Introduction to Mobile Robotics

Bayes Filter – Particle Filter and Monte Carlo Localization

Wolfram Burgard, Michael Krawez

UTN

Particle Filter

- 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 distributions**

- Basic principle
  - Set of state hypotheses (“particles”)
  - Survival-of-the-fittest

Mathematical Description

- Set (actually a multi-set) of weighted samples

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

state hypothesis   importance weight

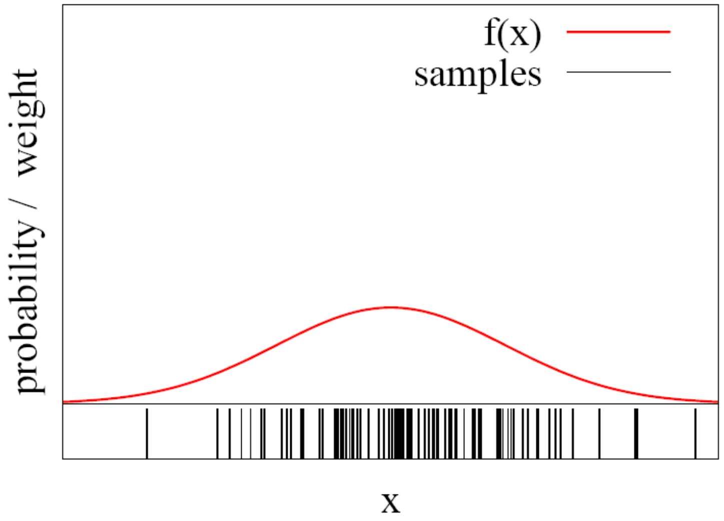
- The samples represent the posterior

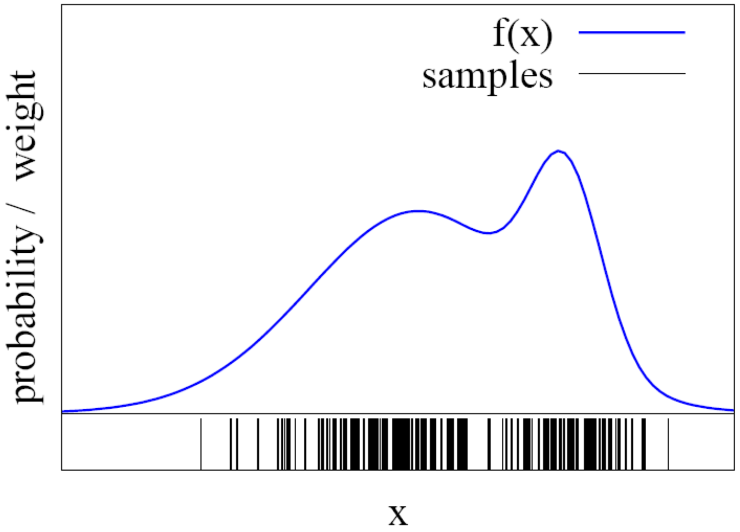
$$\begin{cases} 1 & \text{if } x = s^{[i]} \\ 0 & \text{otherwise} \end{cases}$$

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

Function Approximation

- Particle sets can be used to approximate functions





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

Bayes Filter with Particle Sets

- Measurement update

$$Bel(x) \leftarrow p(z|x)\overline{Bel}(x)$$

$$= p(z|x)\sum_i w_i\ \delta_{s^{[i]}}(x) = \sum_i p\left(z\middle|s^{[i]}\right) w_i\ \delta_{s^{[i]}}(x)$$

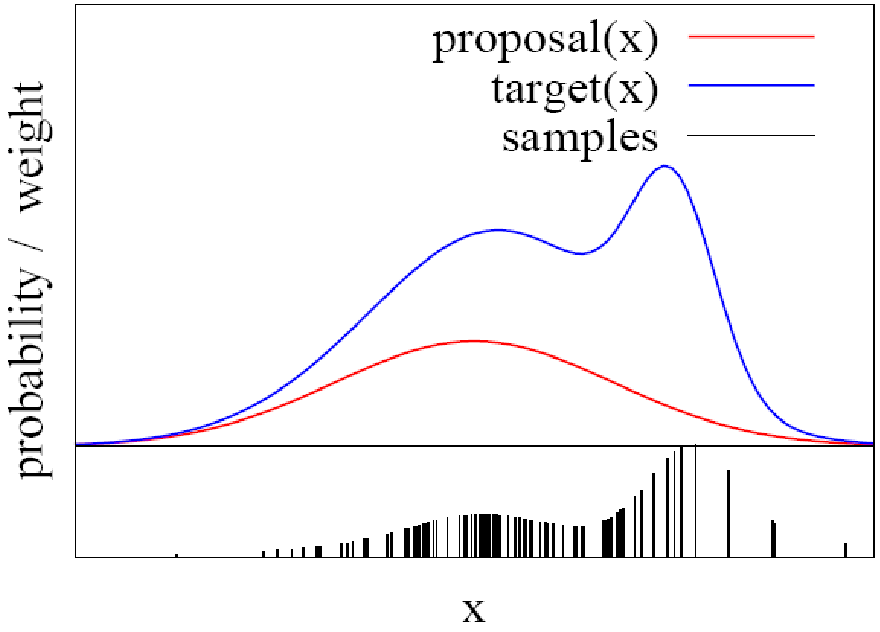
- Motion update

$$\overline{Bel}(x) \leftarrow \int p(x \mid u, x')\, Bel\ (x')dx'$$

$$= \int p(x \mid u, x') \sum_i w_i\ \delta_{s^{[i]}}(x')dx' = \sum_i p(x \mid u, s^{[i]})\ \ w_i$$

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:
  $f(x)>0 \rightarrow g(x)>0$



Particle Filter Algorithm

- Sample the next generation of 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

1. Algorithm **particle_filter**(S_{t-1}, u_t, z_t) returns S_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 $p(x_t \mid x_{t-1}, u_t)$ using $x_{t-1}^{j(i)}$ and u_t
6. $w_t^i = p(z_t \mid x_t^i)$ — ***Compute importance weight***
7. $\eta = \eta + w_t^i$ — ***Update normalization factor***
8. $S_t = S_t \cup \{< x_t^i, w_t^i >\}$ — ***Add to new particle set***
9. **For** $i = 1, \dots, n$
10. $w_t^i = w_t^i / \eta$ — ***Normalize weights***

Particle Filter Algorithm

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

target distribution

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

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

proposal distribution

Importance factor for x^i_t:

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

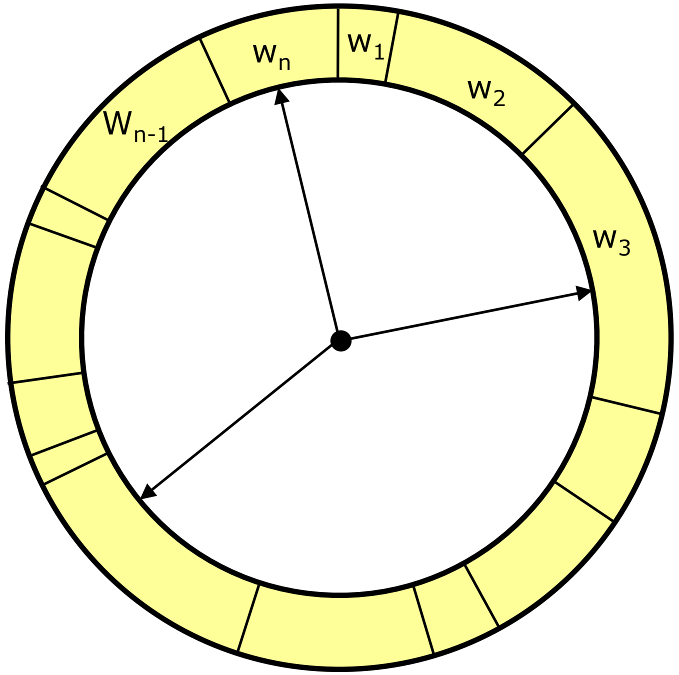
$$= \frac{\eta\, p(z_t \mid x_t)\, p(x_t \mid x_{t-1}, u_t)\, Bel\,(x_{t-1})}{p(x_t \mid x_{t-1}, u_t)\, Bel\,(x_{t-1})}$$

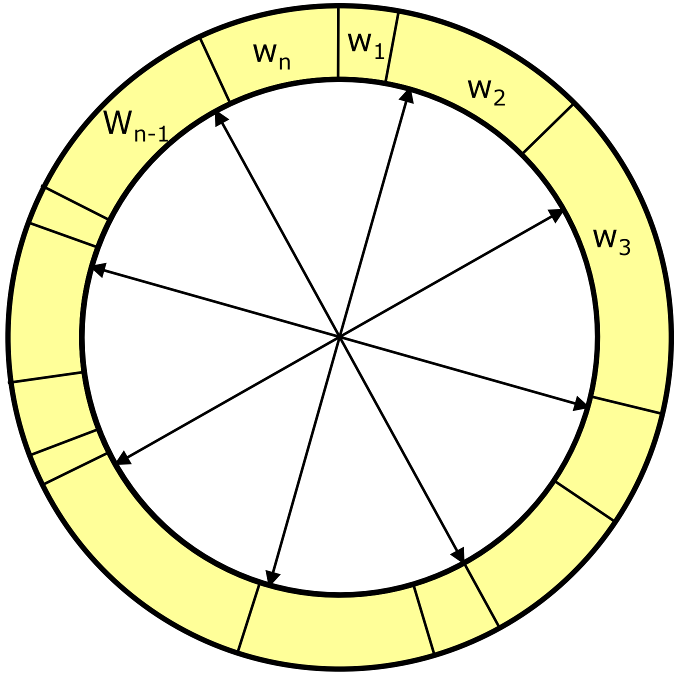
$$\propto p(z_t \mid x_t)$$

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, O(n log(n))

- Stochastic universal sampling
- Systematic resampling
- Linear time complexity O(n)
- 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. $\quad S' = S' \cup \{< x^i, n^{-1} >\}$ — *Insert*
10. $\quad u_{j+1} = u_j + n^{-1}$ — *Increment threshold*

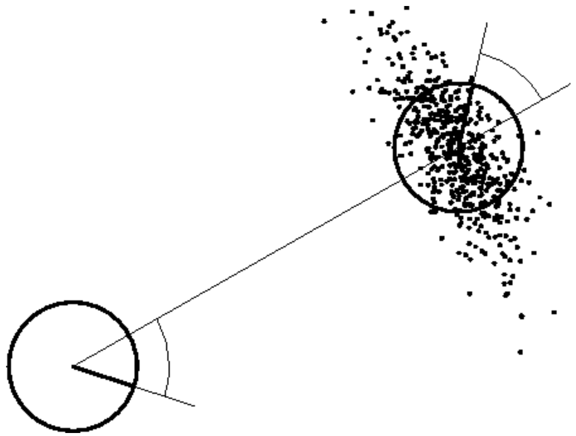
11. **Return** S'

Also called **stochastic universal sampling**

Mobile Robot Localization

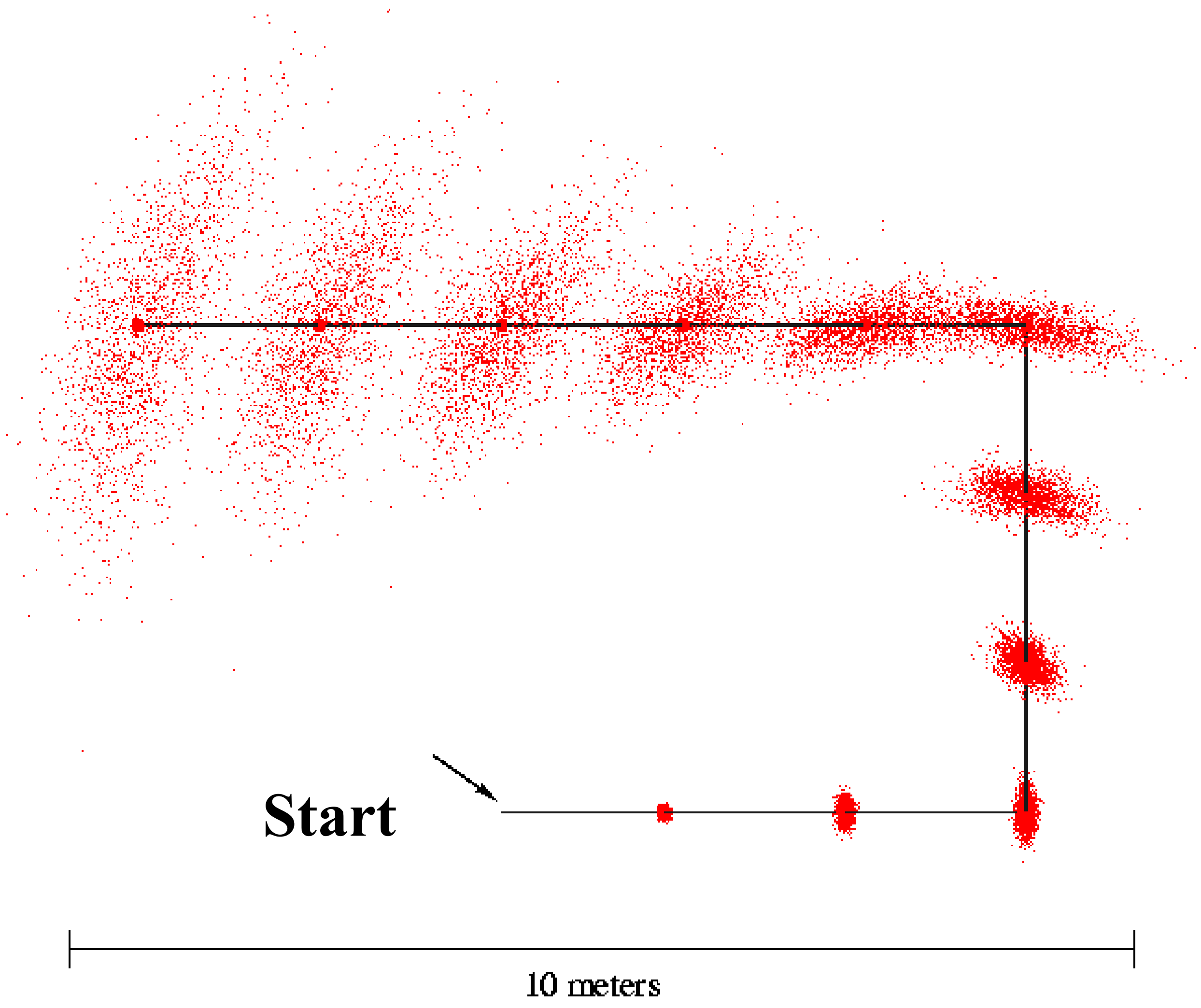
- Each particle is a potential pose of the robot
- Proposal distribution is the motion model of the robot (prediction step)
- The observation model is used to compute the importance weight (correction step)

Motion Model Reminder

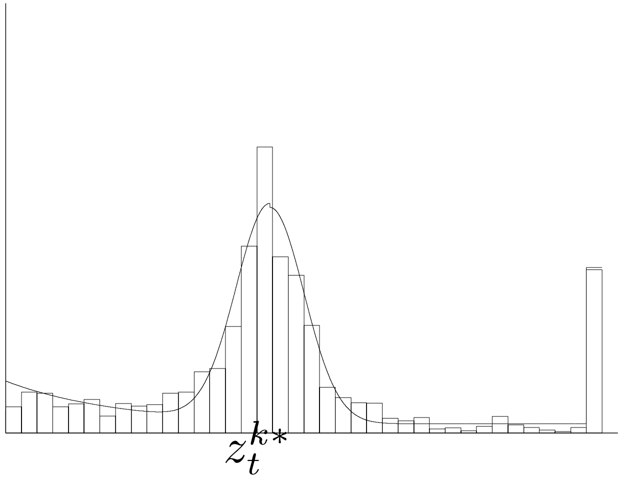


- Uncertainty in the translation of the robot: Gaussian over the traveled distance
- Uncertainty in the rotation of the robot: Gaussians over initial and final rotation
- For each particle, draw a new pose by sampling from these three individual normal distributions

