ELEC 3210
Introduction to Mobile Robotics
Lecture 8

(Machine Learning and Infomation Processing for Robotics)

Huan YIN

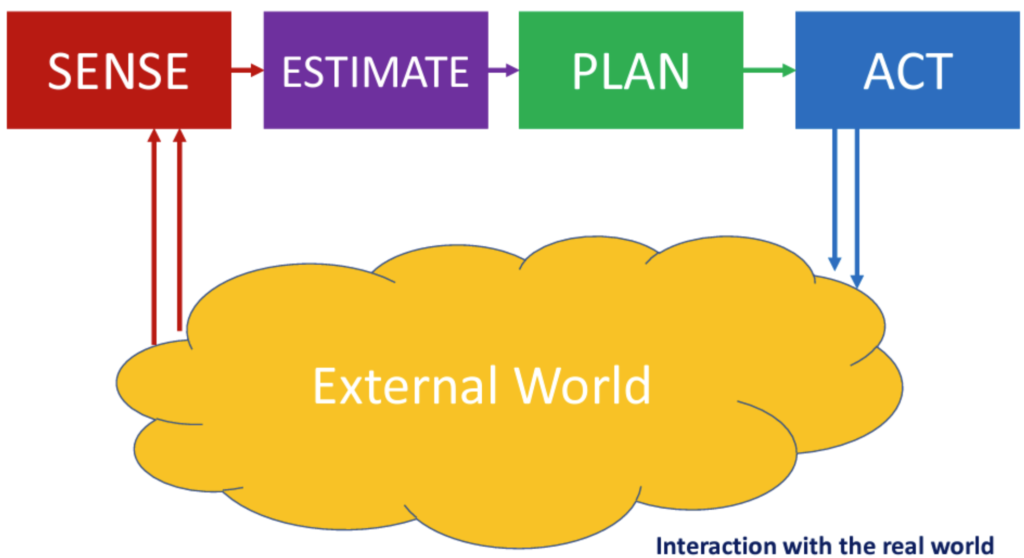
Research Assistant Professor, Dept. of ECE

eehyin@ust.hk

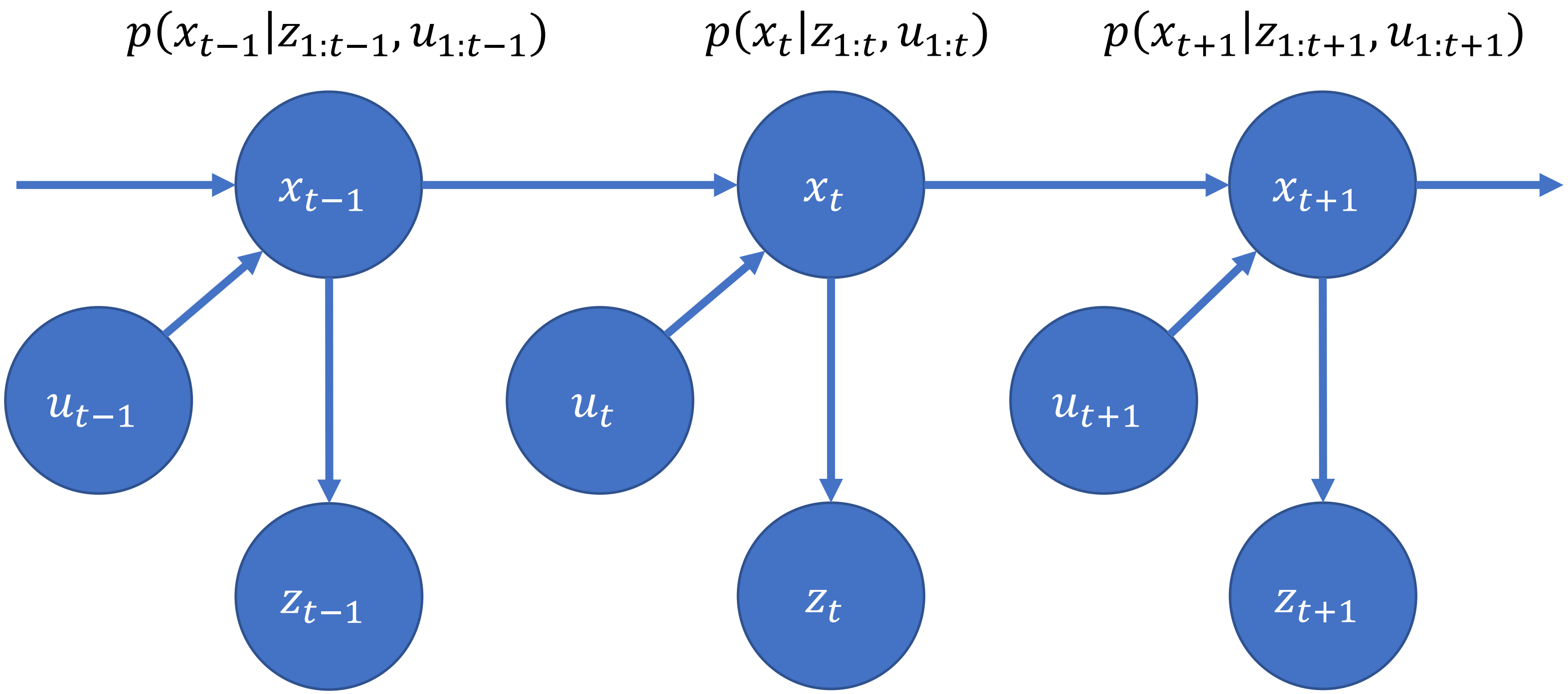
Course Design

- Part 1 - September
  - Navigation, Pose, Odometry, Sensors, ICP, Map etc.
- Part 2 - October
  - Bayes, Particle/Kalman Filter, EKF SLAM, Graph SLAM etc.
- Part 3 - November
  - Visual Perception, Motion Planning, Frontiers in Robotics etc.



Interaction with the real world introduces uncertainty!

Recap L7 - Bayes Filter



Recap L7 - Bayes Filter

- **Prior:** $p(x_0)$
- **Process model:** $p(x_t \mid x_{t-1}, u_t)$
- **Measurement model:** $p(z_t \mid x_t)$

- **Prediction step:**
- $p(x_t \mid z_{1:t-1}, u_{1:t}) = \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1})\, dx_{t-1}$
- **Update step:**
- $p(x_t \mid z_{1:t}, u_{1:t}) = \eta p(z_t \mid x_t) p(x_t \mid z_{1:t-1}, u_{1:t})$

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

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)

Kalman Filter and Particle Filter

- Bayes Filter Implementations

- Kalman Filter **(L9, L10 - EKF SLAM)**
  - Assumption with **Gaussian Distributions**
  - Purely matrix operations
- Particle Filter **(L8)**
  - **Non-Gaussian, arbitrary** distributions
  - Represent belief by random samples

Particle Filter

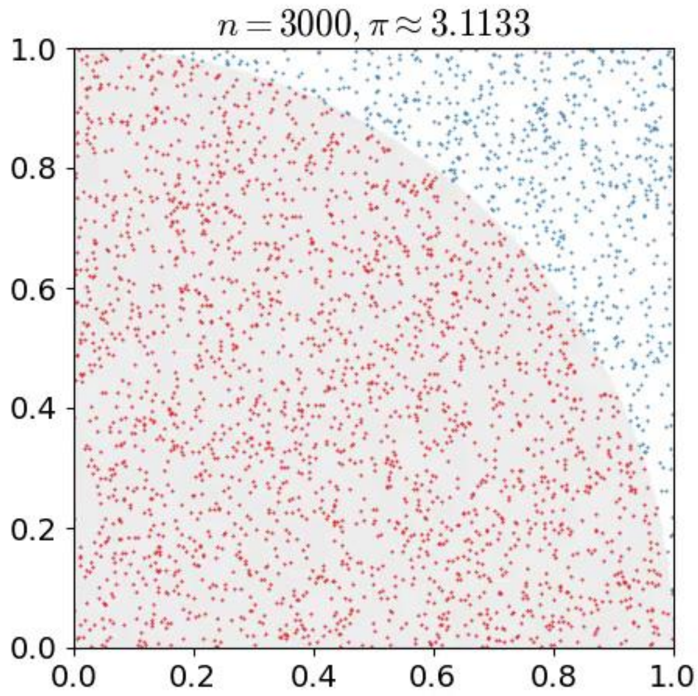
Particle Filter

- Particle Filter for Robot Localization = Monte Carlo Localization
- Monte Carlo
  - a gambling complex located in Monaco
- Monte Carlo Method
  - Estimating probability models of random variables through statistical experiments or random simulation methods

Monte Carlo

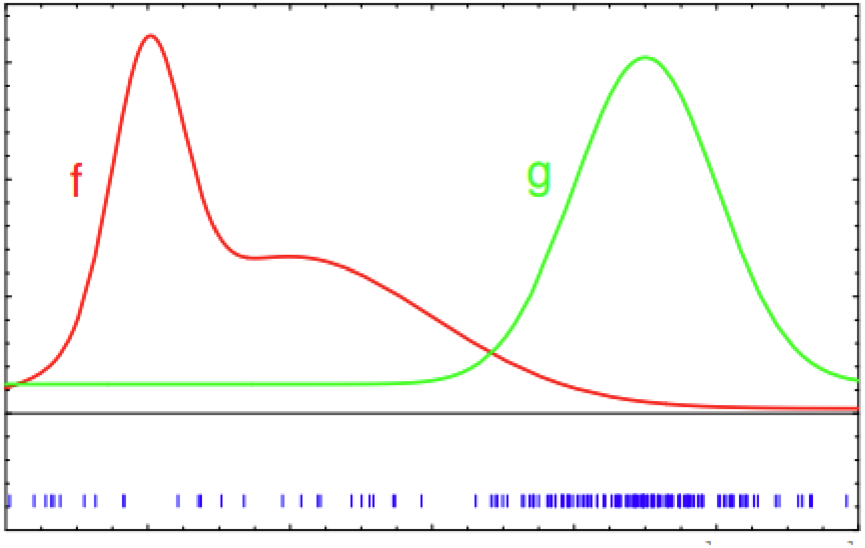
Monte Carlo Method

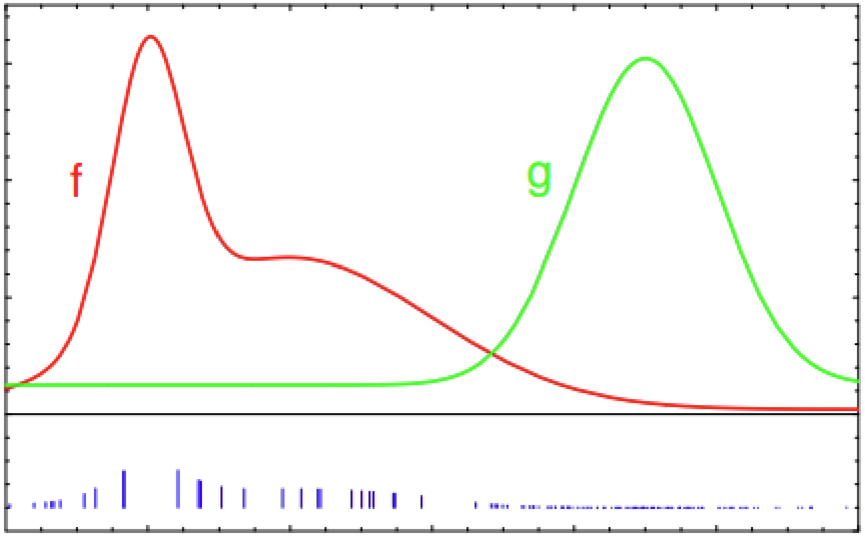
- The relationship between the area of a circle and its circumscribed square: Area ratio = $\pi/4$
- Scatter points across the area
- Count the number of points inside the circle and inside the circumscribed square through statistical experiments. As the points approaches infinity, point raio ≈ area ratio = $\pi/4$
- Get π



Importance Sampling Principle

- g: proposal distribution; f: taget distribution
- generate target from proposal with samples
- Account for the “differences between g and f” using a weight w

