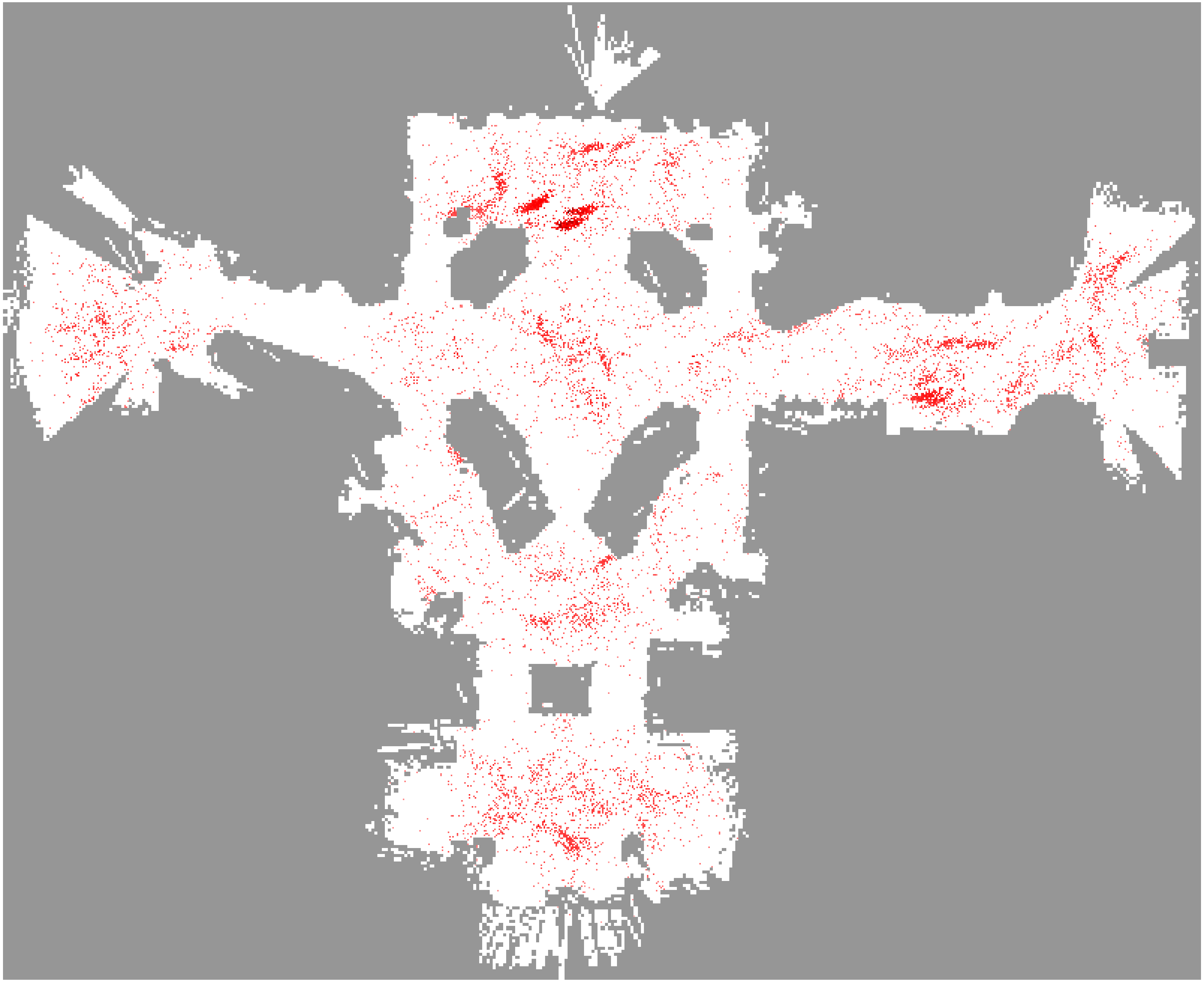


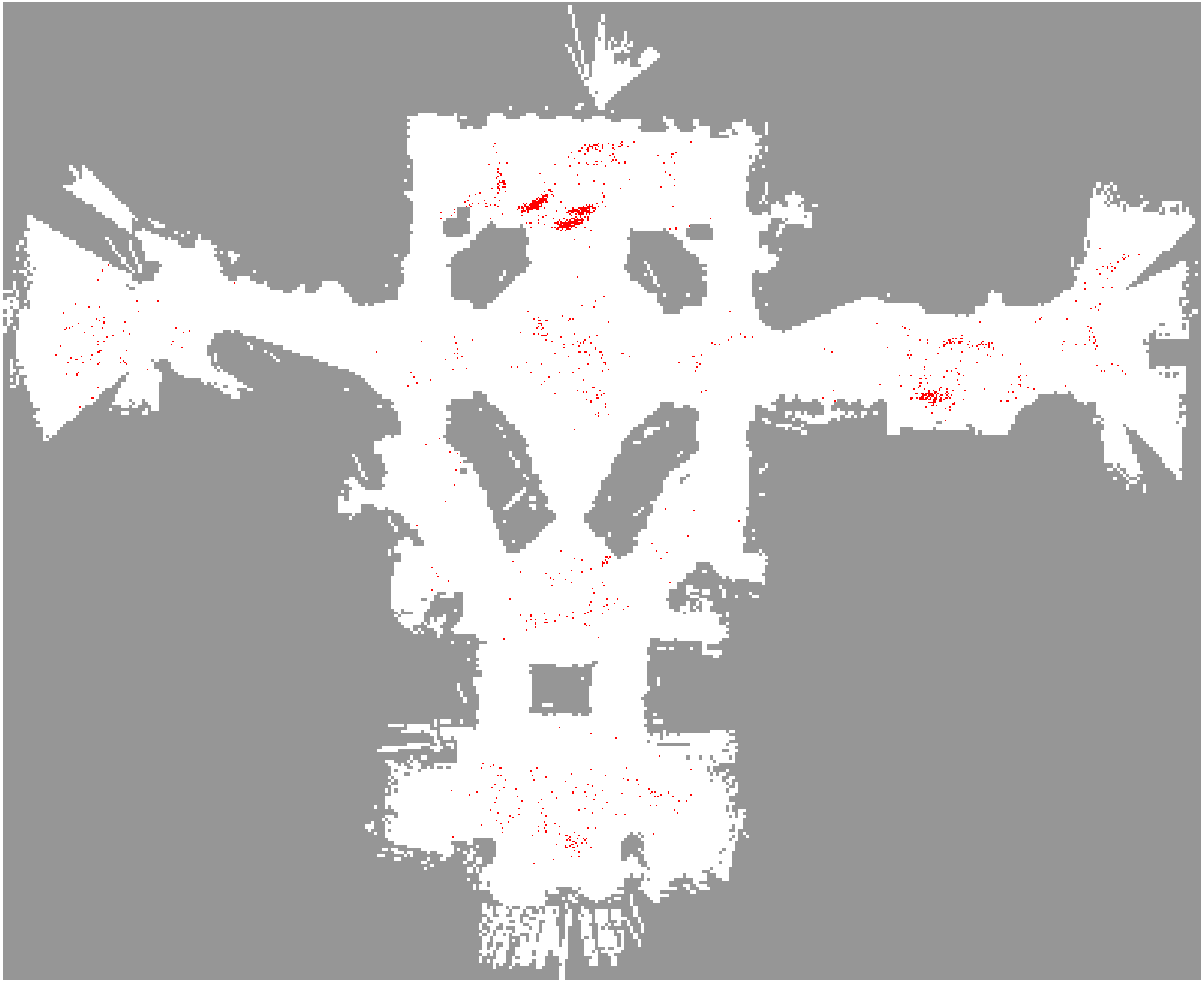


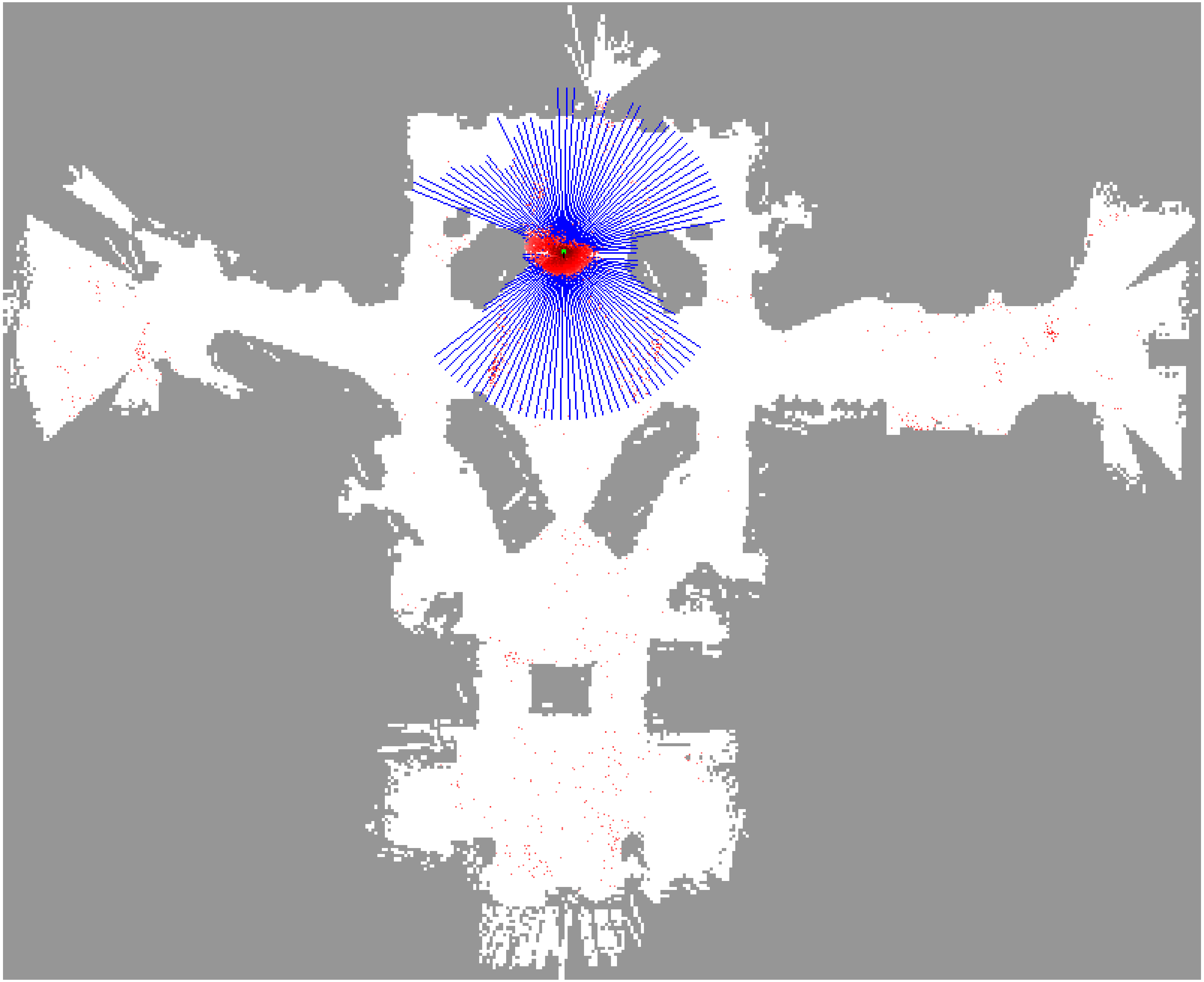