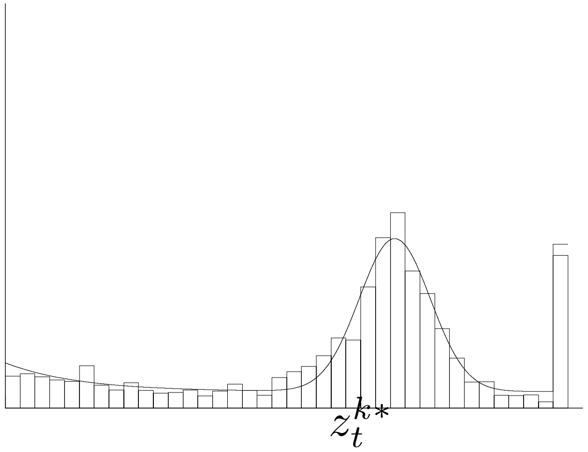
Motion Model Reminder

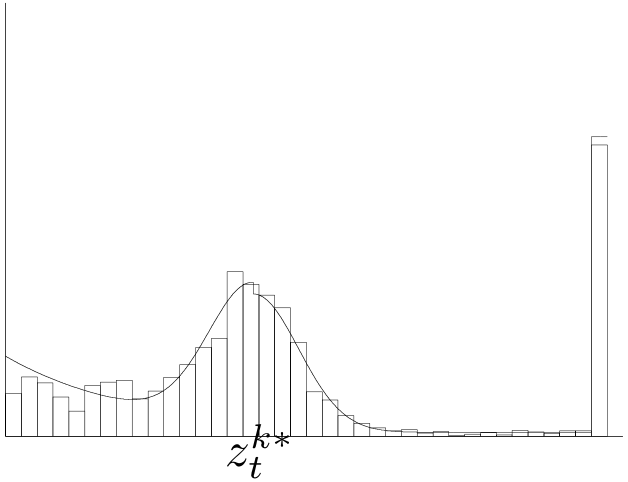


Proximity Sensor Model Reminder

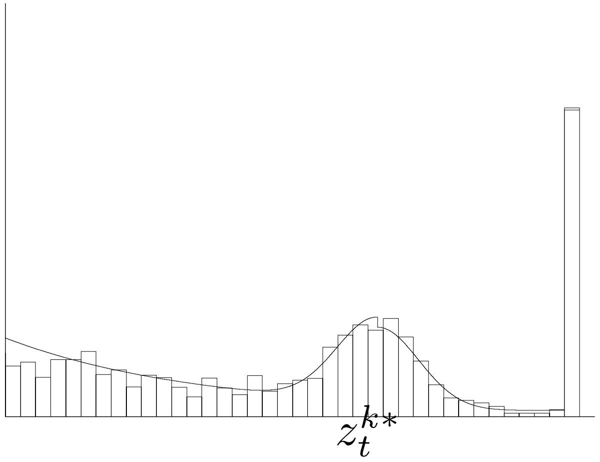


Laser





Sonar



300cm

400cm

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)

Robot Localization using Particle Filters (2)

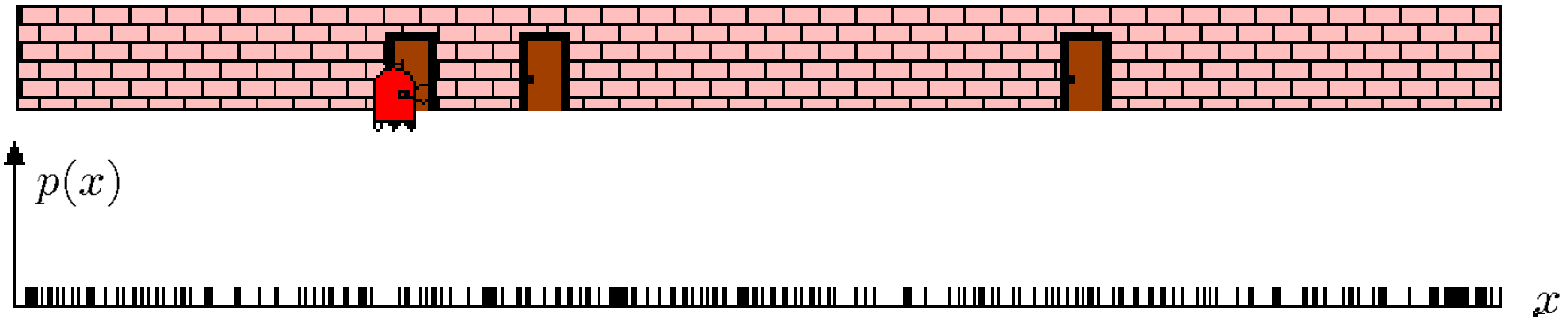
- Particles are drawn from the motion model (proposal distribution)
- Particles are weighted according to the observation model (sensor model)
- Particles are resampled according to the particle weights

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 loose track of the "good" hypotheses
- Resampling ensures that particles stay in the meaningful area of the state space

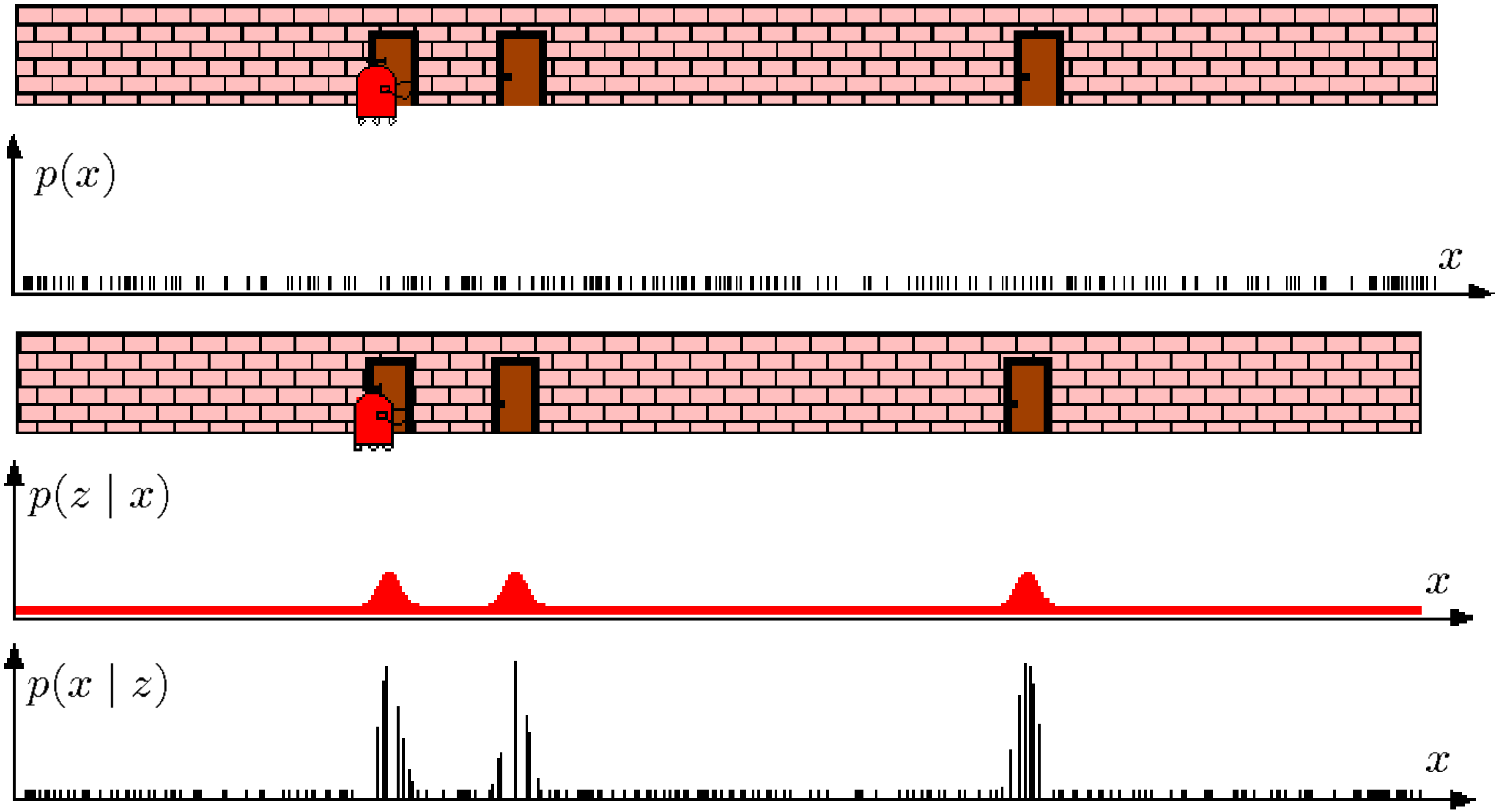
Particle Filters



Sensor Information: Importance Sampling

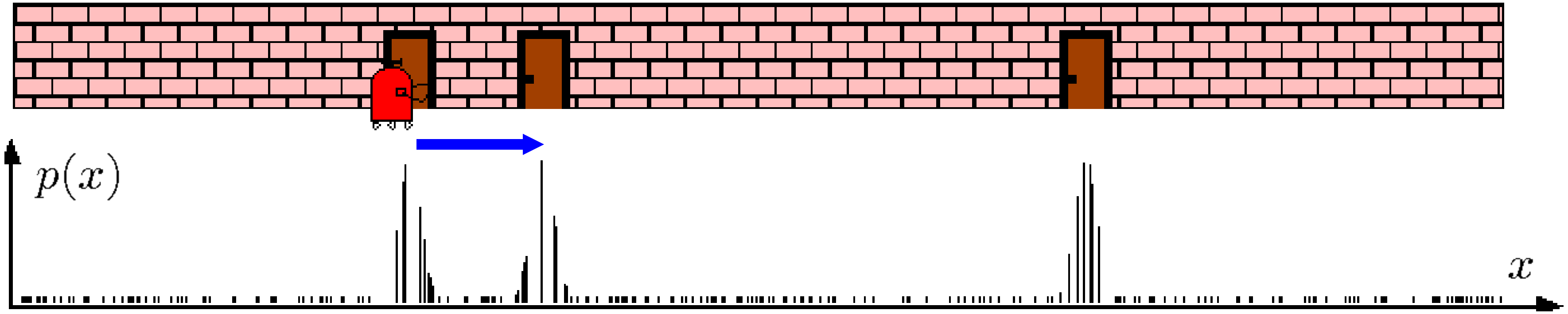
$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

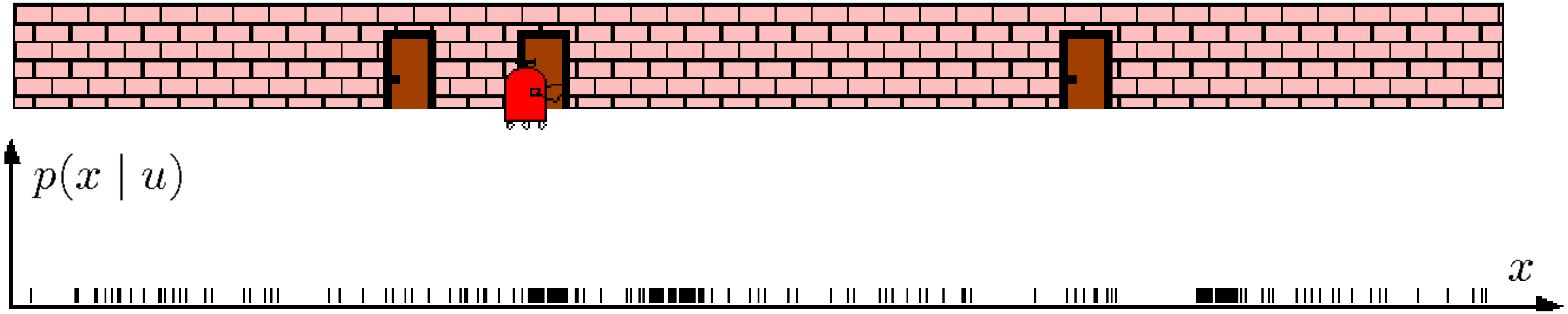
$$w \leftarrow \frac{\alpha\, p(z \mid x)\, Bel^{-}(x)}{Bel^{-}(x)} = \alpha\, p(z \mid x)$$



