

Introduction to Mobile Robotics

Bayes Filter – Particle Filter and Monte Carlo Localization

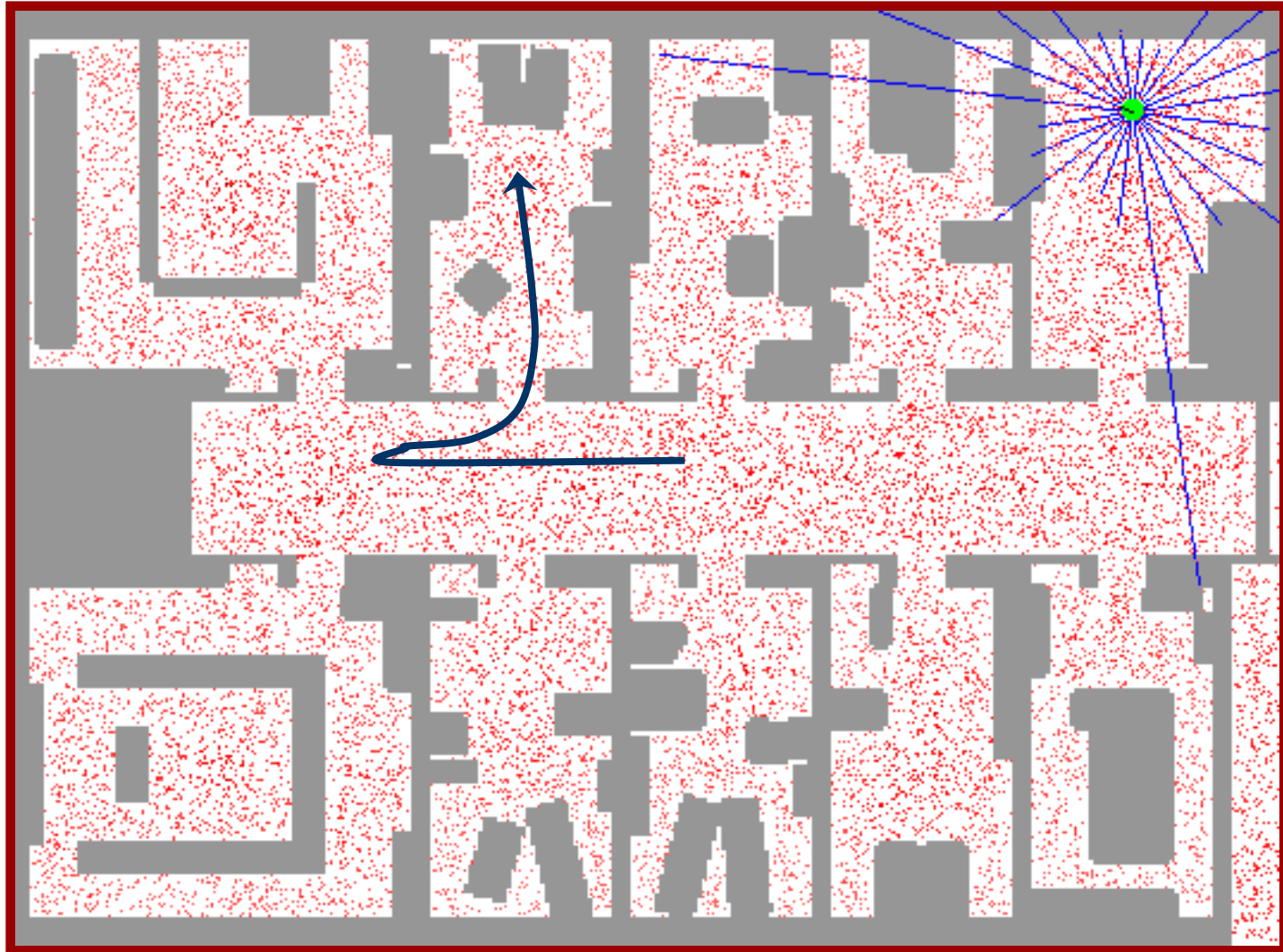
Wolfram Burgard, Diego Tipaldi



Motivation

- Recall: Discrete filter
 - Discretize the continuous state space
 - High memory complexity
 - Fixed resolution (does not adapt to the belief)
- Particle filters are a way to **efficiently** represent **non-Gaussian distribution**
- Basic principle
 - Set of state hypotheses (“particles”)
 - Survival-of-the-fittest

Sample-based Localization (sonar)



Mathematical Description

- Set of weighted samples

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

State hypothesis

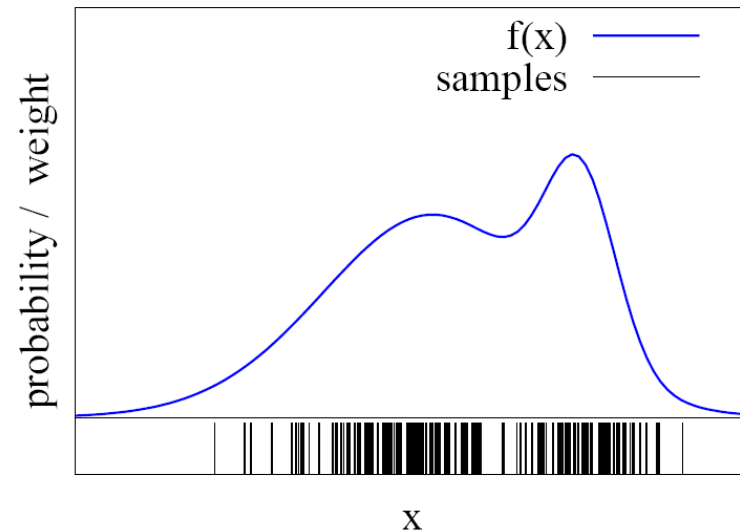
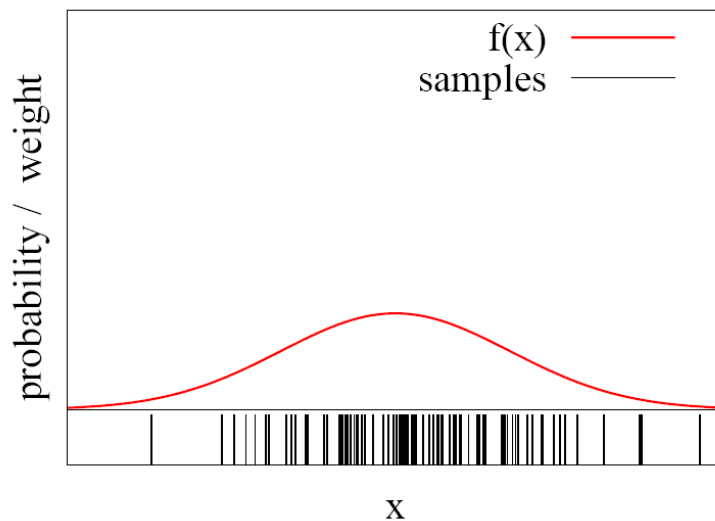
Importance weight

- The samples represent the posterior

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

Function Approximation

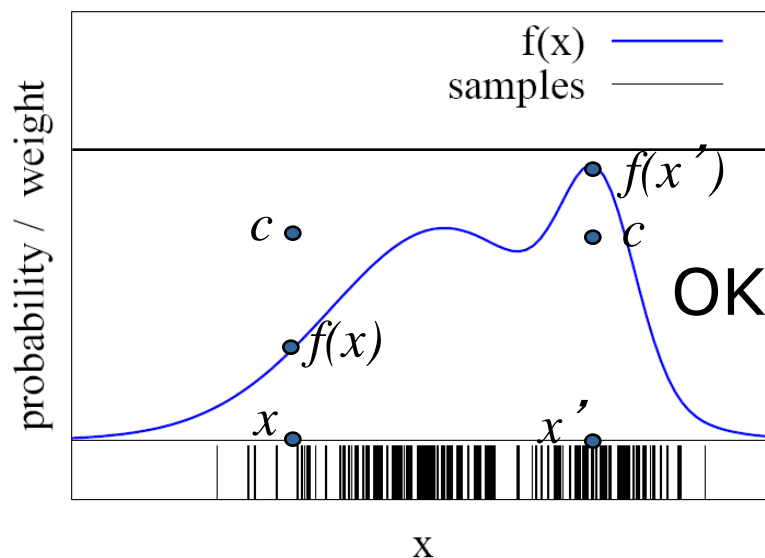
- Particle sets can be used to approximate functions



- The more particles fall into an interval, the higher the probability of that interval
- How to draw samples from a function/distribution?

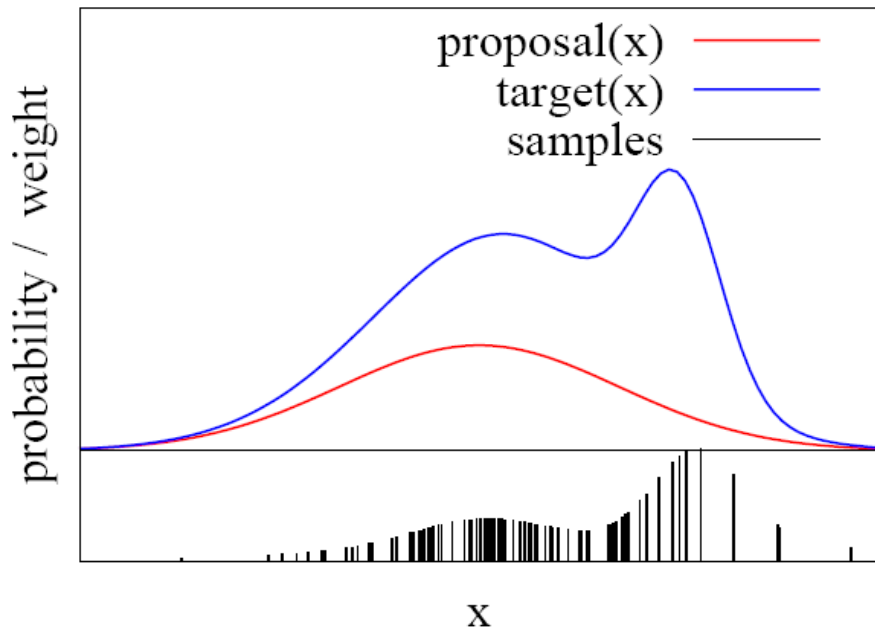
Rejection Sampling

- 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 sample



Importance Sampling Principle

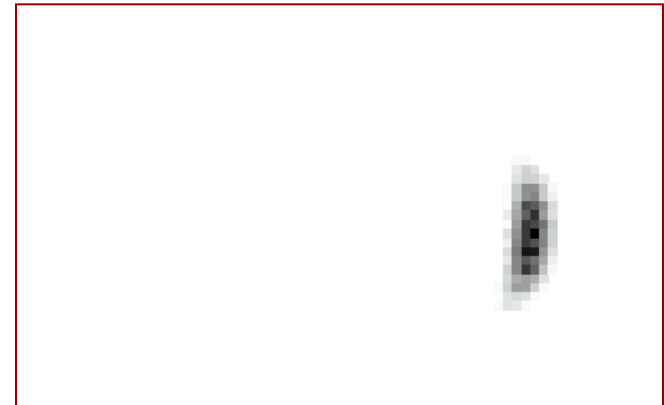
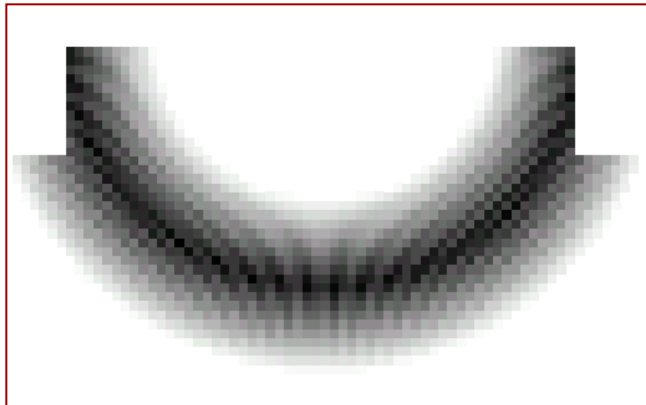
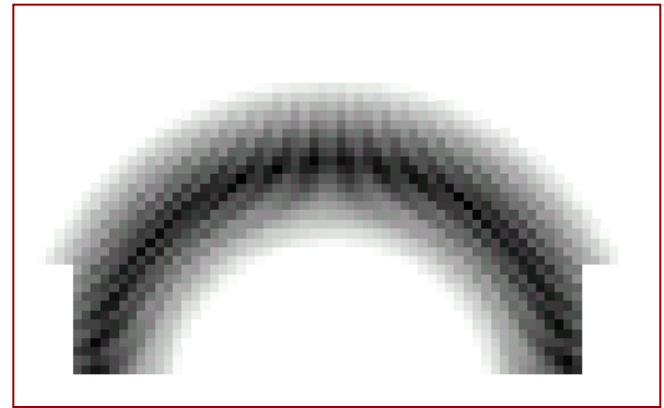
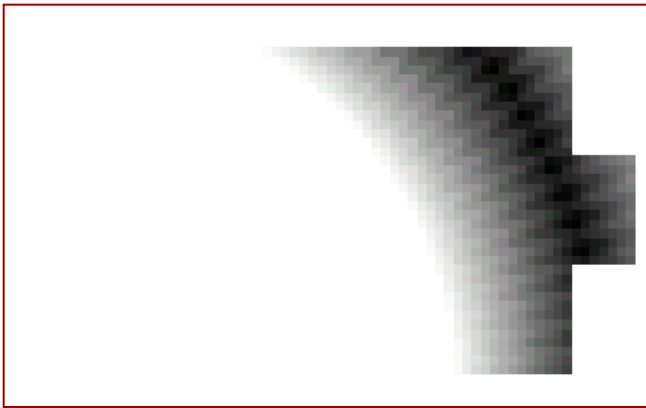
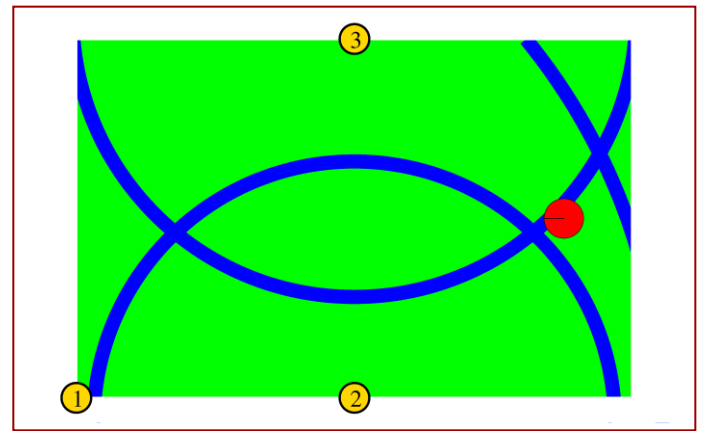
- We can even use a different distribution g to generate samples from f
- By introducing an importance weight w , we can account for the “differences between g and f ”
- $w = f / g$
- f is called target
- g is called proposal
- Pre-condition:
 $f(x) > 0 \rightarrow g(x) > 0$
- Derivation: See webpage



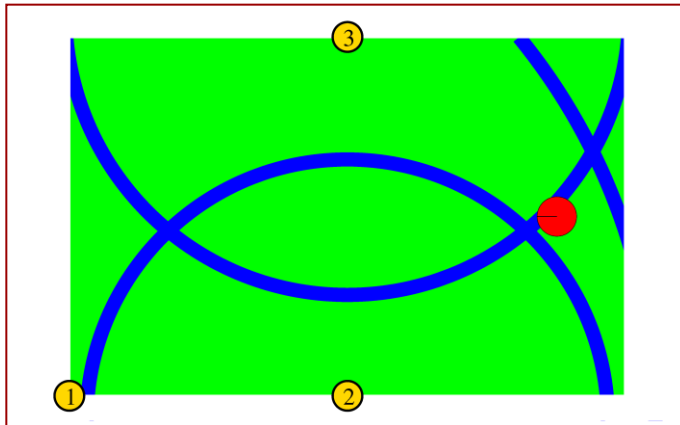
Importance Sampling with Resampling: Landmark Detection Example