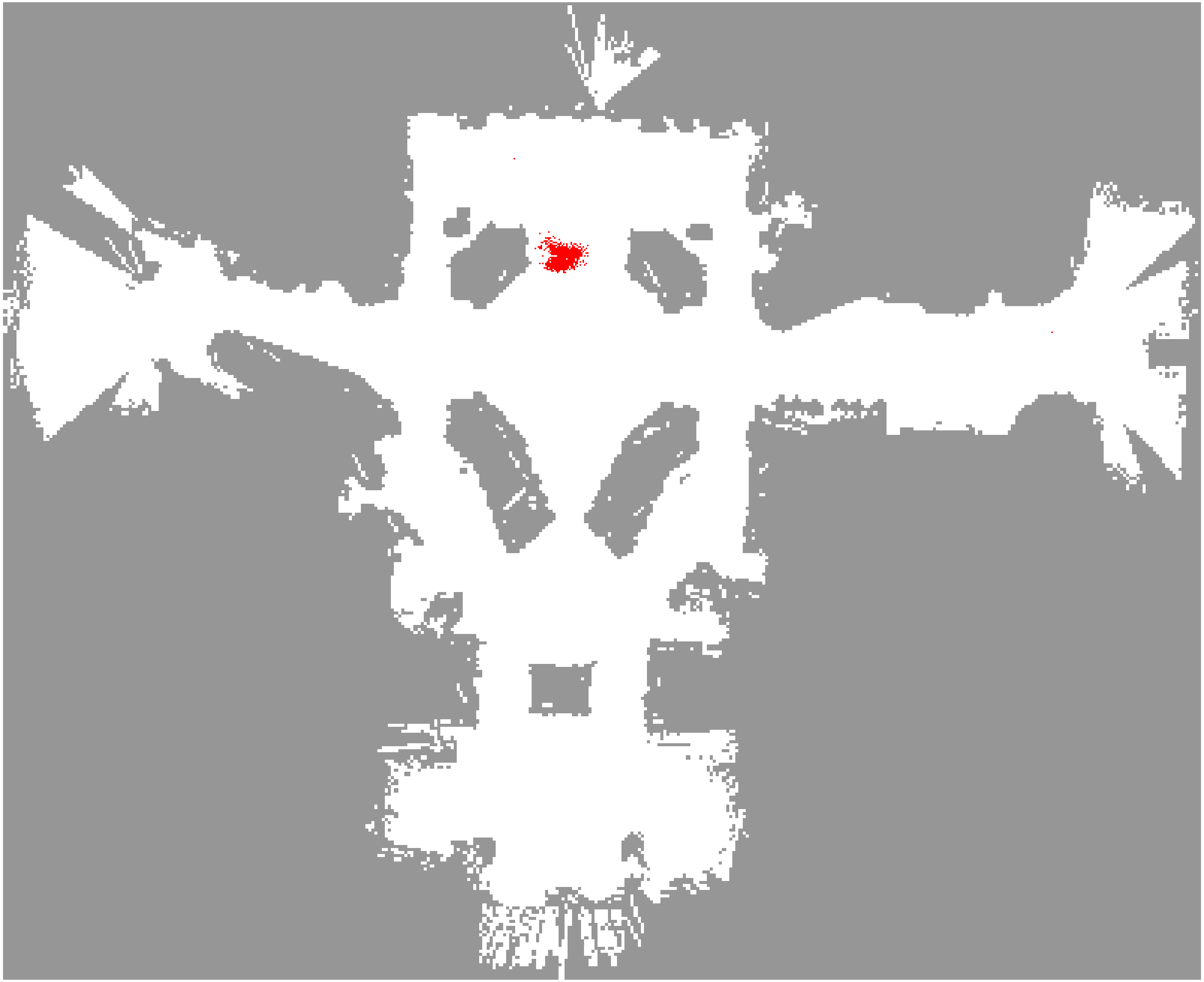


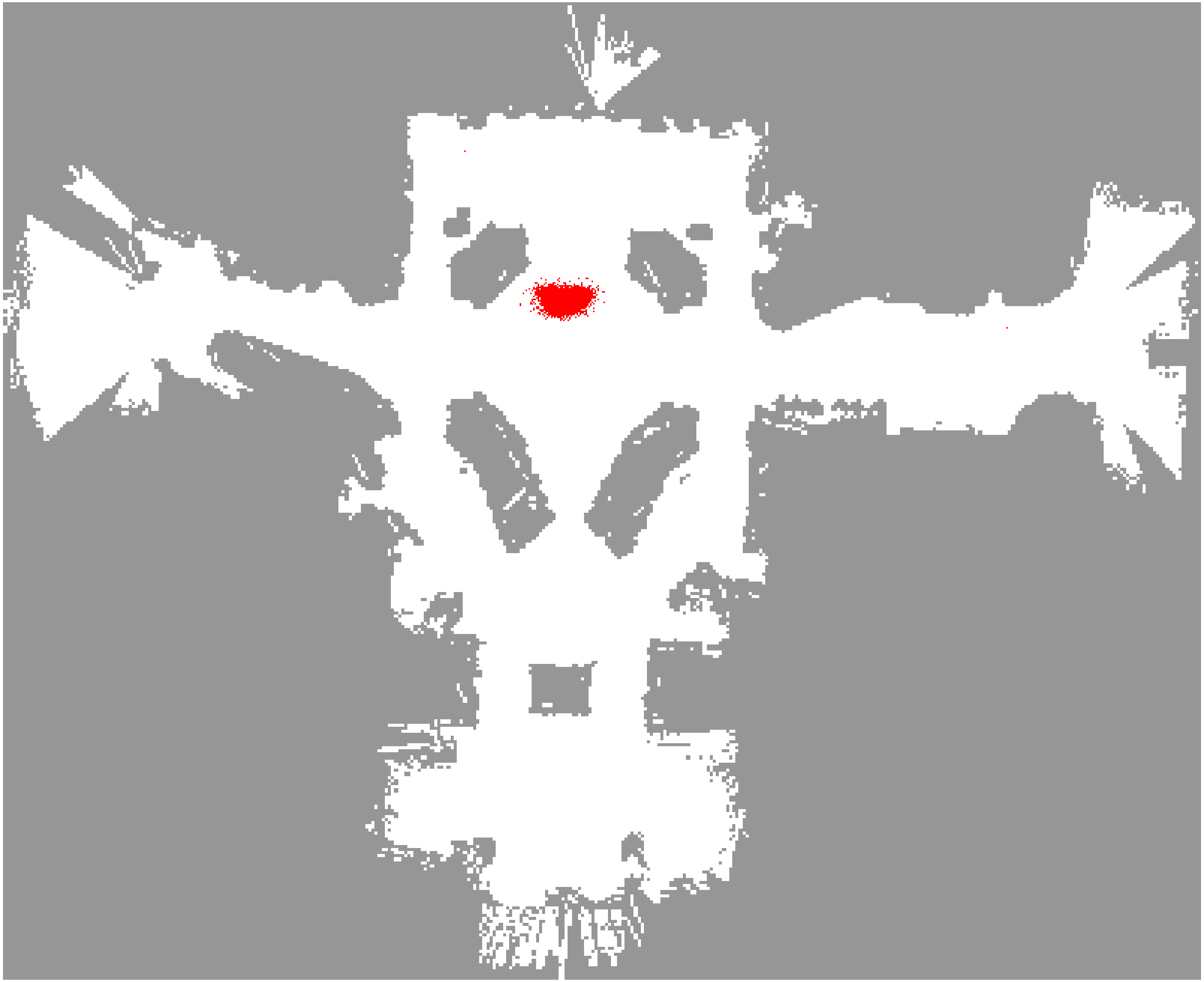


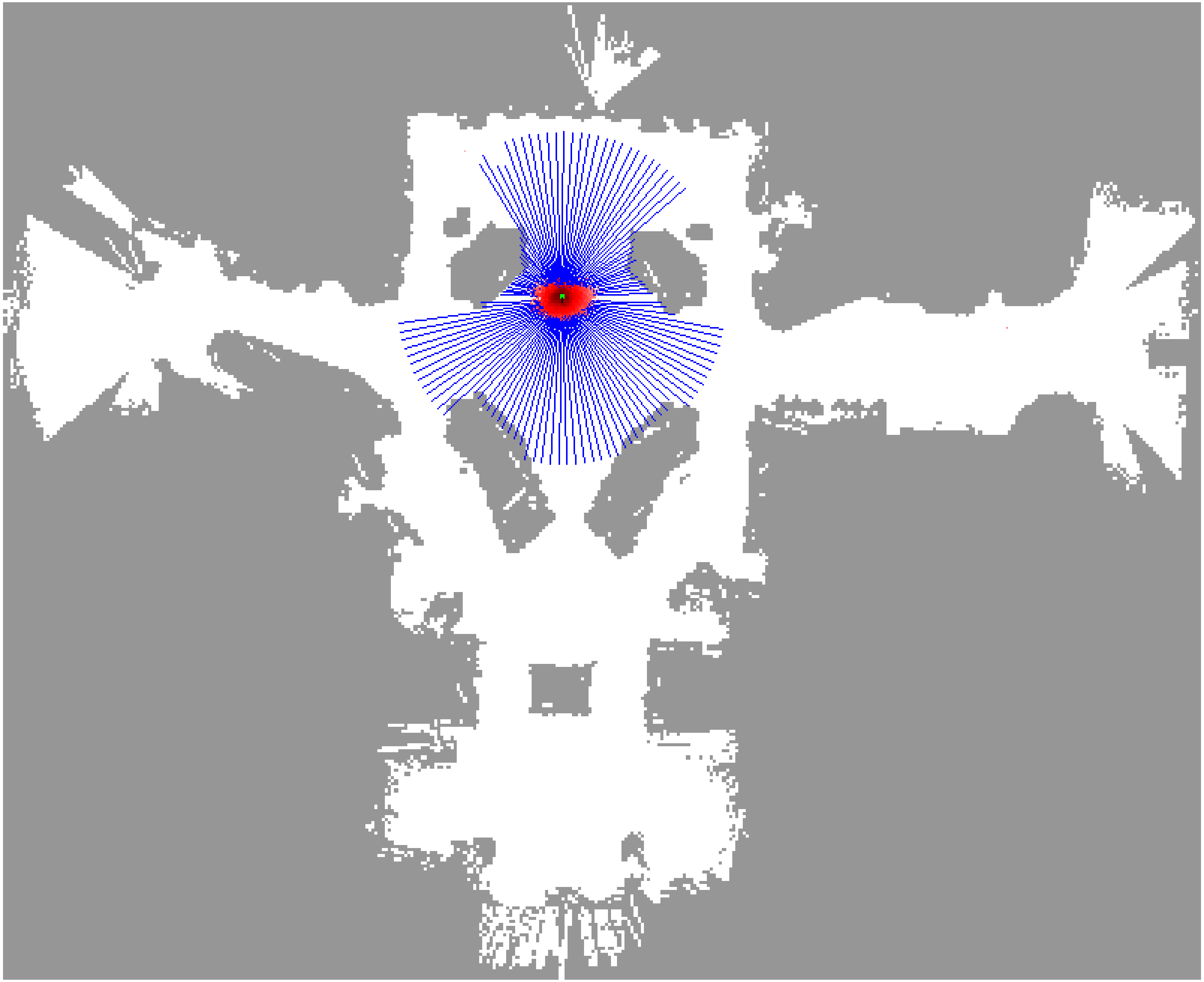