Robot Motion

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

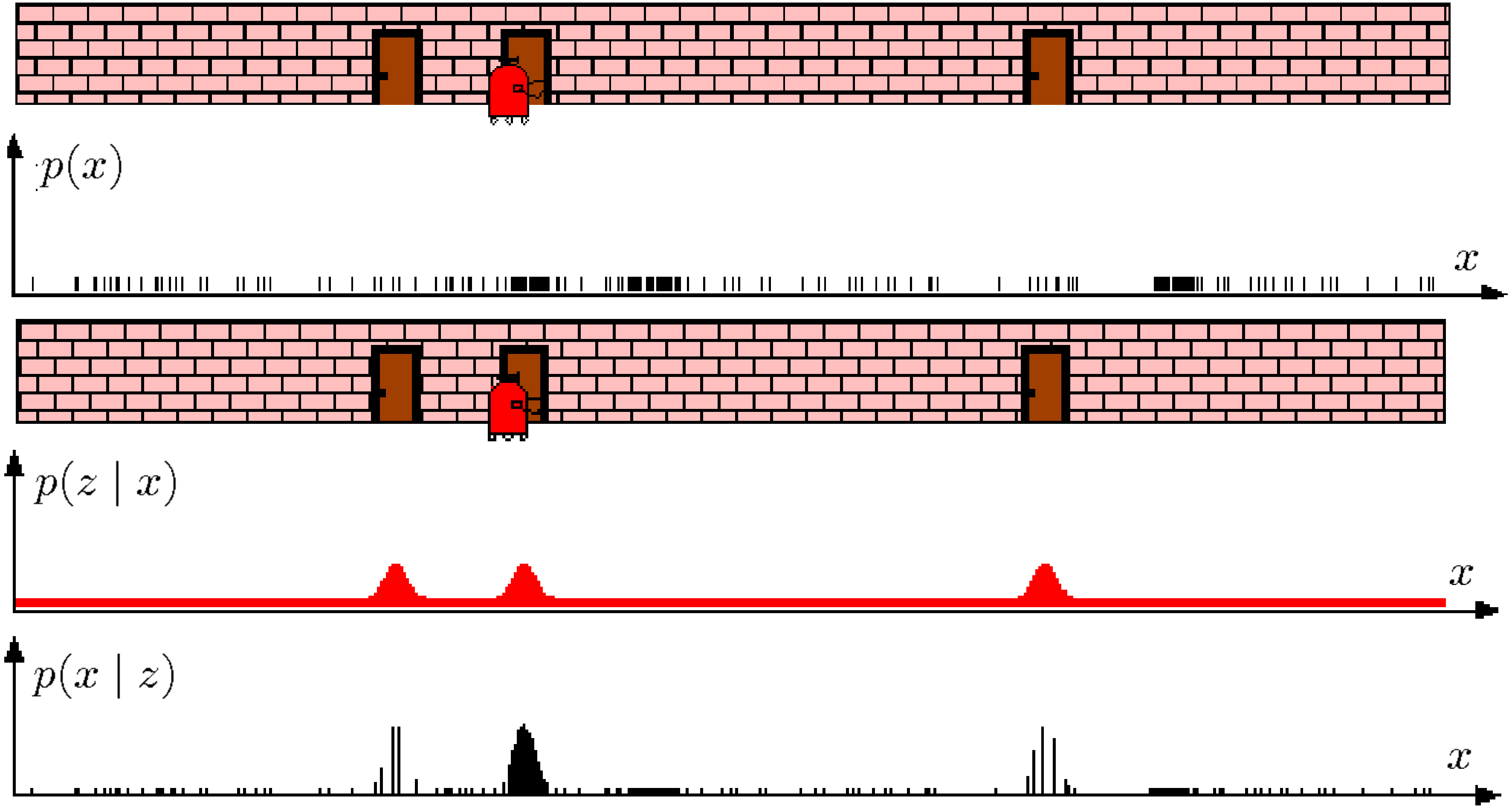




Sensor Information: Importance Sampling

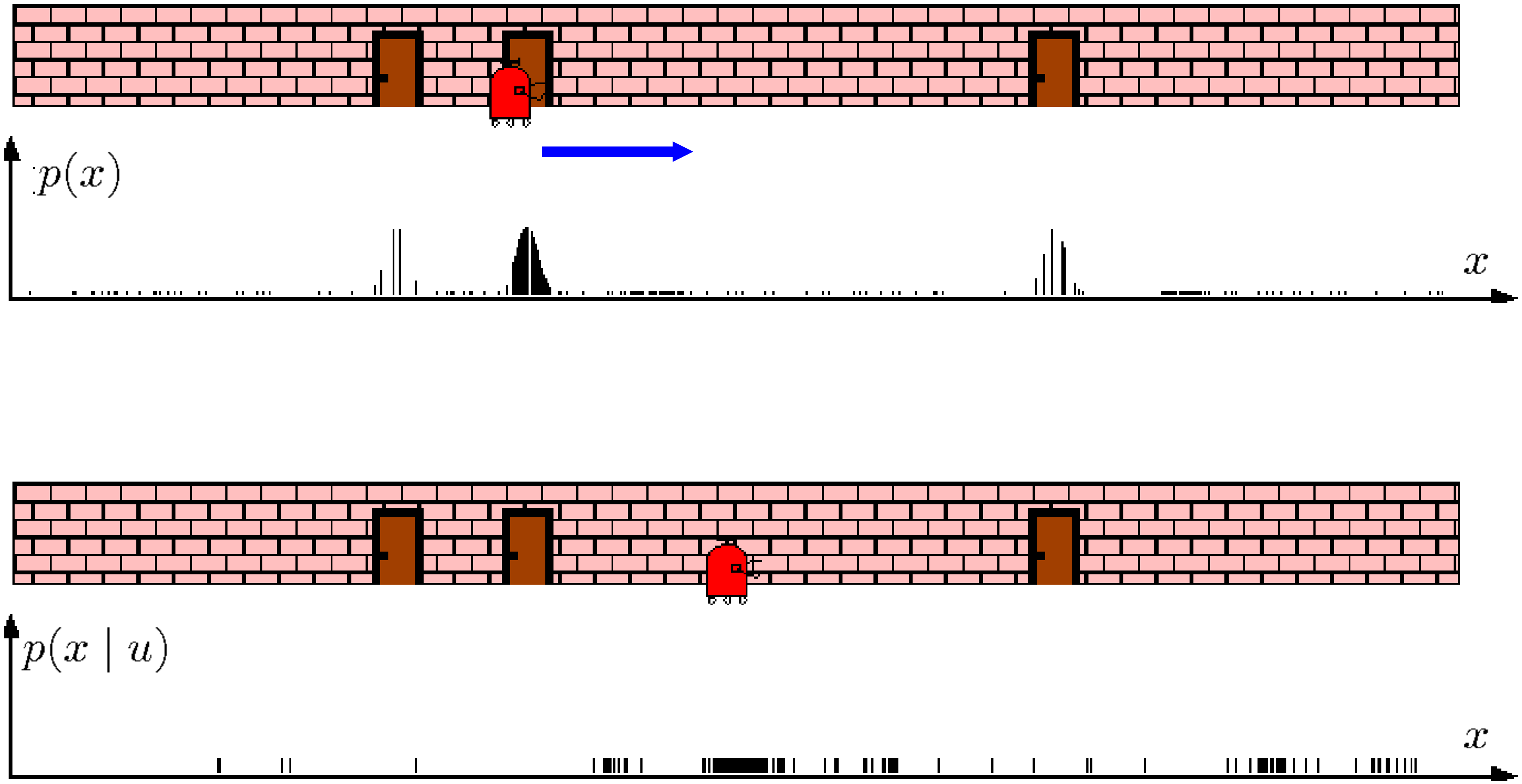
$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

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

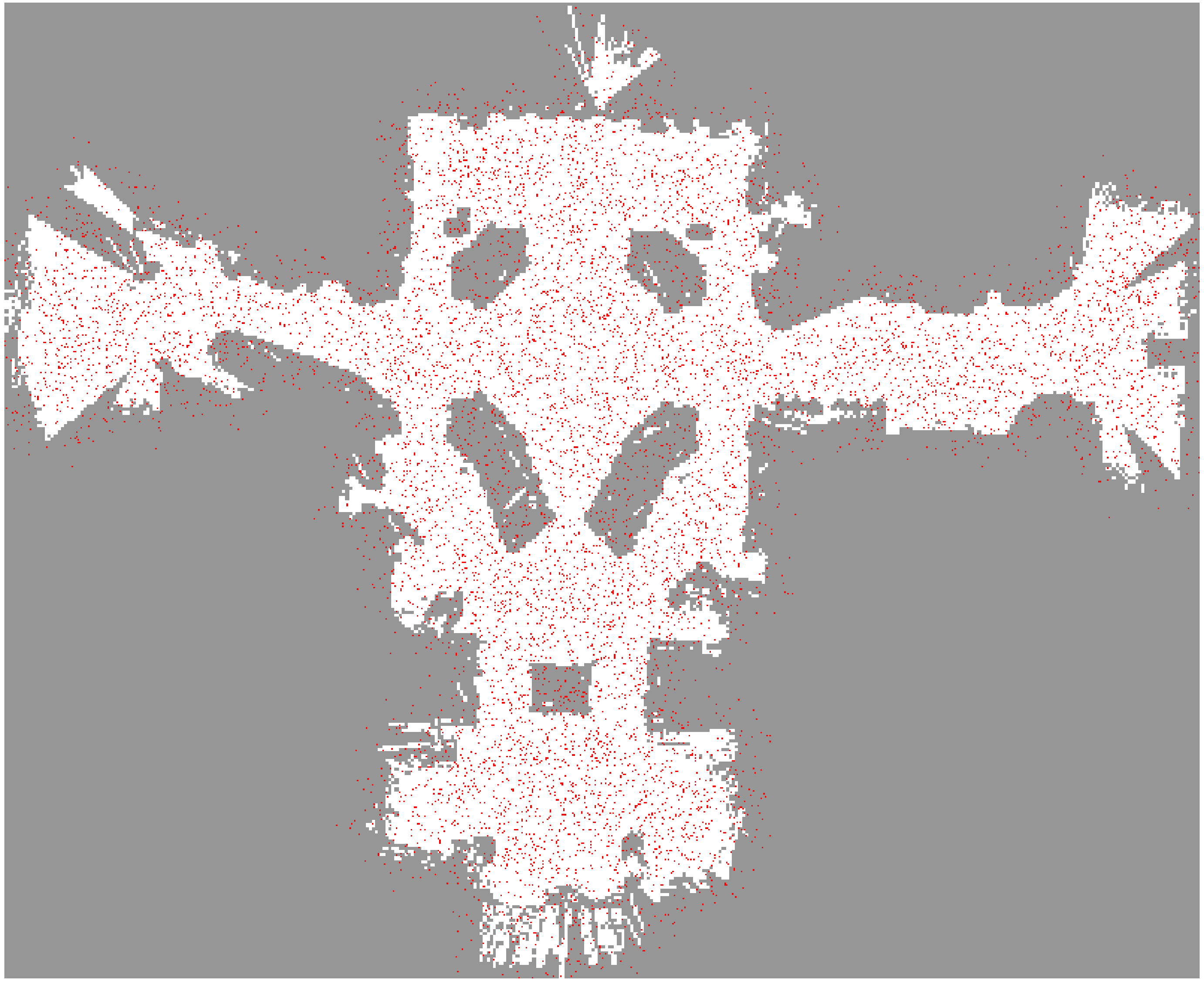


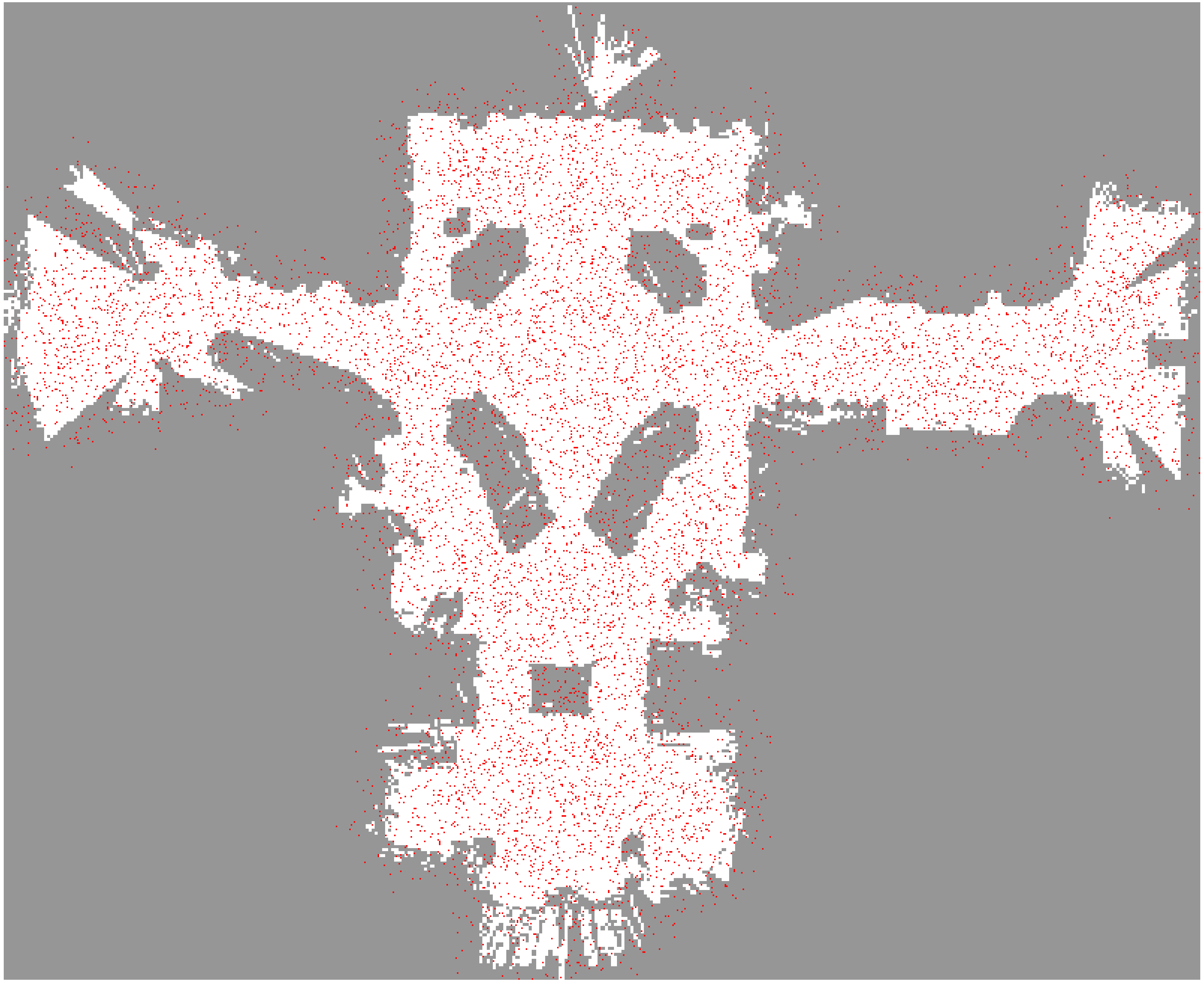
Robot Motion

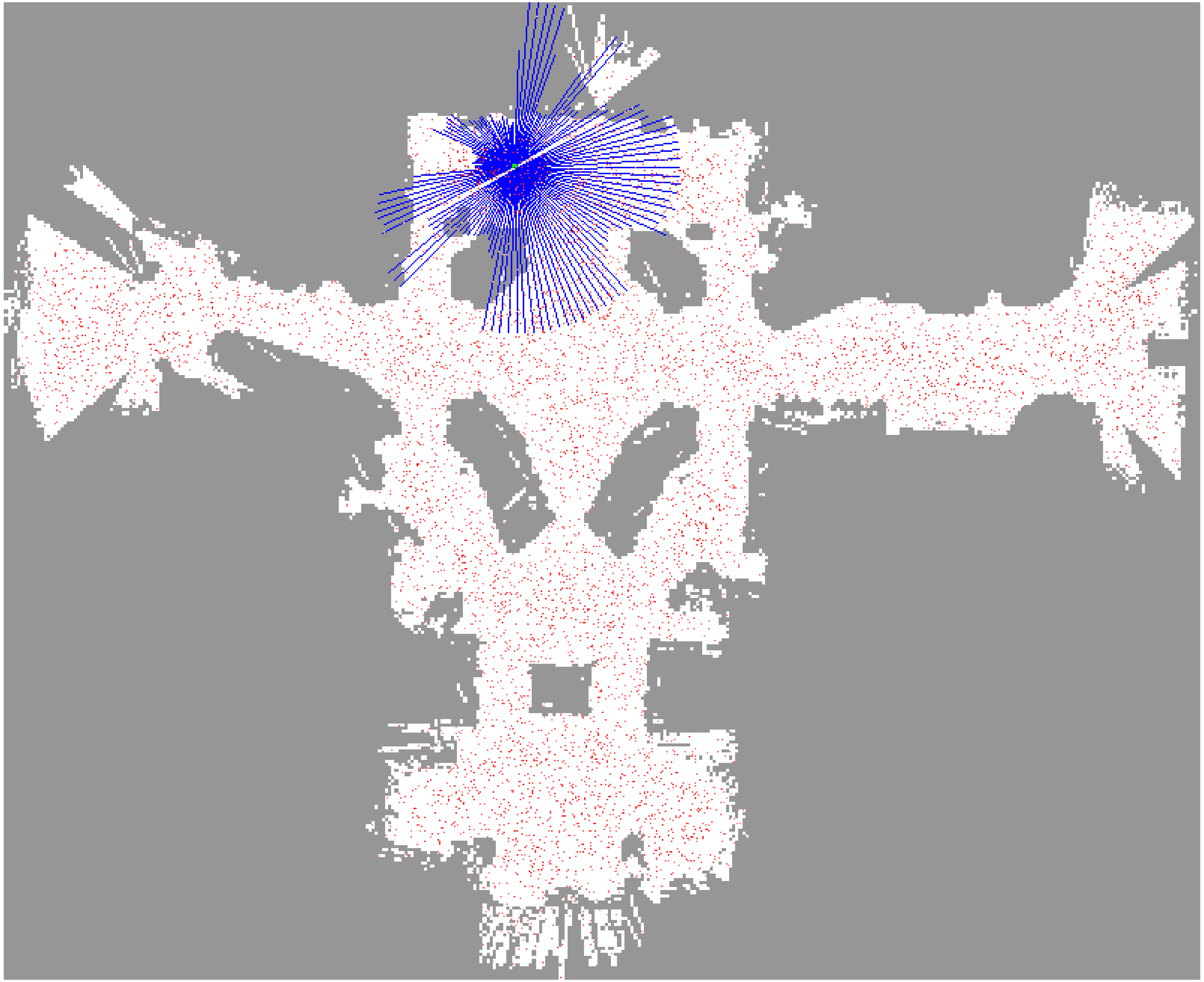
$$Bel^-(x) \leftarrow \int p(x \mid u, x')\, Bel(x')\, \mathrm{d}x'$$