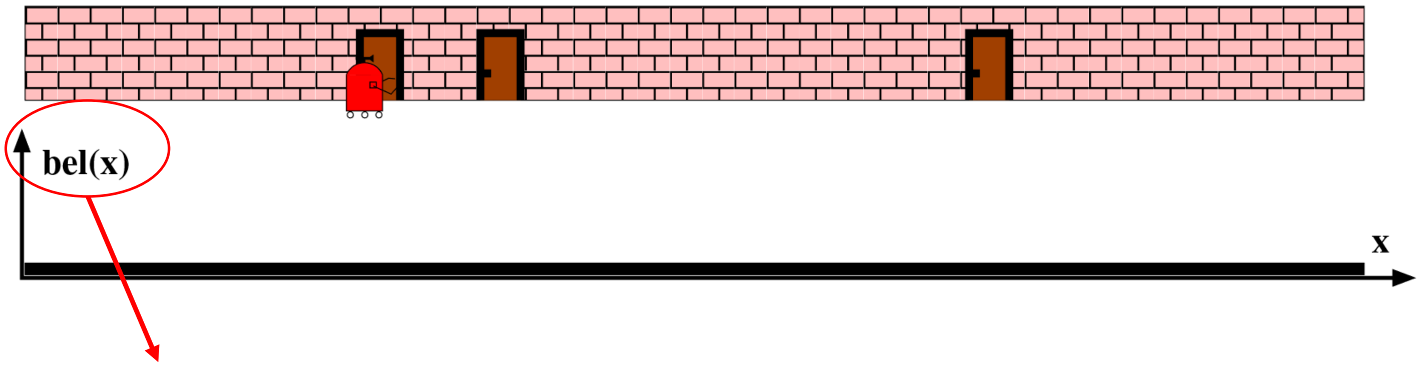




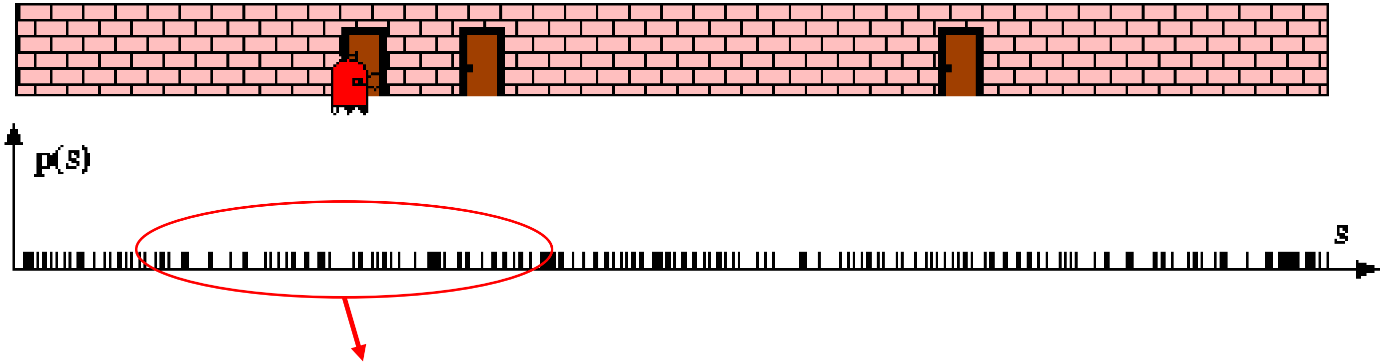
Weight samples: $w = f / g$

Particle Filter

- Represent belief by samples (particles)
- The more particles we use, the better is the estimate



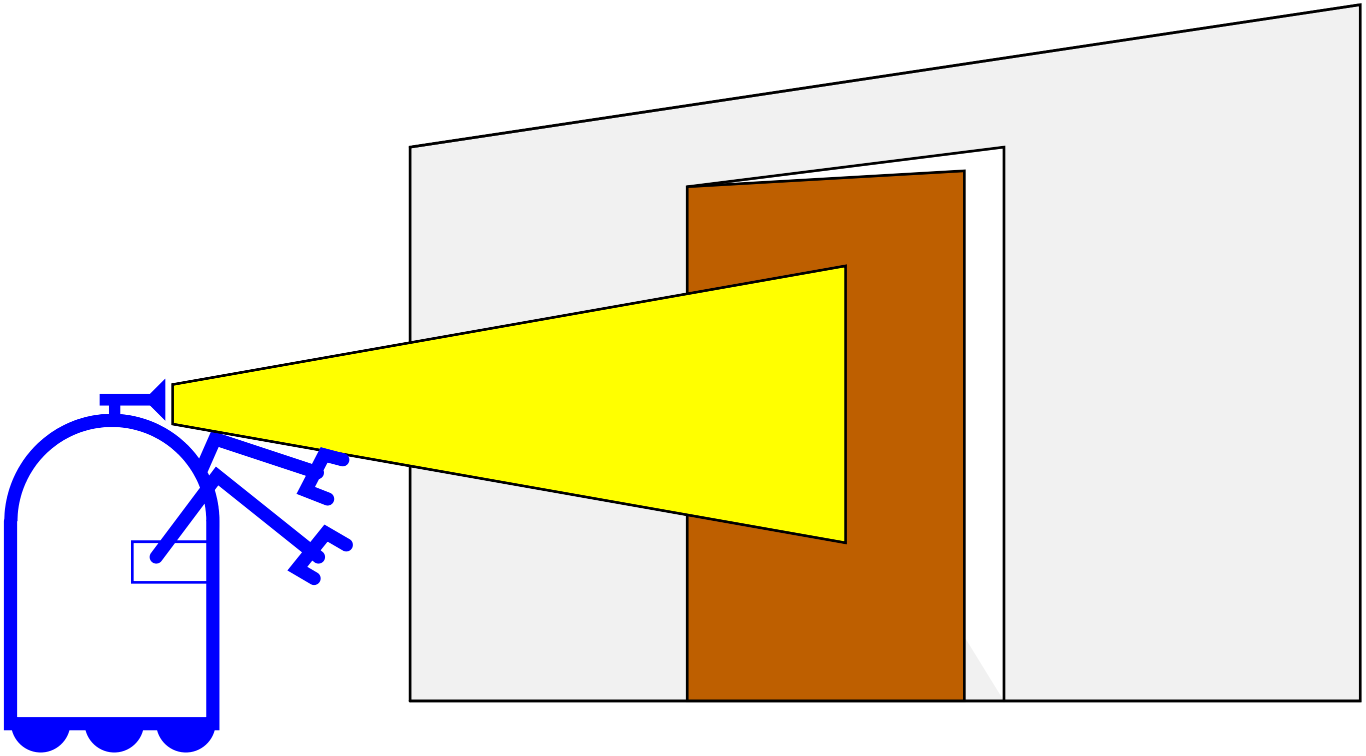
Use “belief” (or probability distribution) to represent robot state (pose)



Use particles to represent the belief

Measurement Model

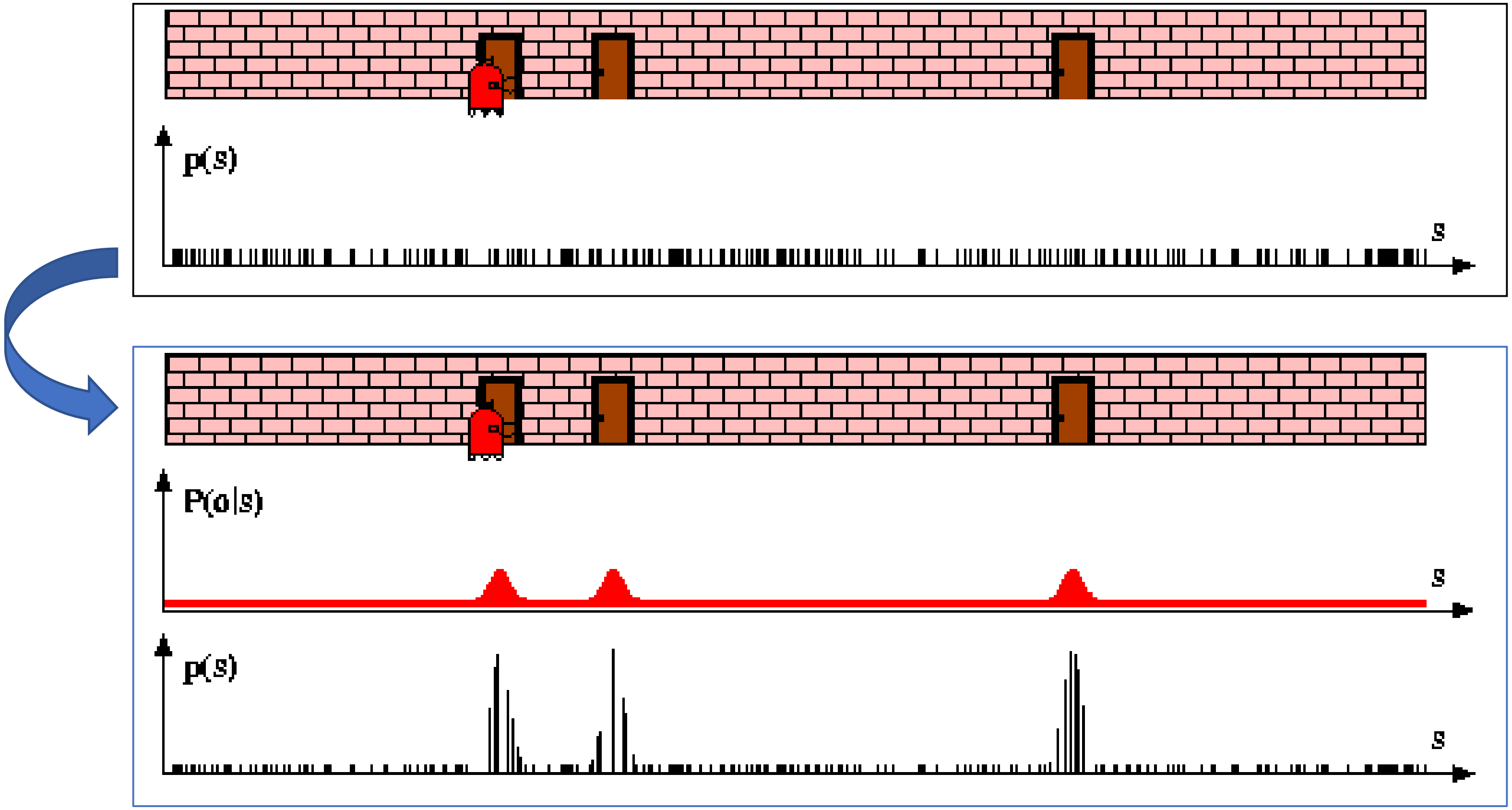
- $P(z|x)$
- Suppose the robot capture an image, and compare against the map



Measurement & Sampling

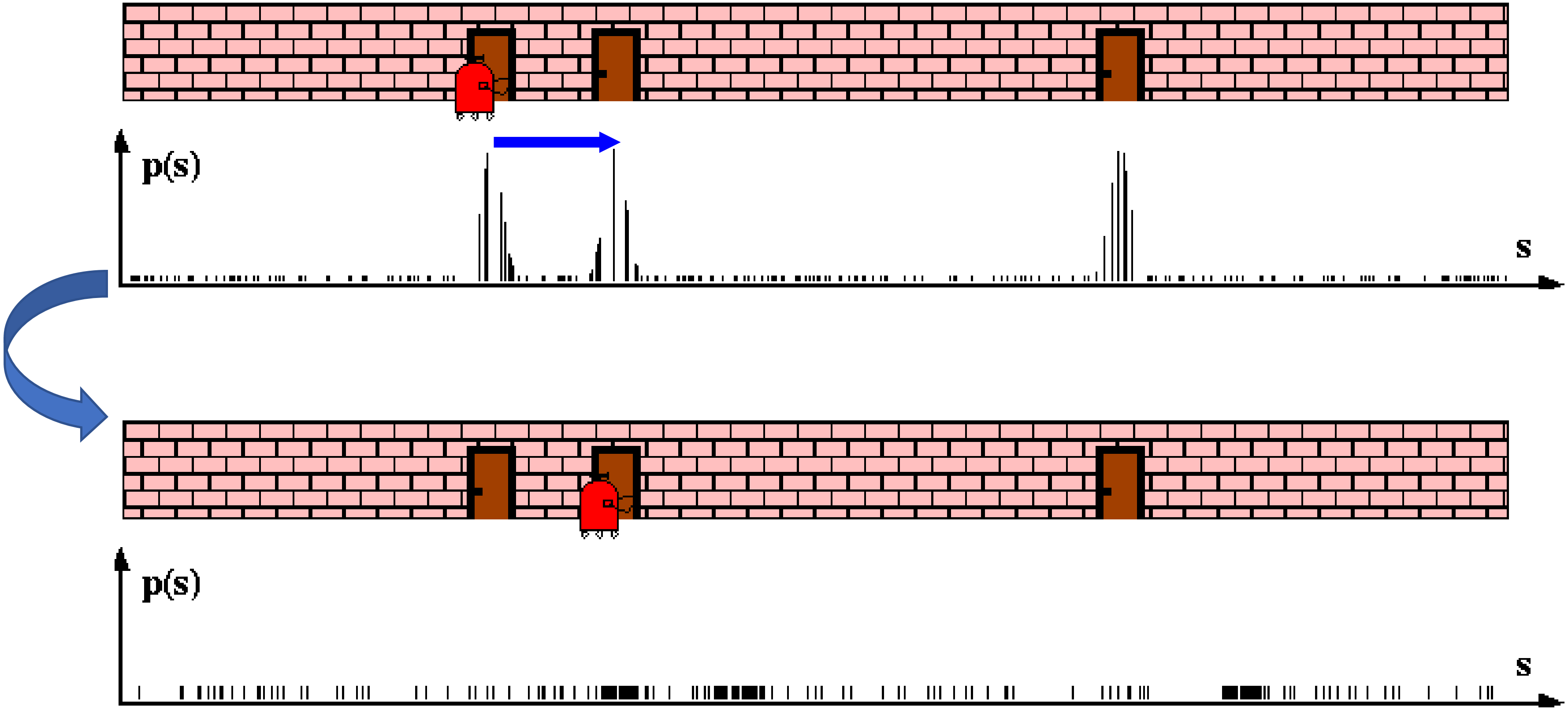
$$w_t \leftarrow \frac{\alpha\, p(z_t|x_t)\, Bel^-(x_t)}{Bel^-(x_t)} = \alpha\, p(z_t|x_t)$$