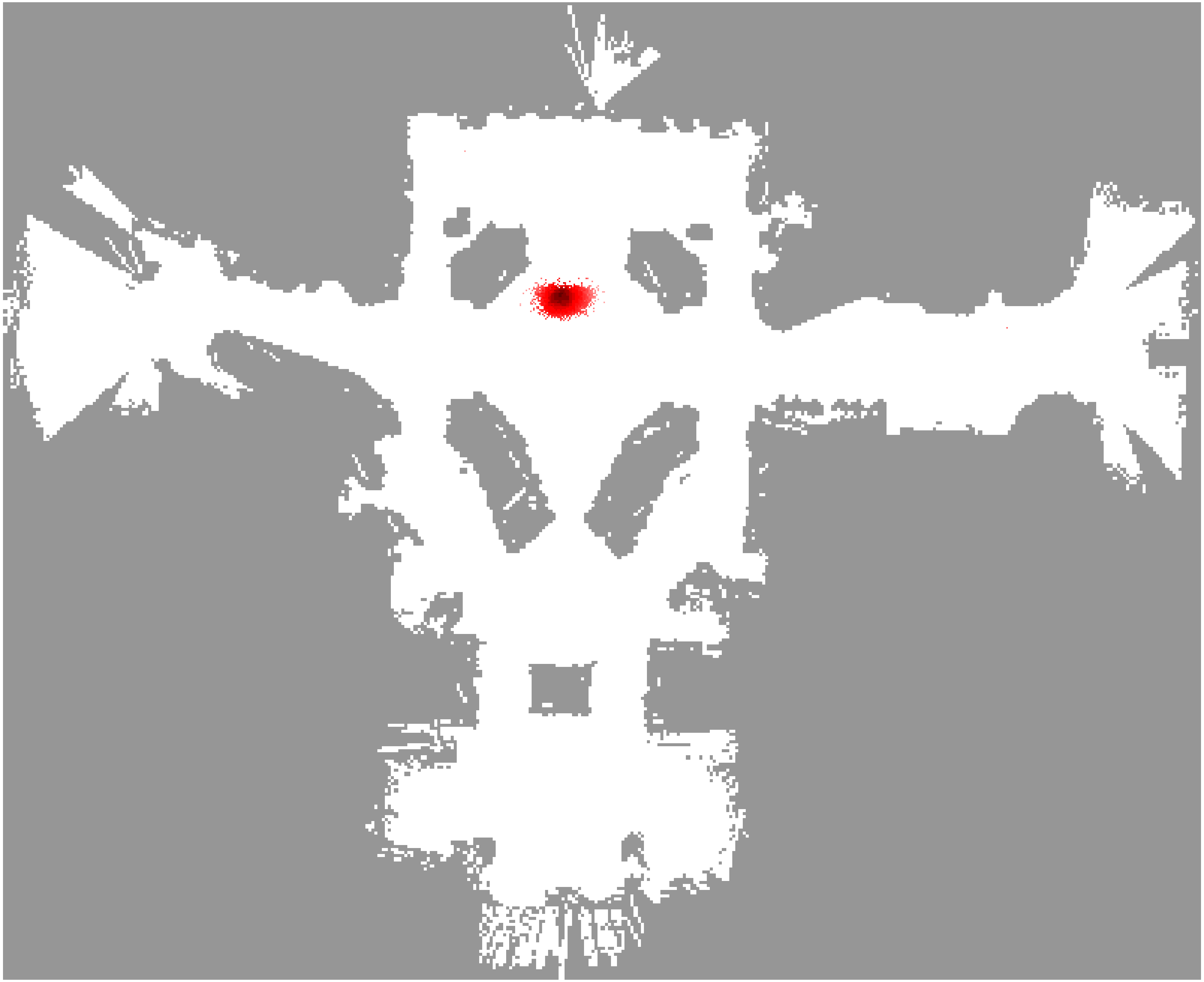


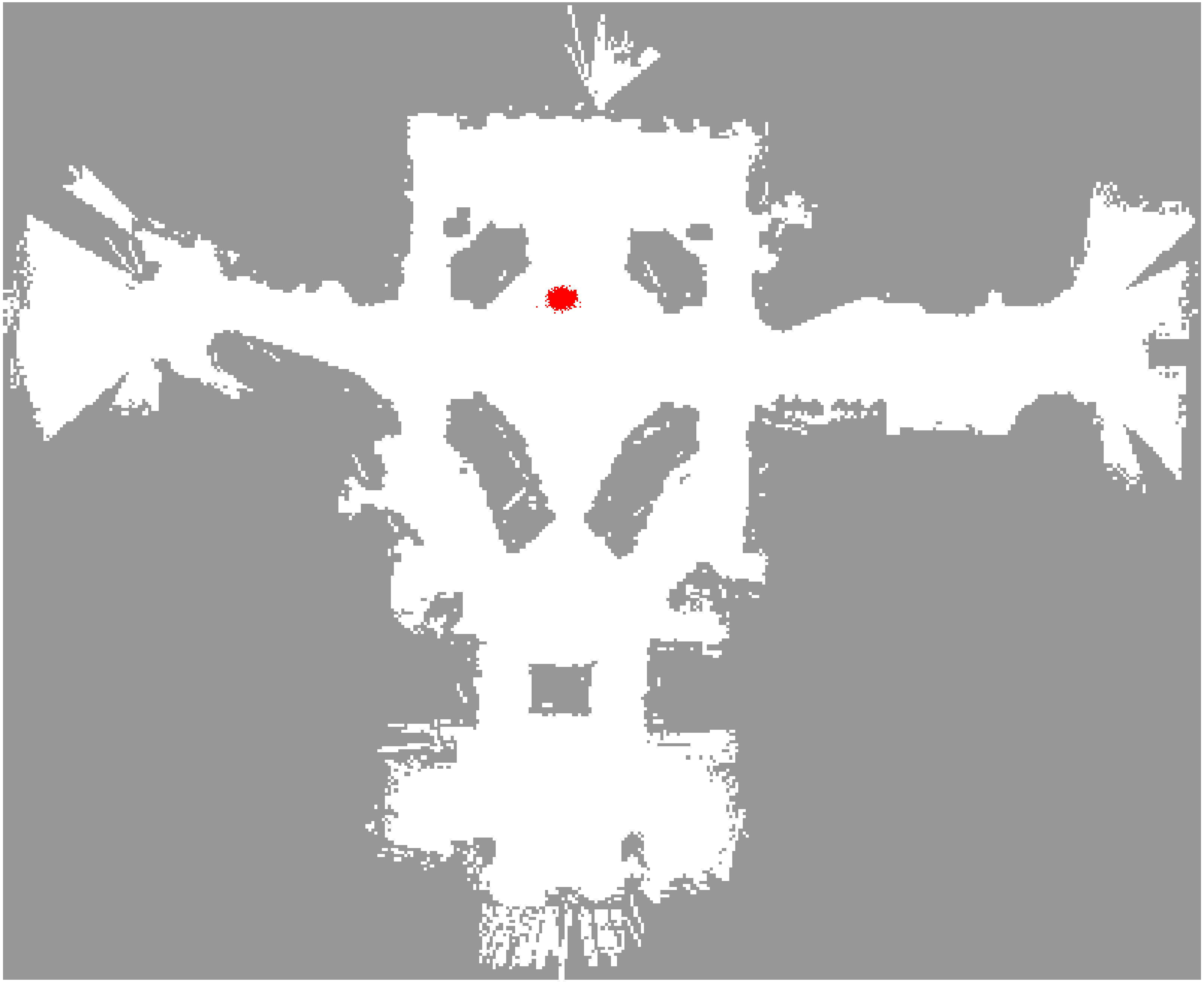