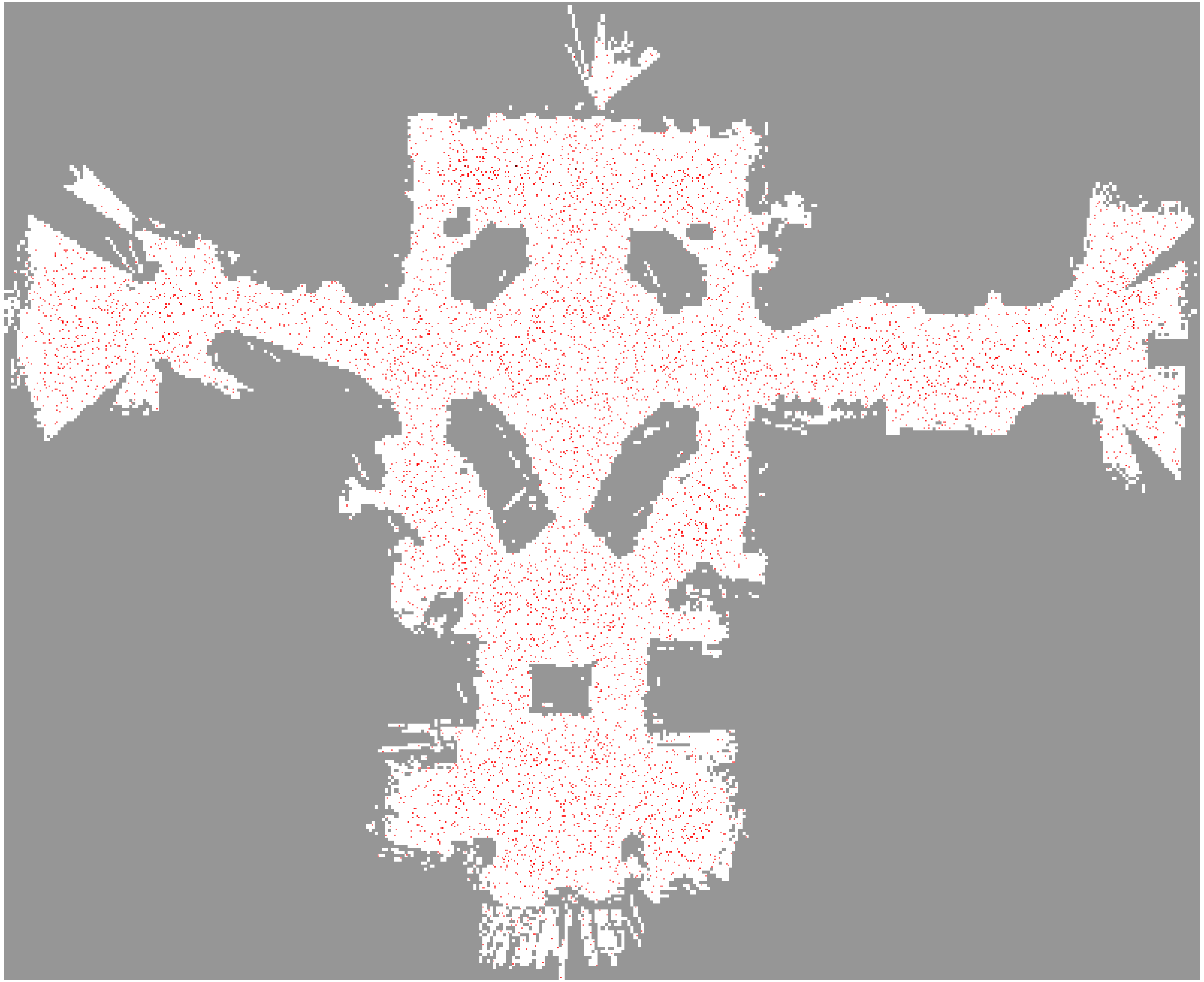


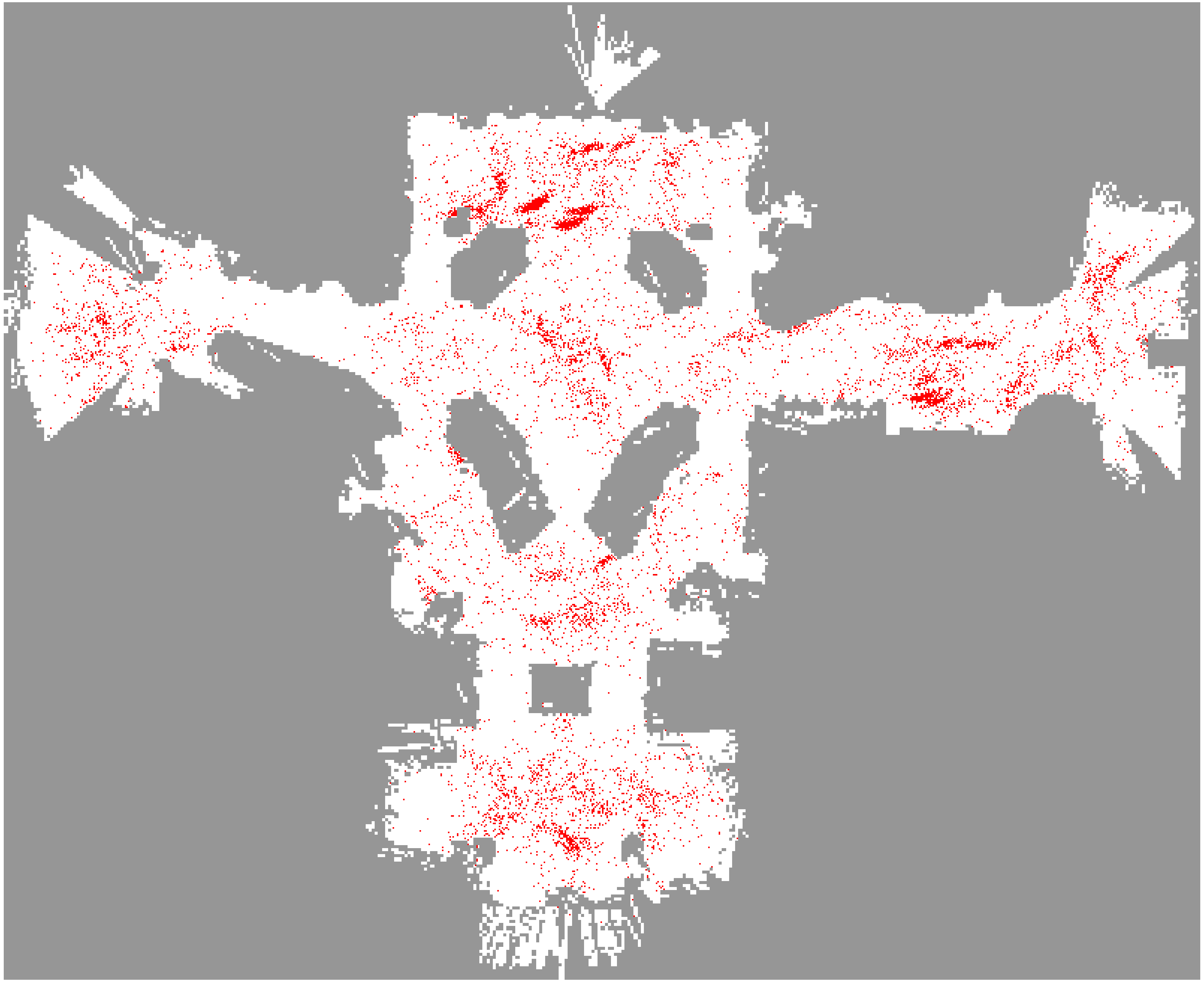
Real World Application (LiDAR)