$$Bel(x_t) \leftarrow \alpha\, p(z_t|x_t)\, Bel^-(x_t)$$



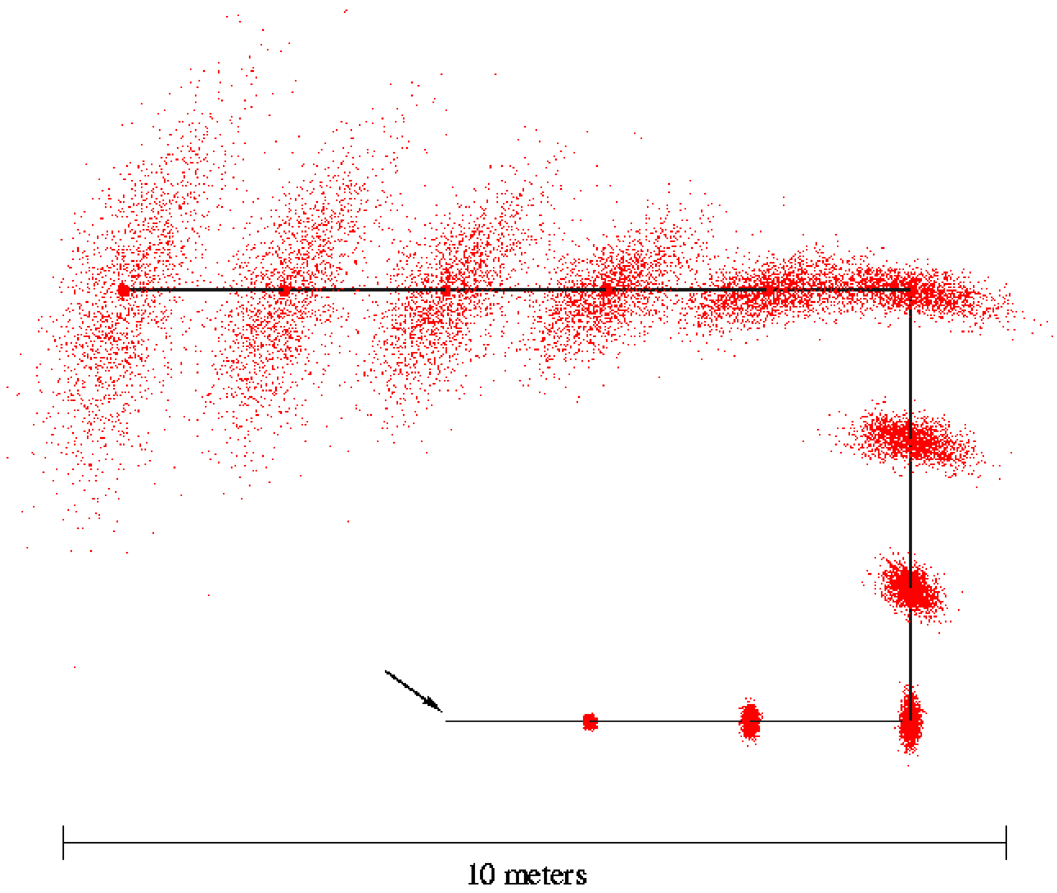
Resampling and Robot Motion

$$Bel^{-}(x_{t+1}) \leftarrow \int p\left(x_{t+1} \middle| x_{t,} u_{t+1}\right) Bel(x_t)\, \mathrm{d}x_t$$



Motion (Process) Model

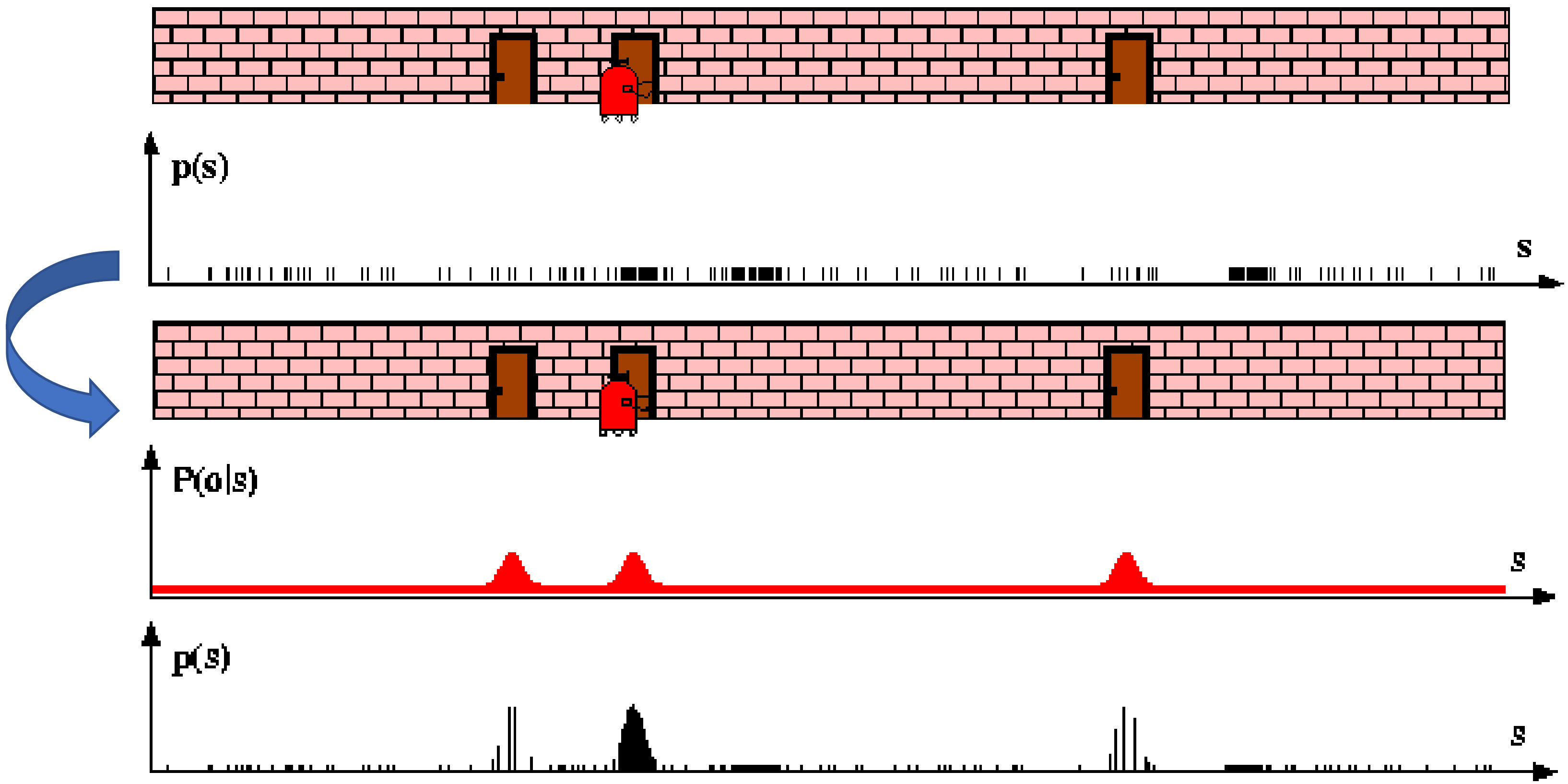
- Motion model using wheeled kinematics in Lecture 3
- Or with Iterative closest point in Lecture 5



Measurement & Sampling

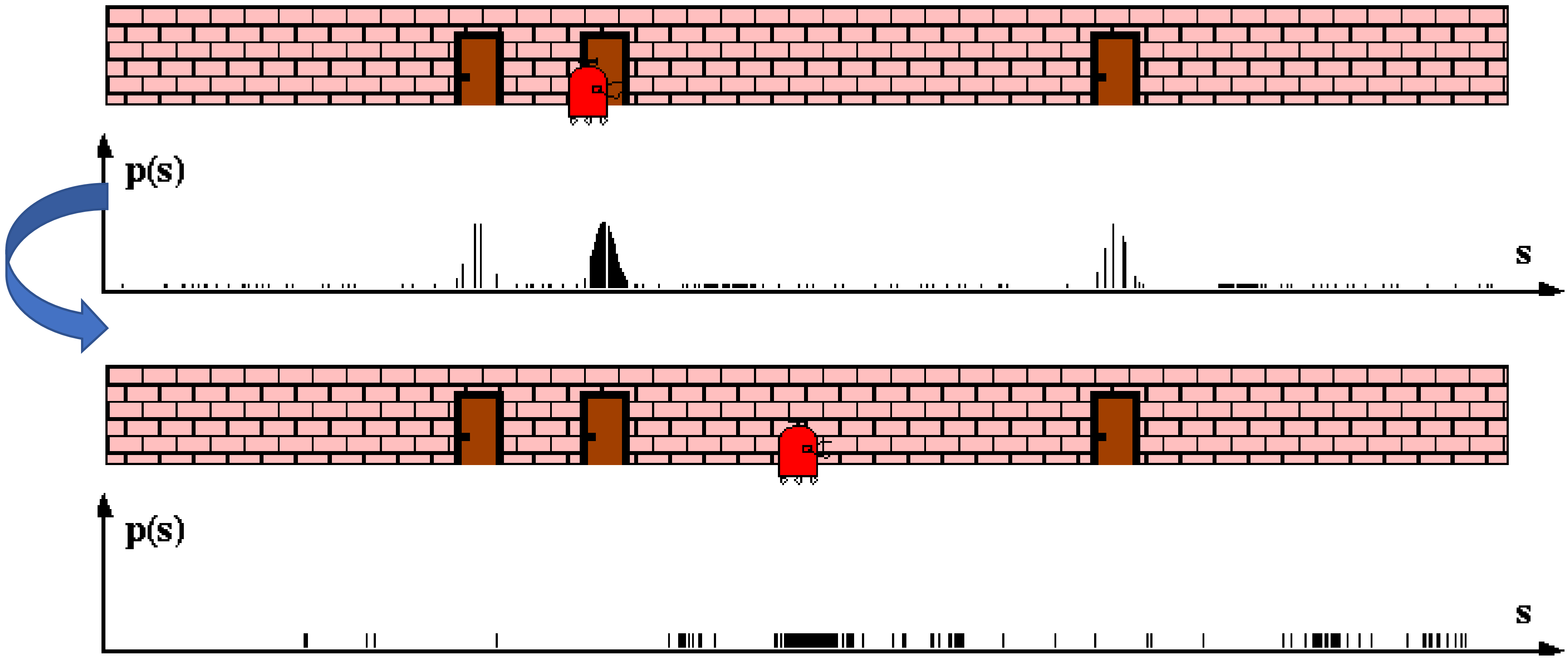
$$w_{t+1} \leftarrow \frac{\alpha\, p(z_{t+1}|x_{t+1})\, Bel^{-}(x_{t+1})}{Bel^{-}(x_{t+1})} = \alpha\, p(z_{t+1}|x_{t+1})$$

$$Bel(x_{t+1}) \leftarrow \alpha\, p(z_{t+1}|x_{t+1})\, Bel^{-}(x_{t+1})$$



Resampling and Robot Motion

$$Bel^{-}(x_{t+2}) \leftarrow \int p(x_{t+2} | x_{t+1}, u_{t+1}) Bel(x_{t+1})\, \mathrm{d}x_{t+1}$$