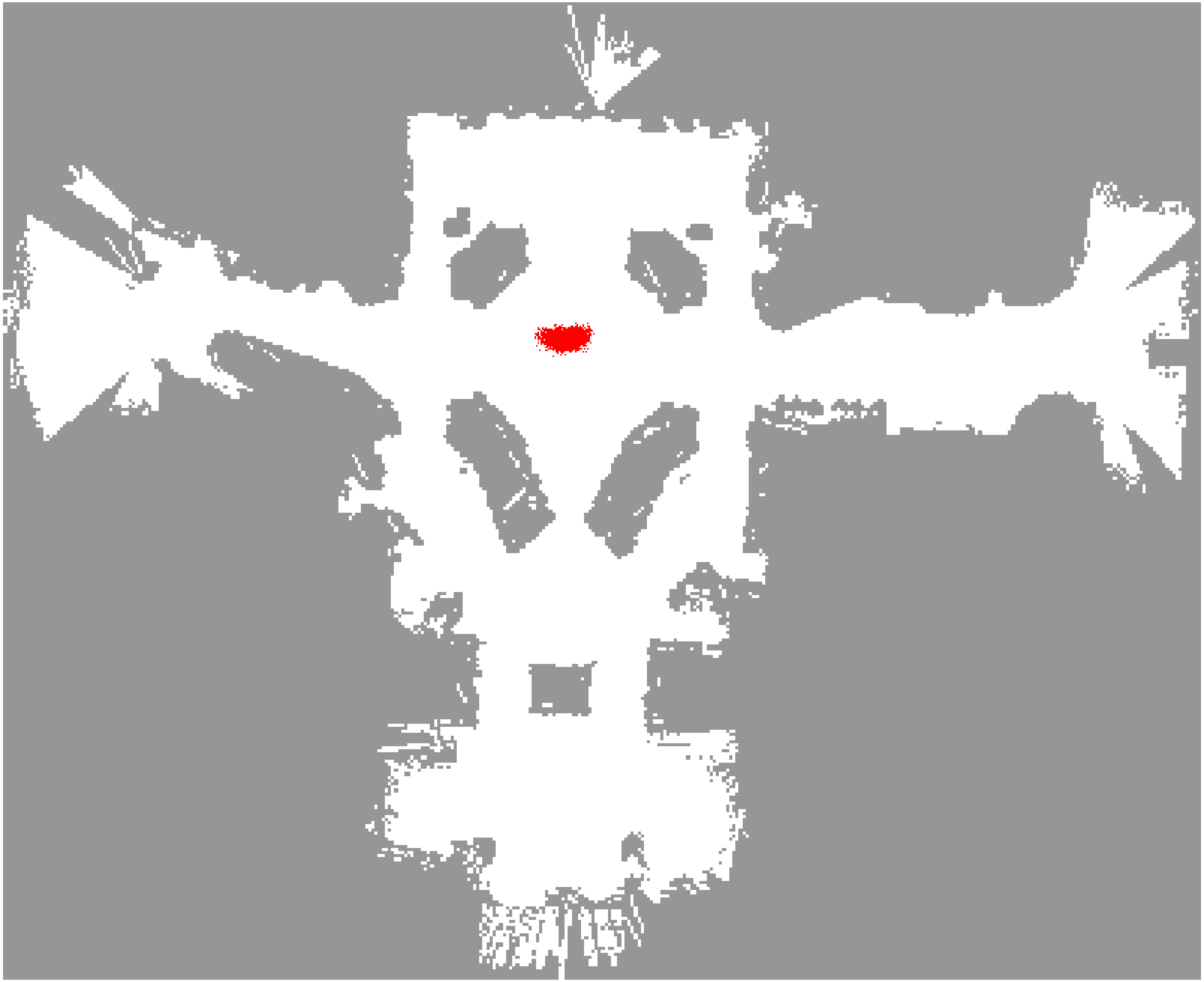


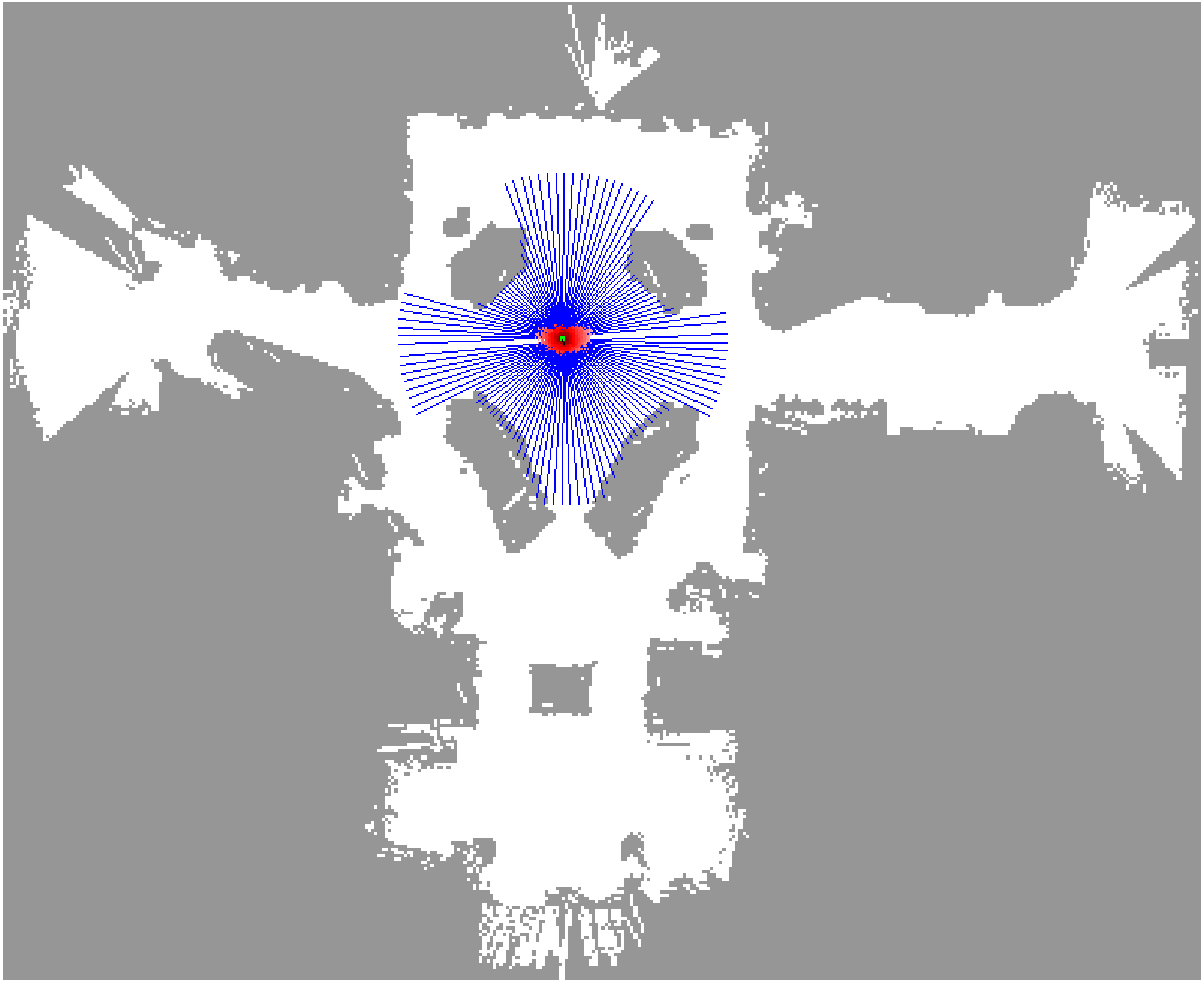


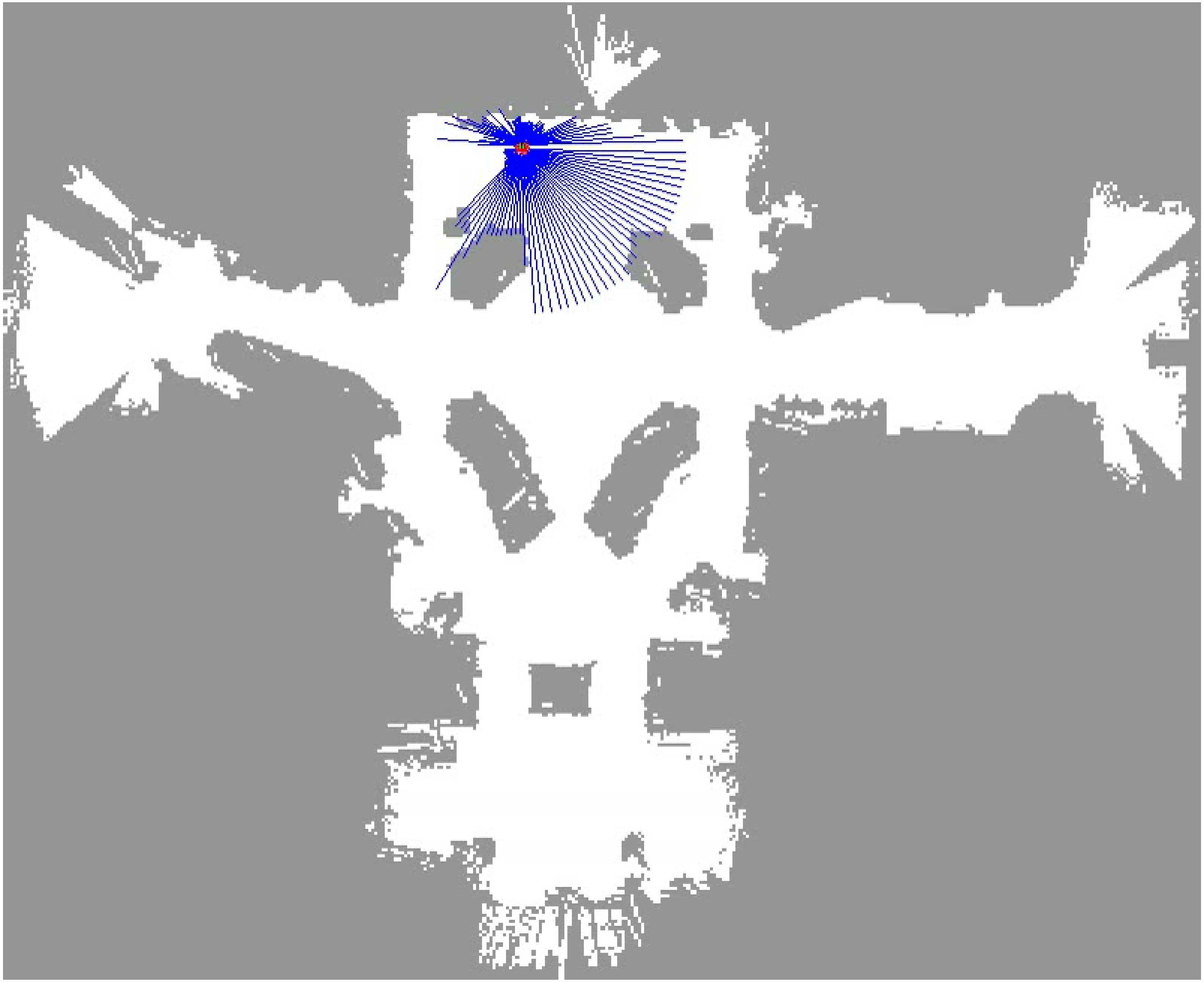