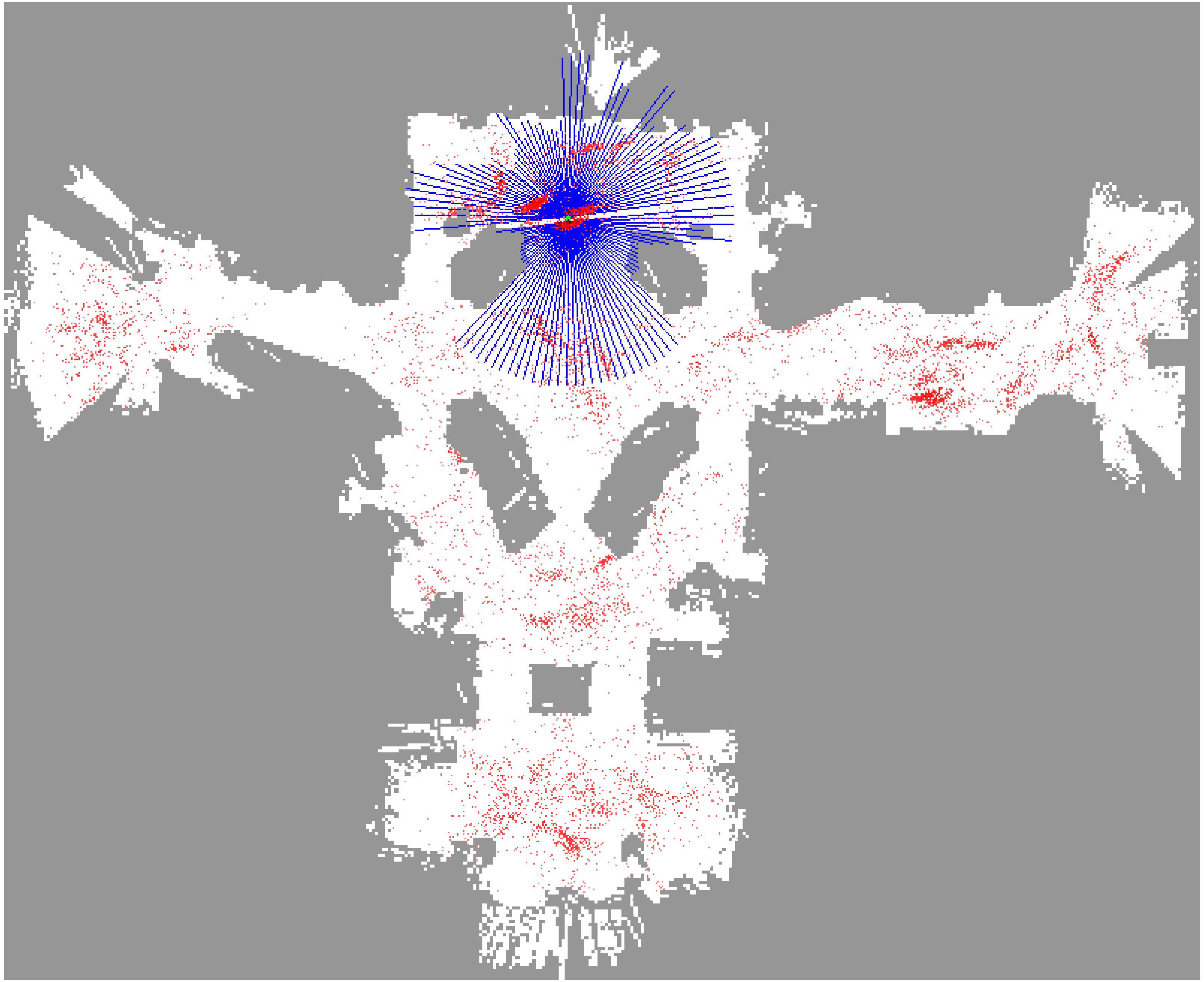


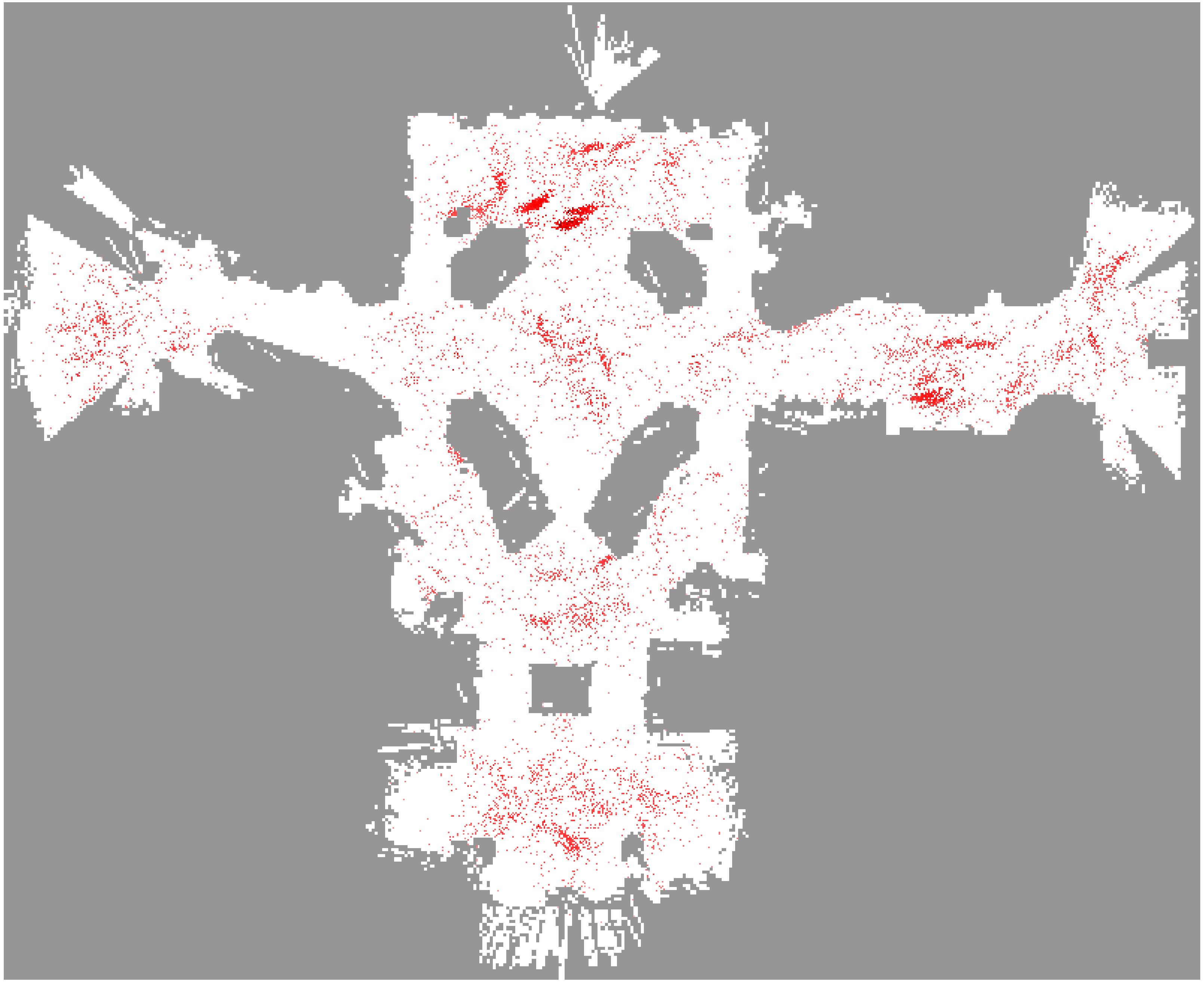


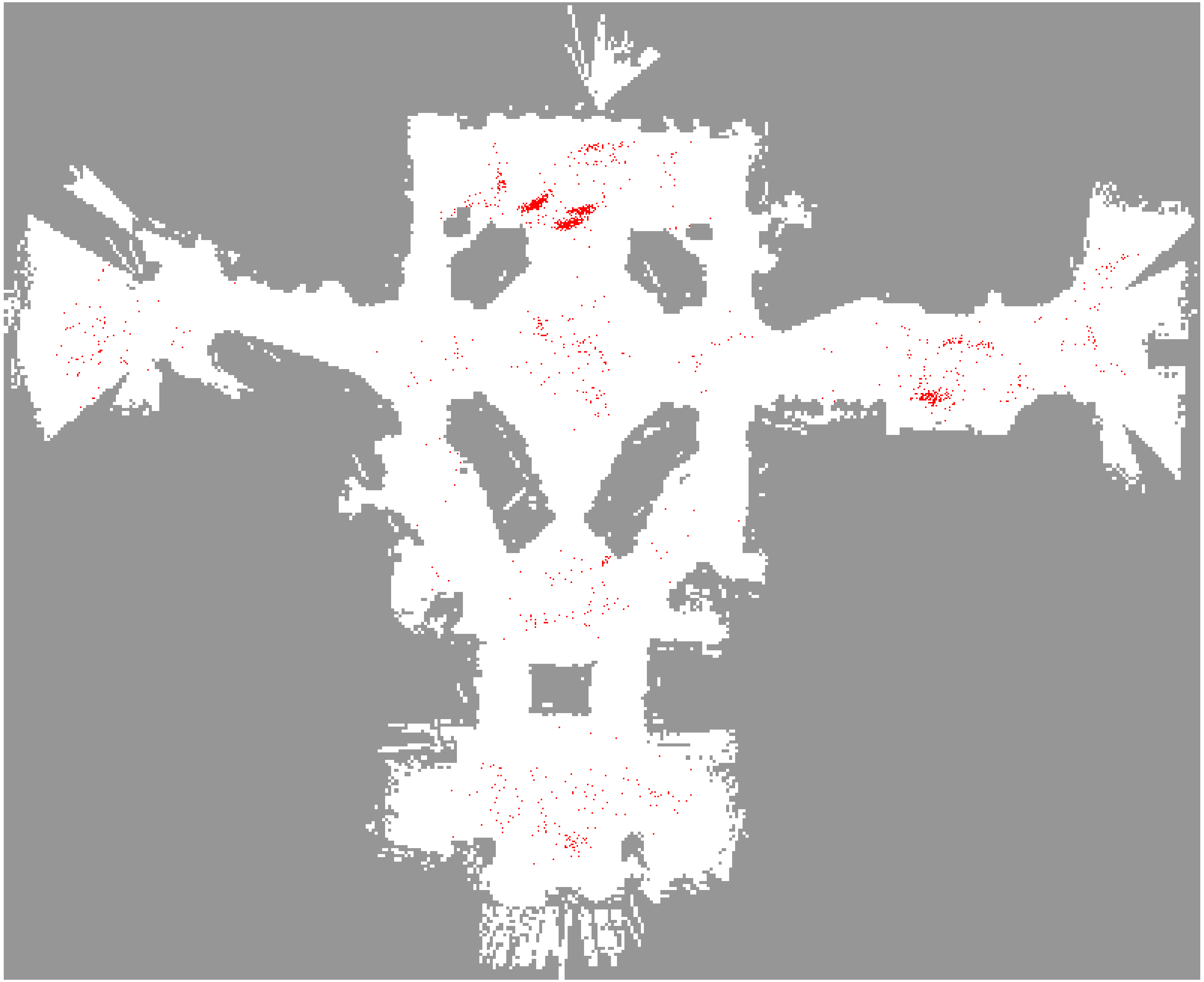


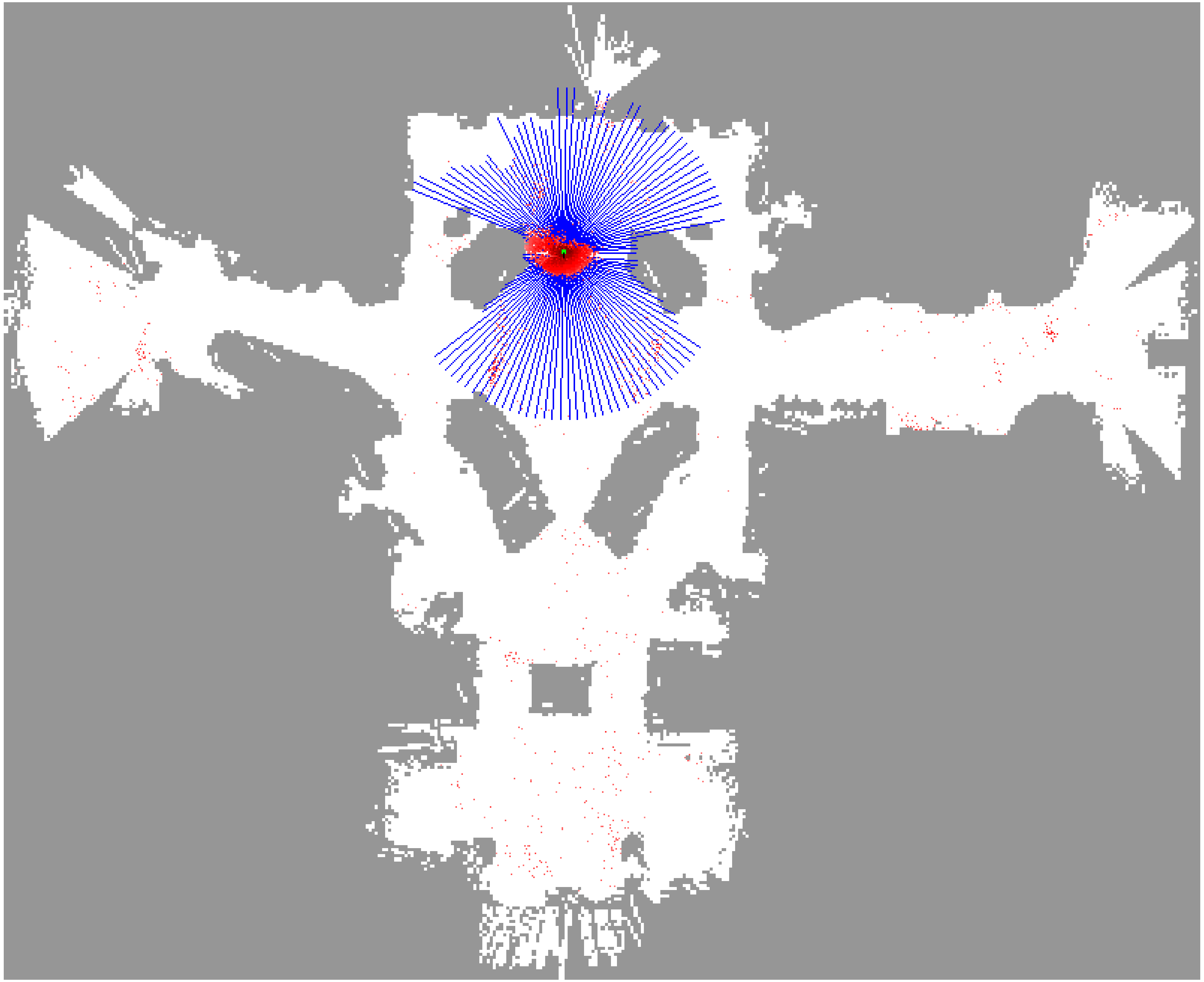


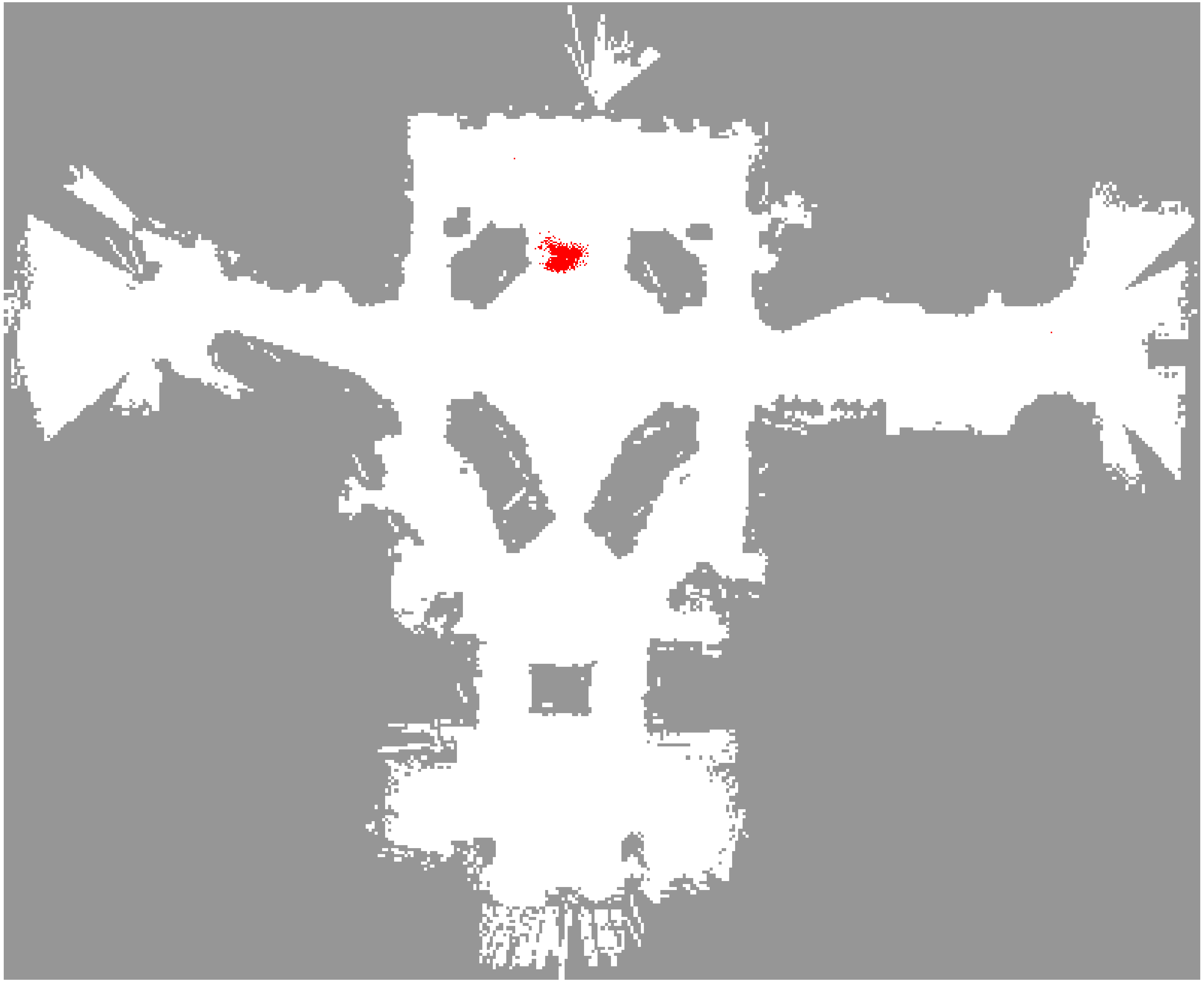


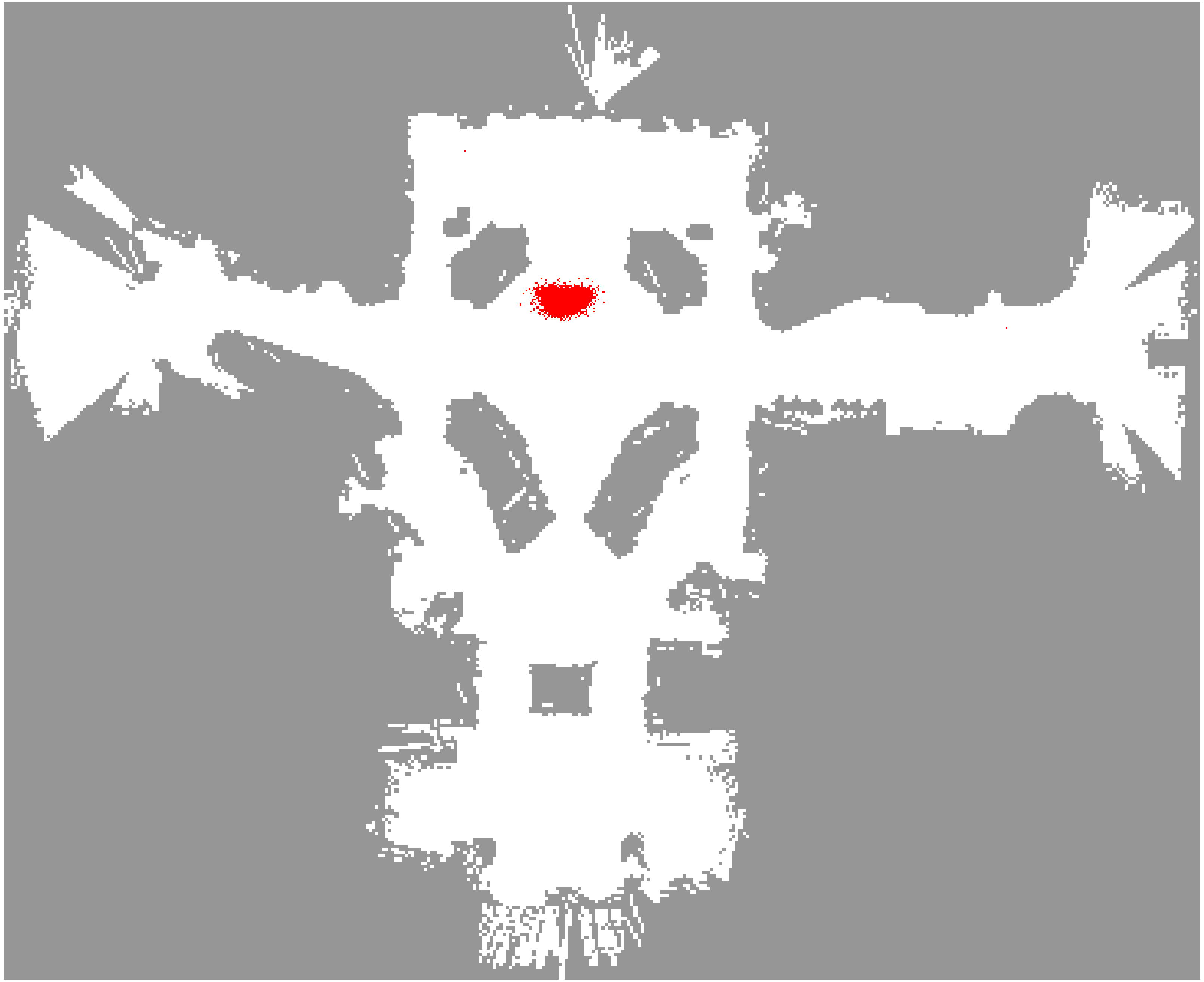


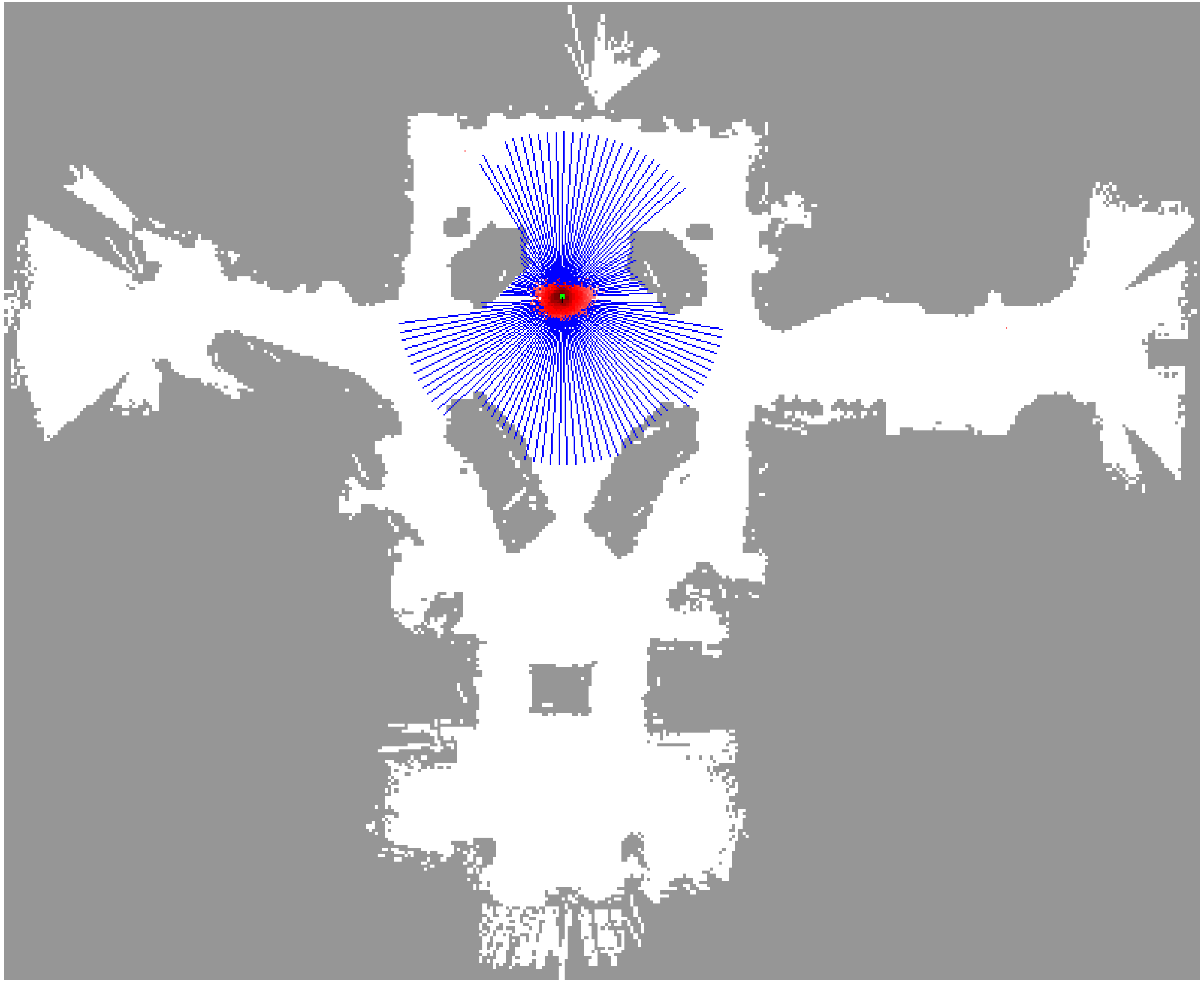