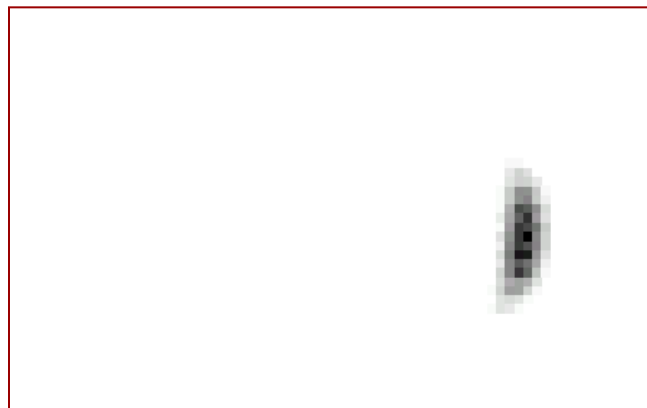
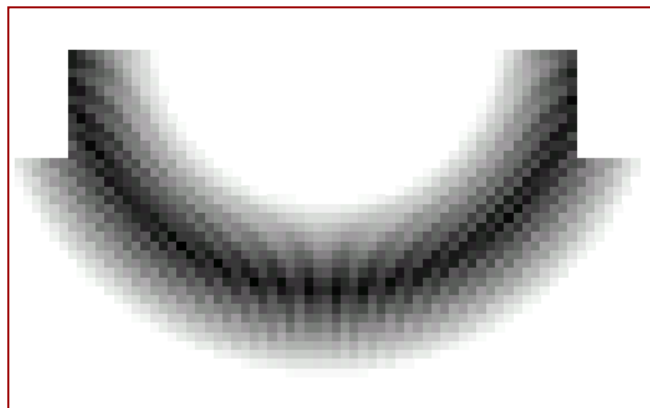
Distributions



Distributions

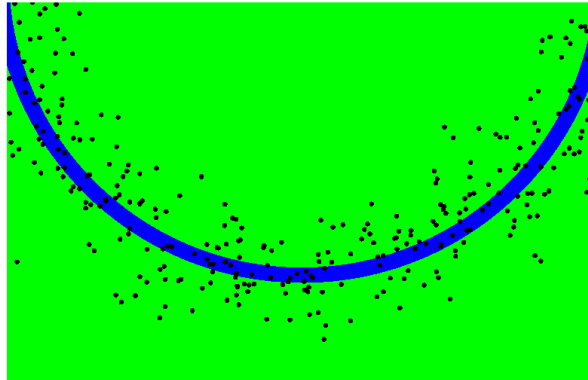
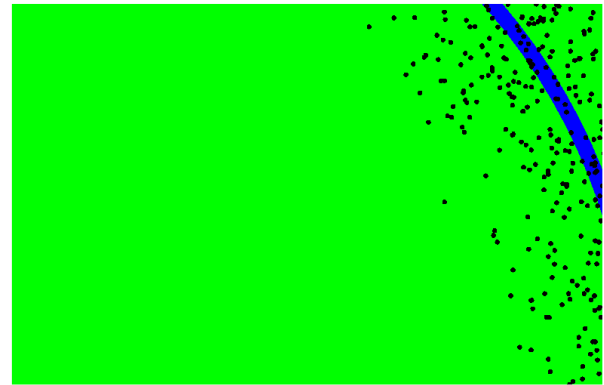
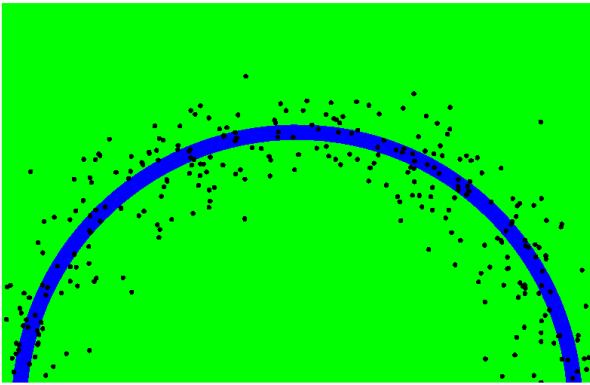


Wanted: samples distributed according to $p(x | z_1, z_2, z_3)$



This is Easy!

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



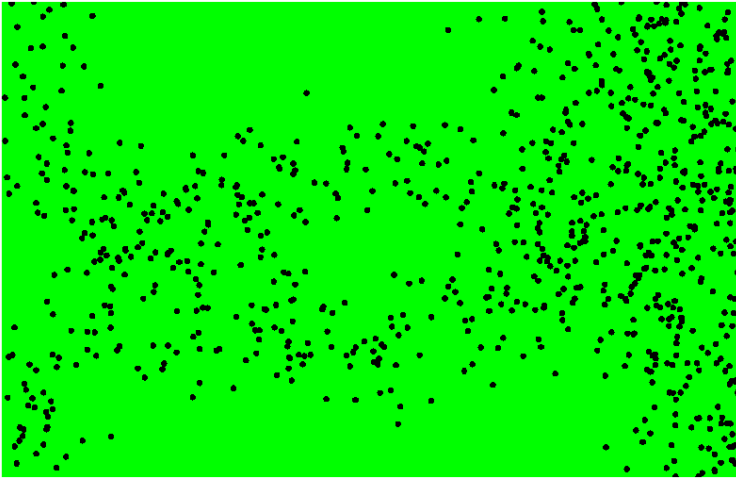
Importance Sampling

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

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

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

Importance Sampling with Resampling

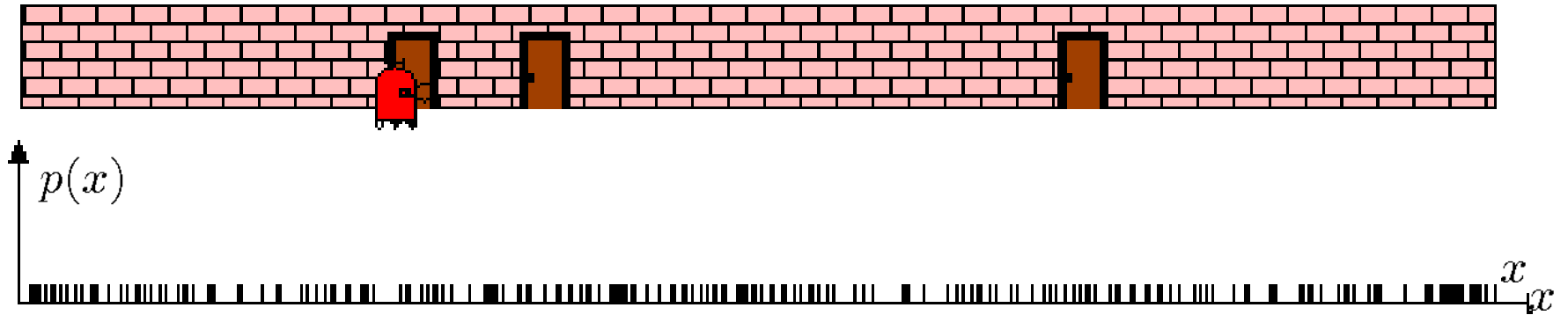


Weighted samples



After resampling

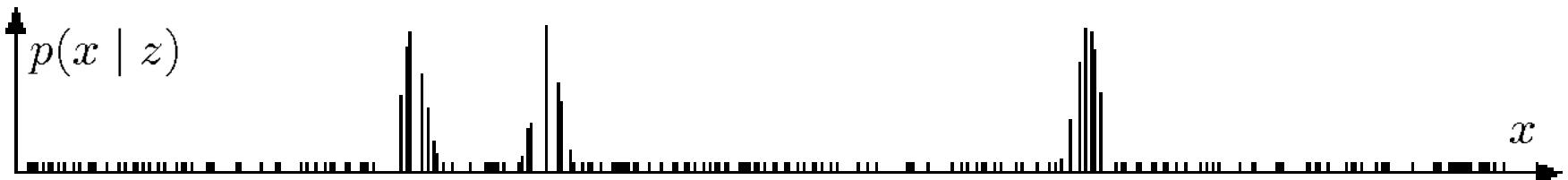
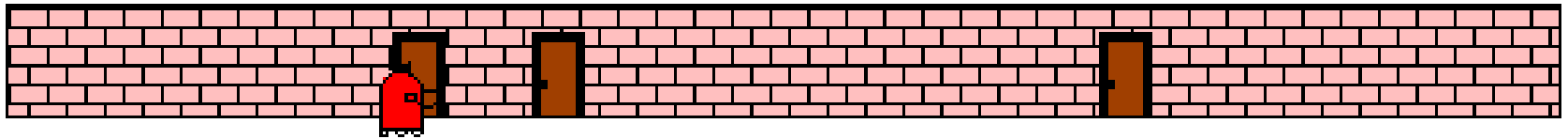
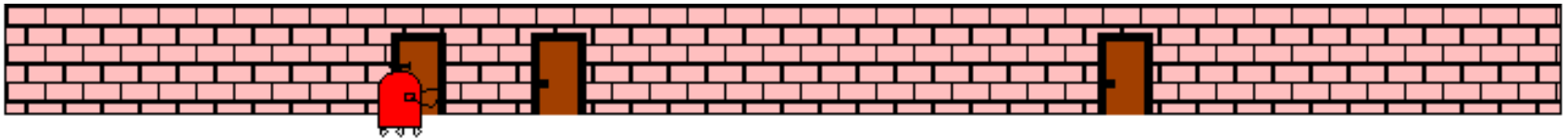
Particle Filters



Sensor Information: Importance Sampling

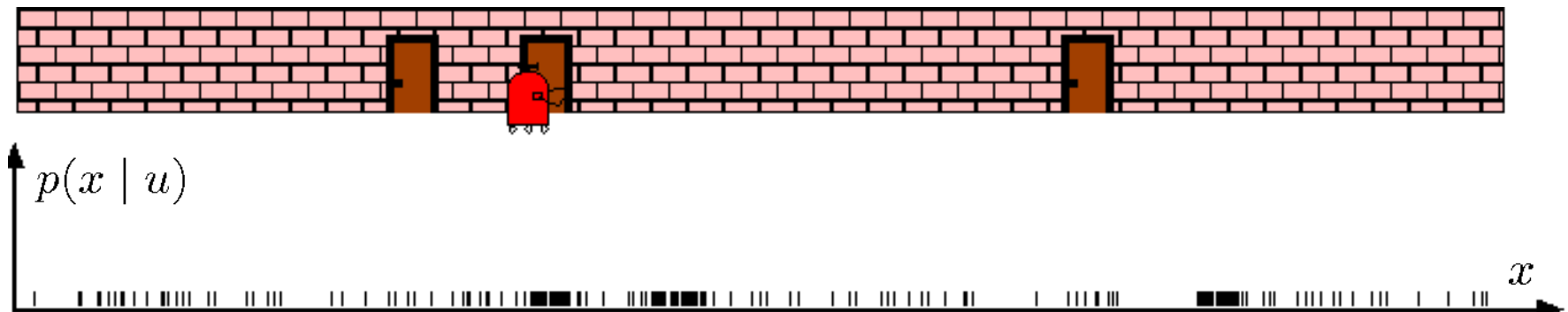
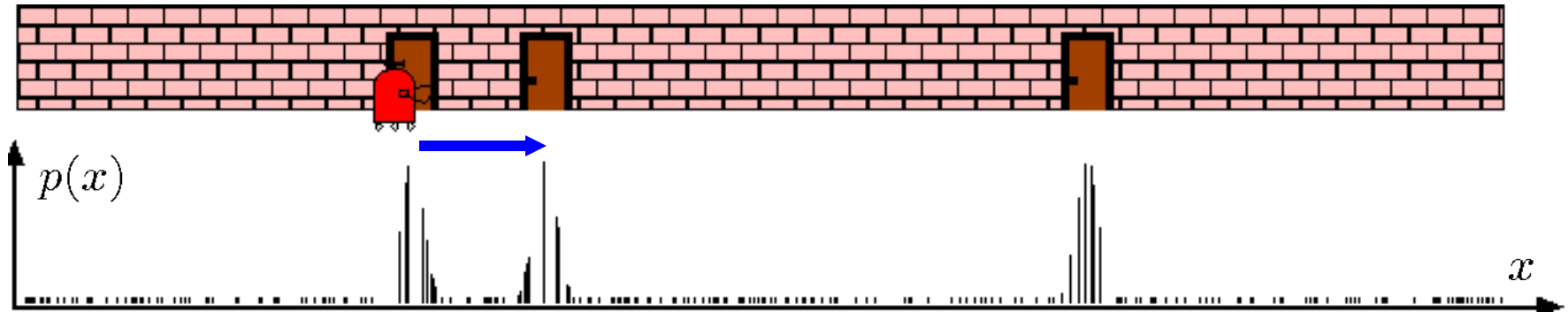
$$Bel(x) \leftarrow \alpha p(z|x) Bel^-(x)$$

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



Robot Motion

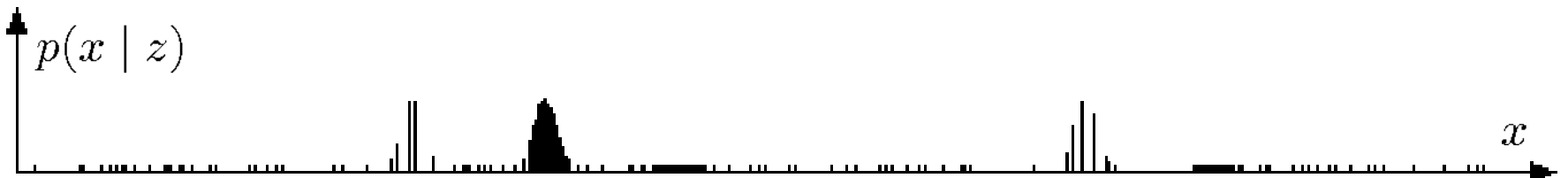
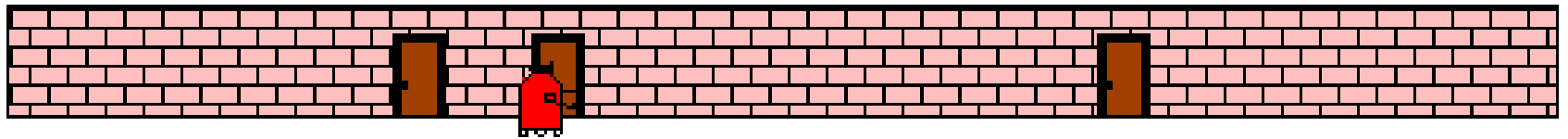
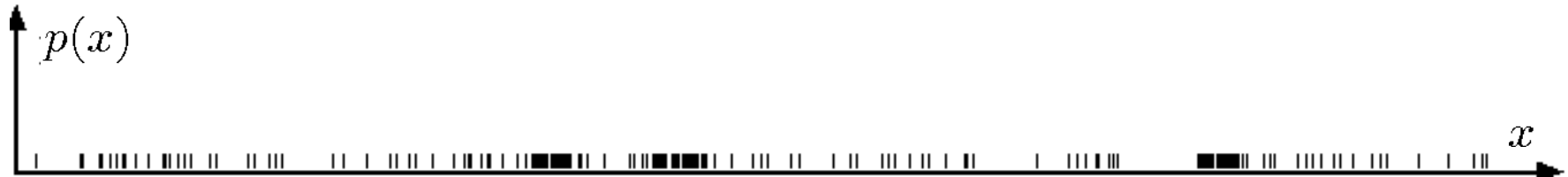
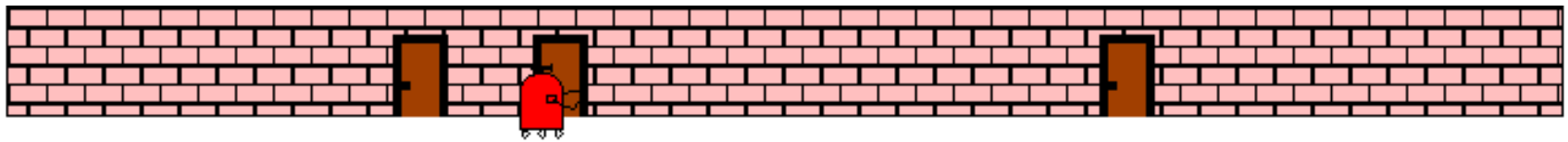
$$Bel^-(x) \leftarrow \int p(x | u, x') Bel(x') dx'$$