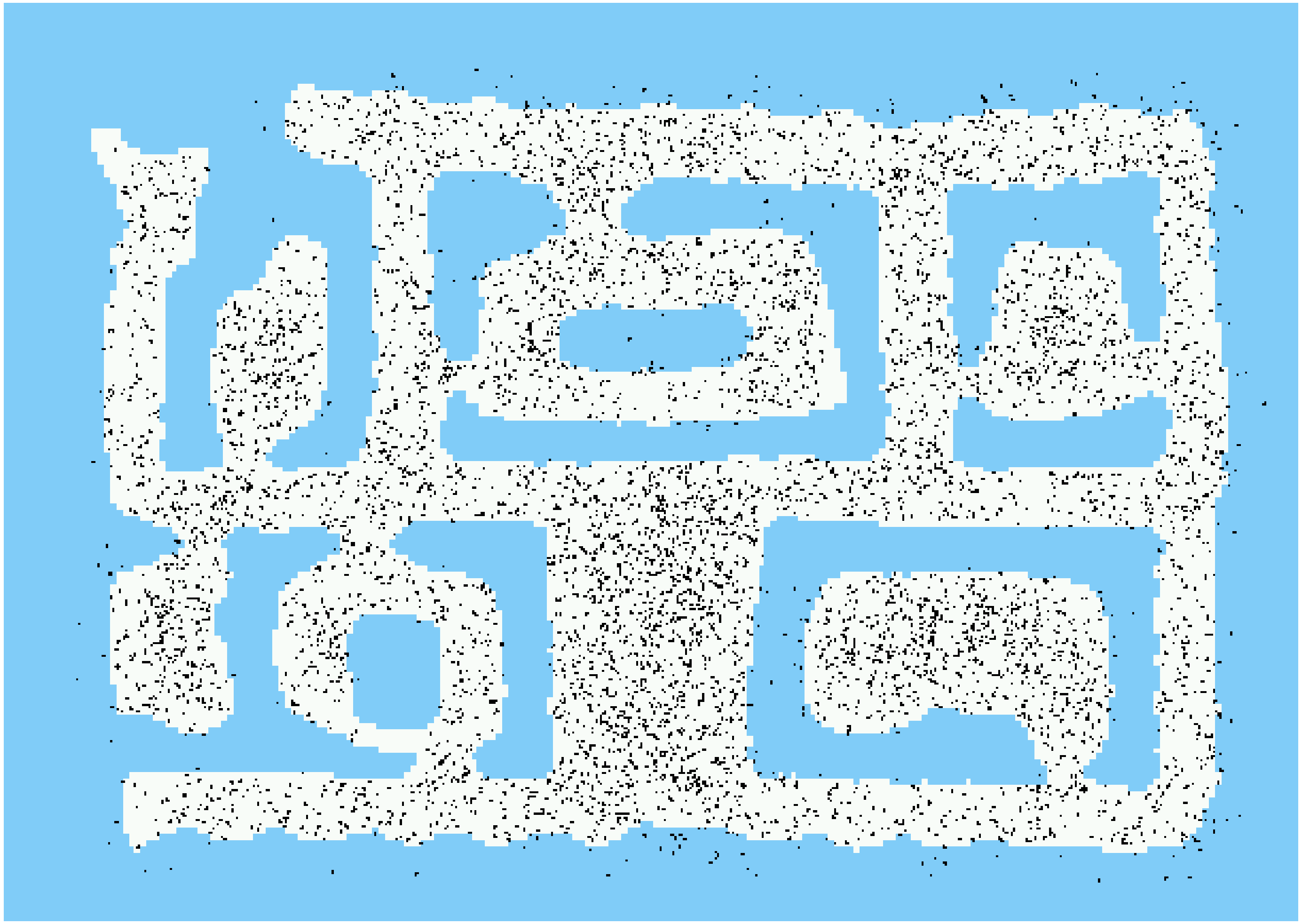




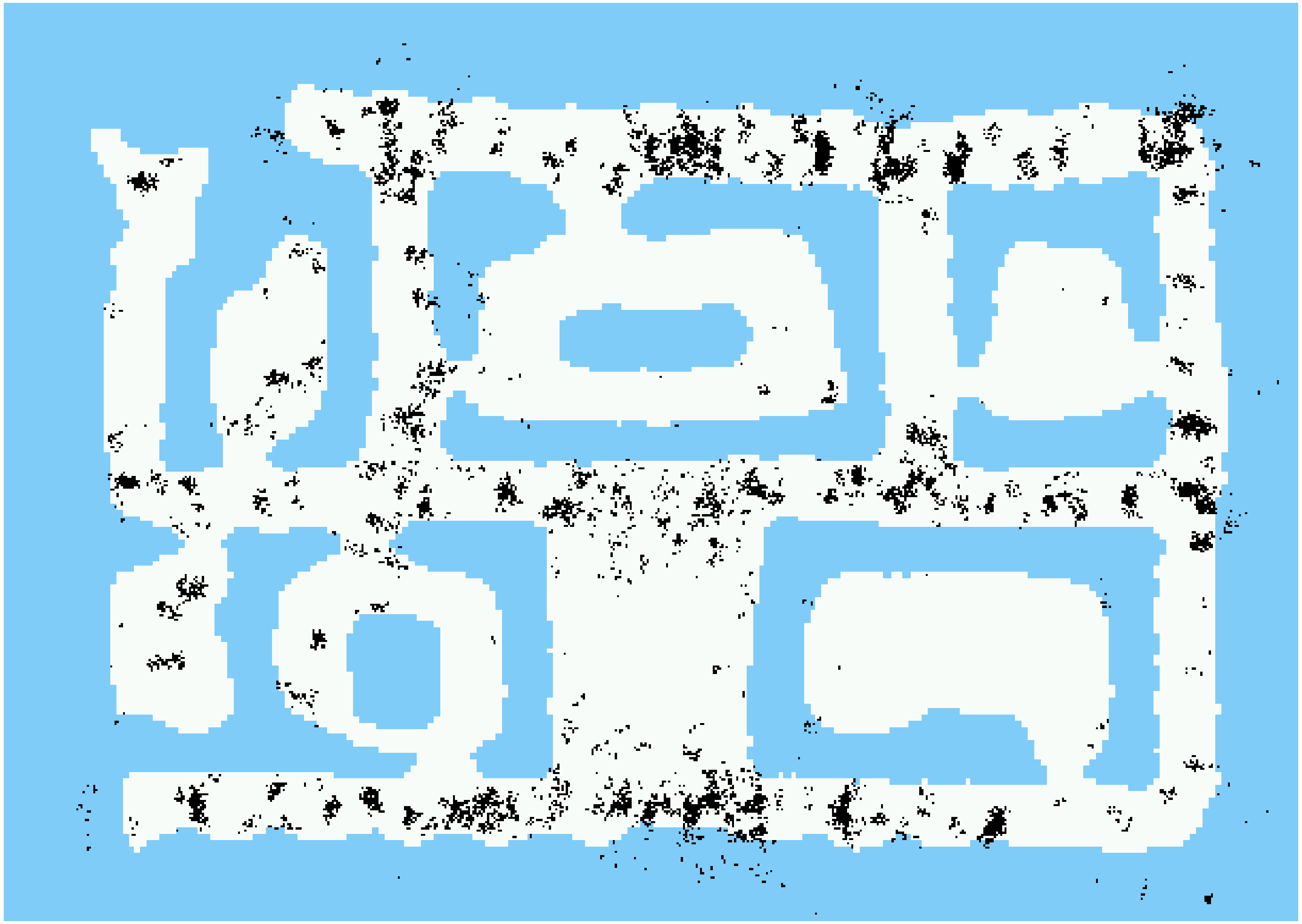




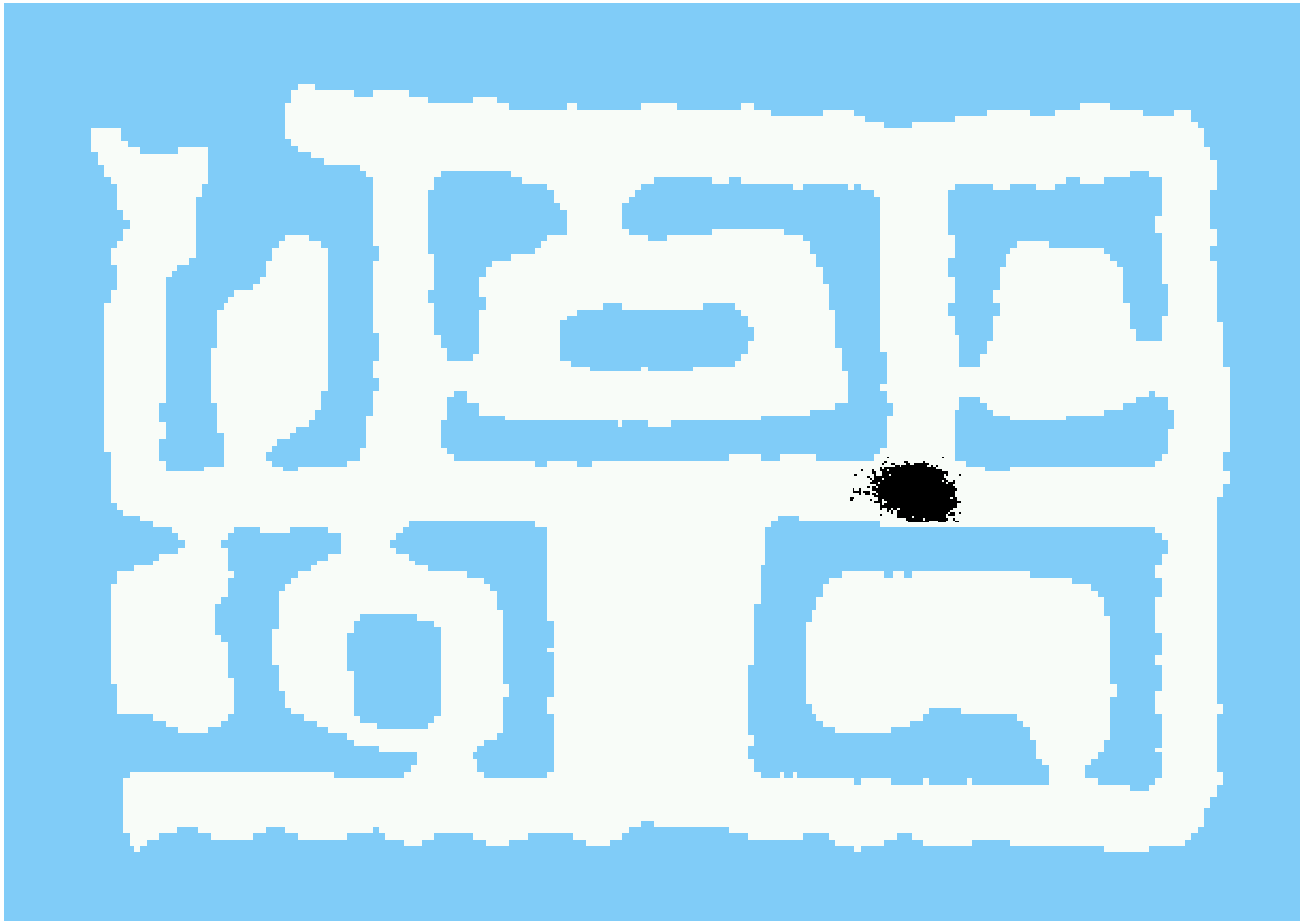