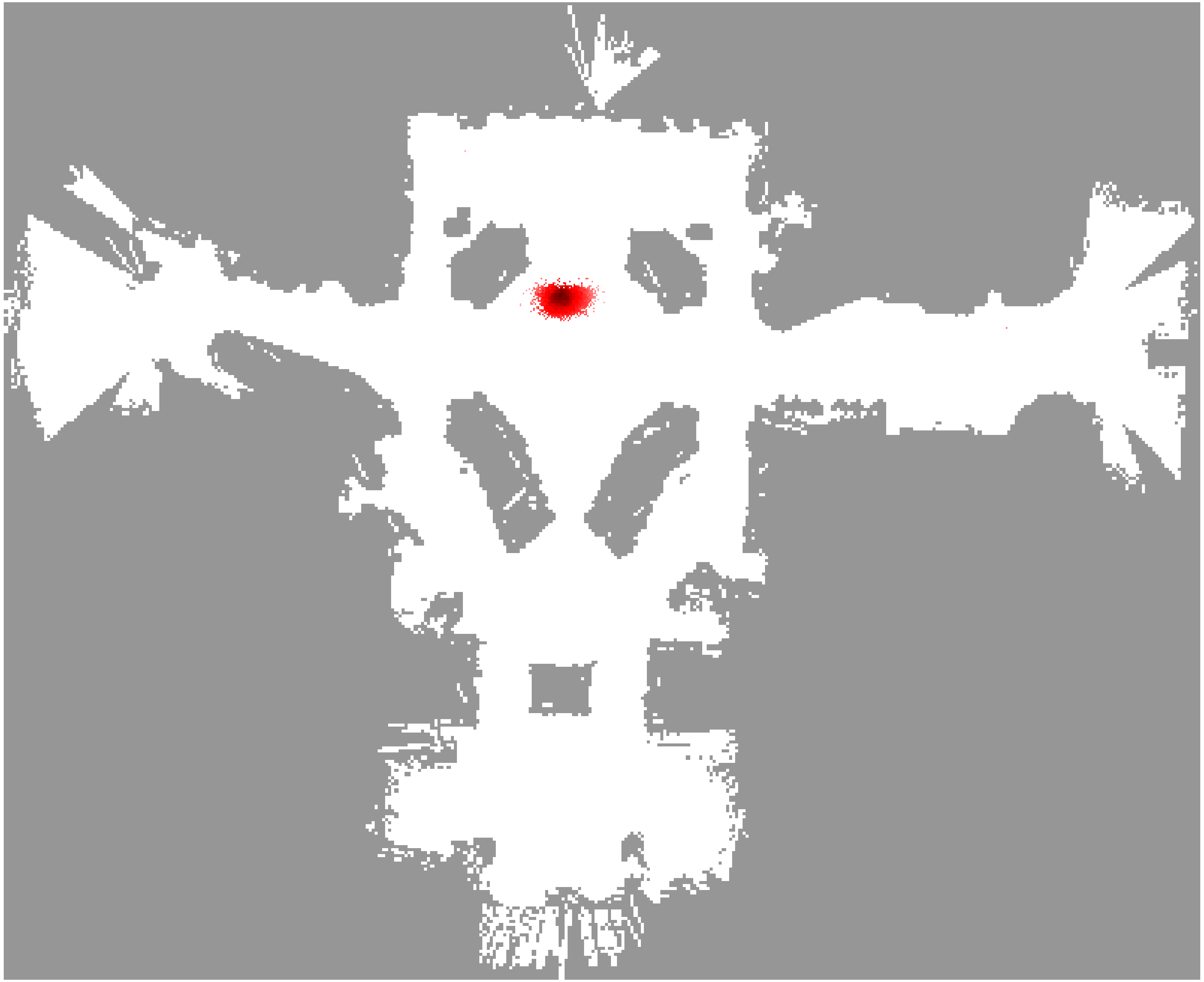


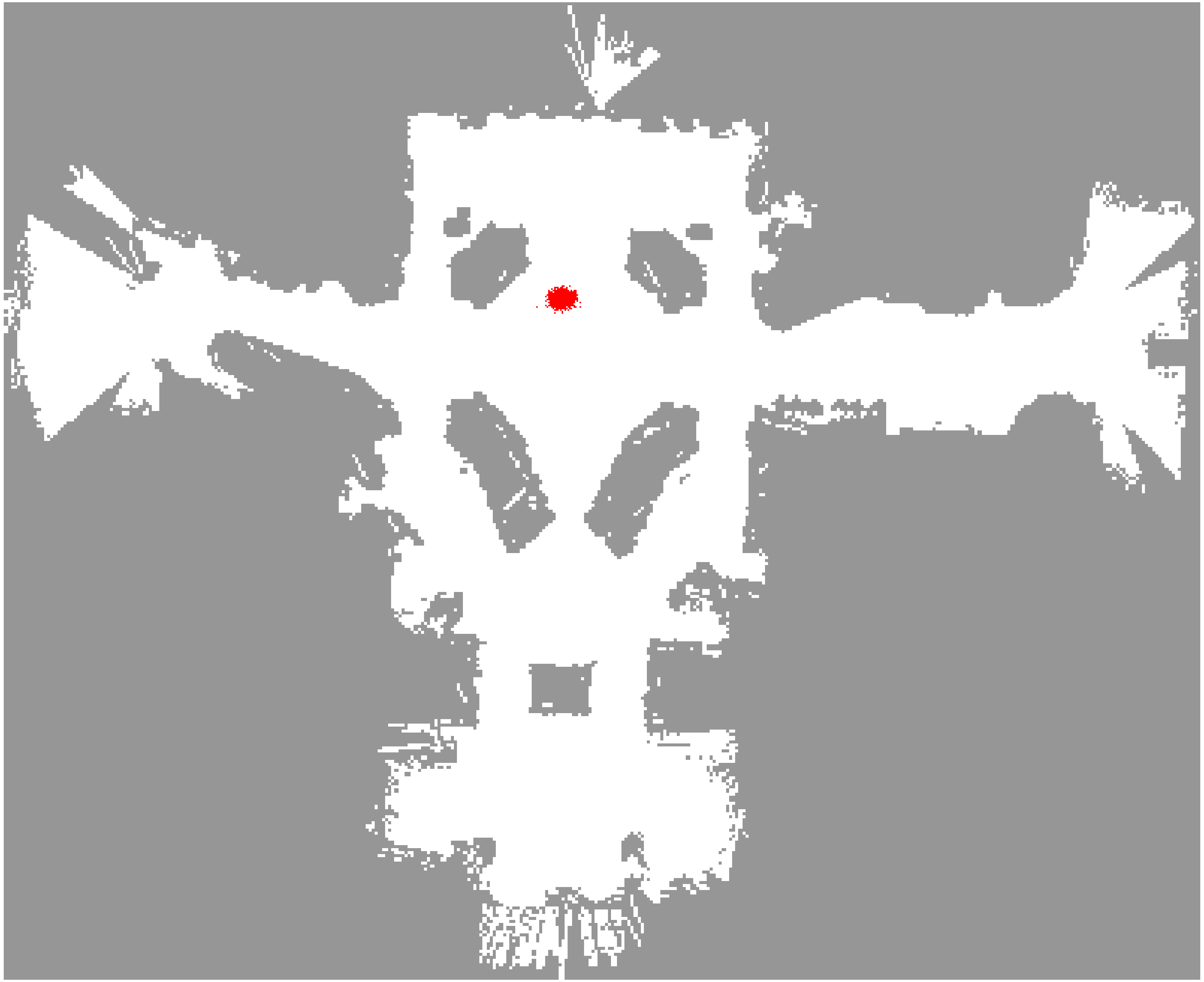


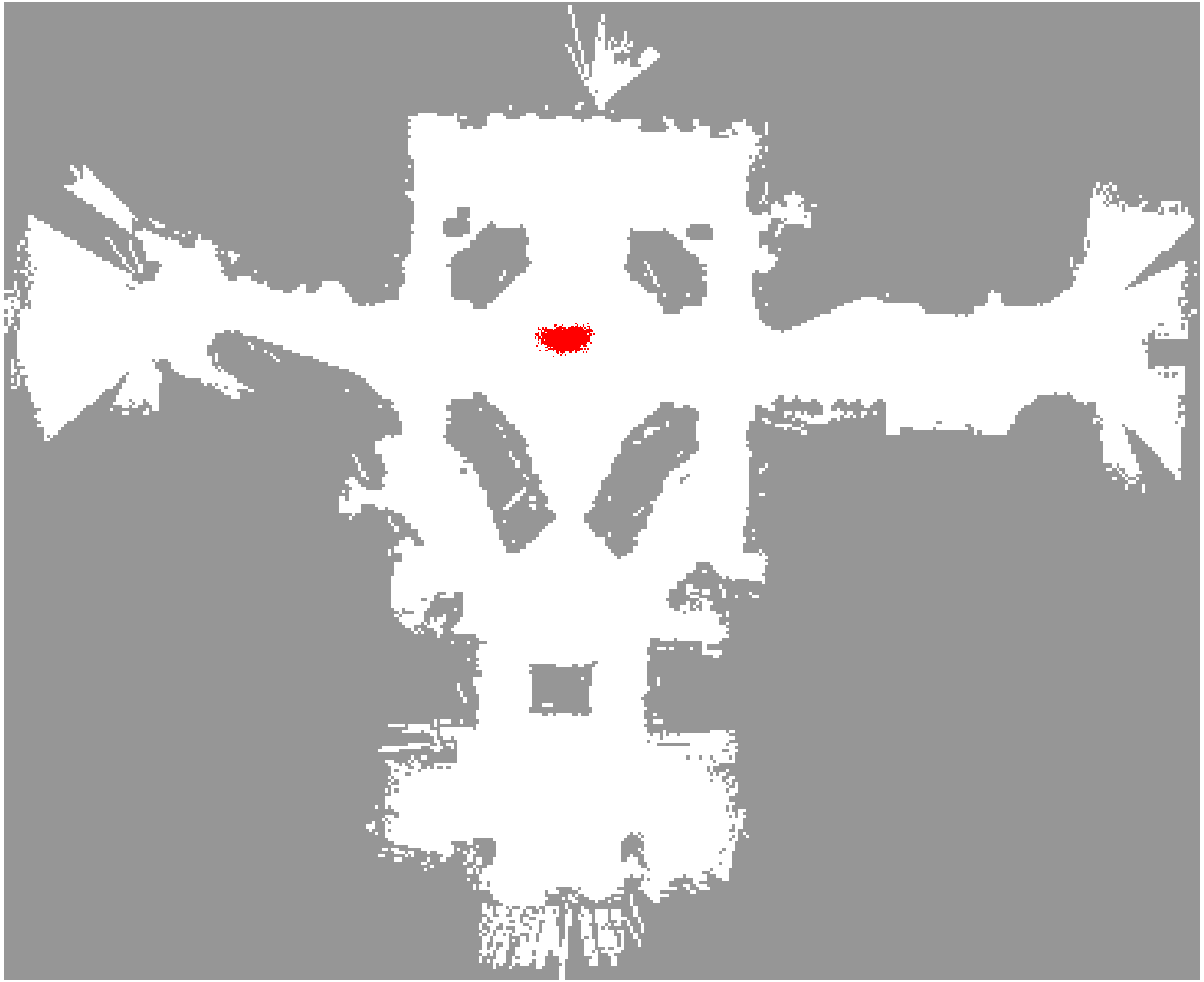


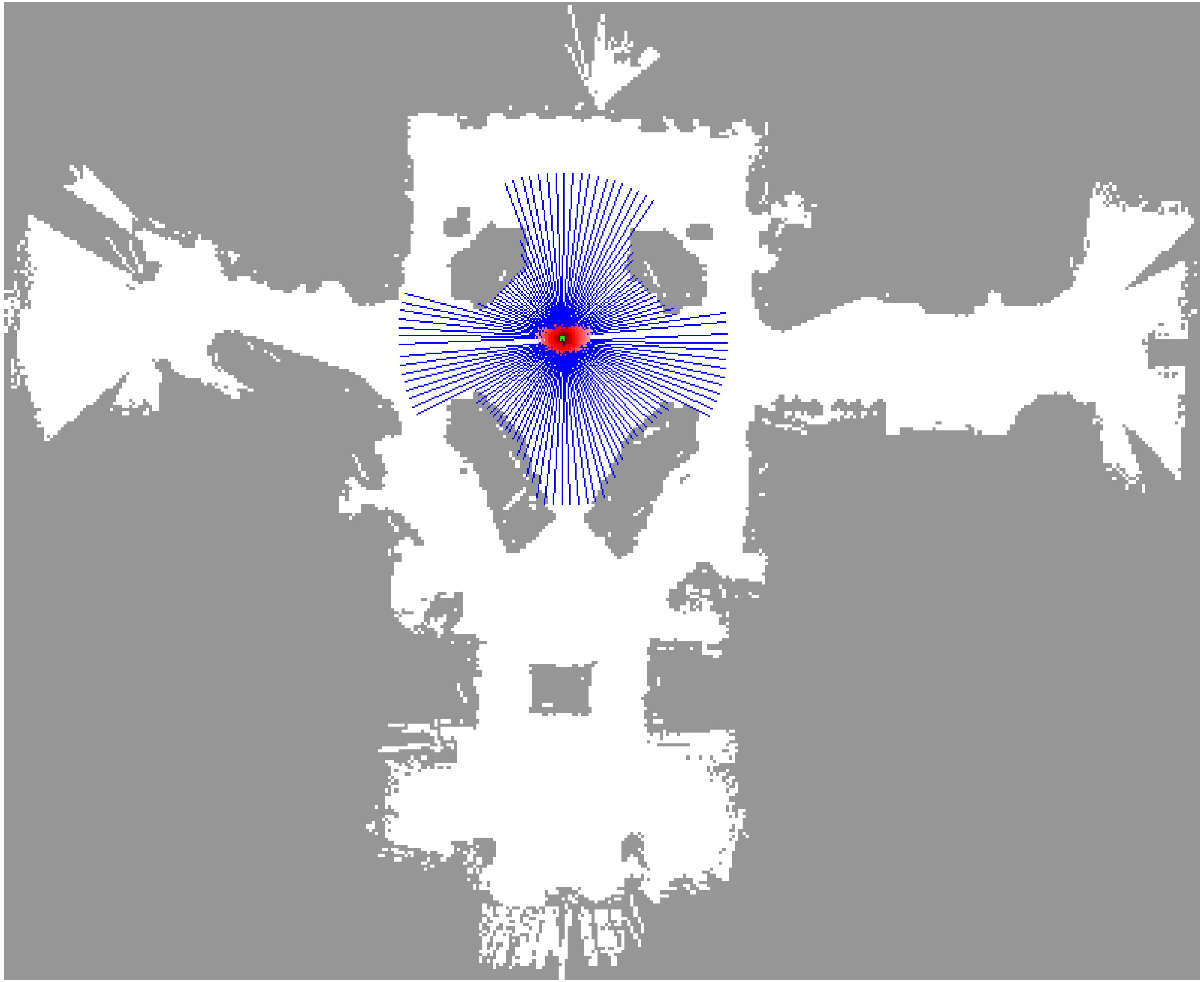




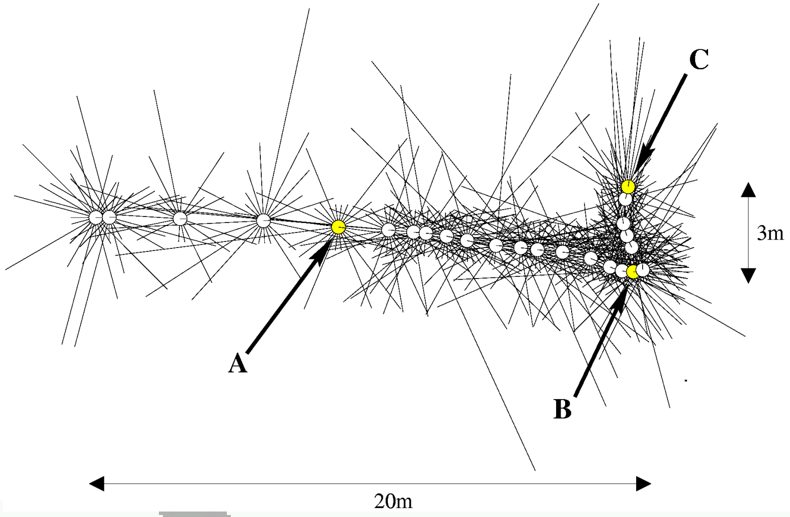


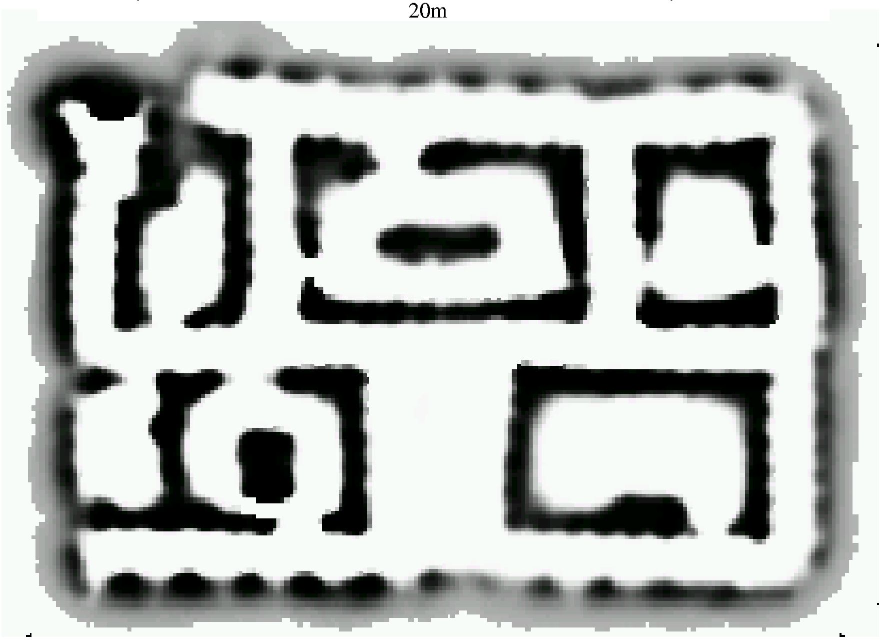


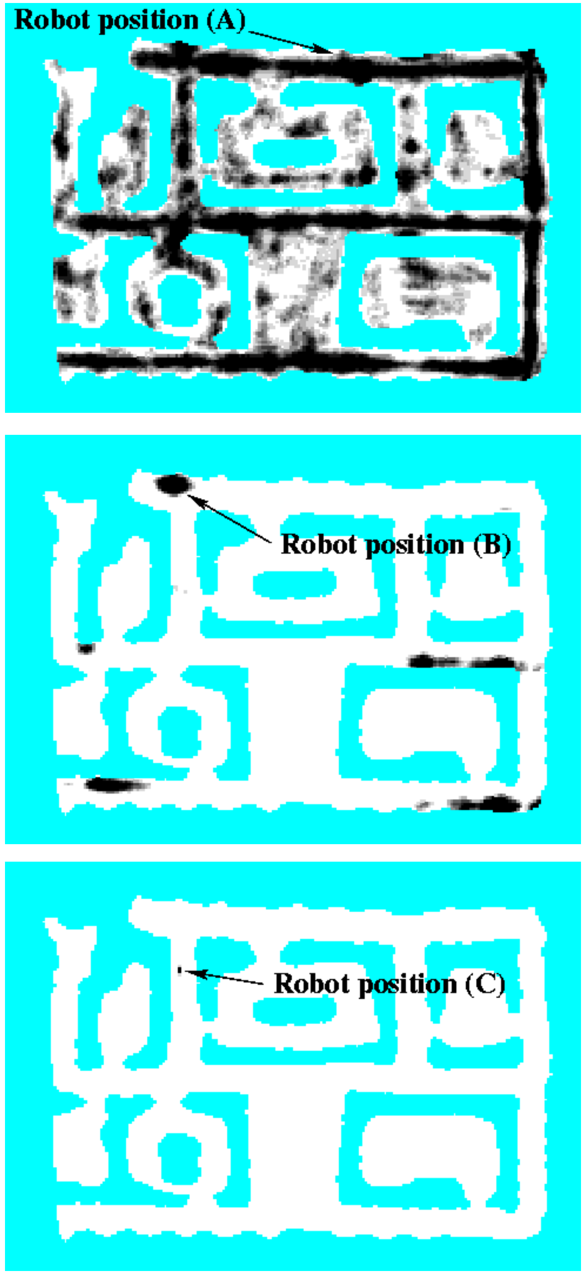