Sensor Information: Importance Sampling

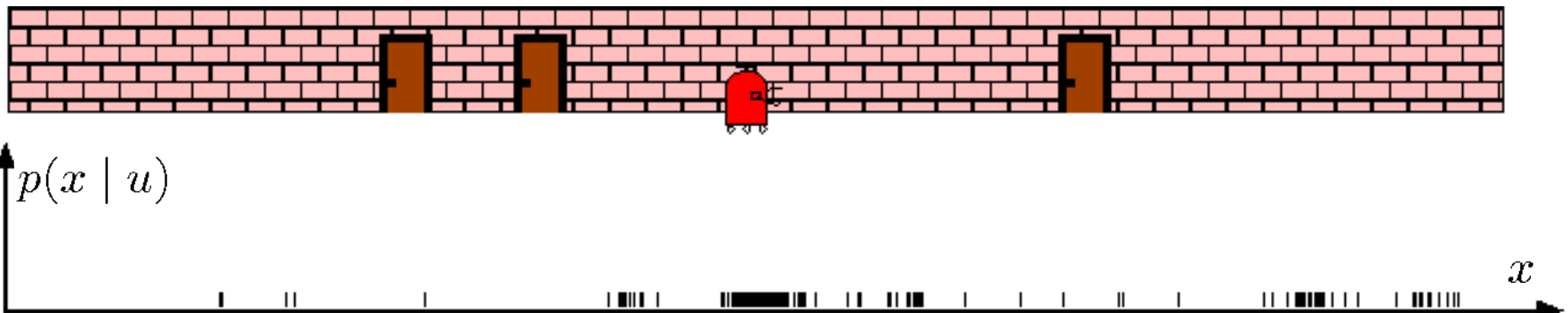
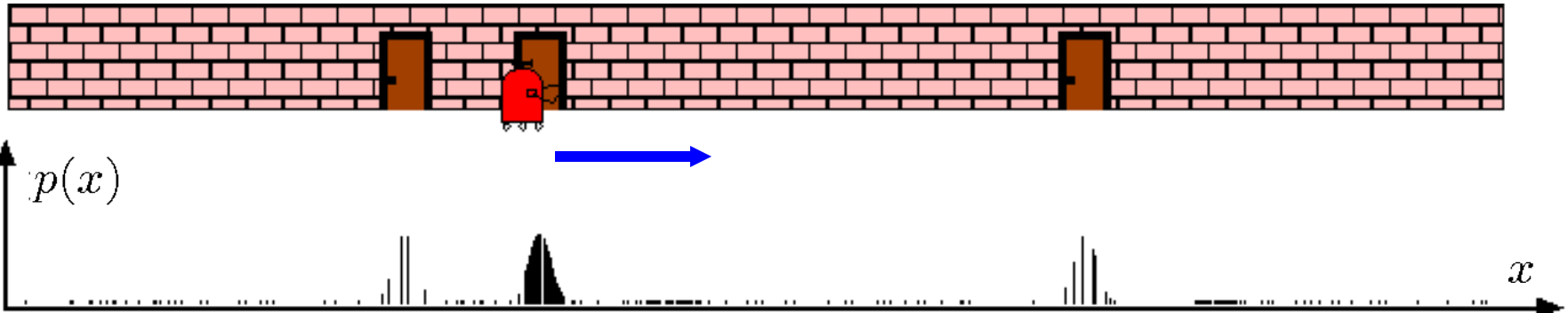
$$Bel(x) \leftarrow \alpha p(z|x) Bel^-(x)$$

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



Robot Motion

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



Particle Filter Algorithm

- Sample the next generation for particles using the proposal distribution
- Compute the importance weights :
$$\textit{weight} = \textit{target distribution} / \textit{proposal distribution}$$
- Resampling: “Replace unlikely samples by more likely ones”

Particle Filter Algorithm

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

Particle Filter Algorithm

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

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

draw x_t^i from $p(x_t | x_{t-1}^i, u_t)$

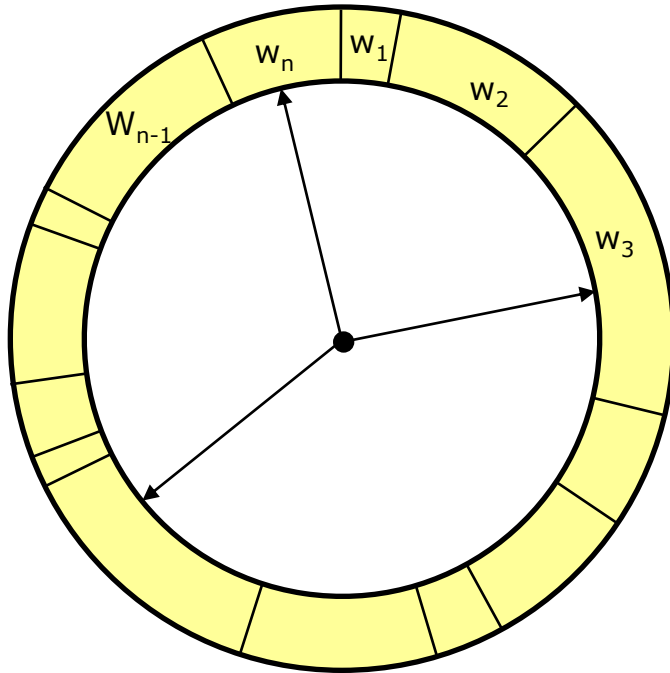
Importance factor for x_t^i :

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

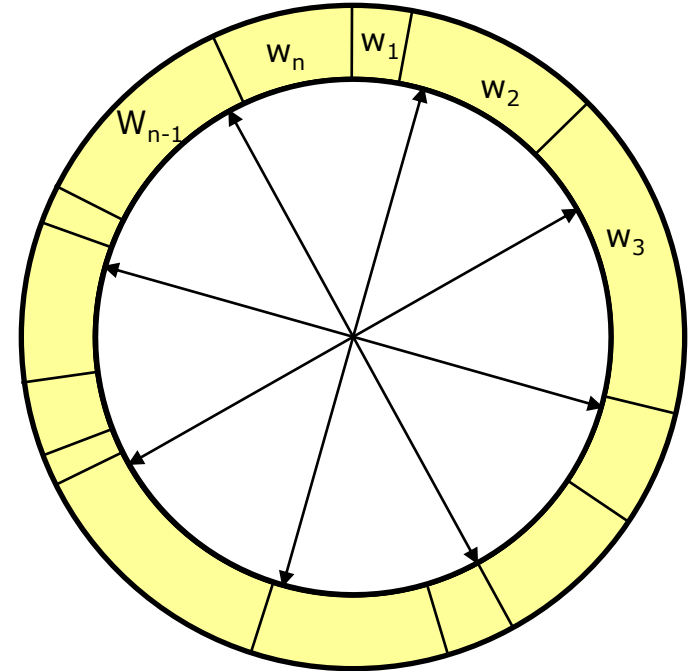
Resampling

- **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



- 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** $i = 2 \dots n$ *Generate cdf*

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

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

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

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

8. $i = i + 1$

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

10. $u_{j+1} = u_j + n^{-1}$ *Increment threshold*

11. **Return** S'

Also called **stochastic universal sampling**

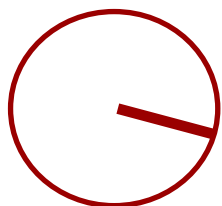
Mobile Robot Localization

- 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)

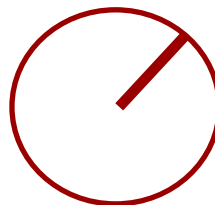
[For details, see PDF file on the lecture web page]

Motion Model Reminder

start pose

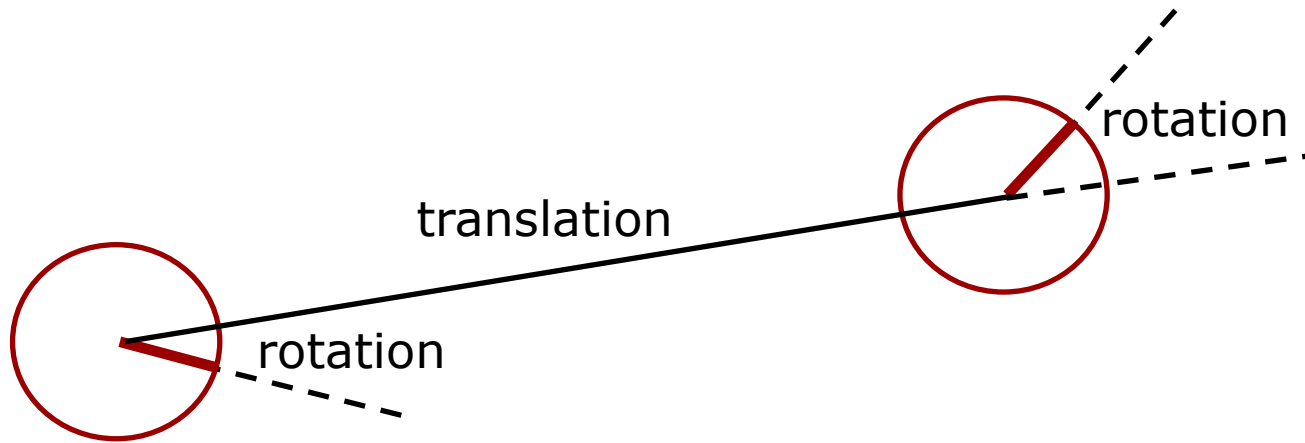


end pose



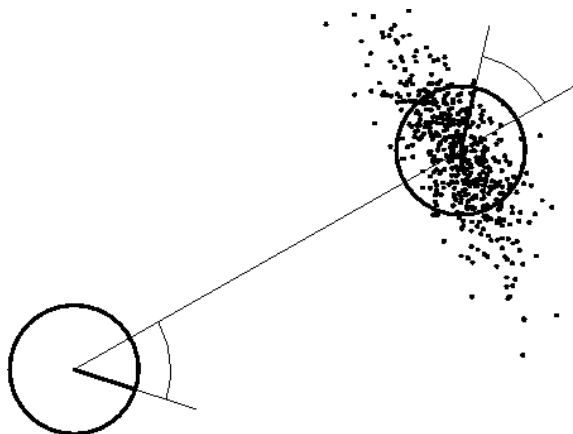
According to the estimated motion

Motion Model Reminder



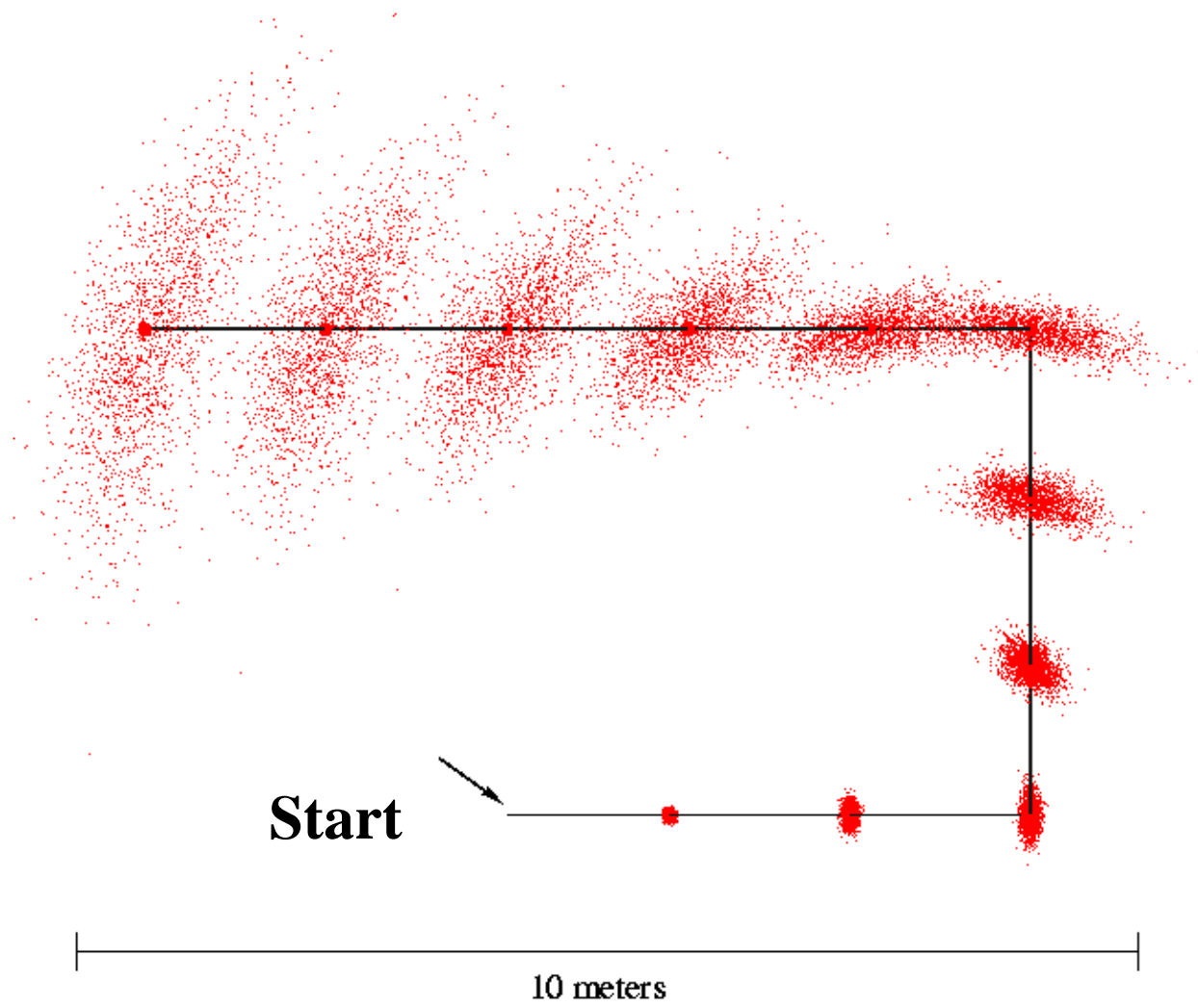
- Decompose the motion into
 - Traveled distance
 - Start rotation
 - End rotation

Motion Model Reminder

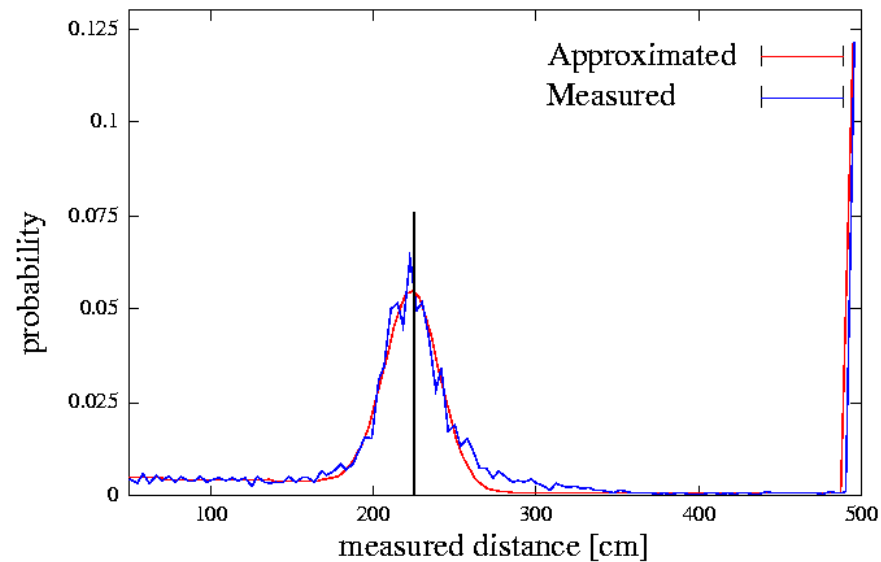


- Uncertainty in the translation of the robot:
Gaussian over the traveled distance
- Uncertainty in the rotation of the robot:
Gaussians over start and end rotation
- For each particle, draw a new pose by sampling from these three individual normal distributions

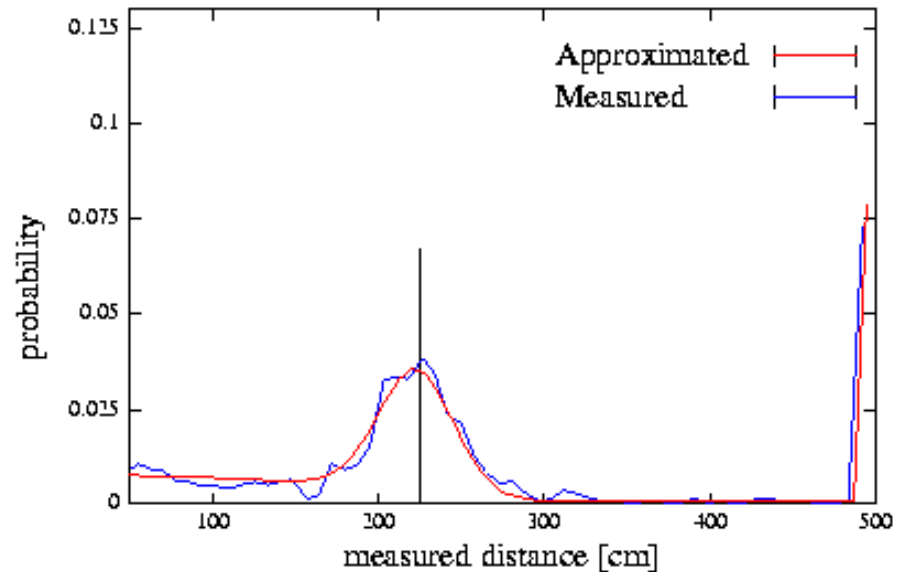
Motion Model Reminder



Proximity Sensor Model Reminder



Laser sensor



Sonar sensor

Mobile Robot Localization Using Particle Filters (1)

- Each particle is a potential pose of the robot
- The set of weighted particles approximates the posterior belief about the robot's pose (target distribution)