Particle Filter

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

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:

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

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

Particle Filter

1. Algorithm **particle_filter**(S_{t-1}, u_{t-1} z_t):
2. $S_t = \varnothing, \quad \eta = 0$
3. **For** $i = 1 \ldots n$ ***Generate new samples***
4. Sample index *j(i)* from the discrete distribution given by w_{t-1}
5. Sample x_t^i from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x_{t-1}^{j(i)}$ and u_{t-1}
6. $w_t^i = p(z_t \mid x_t^i)$ ***Compute importance weight***
7. $\eta = \eta + w_t^i$ ***Update normalization factor***
8. $S_t = S_t \cup \{< x_t^i, w_t^i >\}$ ***Insert***
9. **For** $i = 1 \ldots n$
10. $w_t^i = w_t^i / \eta$ ***Normalize weights***

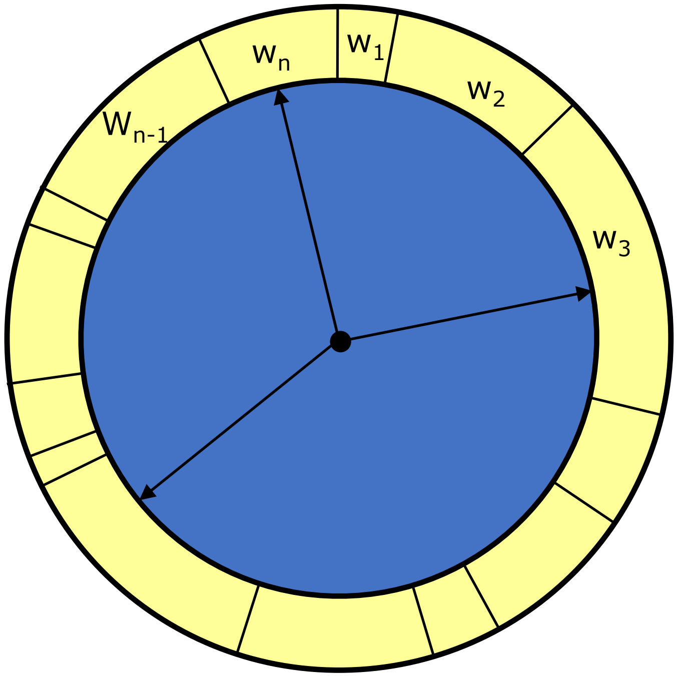
Particle Filter

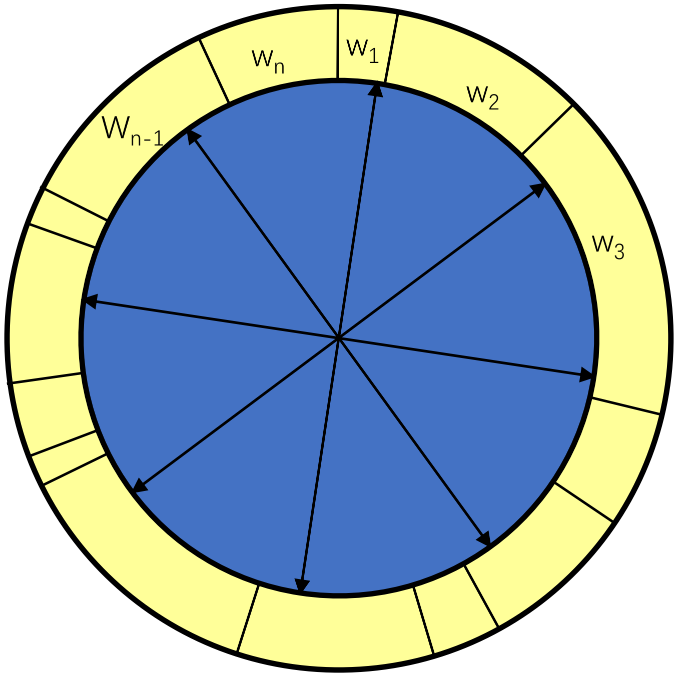
1. Algorithm **particle_filter**(S_{t-1}, u_{t-1} z_t):
2. $S_t = \emptyset, \quad \eta = 0$
3. **For** $i = 1 \ldots n$ — ***Generate new samples*** **HOW?**
4. Sample index *j(i)* from the discrete distribution given by w_{t-1}
5. Sample x_t^i from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x_{t-1}^{j(i)}$ and u_{t-1}
6. $w_t^i = p(z_t \mid x_t^i)$ — ***Compute importance weight***
7. $\eta = \eta + w_t^i$ — ***Update normalization factor***
8. $S_t = S_t \cup \{< x_t^i, w_t^i >\}$ — ***Insert***
9. **For** $i = 1 \ldots n$
10. $w_t^i = w_t^i / \eta$ — ***Normalize weights***

Resampling

- **Given**: Set S of weighted samples
- **Wanted** : Random sample, where the probability of drawing x_t^i is given by w_t^i
- Infomally “Replace unlikely samples by more likely ones”
- Survival of the fittest

Resampling



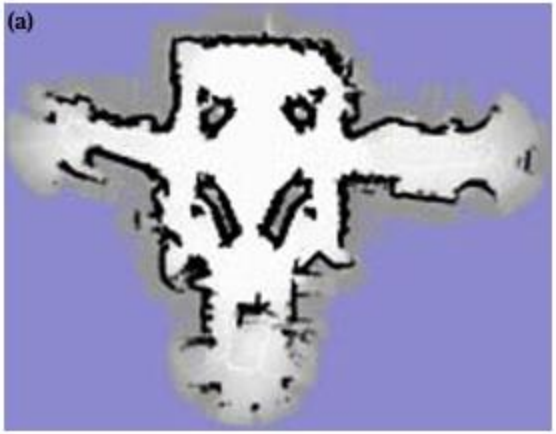


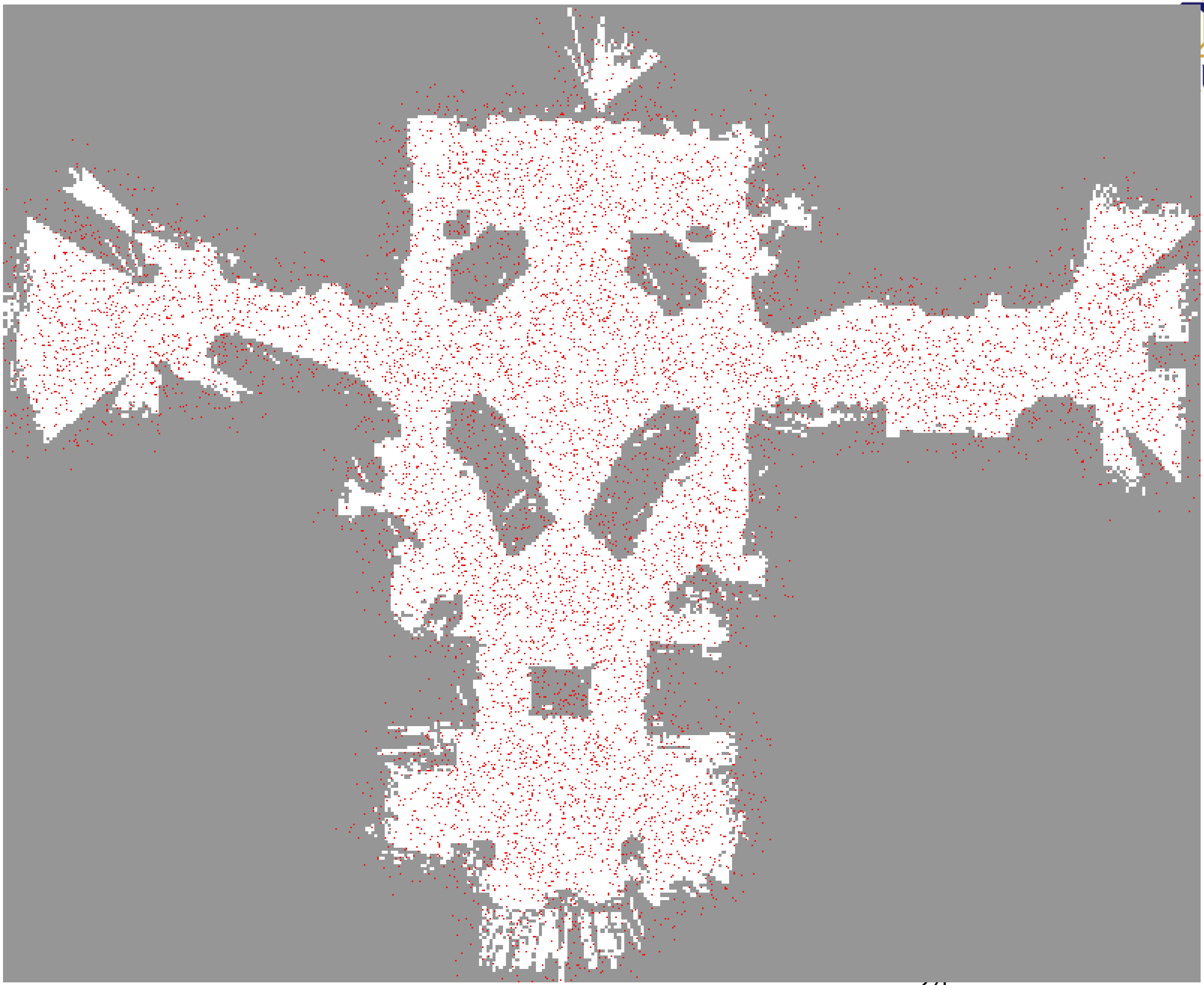
- Roulette Wheel
- Binary Search
- O (n logn)

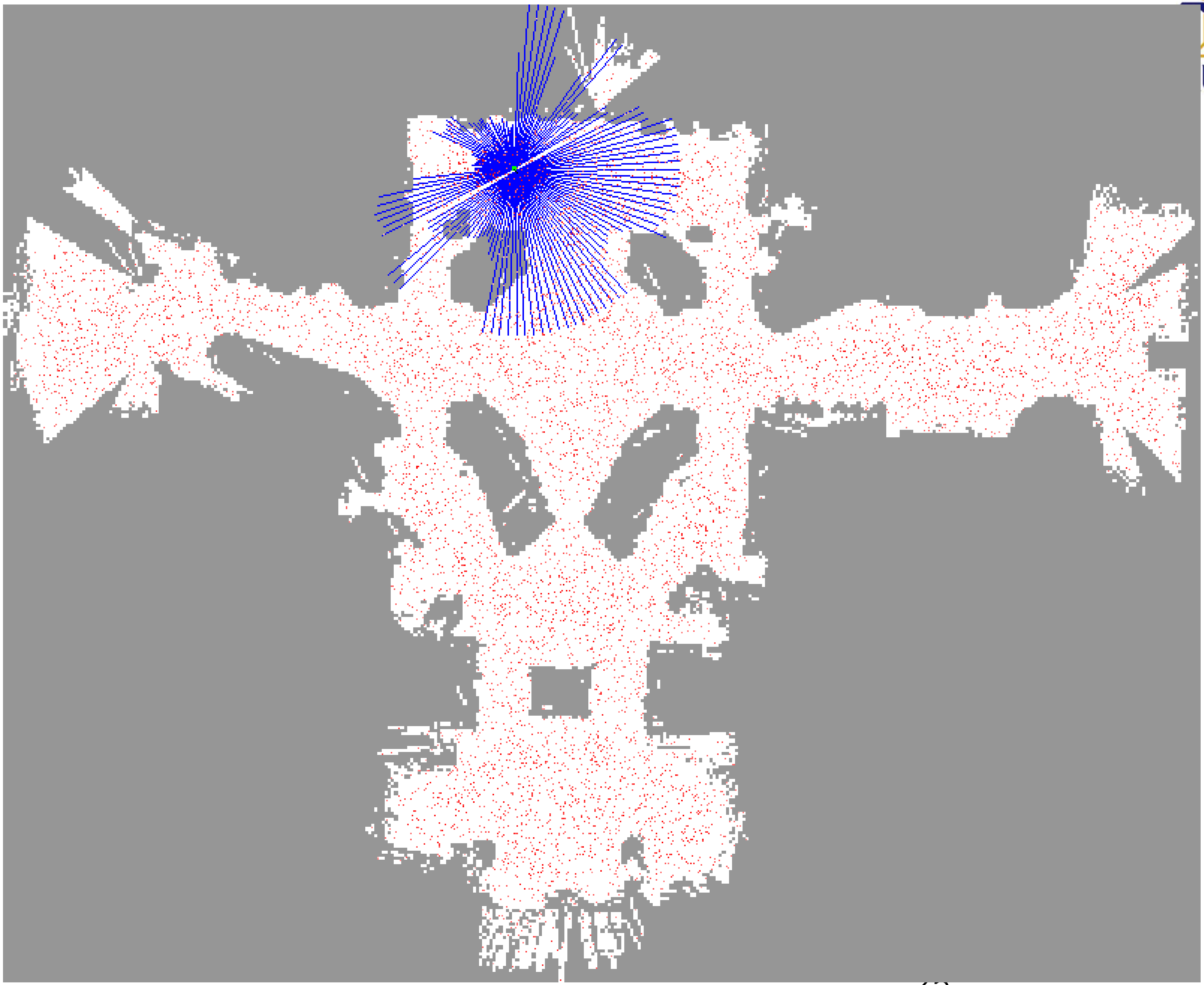
- Stochastic universal sampling
- Systematic resampling
- Linear time complexity