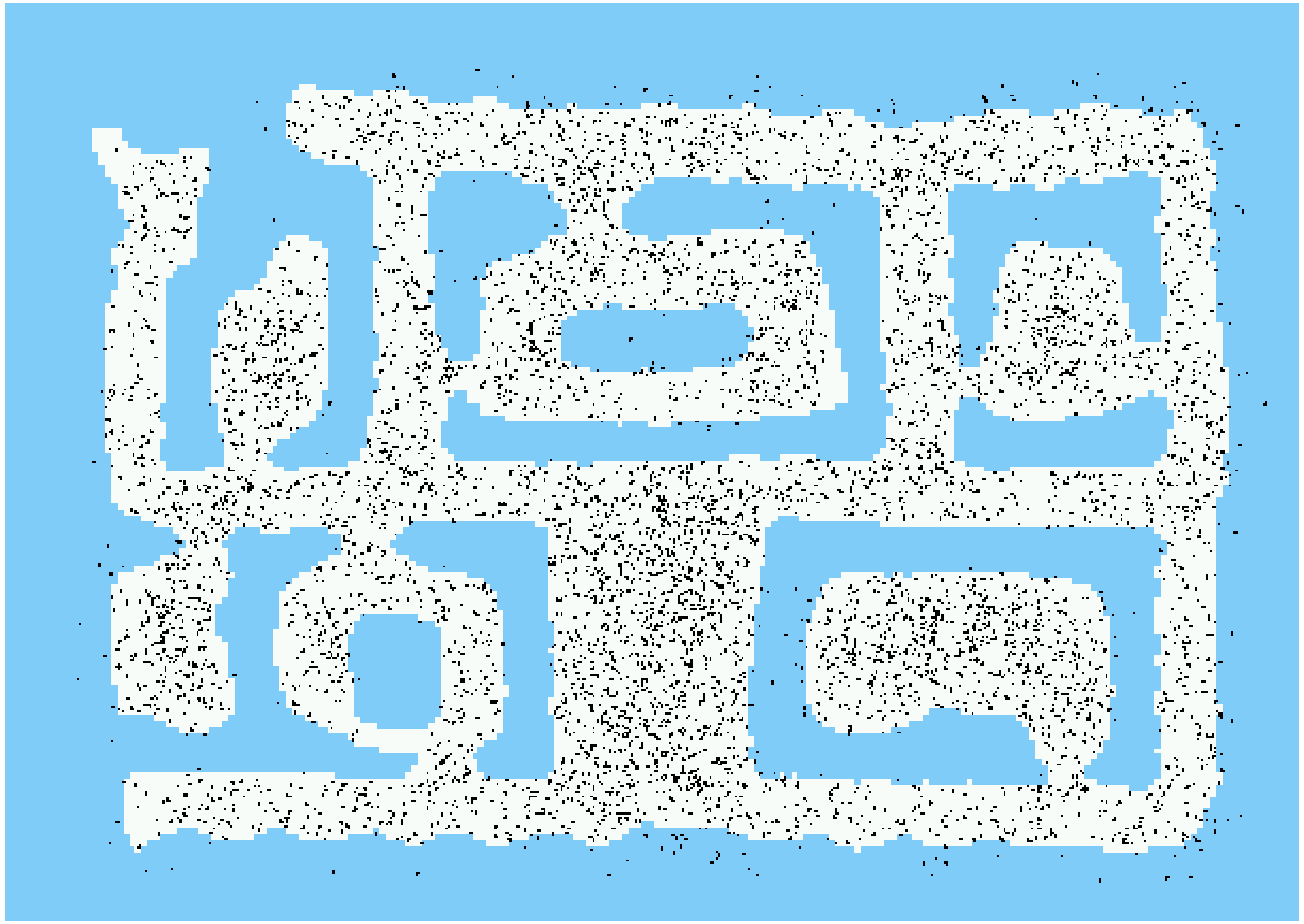
Discrete Filters Reminder (Ultrasound)



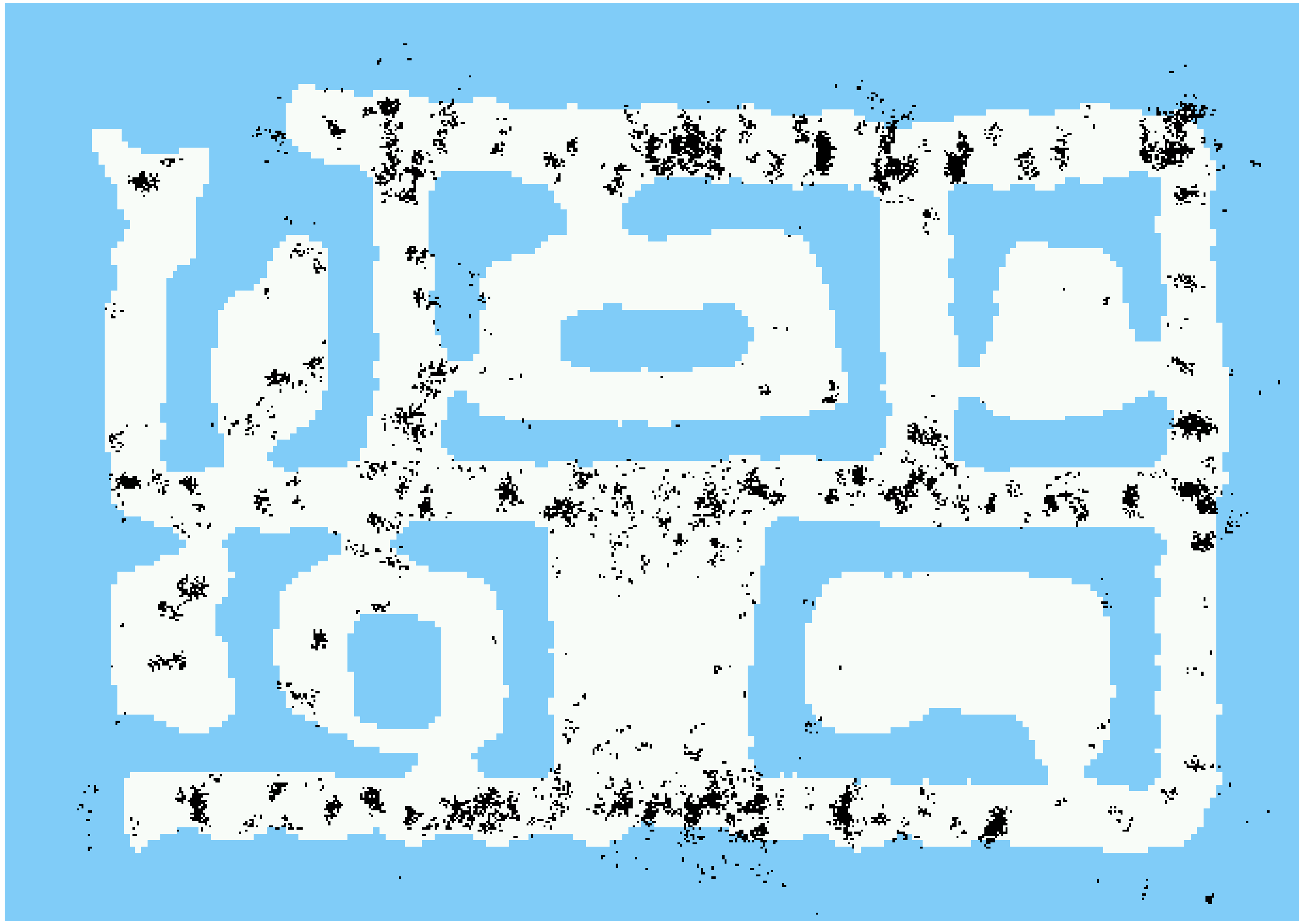




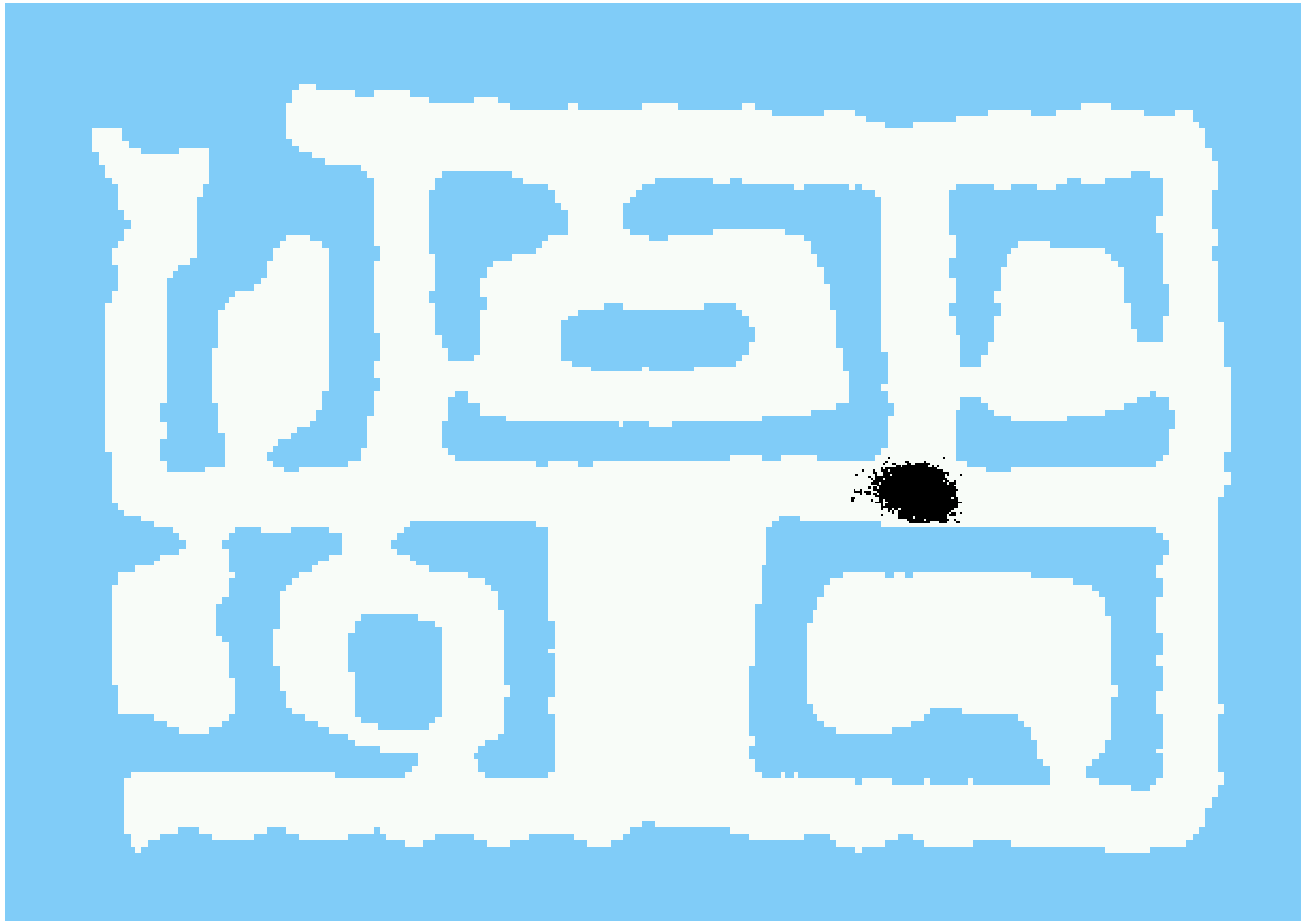
Initial Distribution



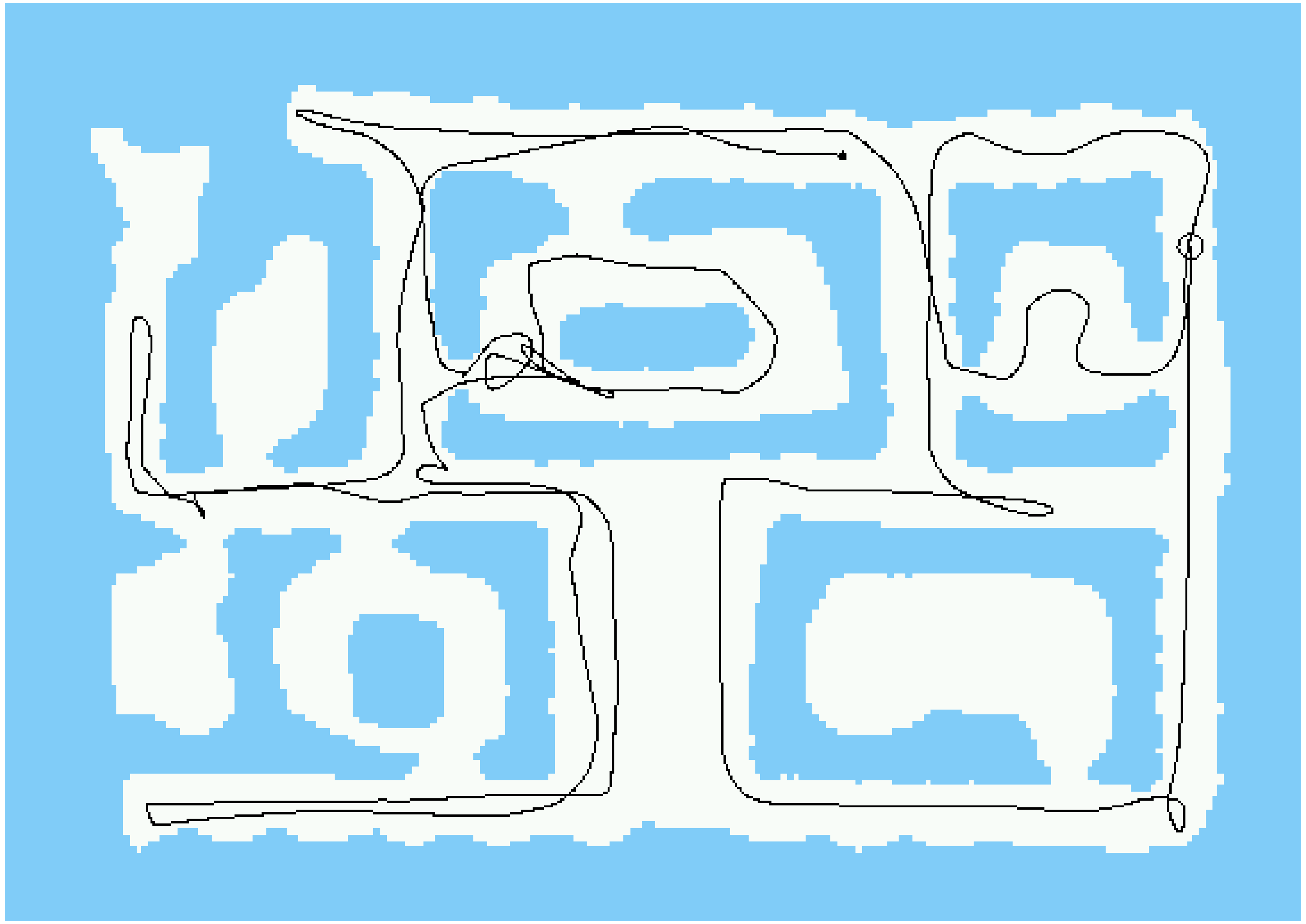
After Incorporating Ten Ultrasound Scans