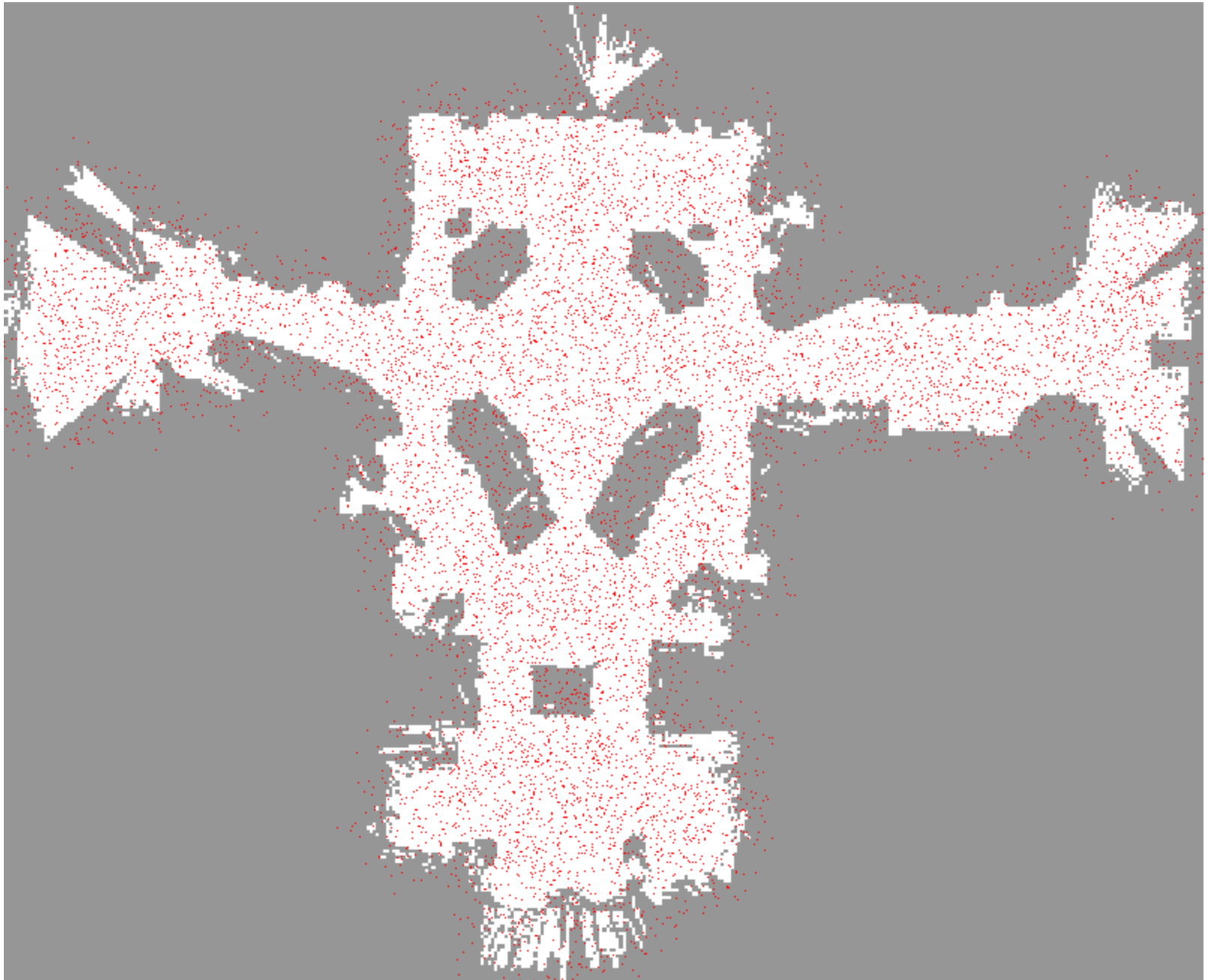
Mobile Robot Localization Using Particle Filters (2)

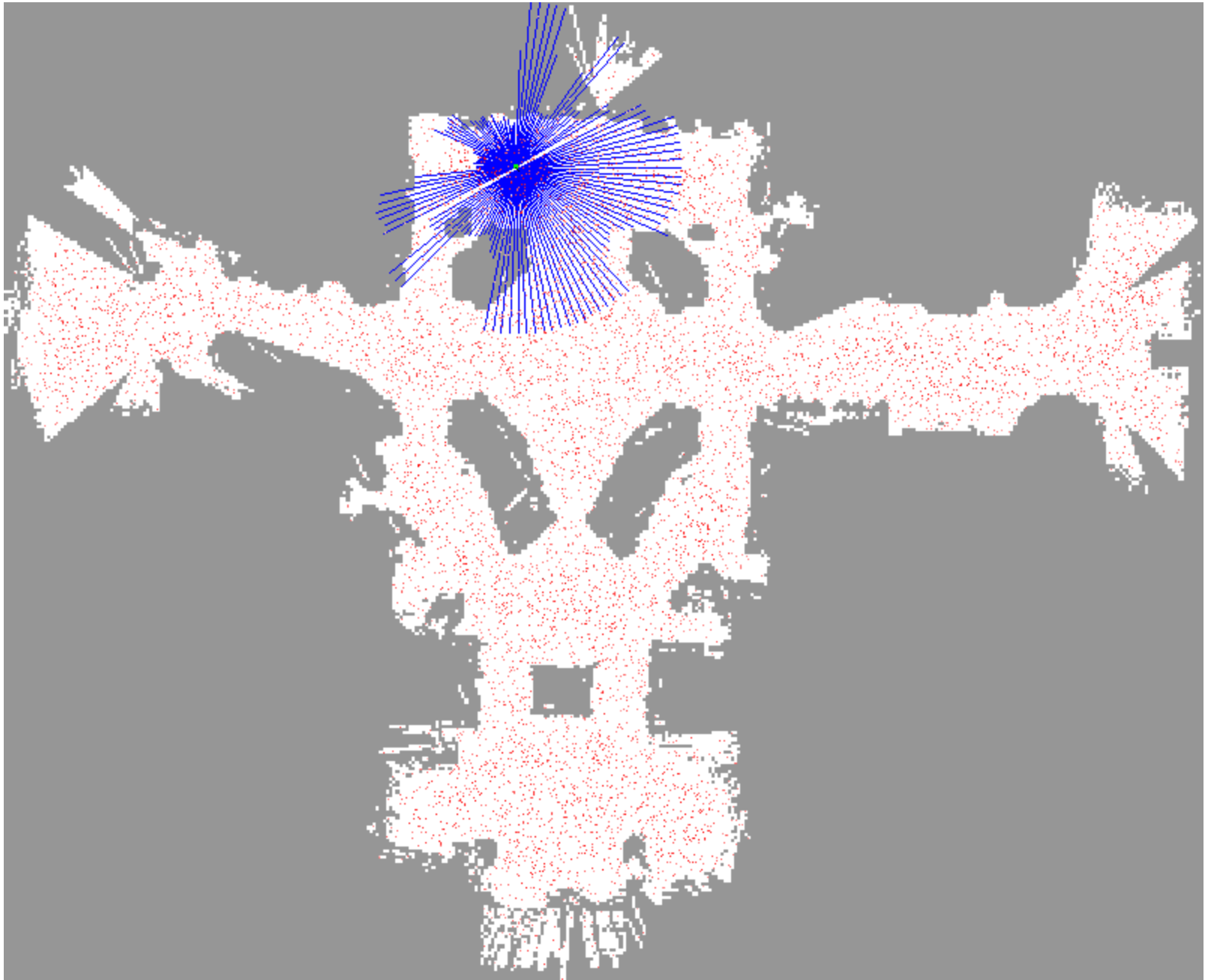
- Particles are drawn from the motion model (proposal distribution)
- Particles are weighted according to the observation model (sensor model)
- Particles are resampled according to the particle weights

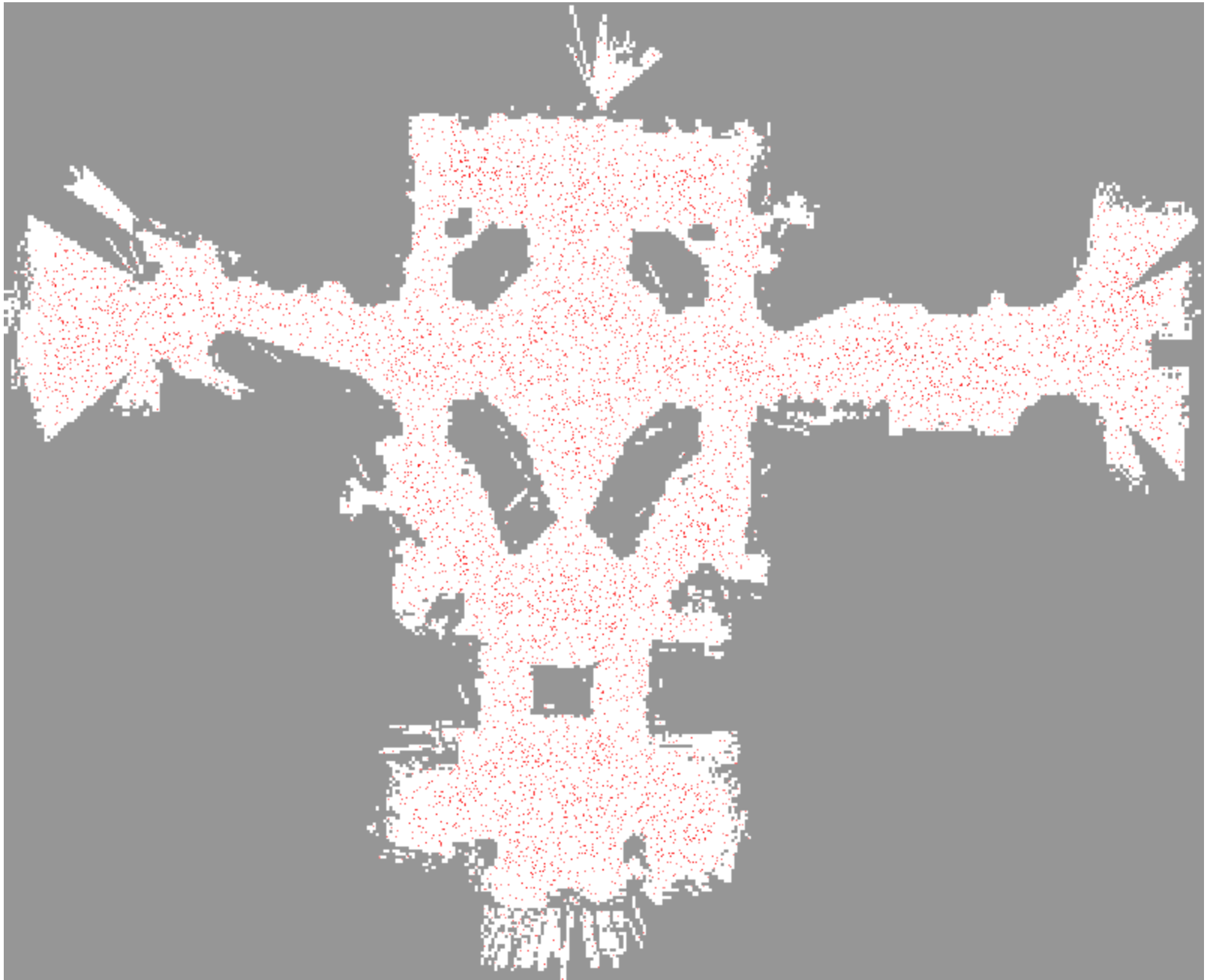
Mobile Robot Localization Using Particle Filters (3)

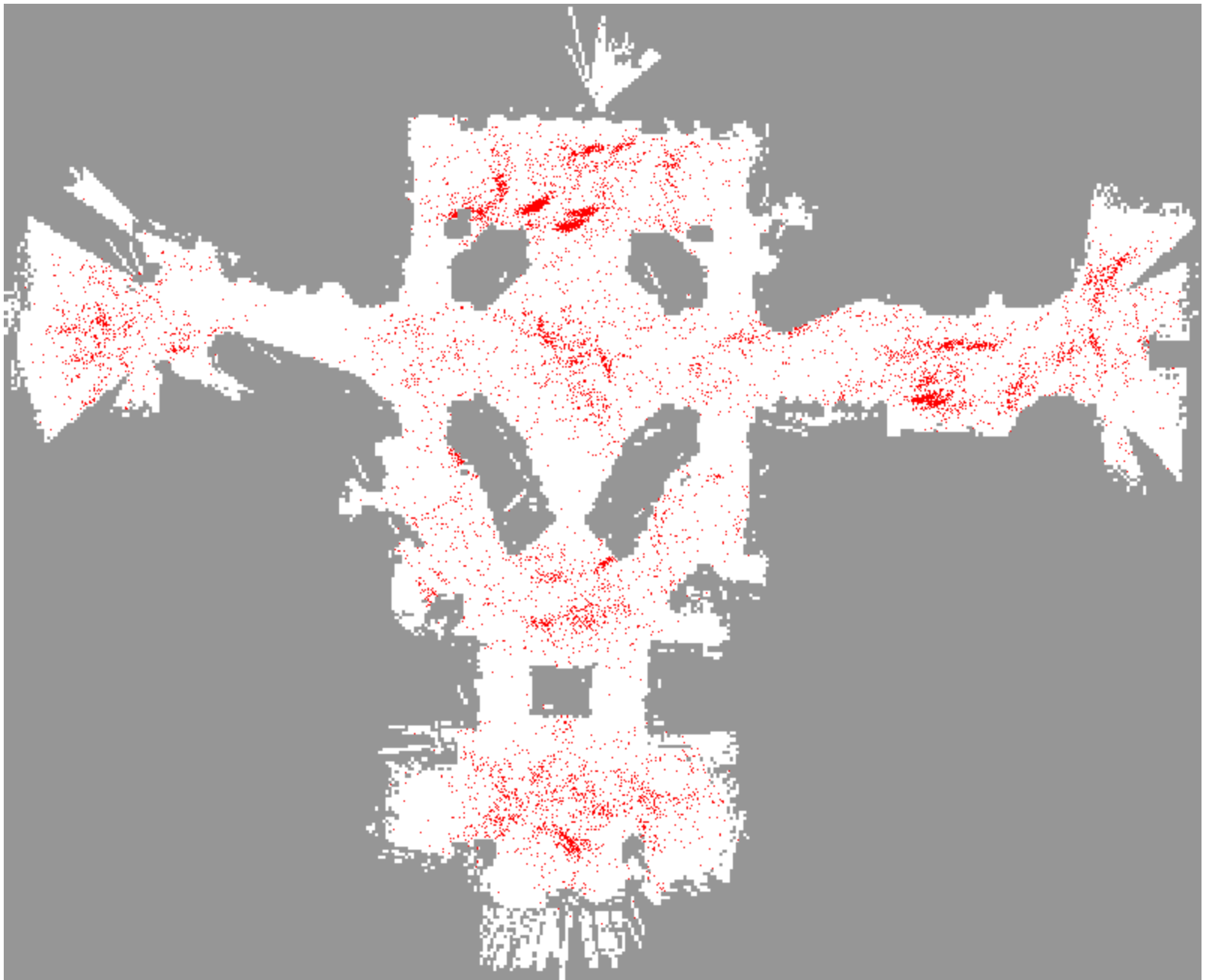
Why is resampling needed?