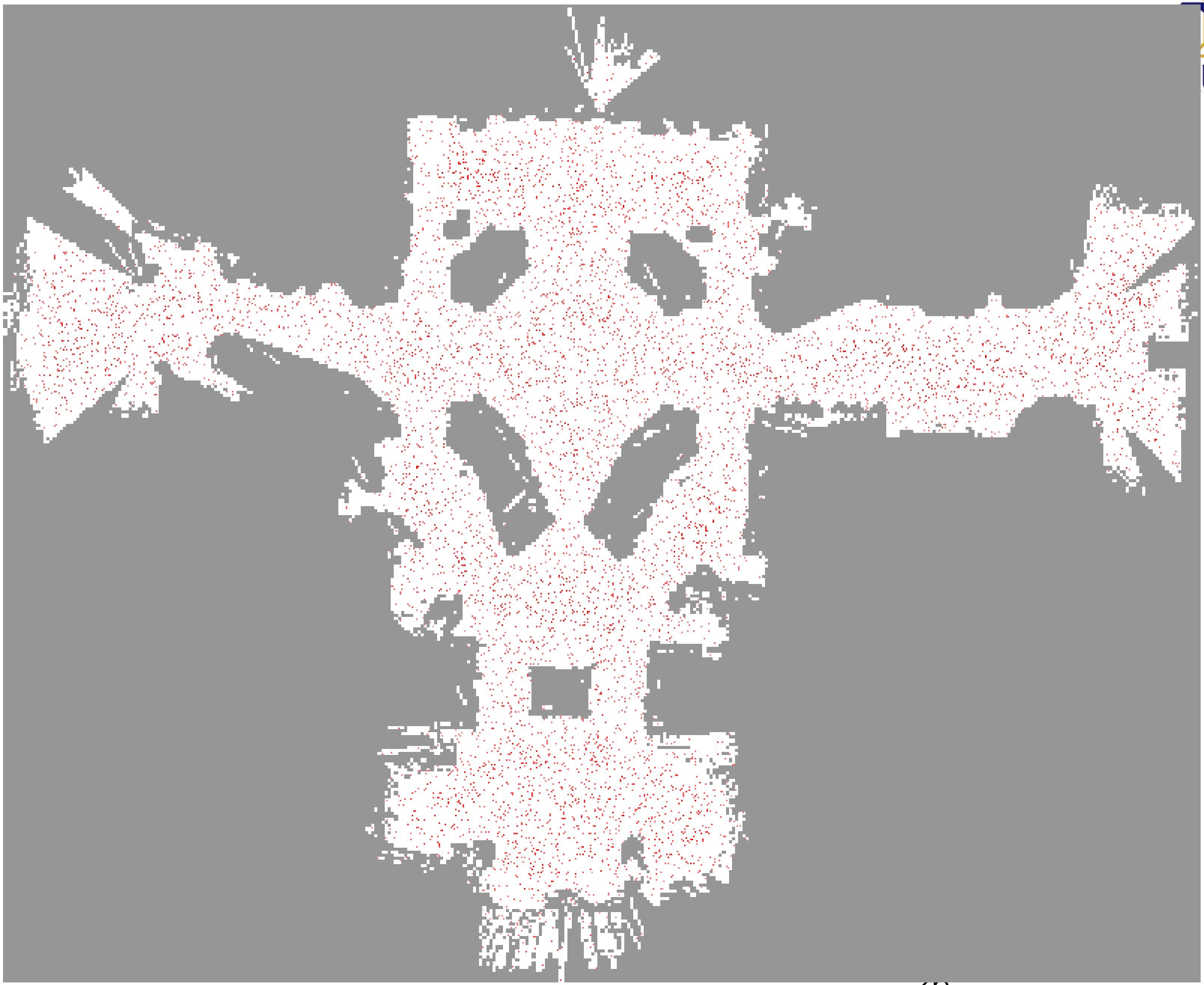
Example

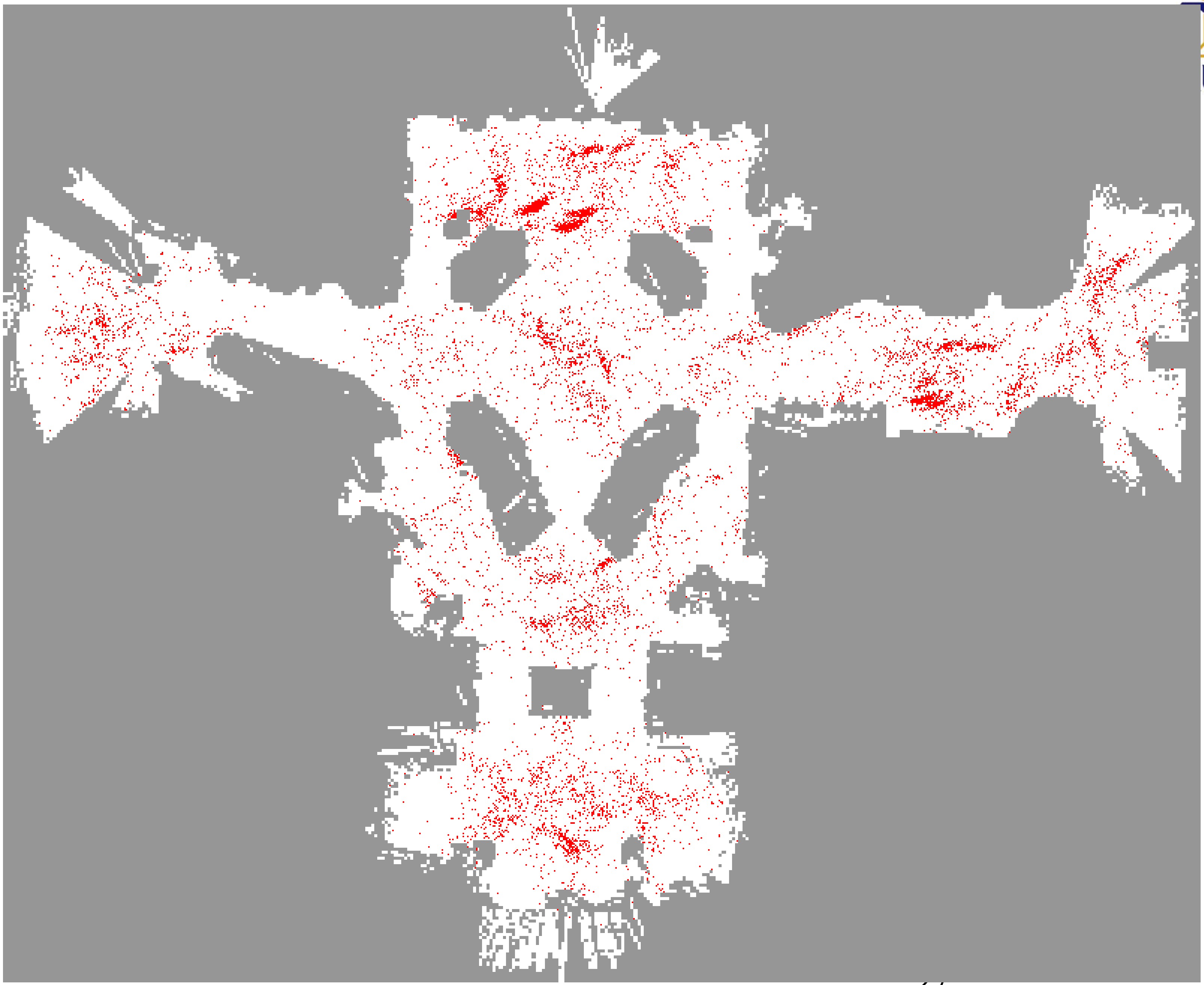
- Mobile tour guide robot in Museum, ~1998
- Monte Carlo Localization on a known map
  - Lecture 3 - Map-based localization
  - Lecture 9 - Global grid map