Initial Distribution



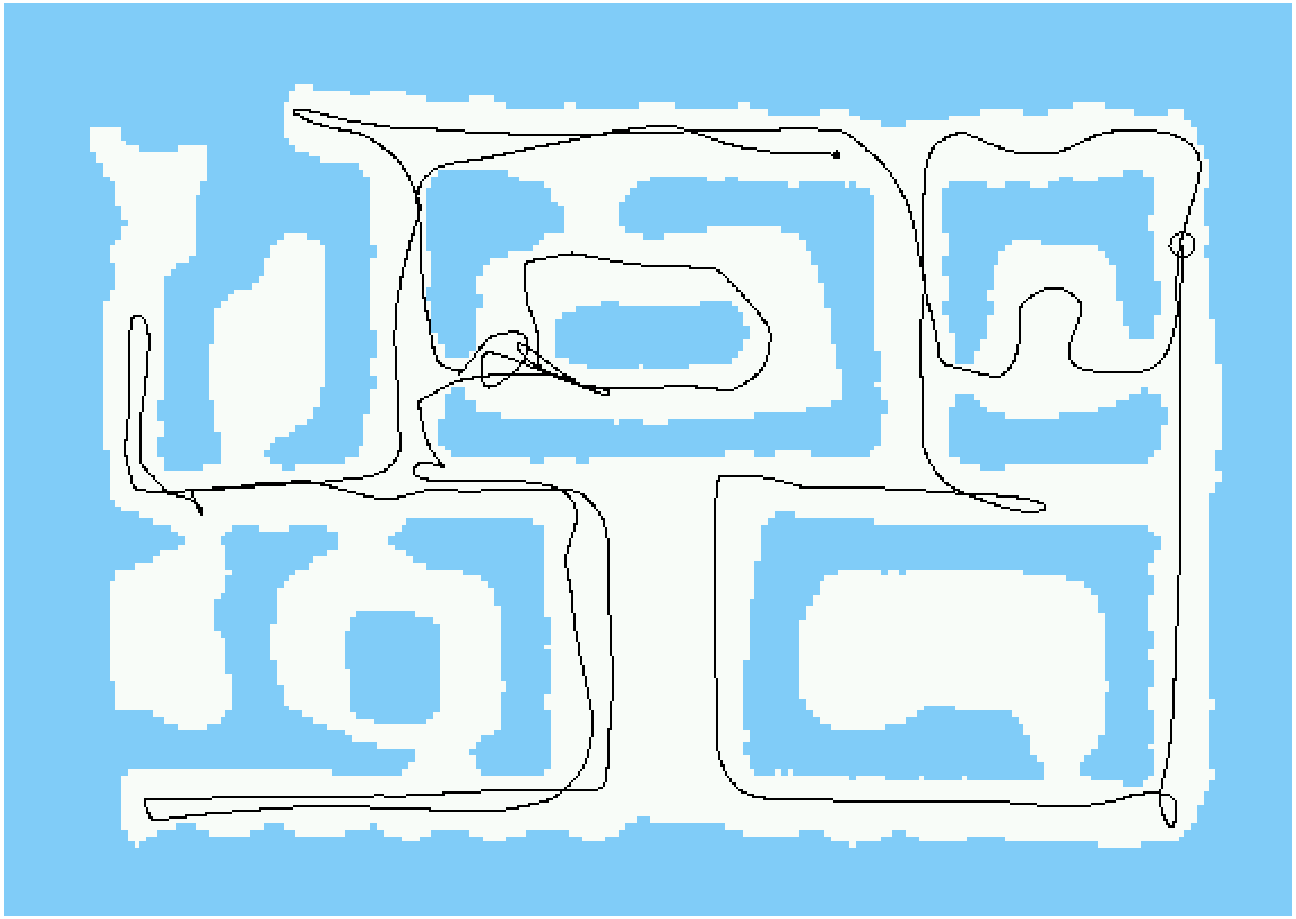
After Incorporating Ten Ultrasound Scans