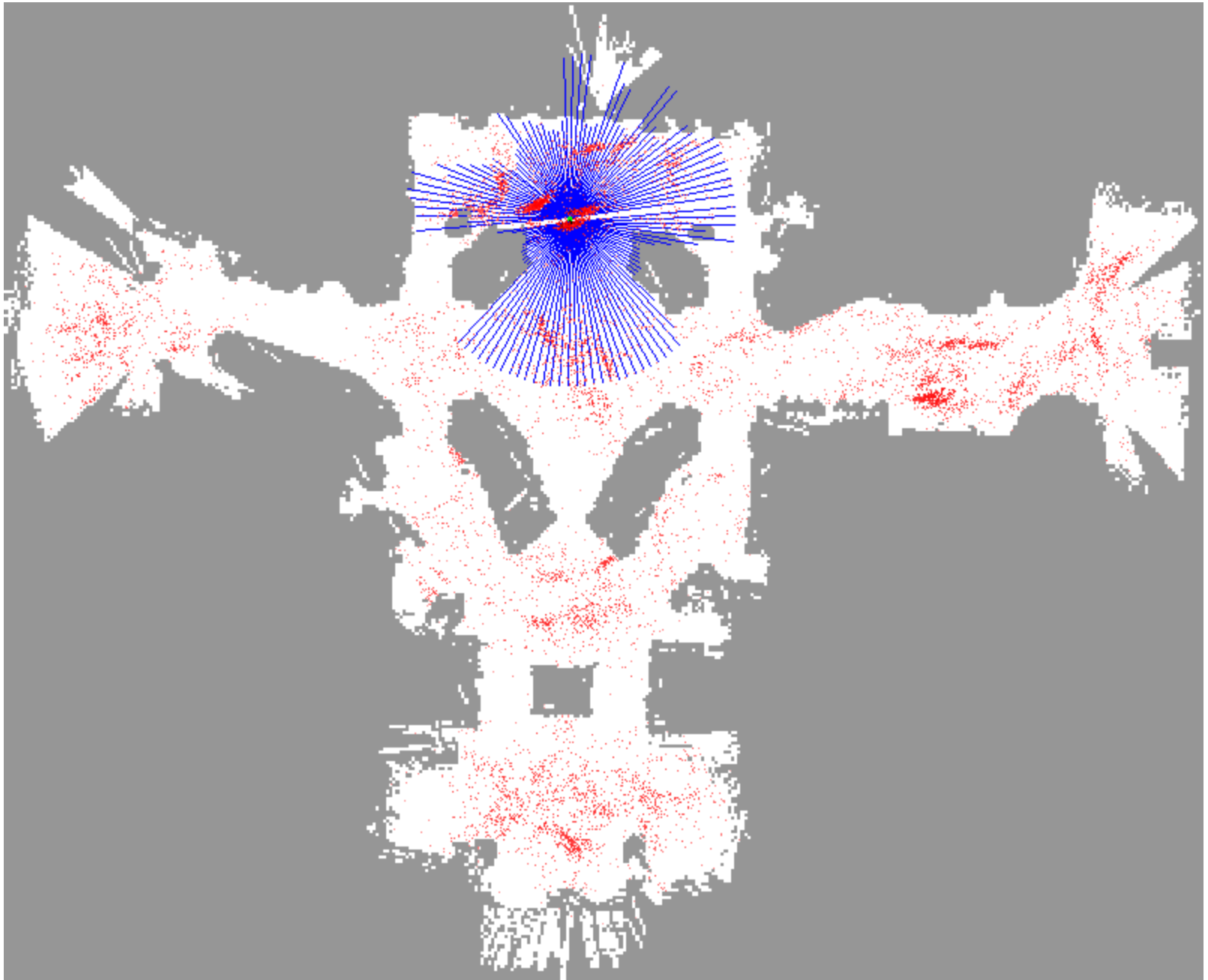
- We only have a finite number of particles
- Without resampling: The filter is likely to lose track of the “good” hypotheses
- Resampling ensures that particles stay in the meaningful area of the state space