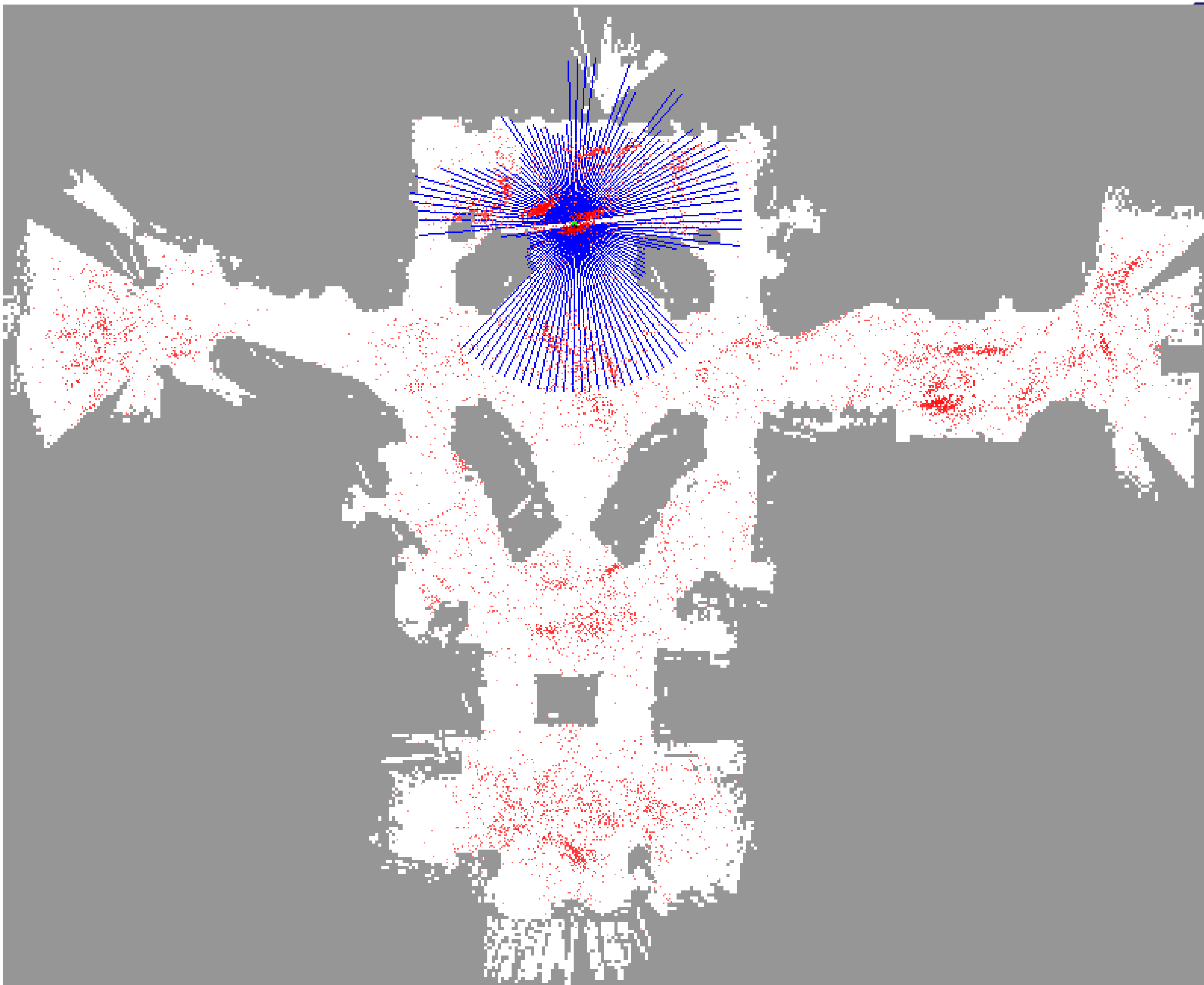


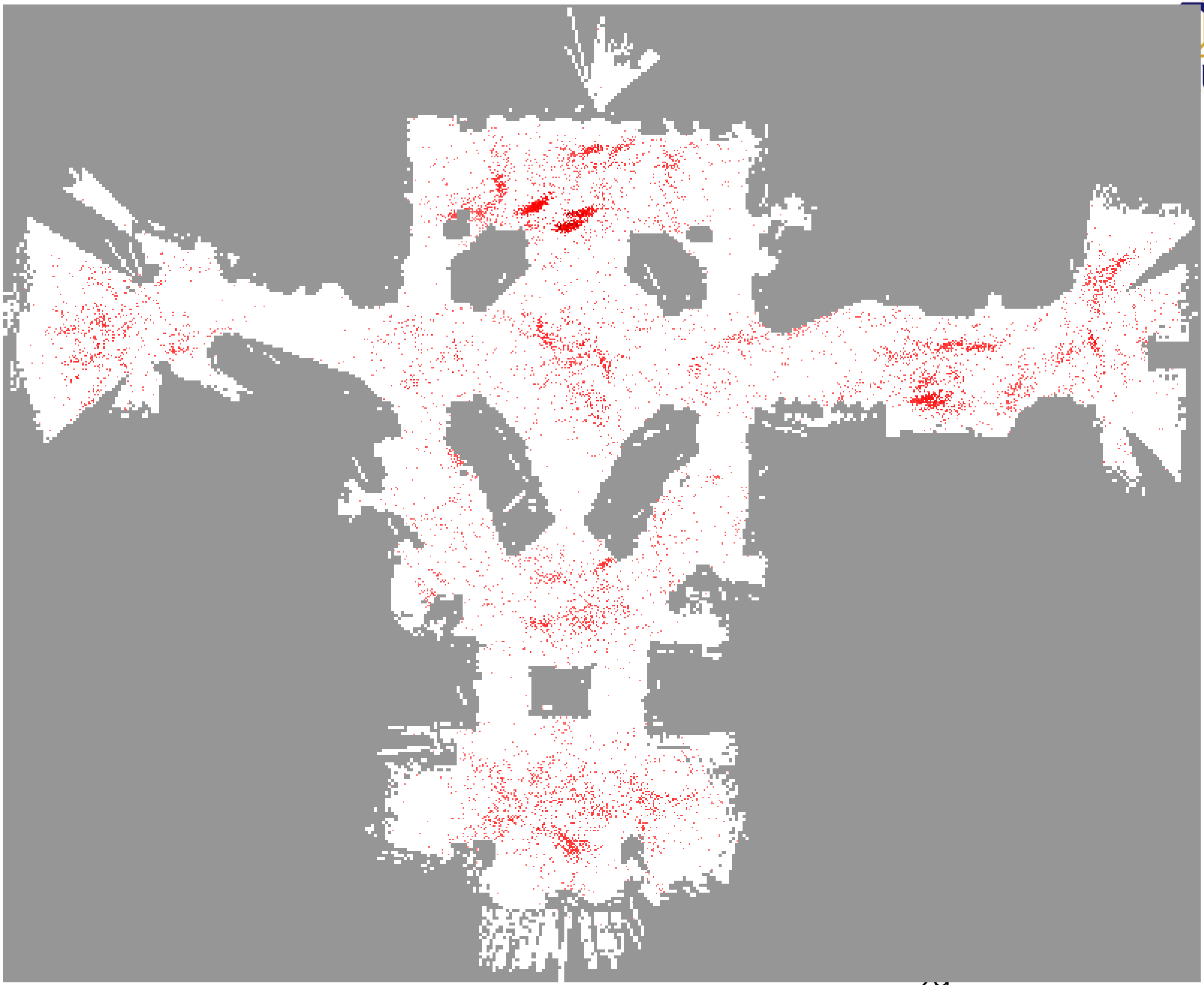


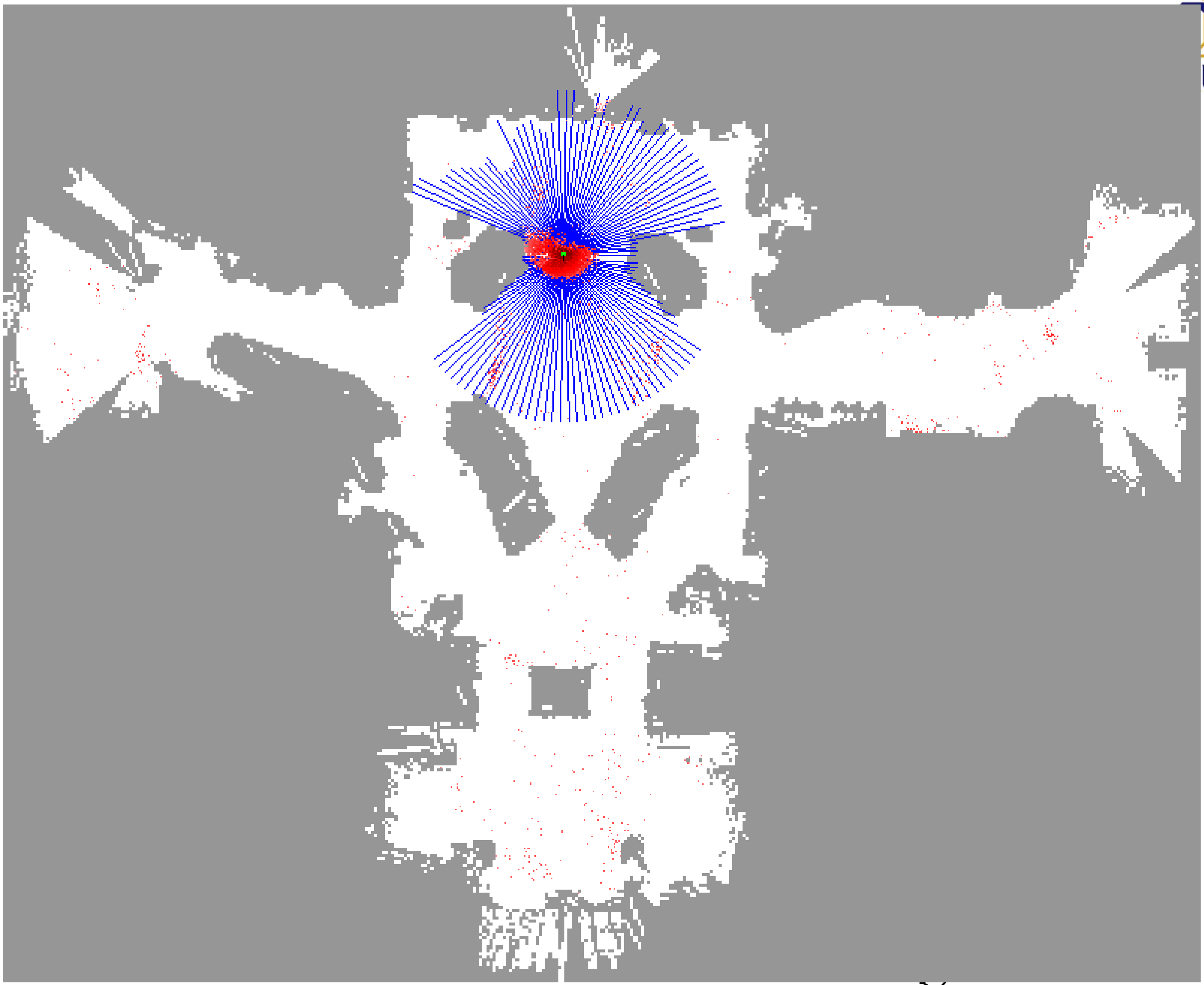


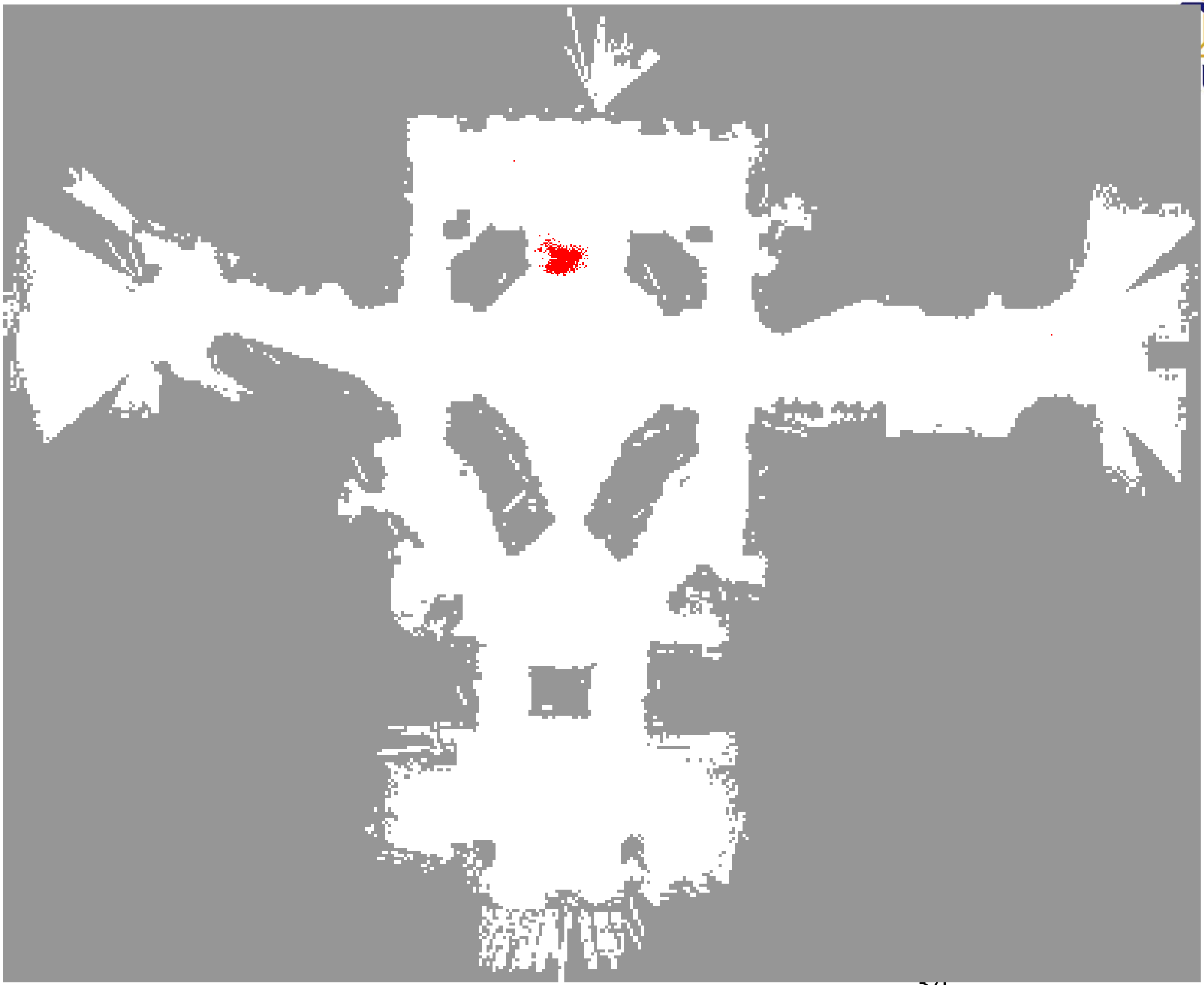