After Incorporating 65 Ultrasound Scans

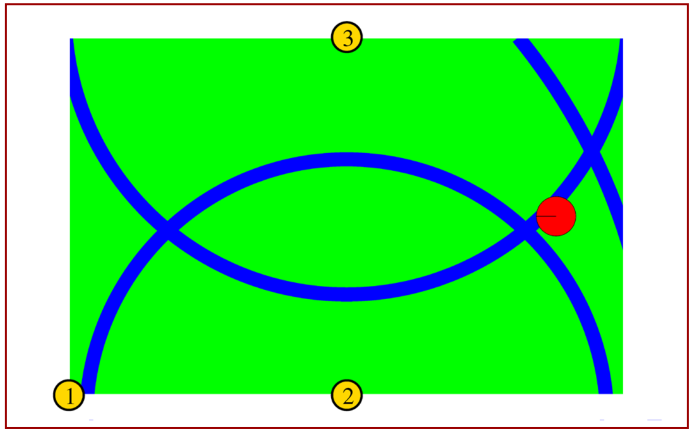


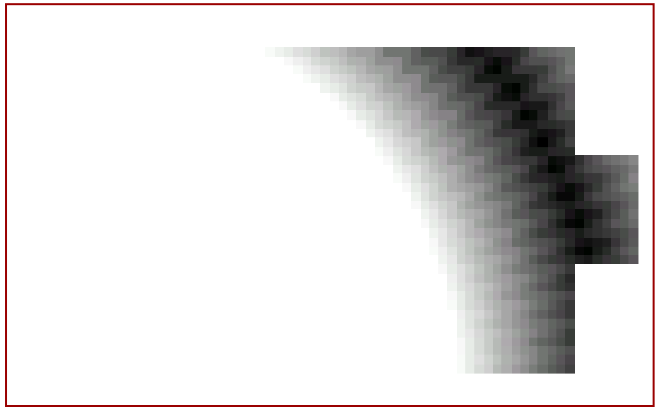
Estimated Path

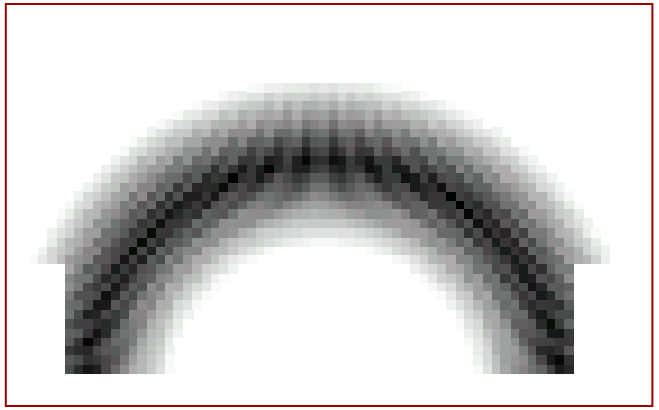


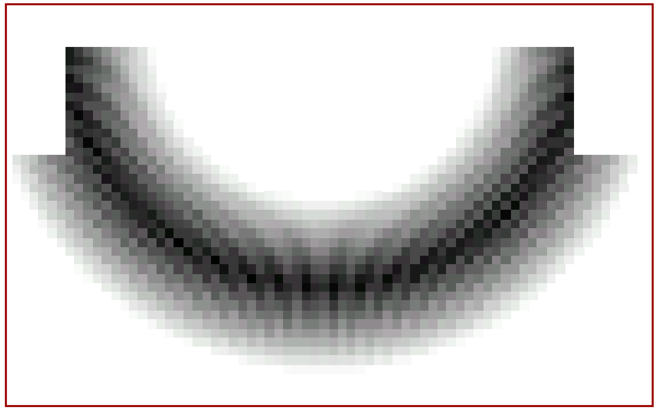
Importance Sampling with Resampling: Landmark Detection Example

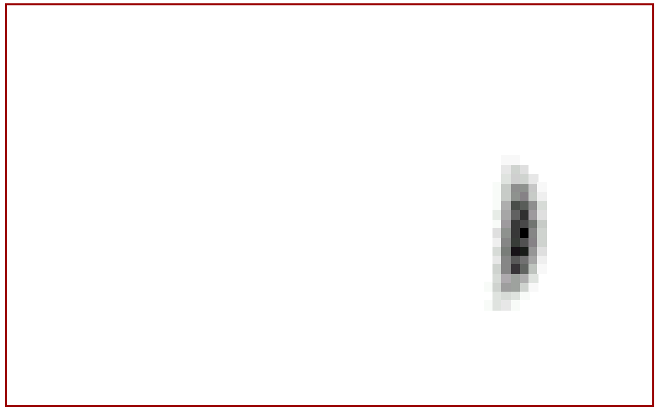
Distributions



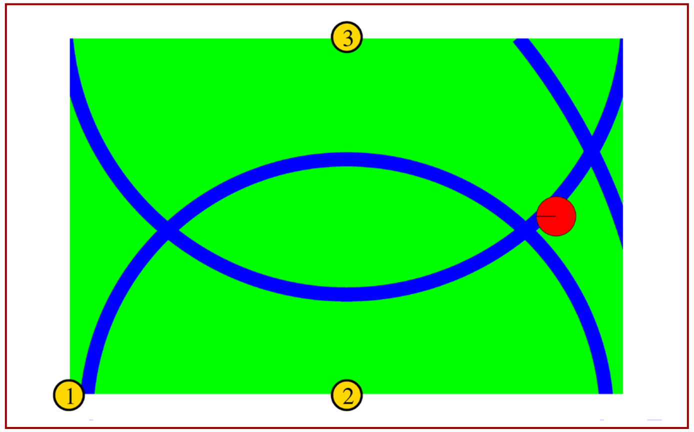


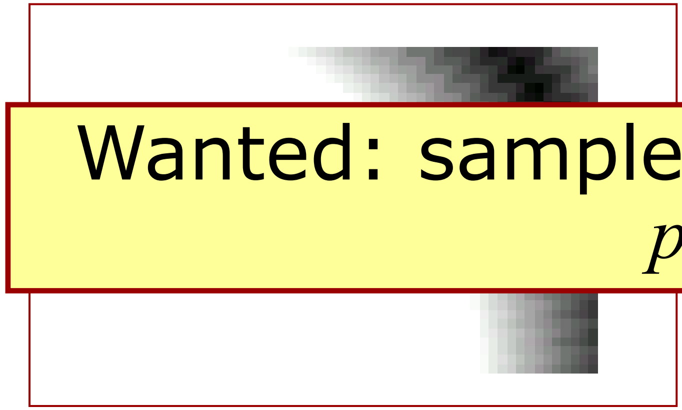


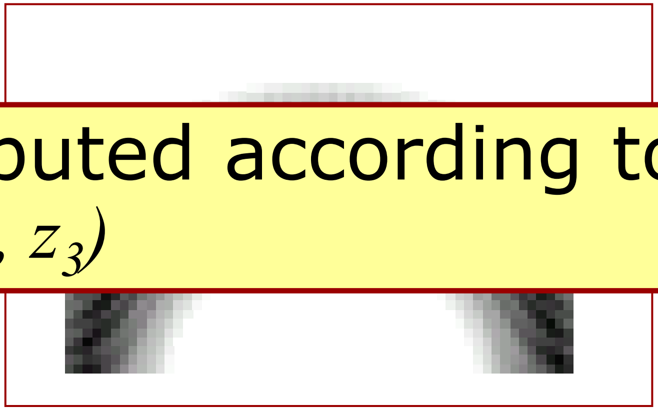




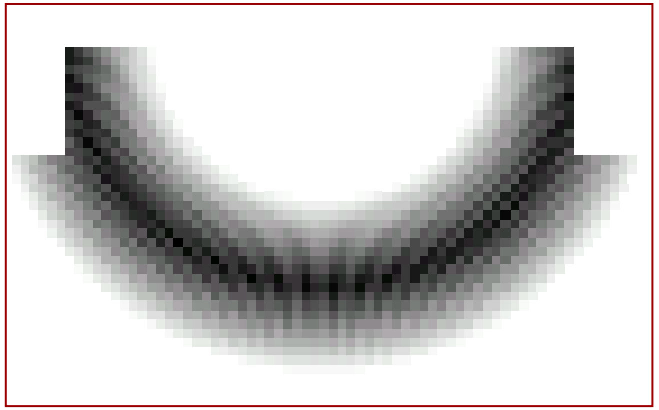
Distributions

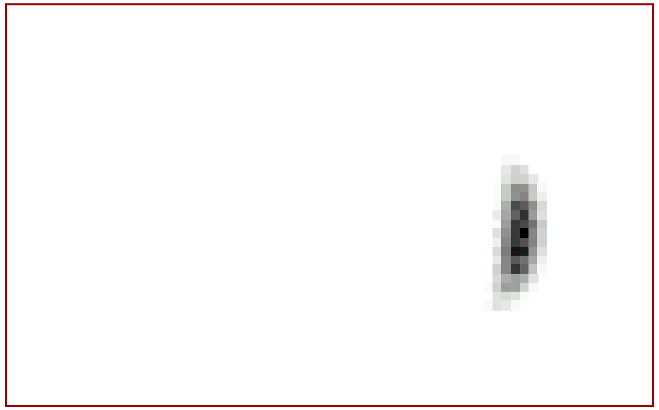






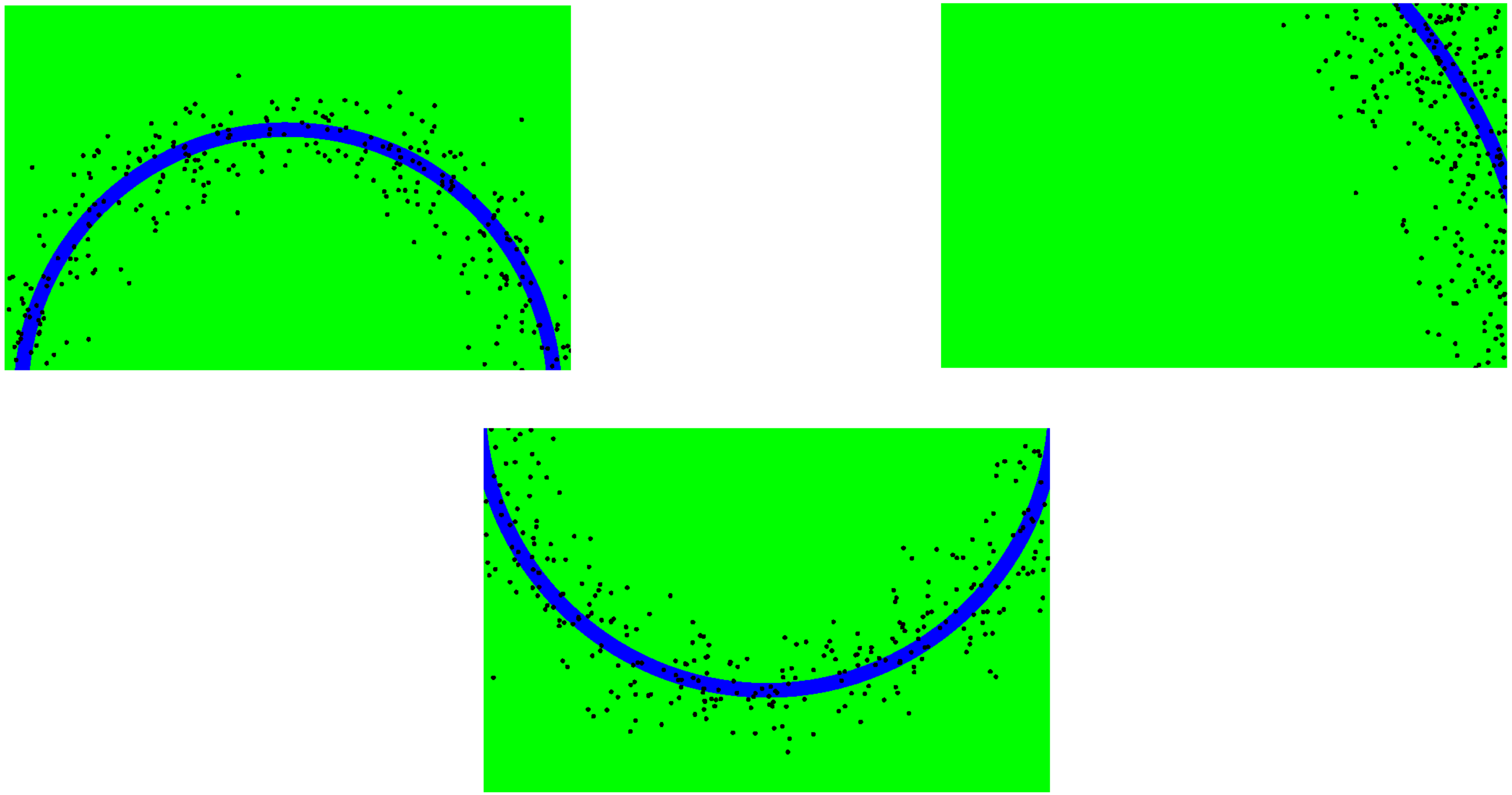
Wanted: samples distributed according to
$p(x \mid z_1, z_2, z_3)$



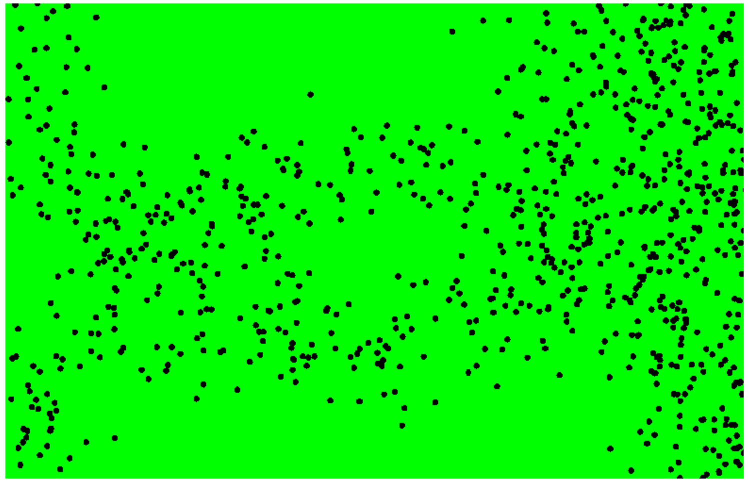


This is Easy!

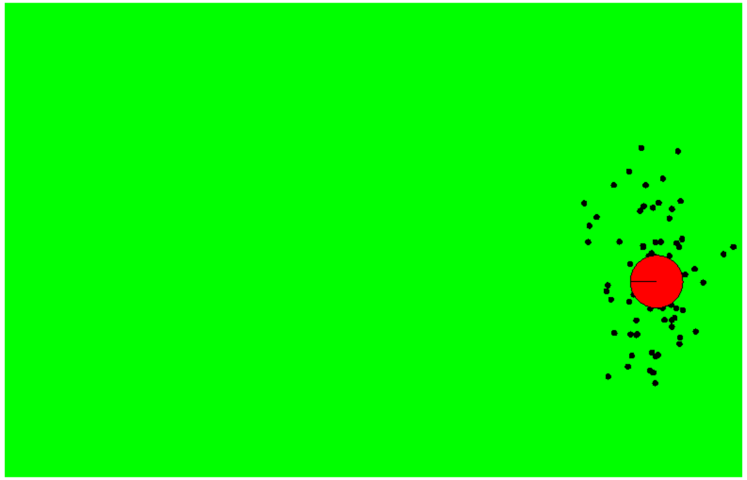
- We can draw samples from $p(x|z_i)$ by adding noise to the detection parameters.



Importance Sampling with Resampling



Weighted samples



After resampling

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 with randomly chosen poses
- This corresponds to the assumption that the robot can be teleported at any point in time to an arbitrary location
- Alternatively, insert such samples inverse proportional to the average likelihood of the observations (the lower this likelihood the higher the probability that the current estimate is wrong).

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 arbitrary and thus also non-Gaussian distributions
- Proposal to draw new samples
- Weights are computed to account for the difference between the proposal and the target
- Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter

Summary – Particle Filter Localization

- In the context of localization, the particles are propagated according to the motion model.
- They are then weighted according to the likelihood model (likelihood of the observations).
- In a re-sampling step, new particles are drawn with a probability proportional to the likelihood of the observation.
- This leads to one of the most popular approaches to mobile robot localization