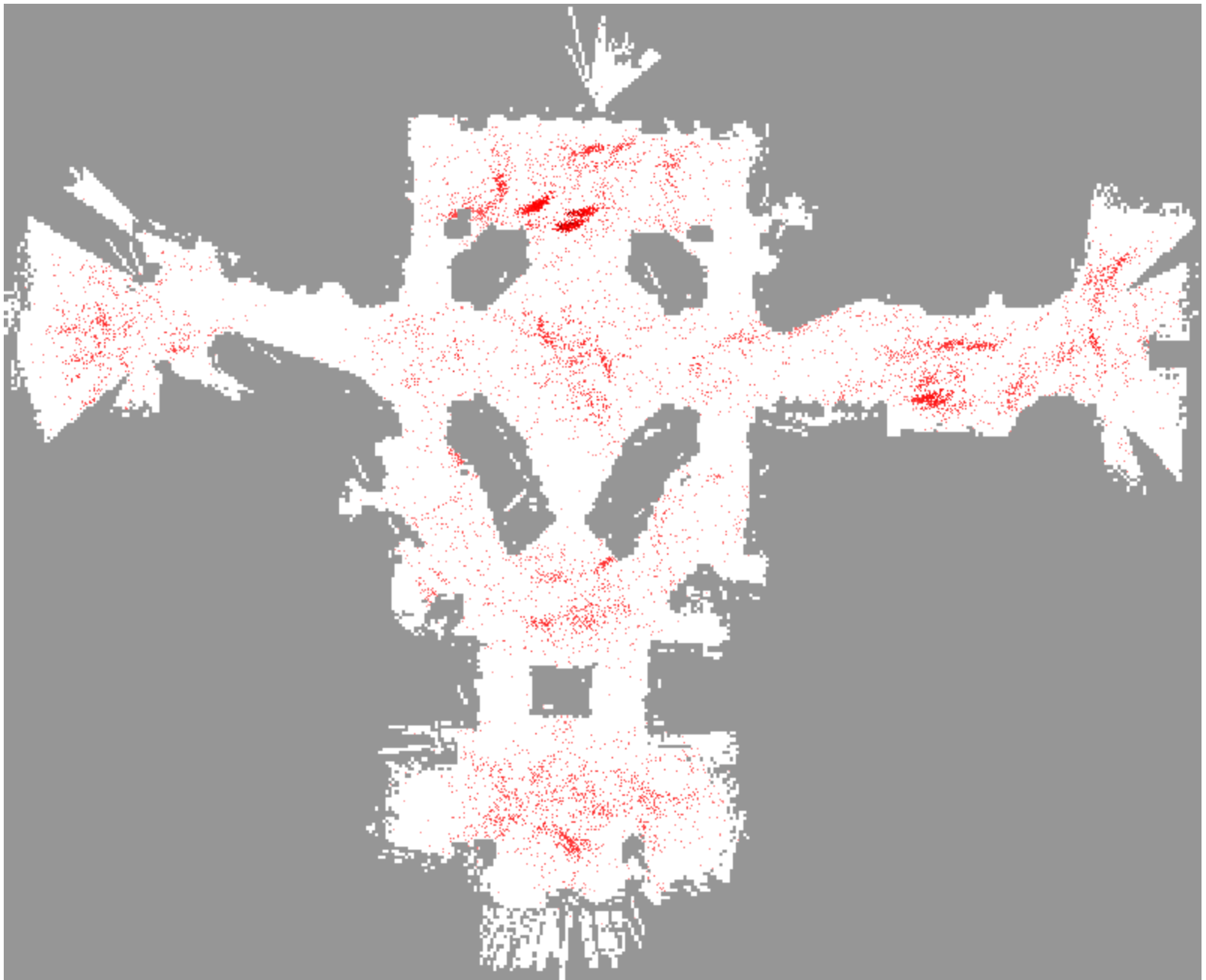


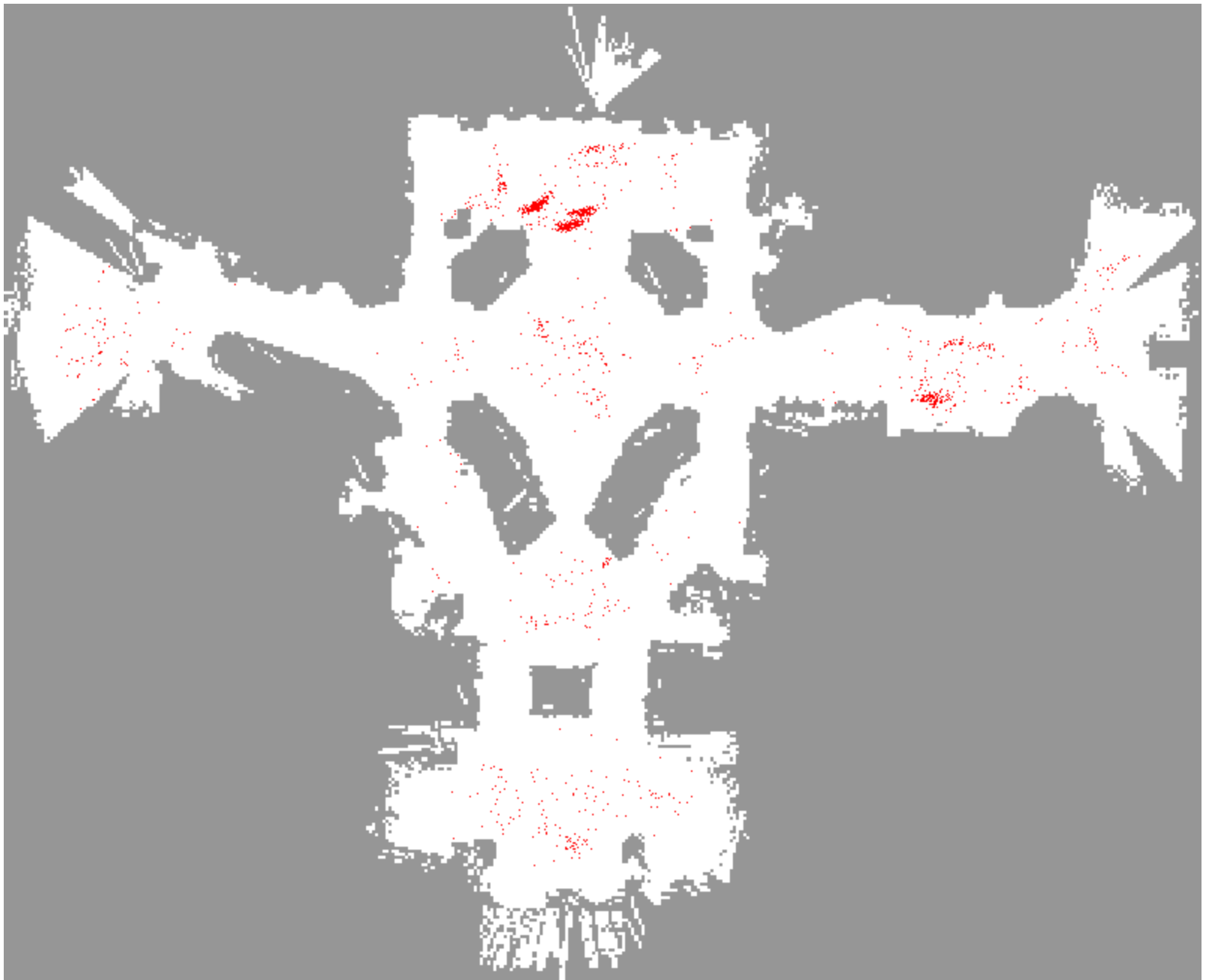




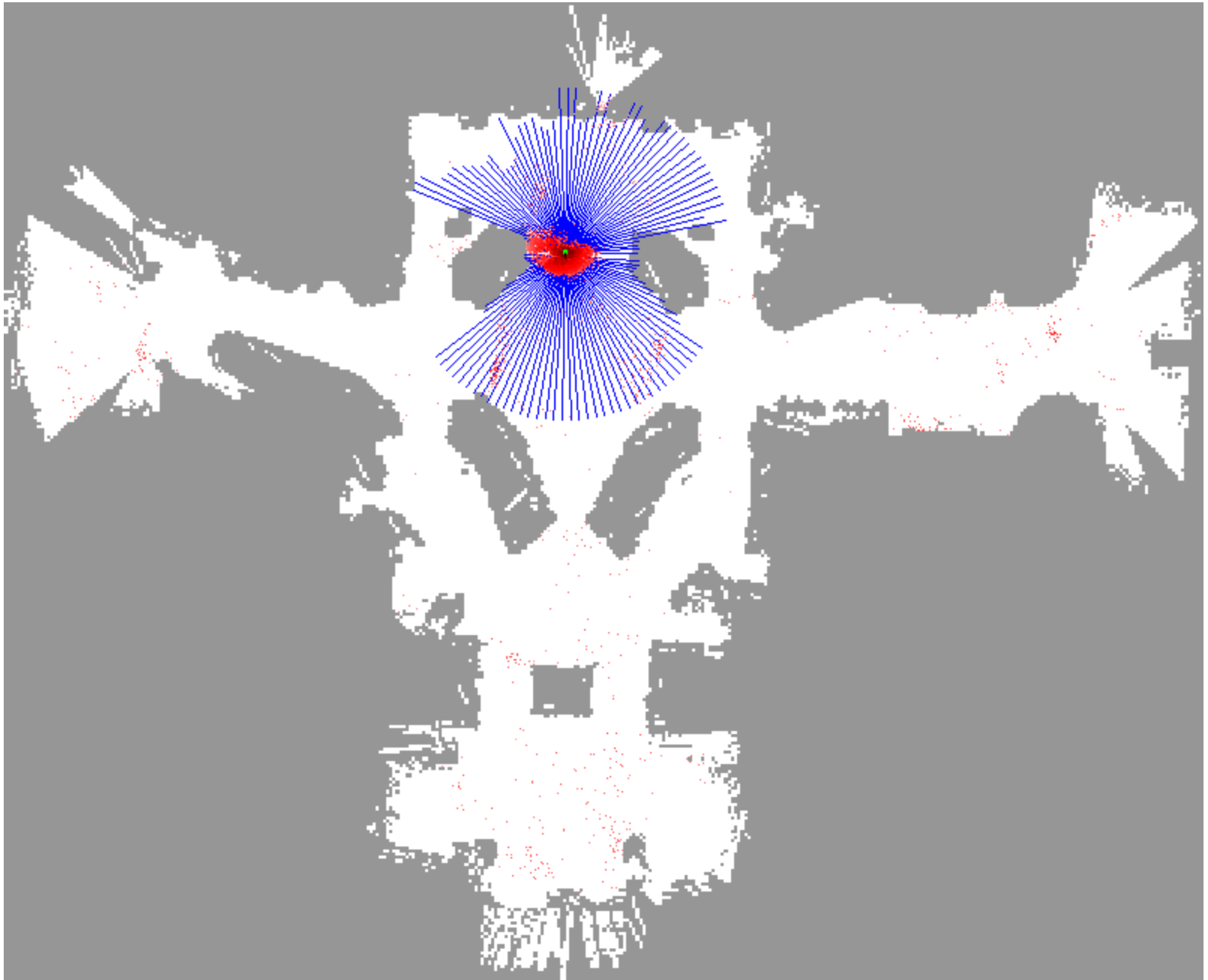




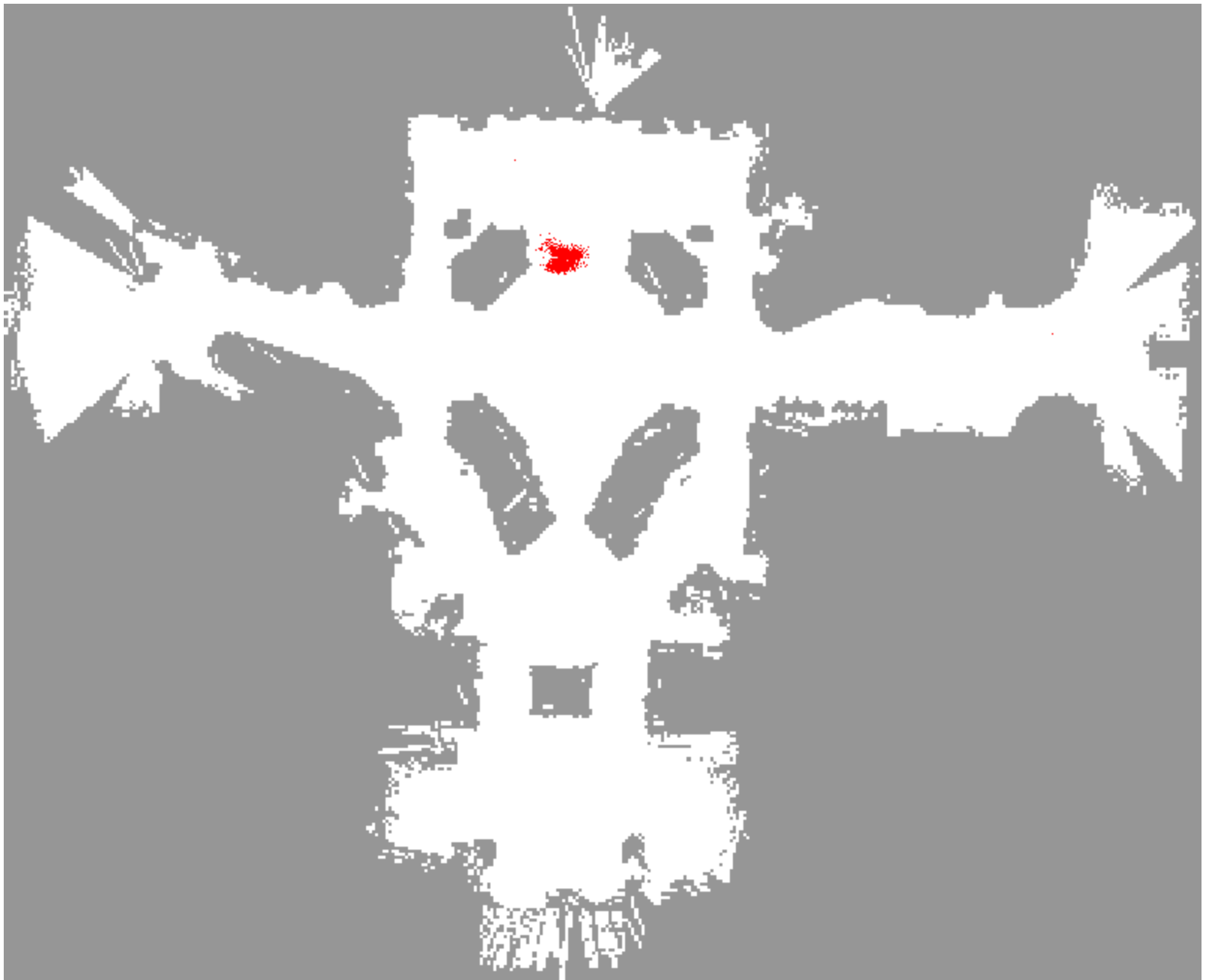


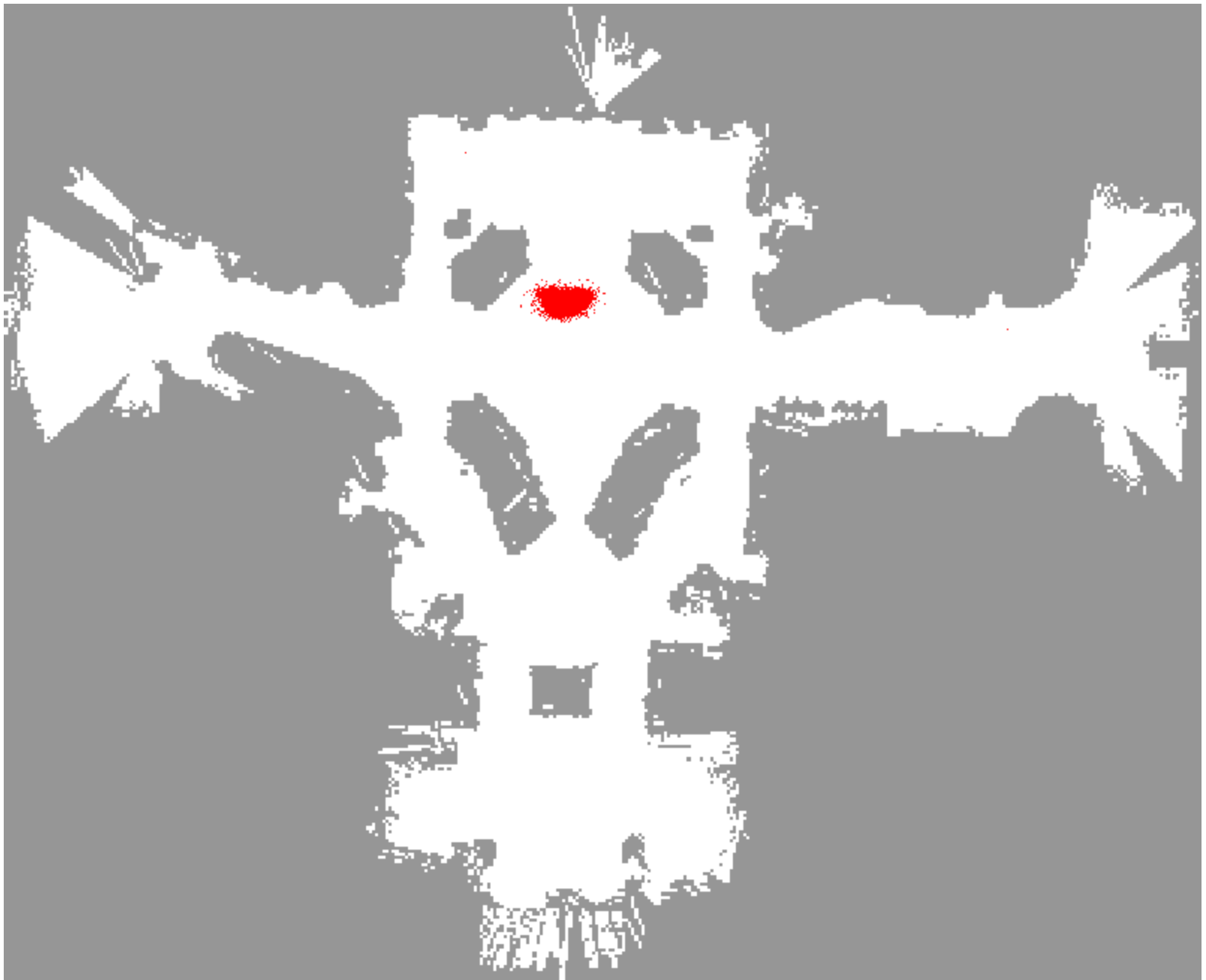


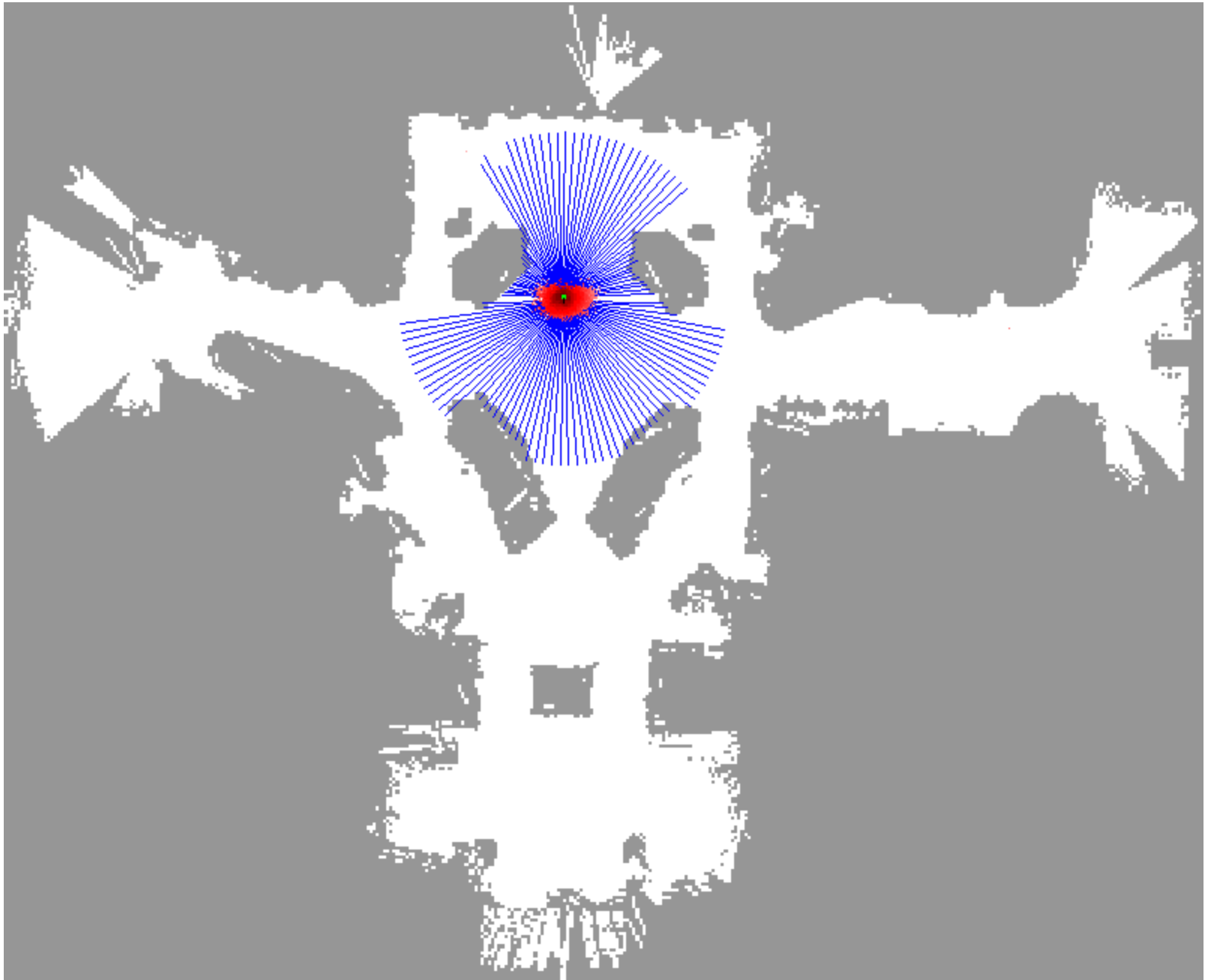


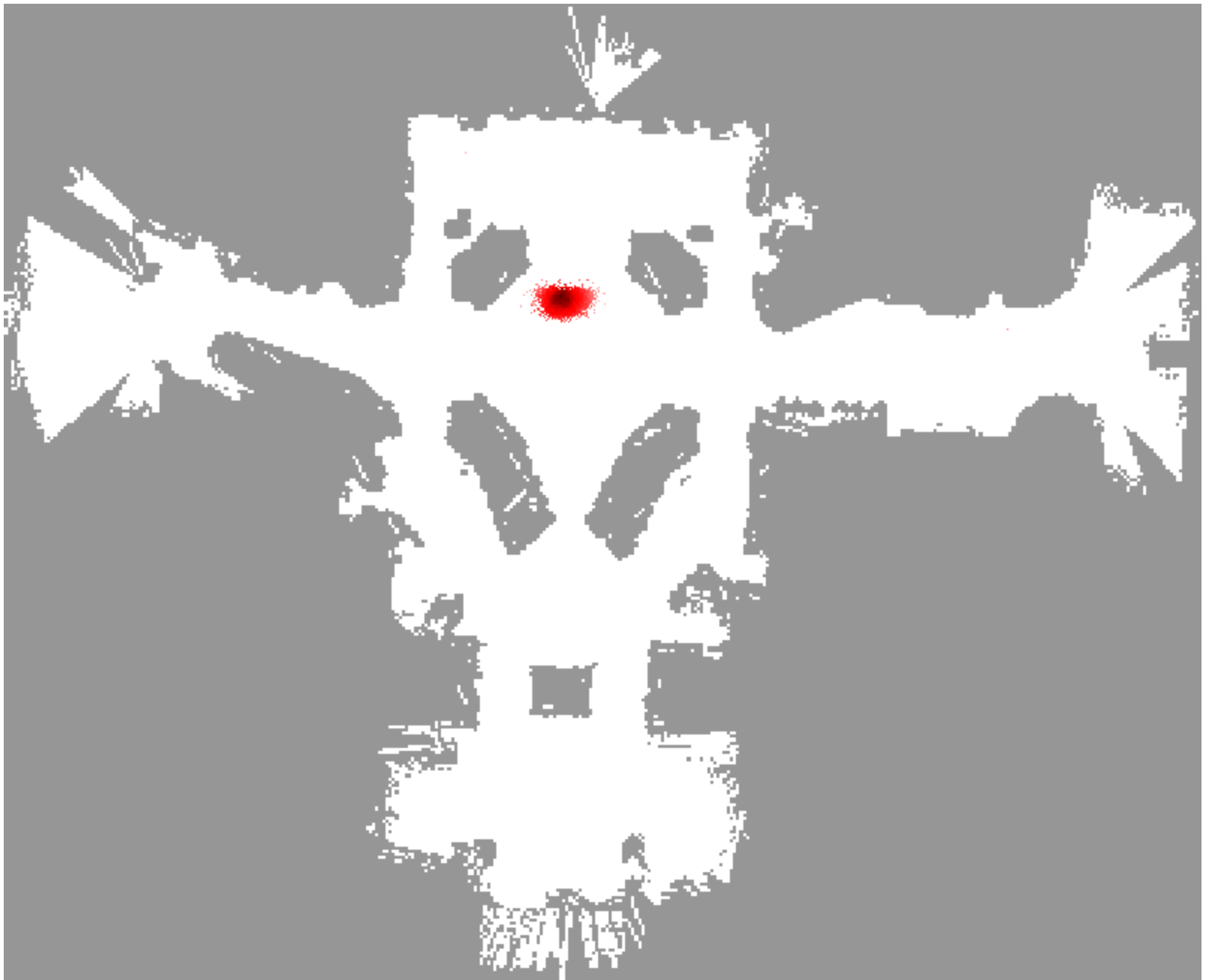