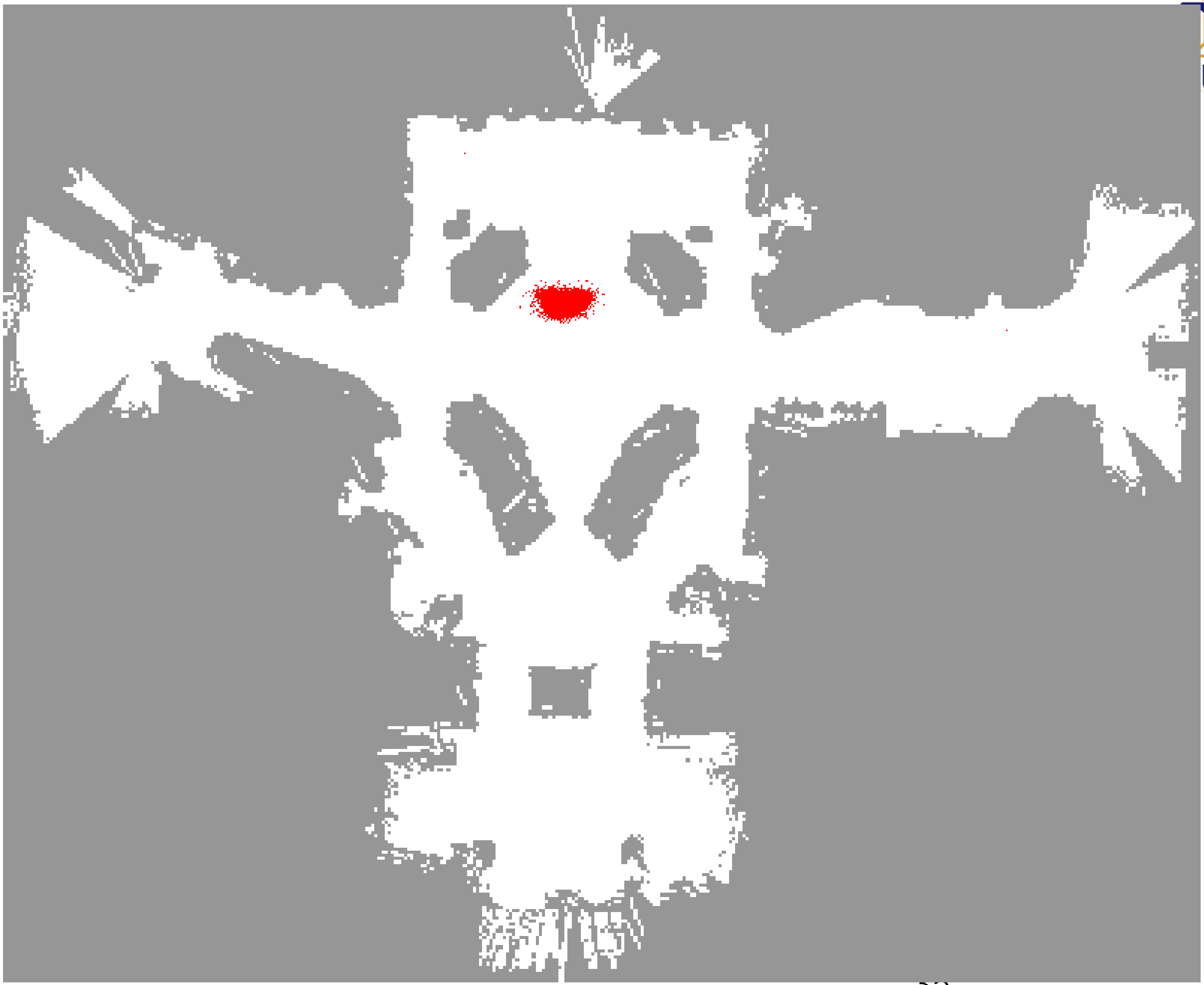


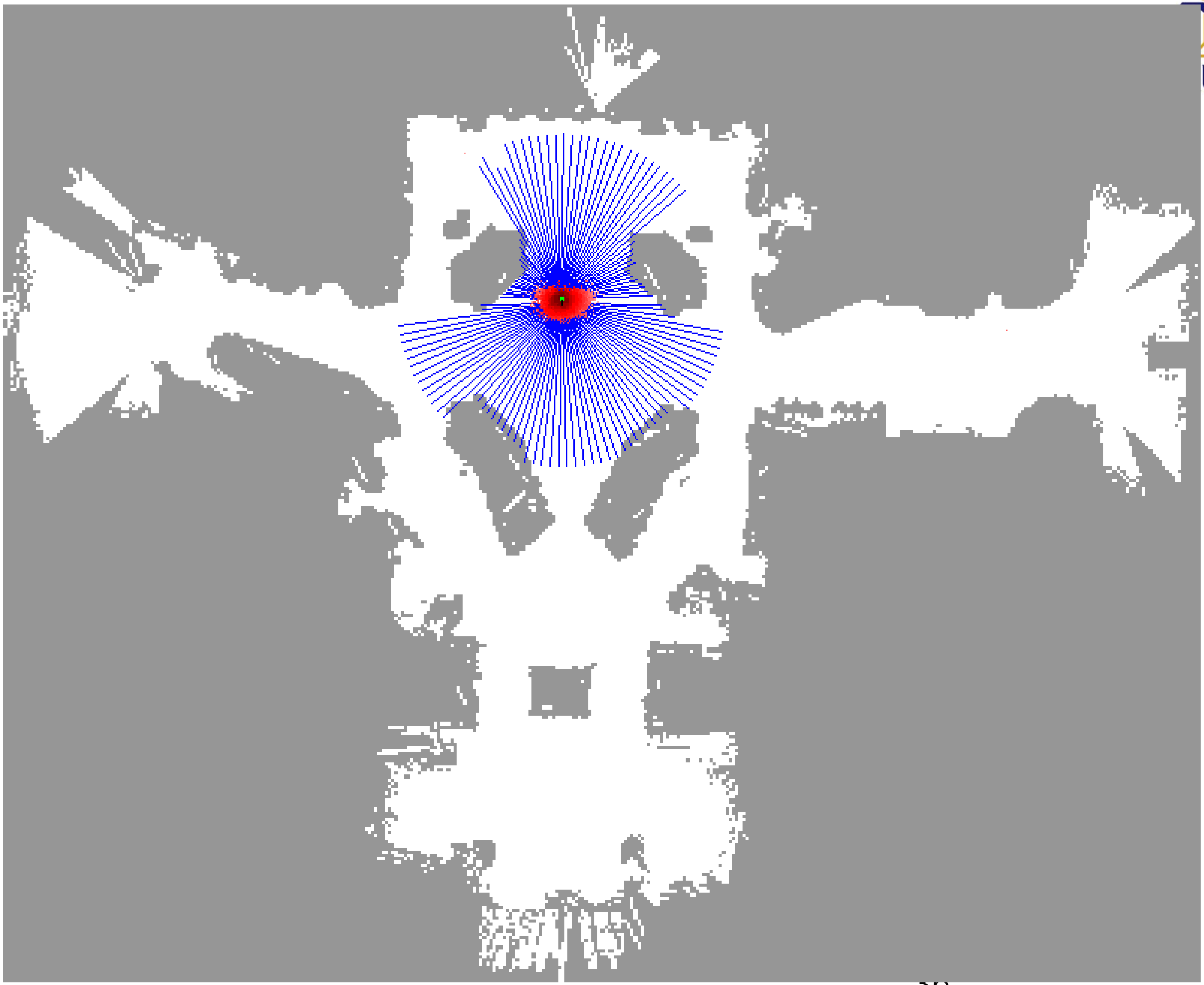


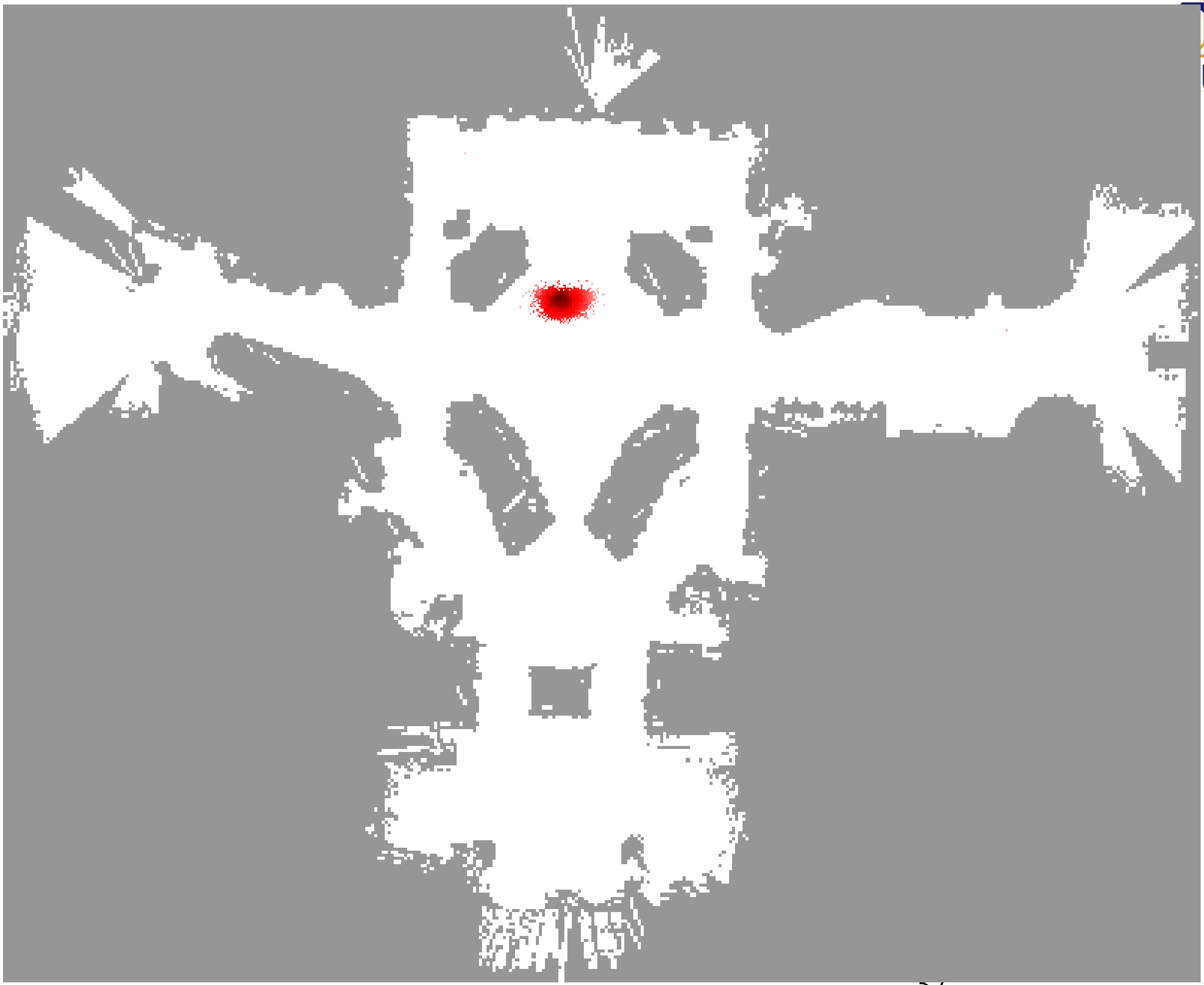