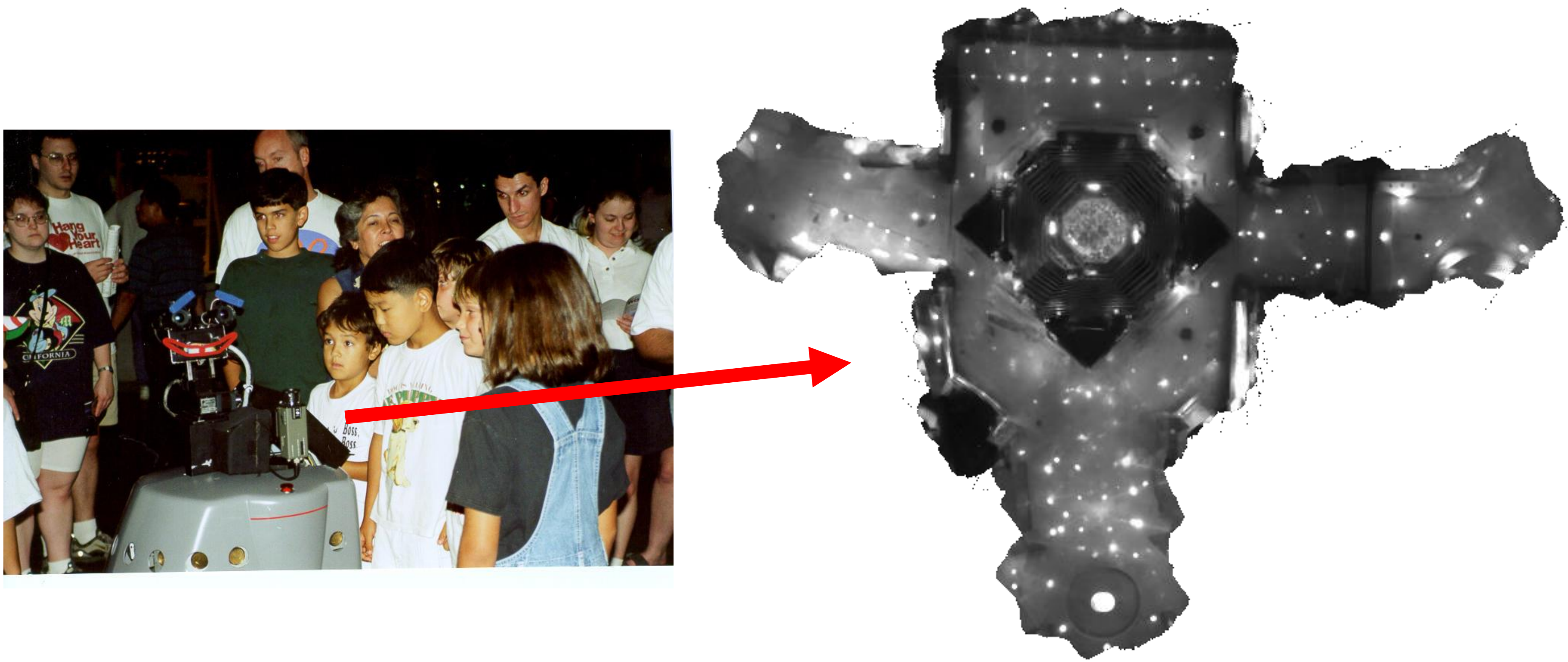
After Incorporating 65 Ultrasound Scans



Estimated Path

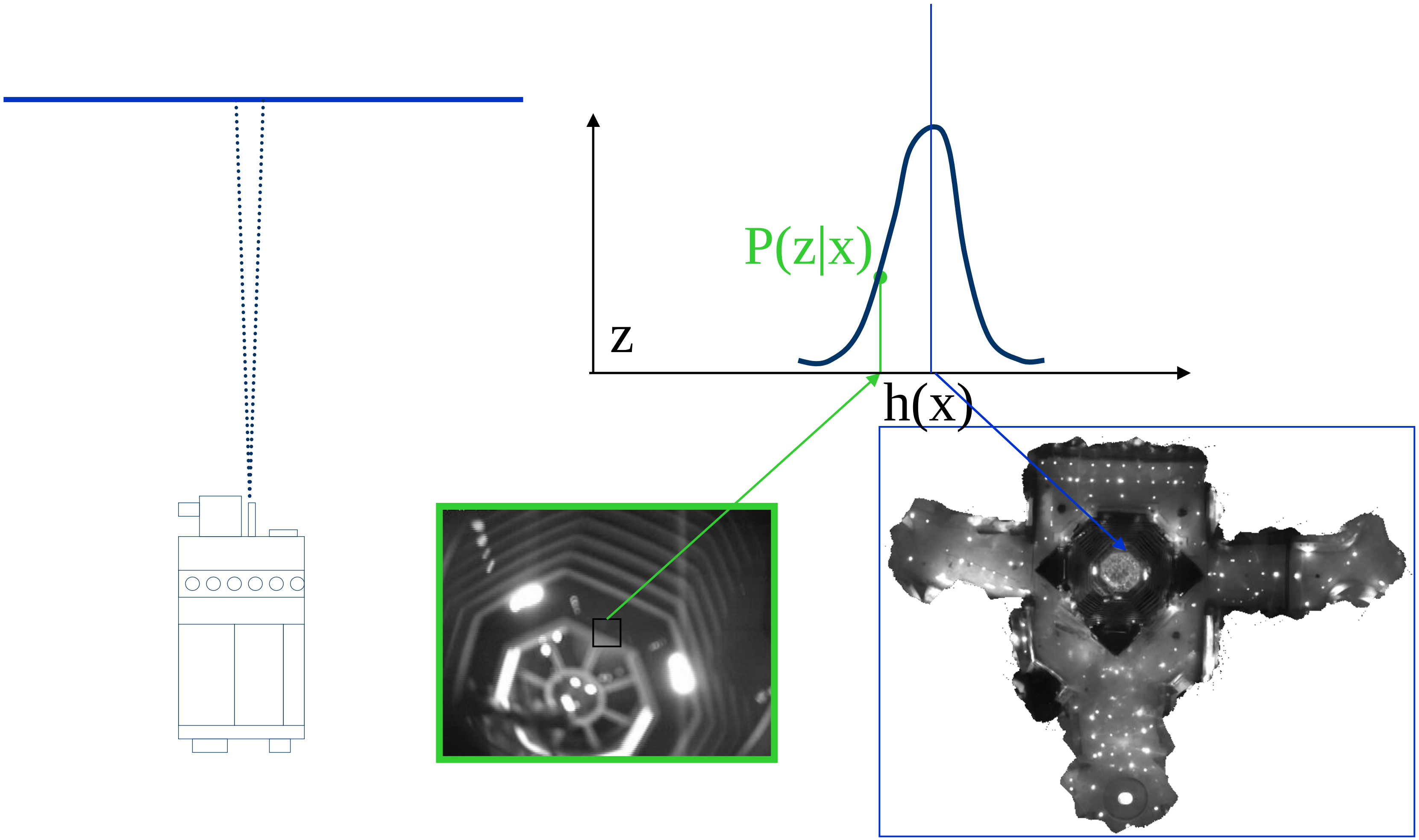


Using Ceiling Maps for Localization



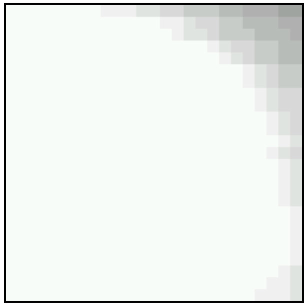
[Dellaert et al. 99]

Vision-based Localization

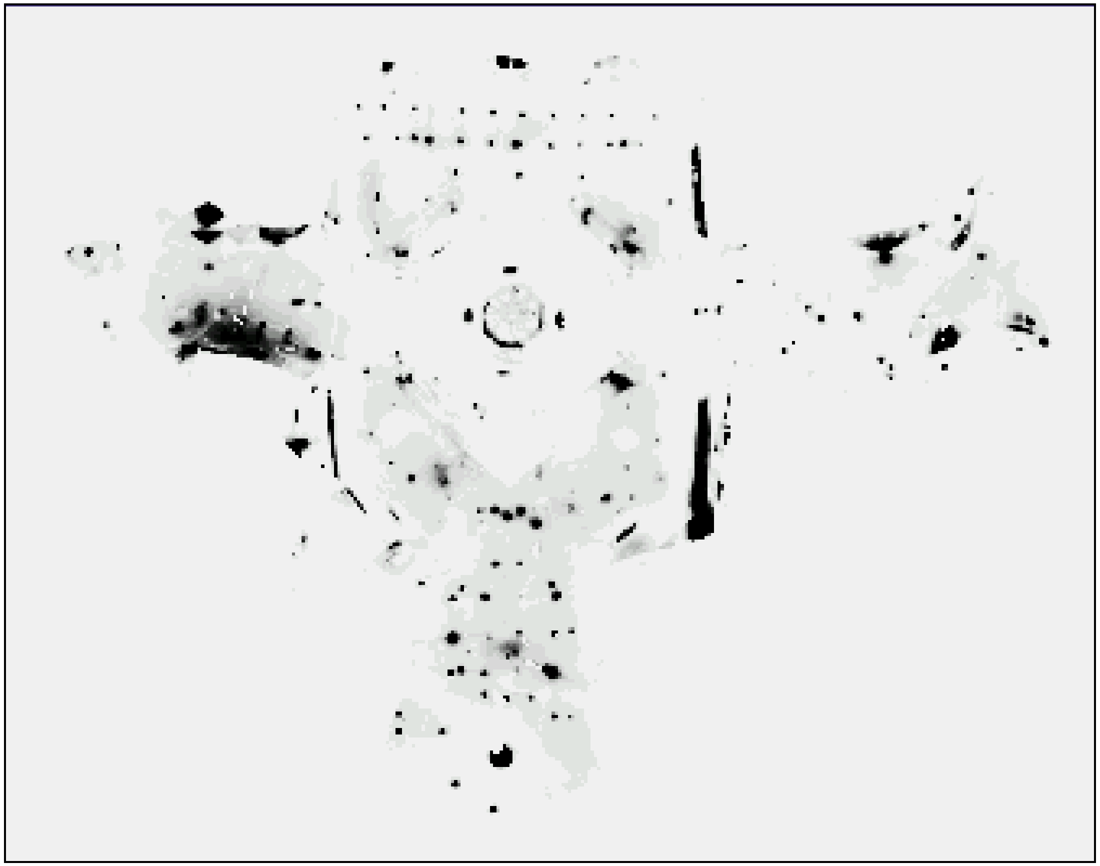


Under a Light

Measurement z:

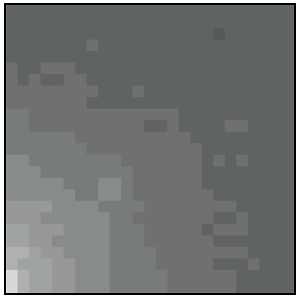


P(z|x):