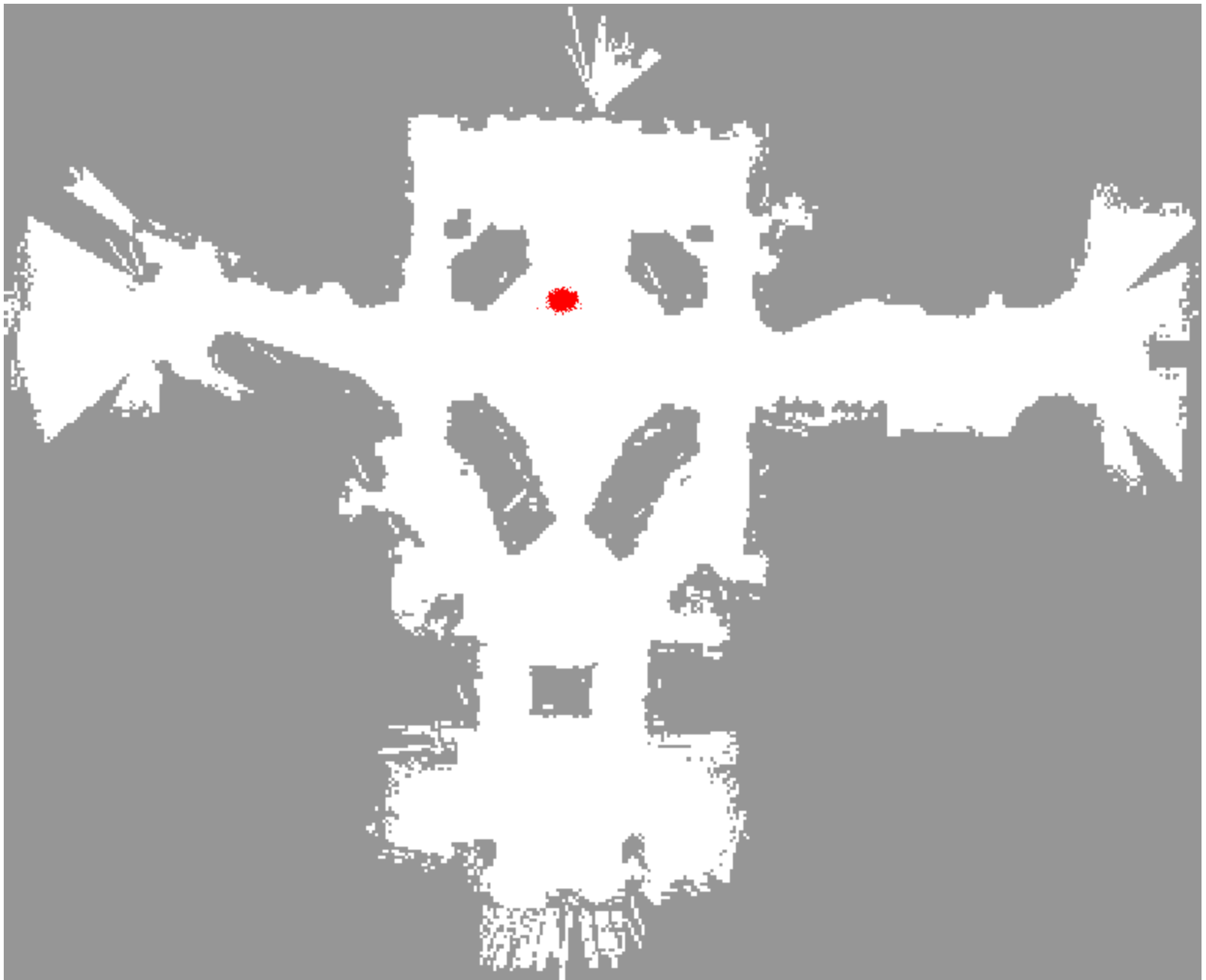


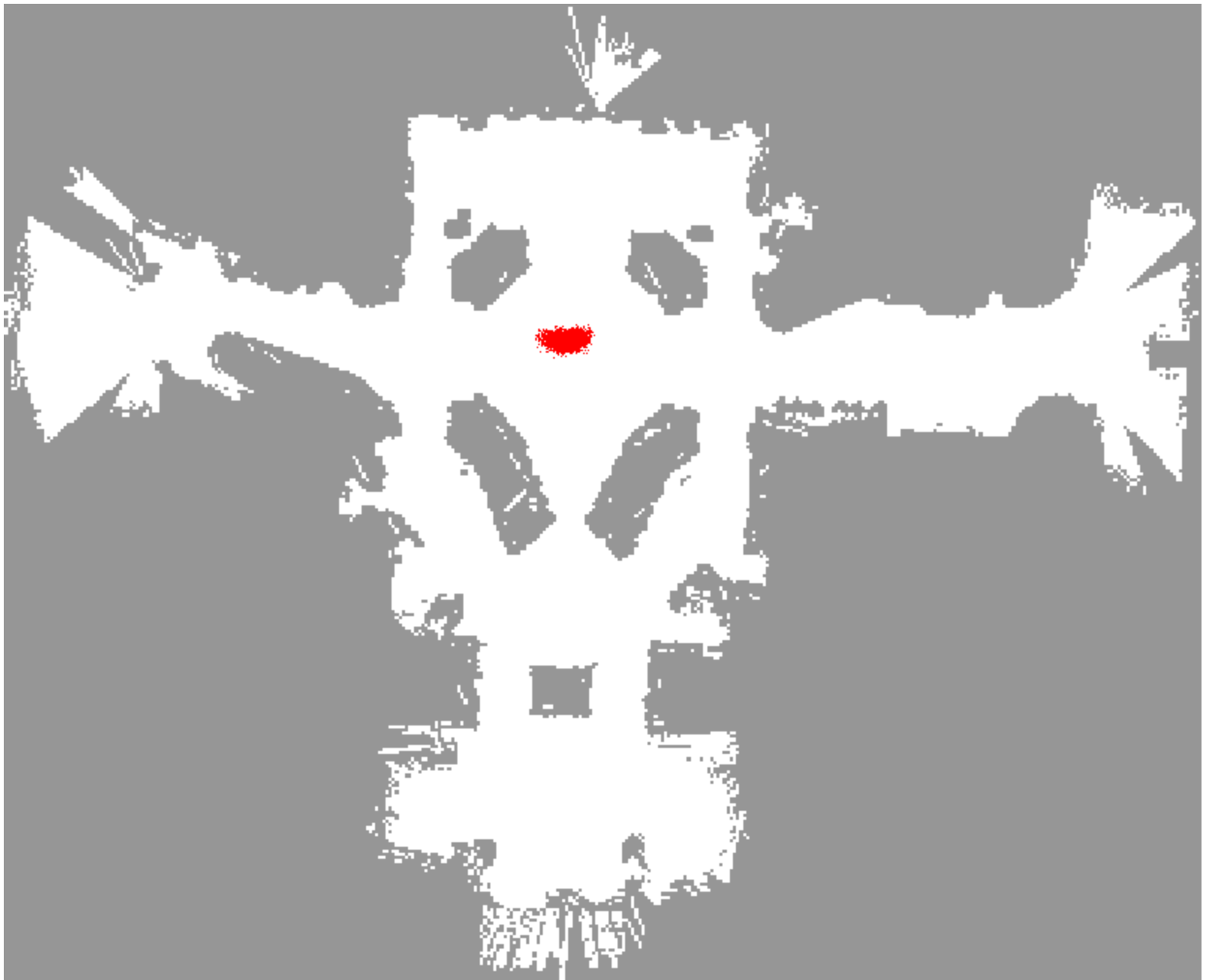


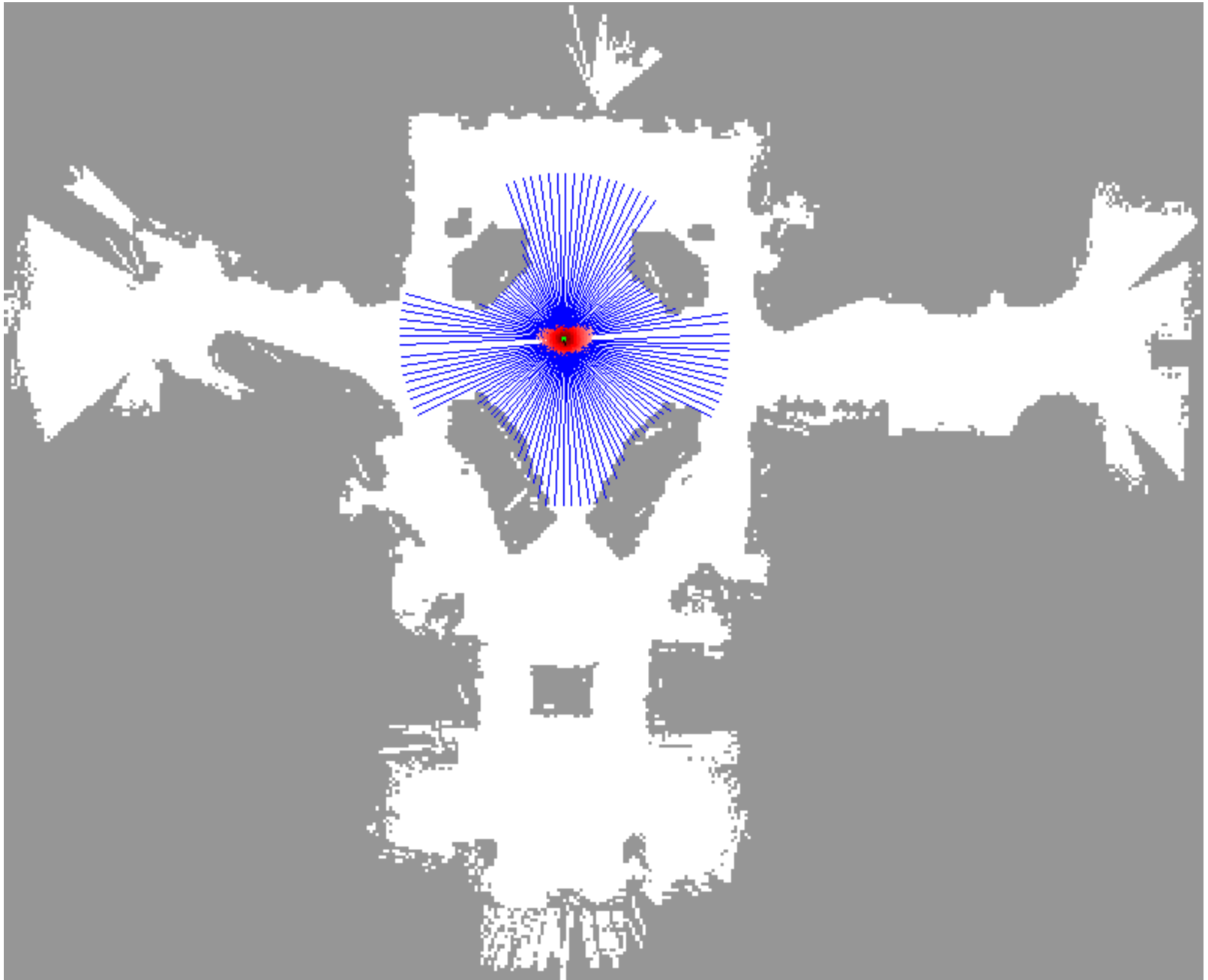




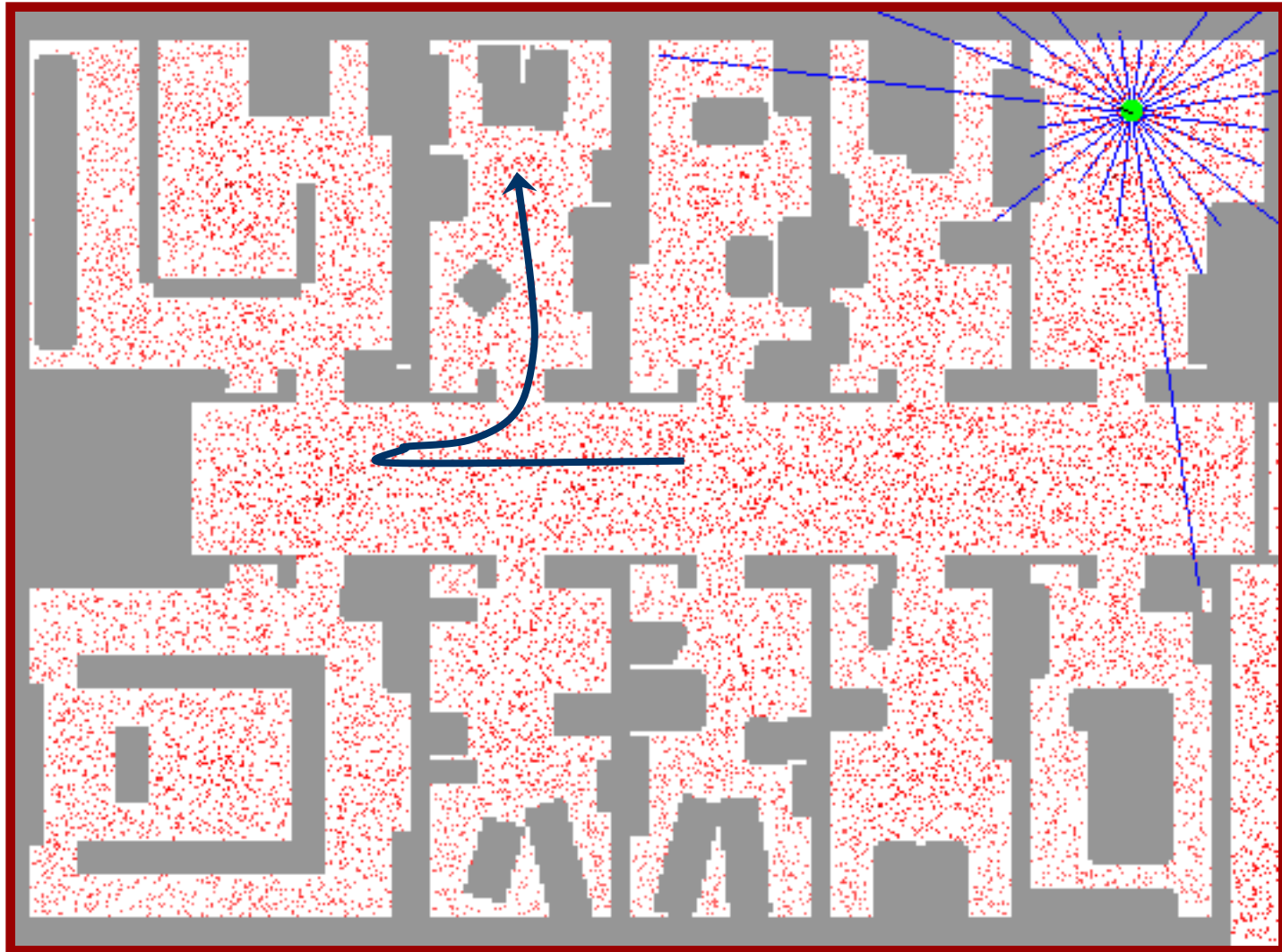




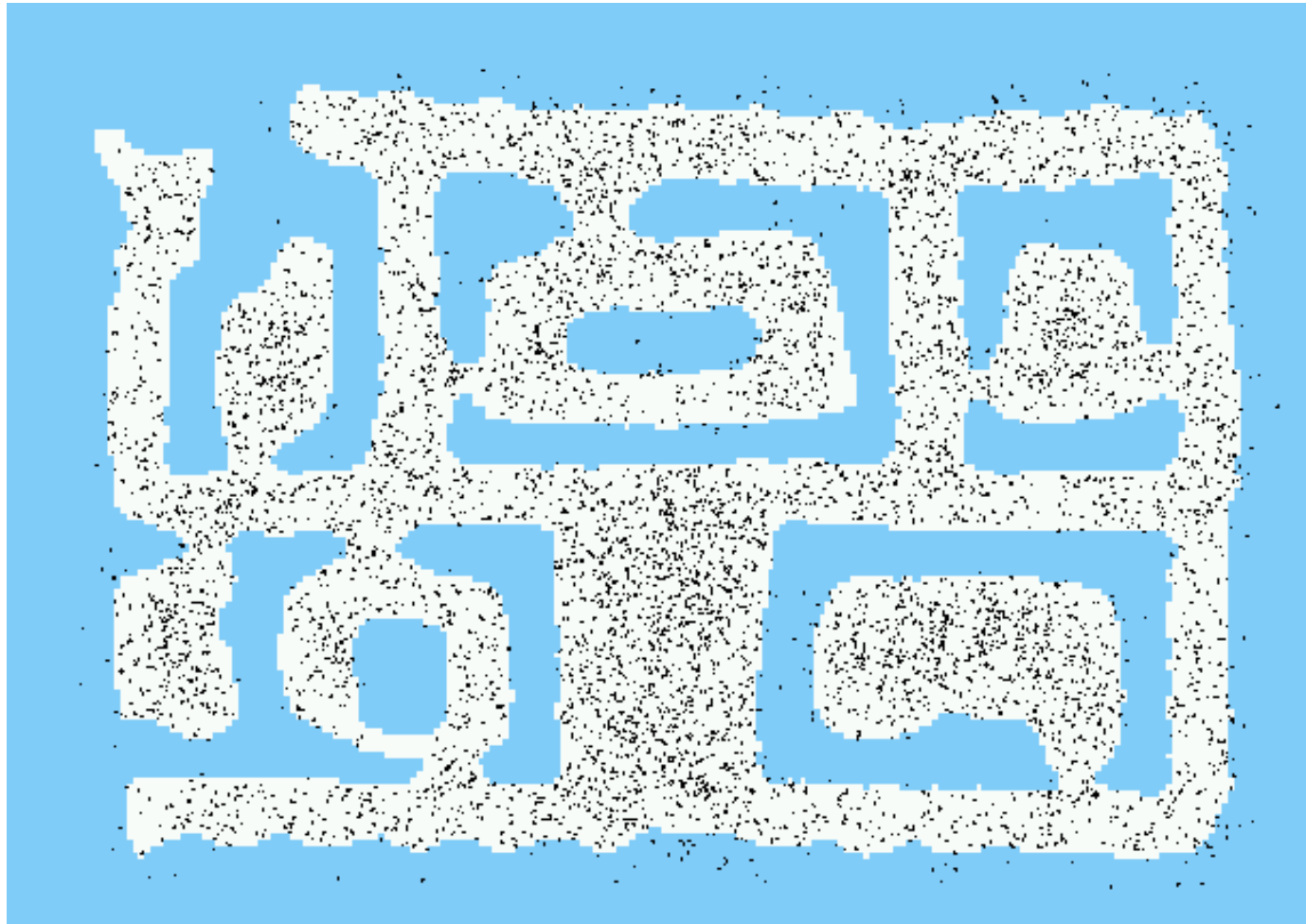




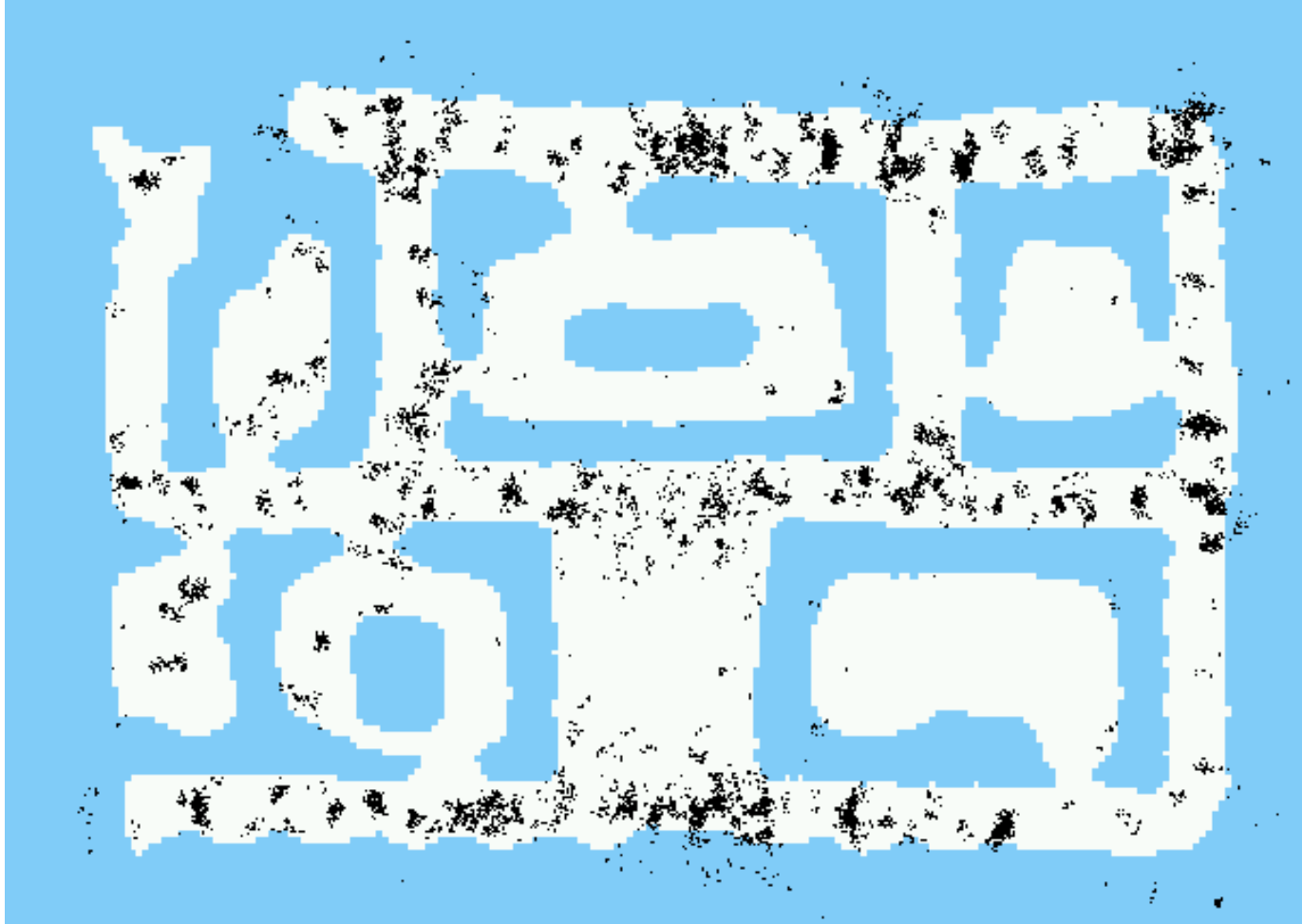
Sample-based Localization (sonar)



Initial Distribution



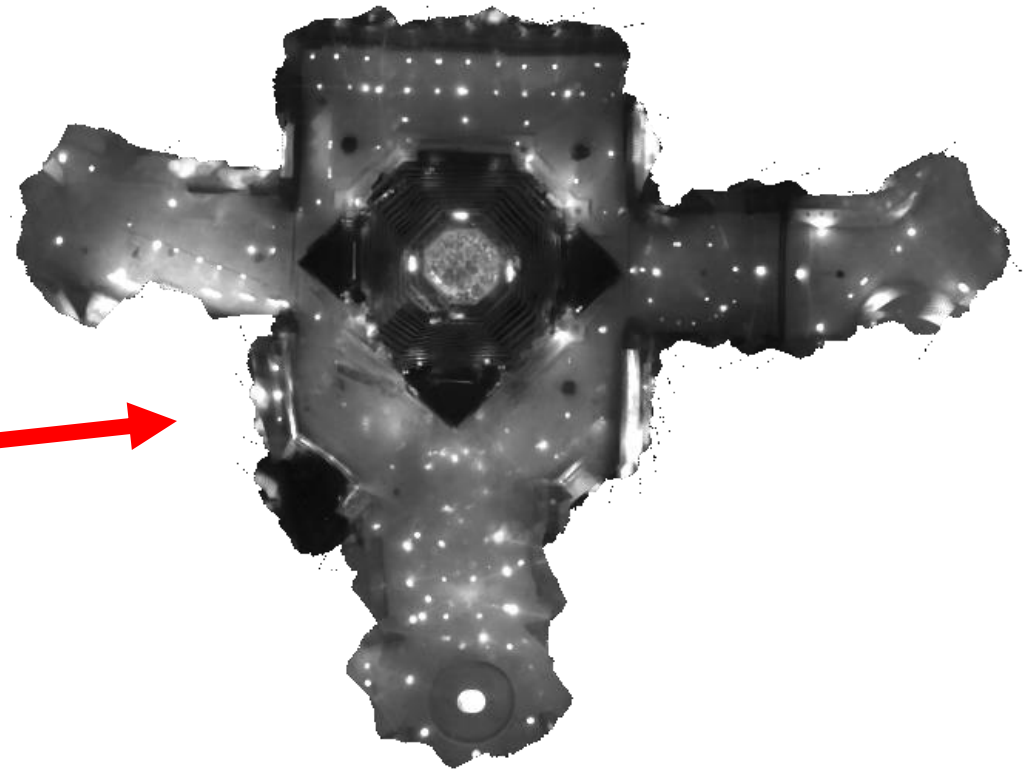
After Incorporating Ten Ultrasound Scans



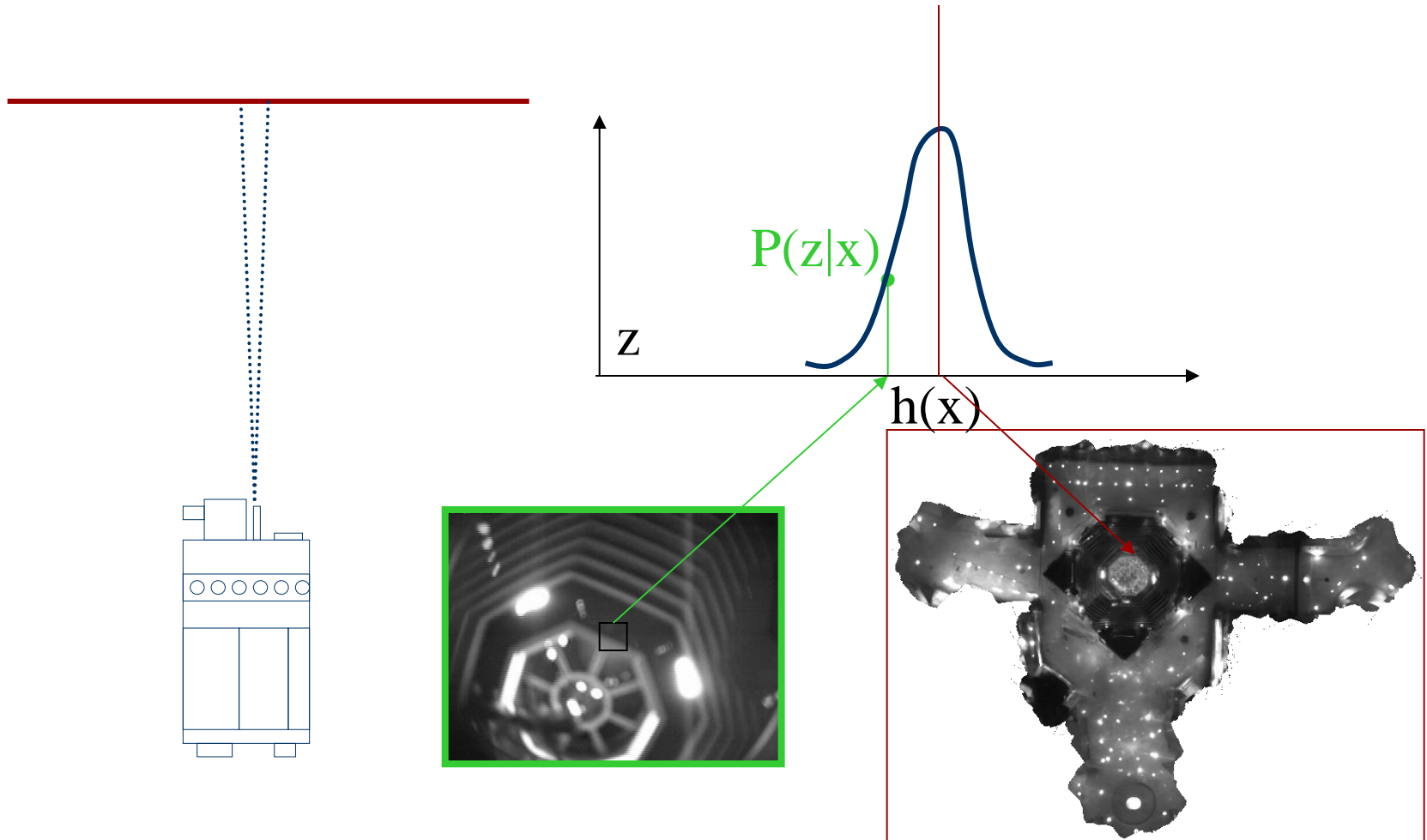
After Incorporating 65 Ultrasound Scans



Using Ceiling Maps for Localization

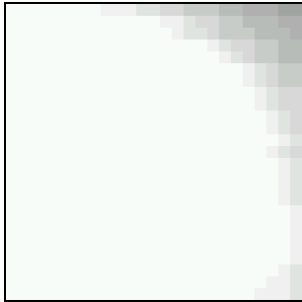


Vision-based Localization



Under a Light

Measurement z :

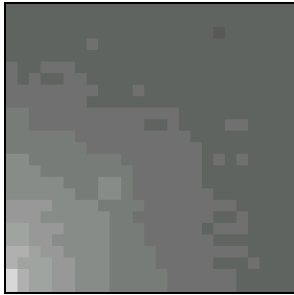


$P(z/x)$:

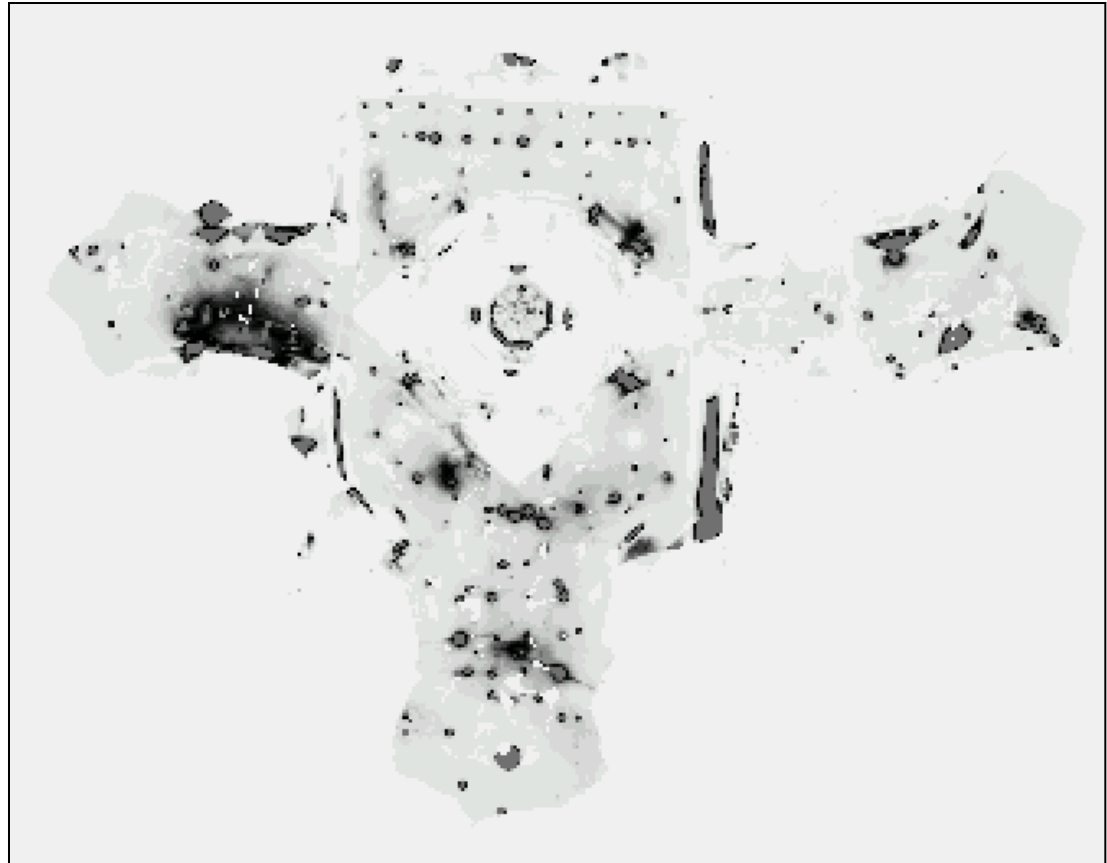


Next to a Light

Measurement z :



$P(z/x)$:

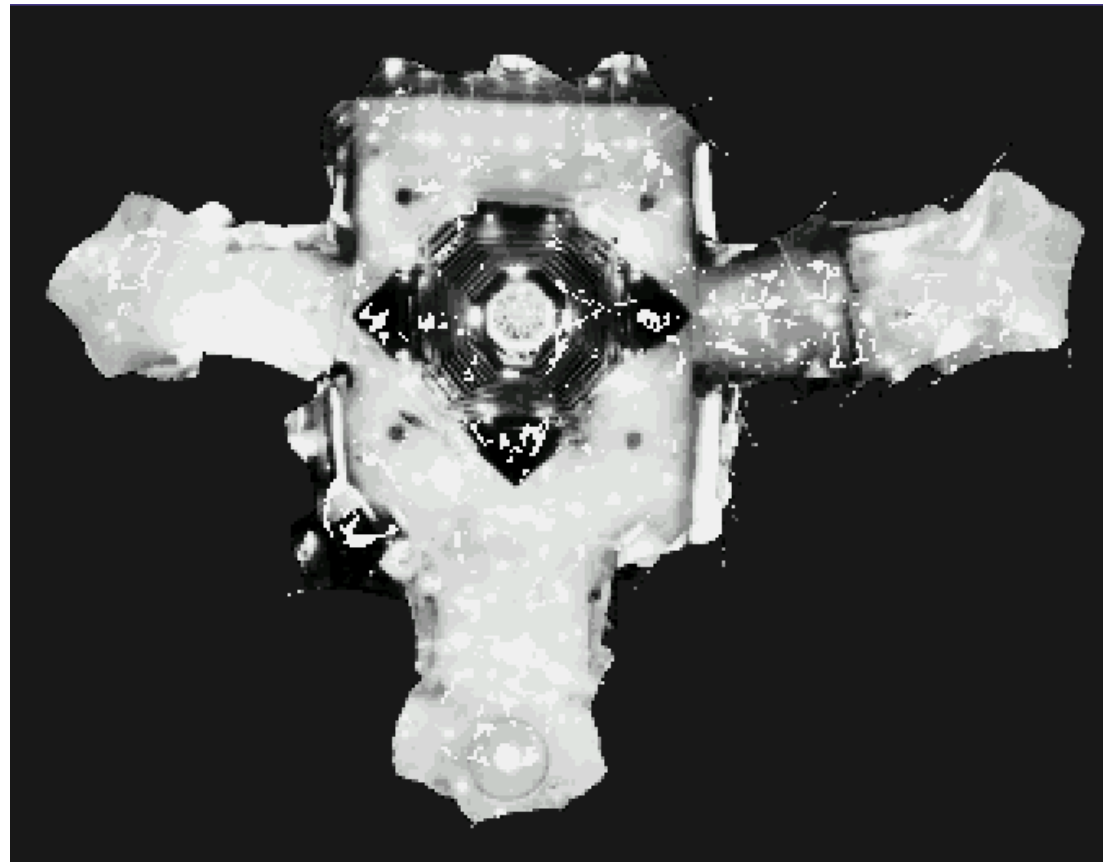


Elsewhere

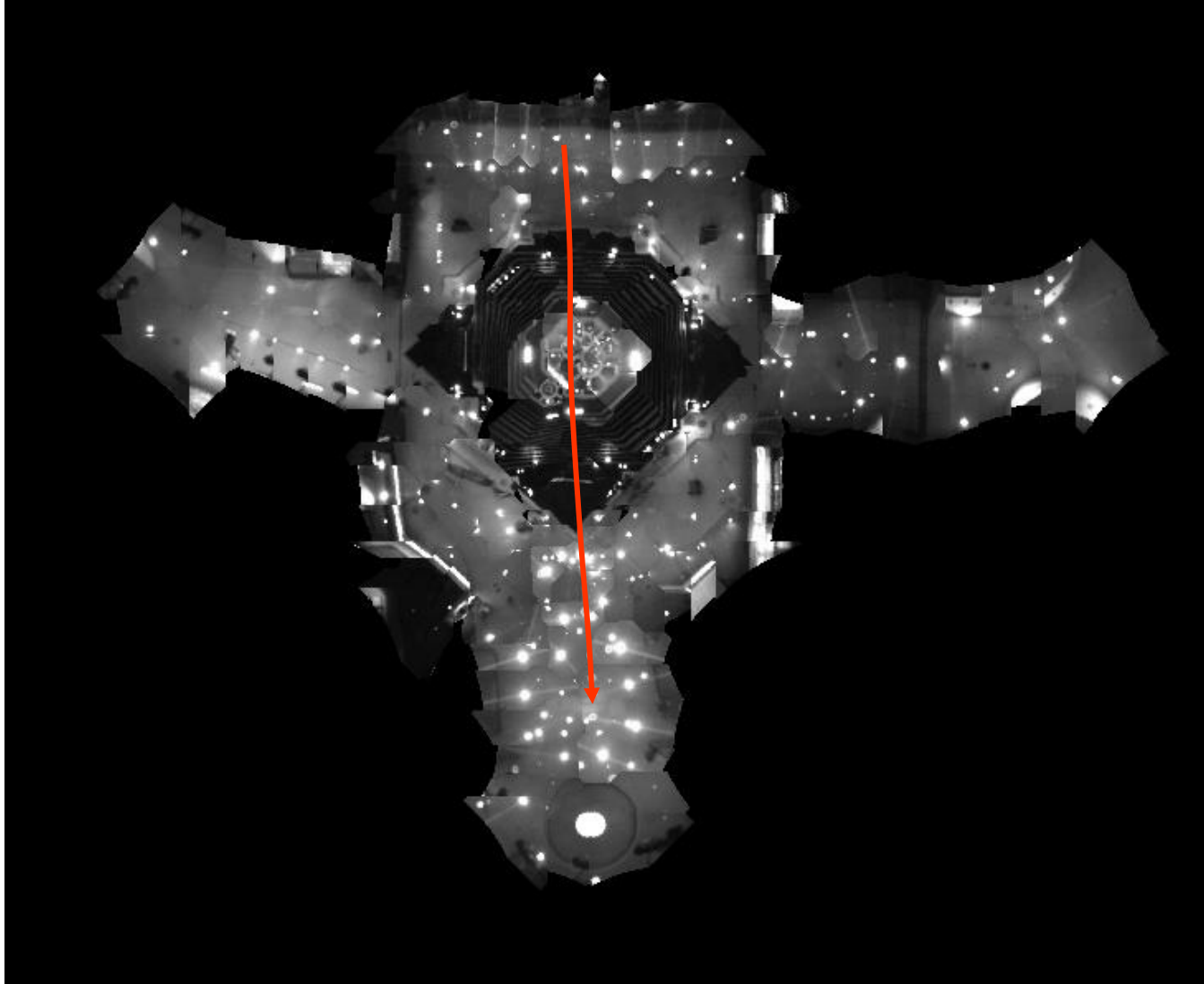
Measurement z :



$P(z/x)$:



Global Localization Using Vision



Limitations

- The approach described so far is able
 - to track the pose of a mobile robot and
 - to globally localize the robot
- How can we deal with localization errors (i.e., the kidnapped robot problem)?

Approaches

- Randomly insert a fixed number of samples
- This assumes that the robot can be teleported at any point in time
- Alternatively, insert random samples proportional to the average likelihood of the particles

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 – PF 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.