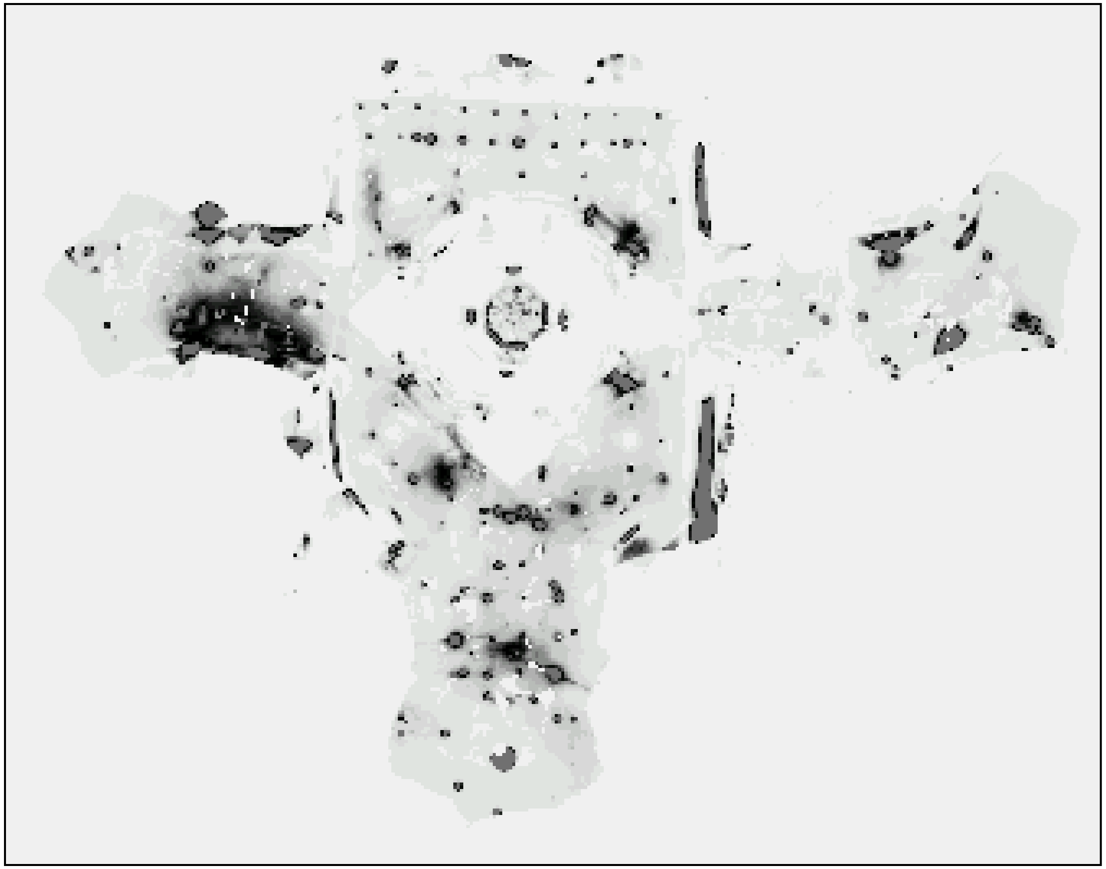


Next to a Light

Measurement z:



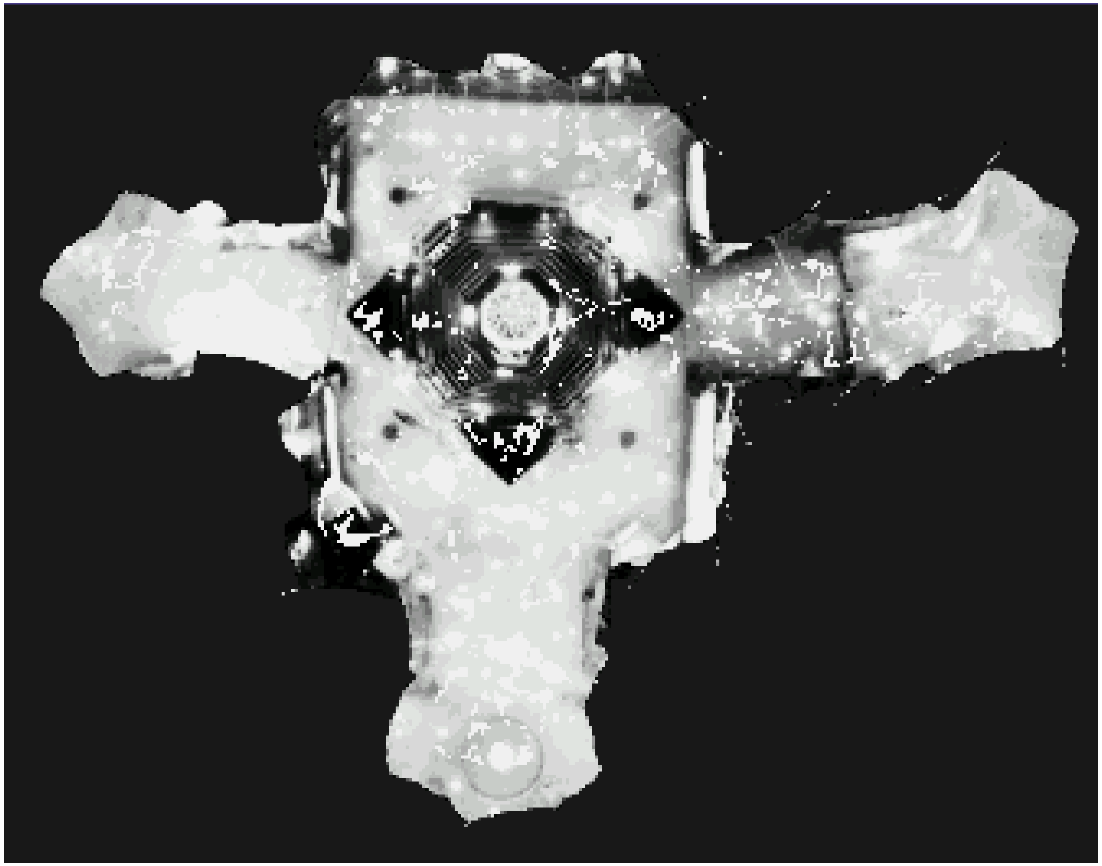
$P(z|x)$:



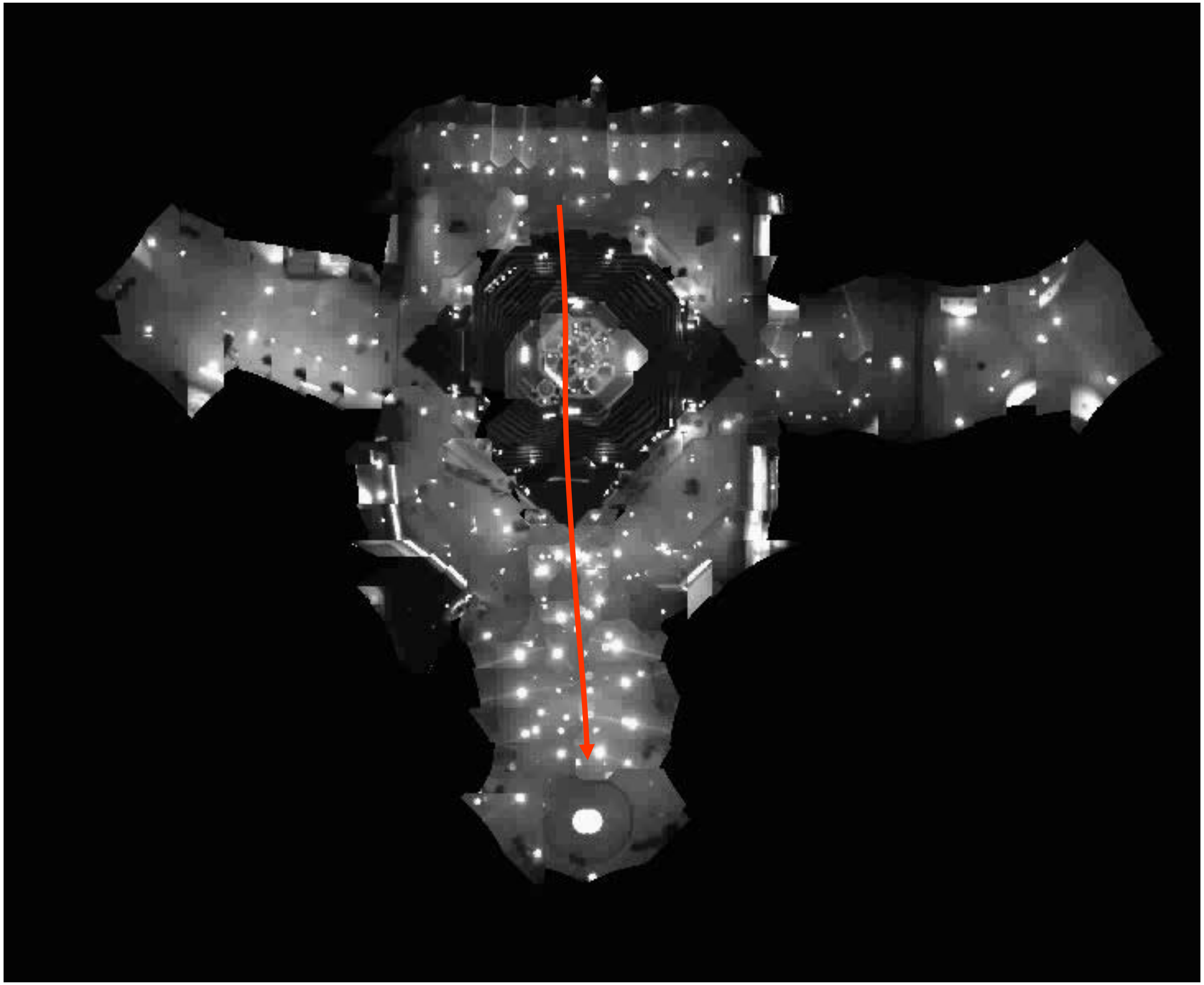
Elsewhere

Measurement z:

P(z|x):



Global Localization Using Vision



Summary – Particle Filters

- Particle filters are an implementation of recursive Bayesian filtering
- They represent the posterior by a set of weighted samples
- They can model non-Gaussian distributions
- Proposal to draw new samples
- Weight to account for the differences between the proposal and the target
- Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter

Summary – Monte Carlo Localization

- In the context of localization, the particles are propagated according to the motion model.
- They are then weighted according to the likelihood of the observations.
- In a re-sampling step, new particles are drawn with a probability proportional to the likelihood of the observation.