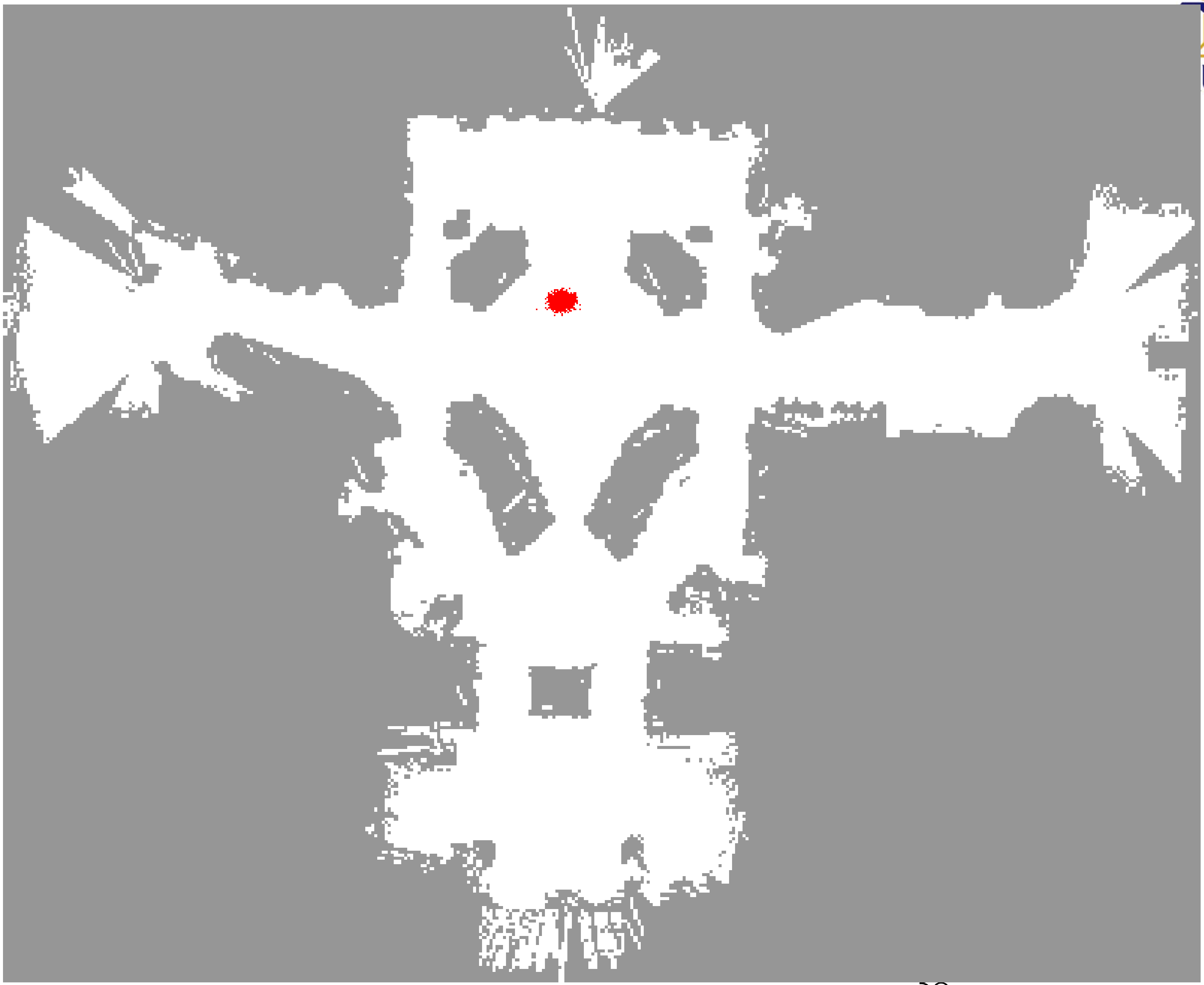


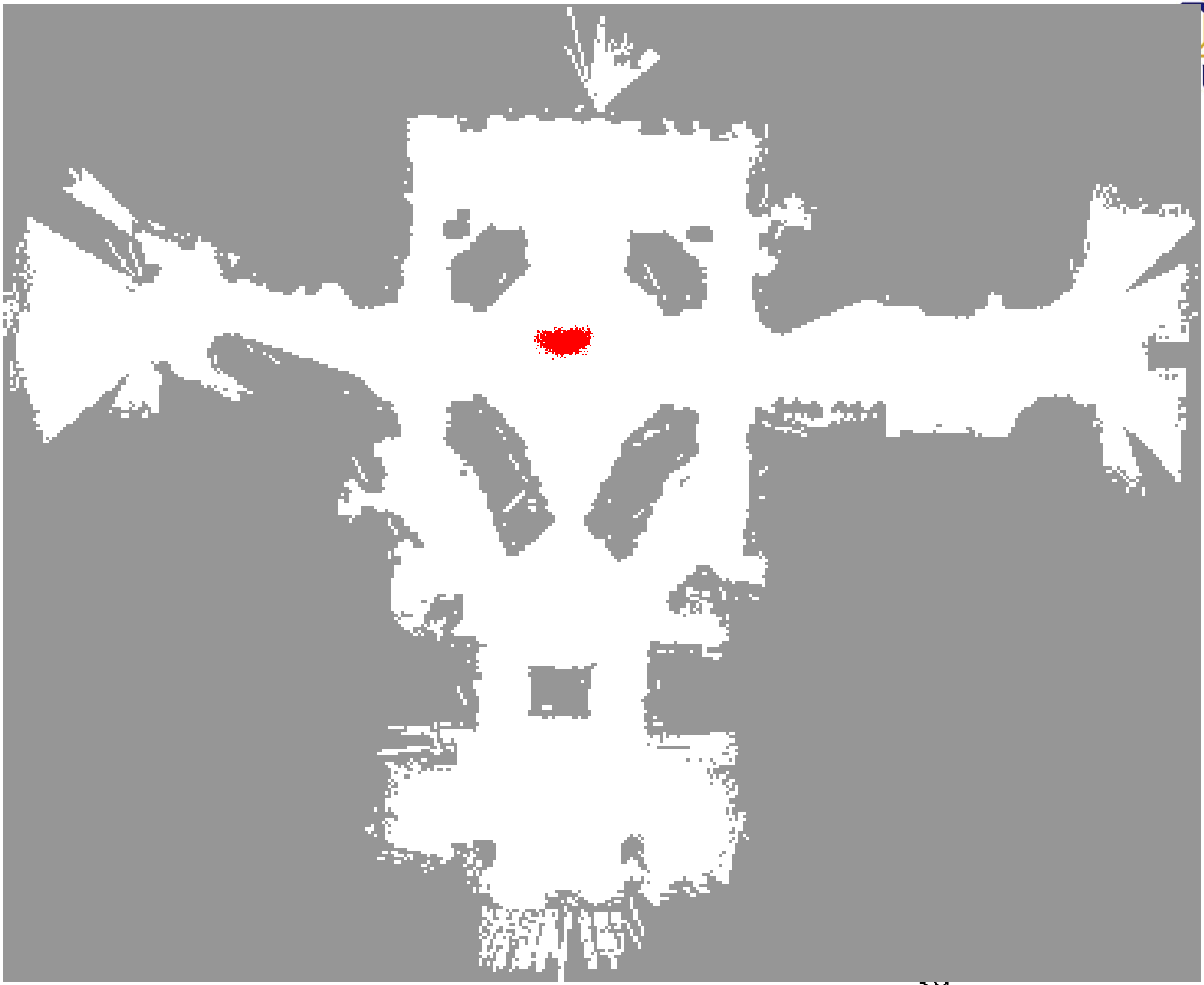


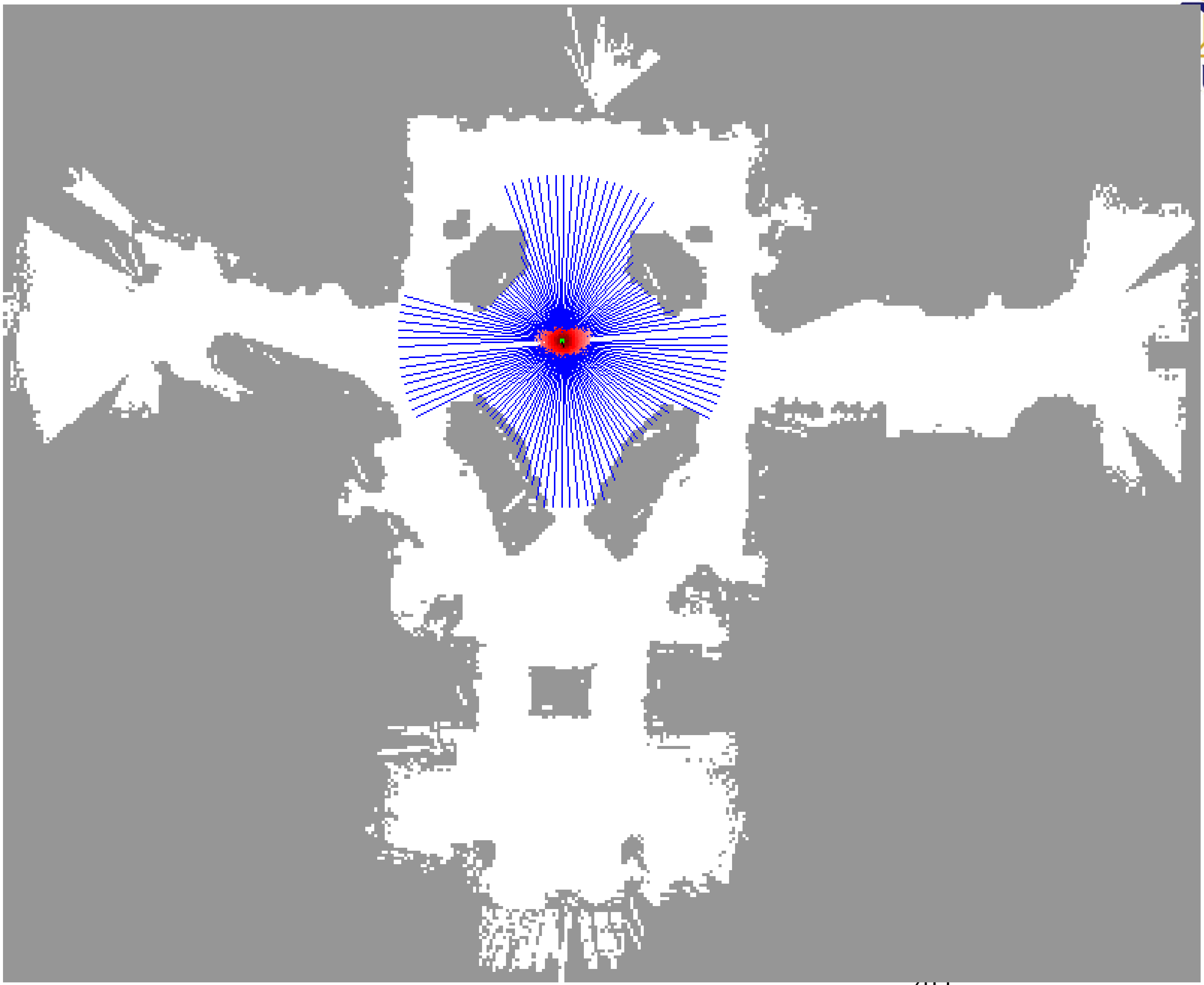


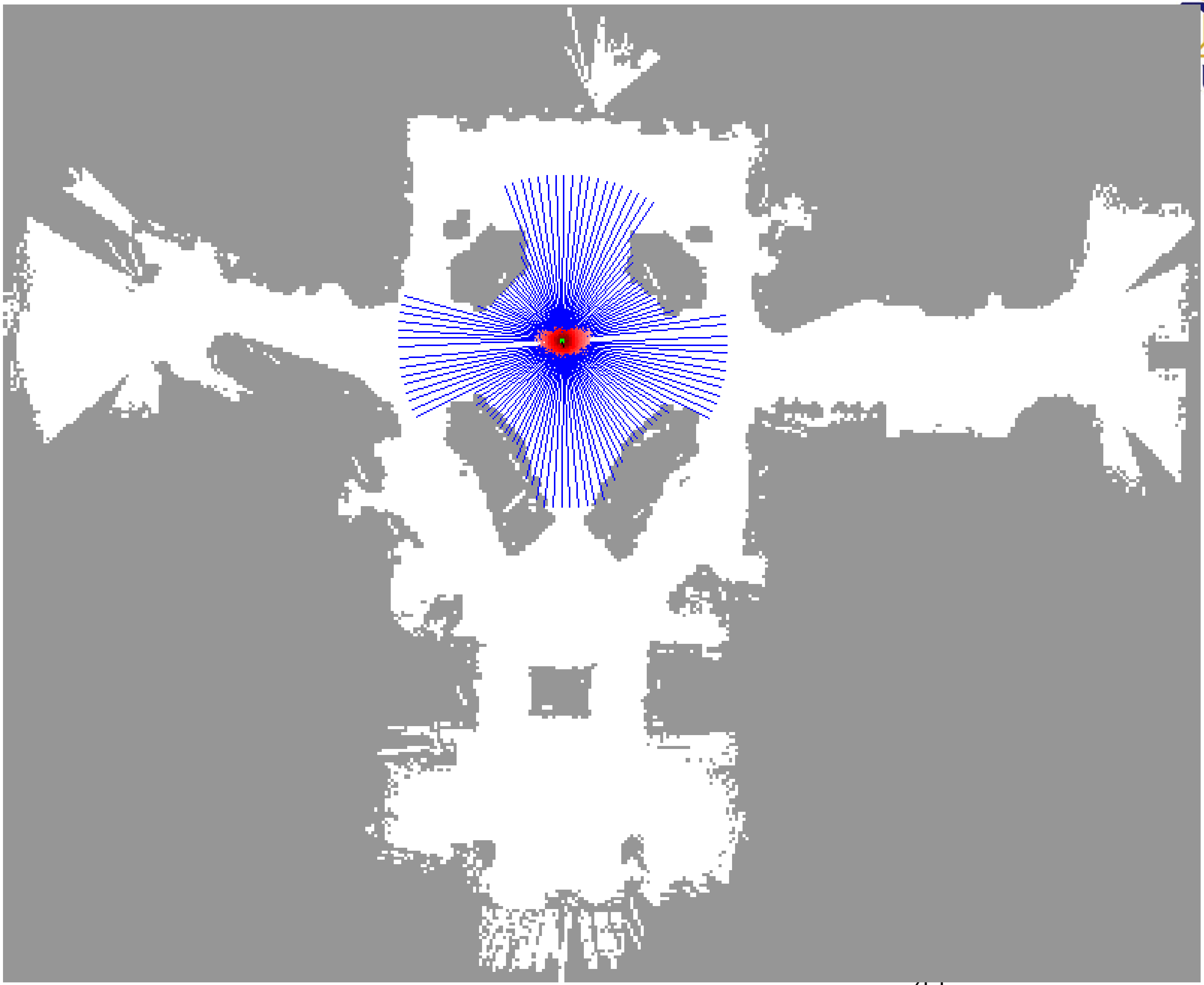




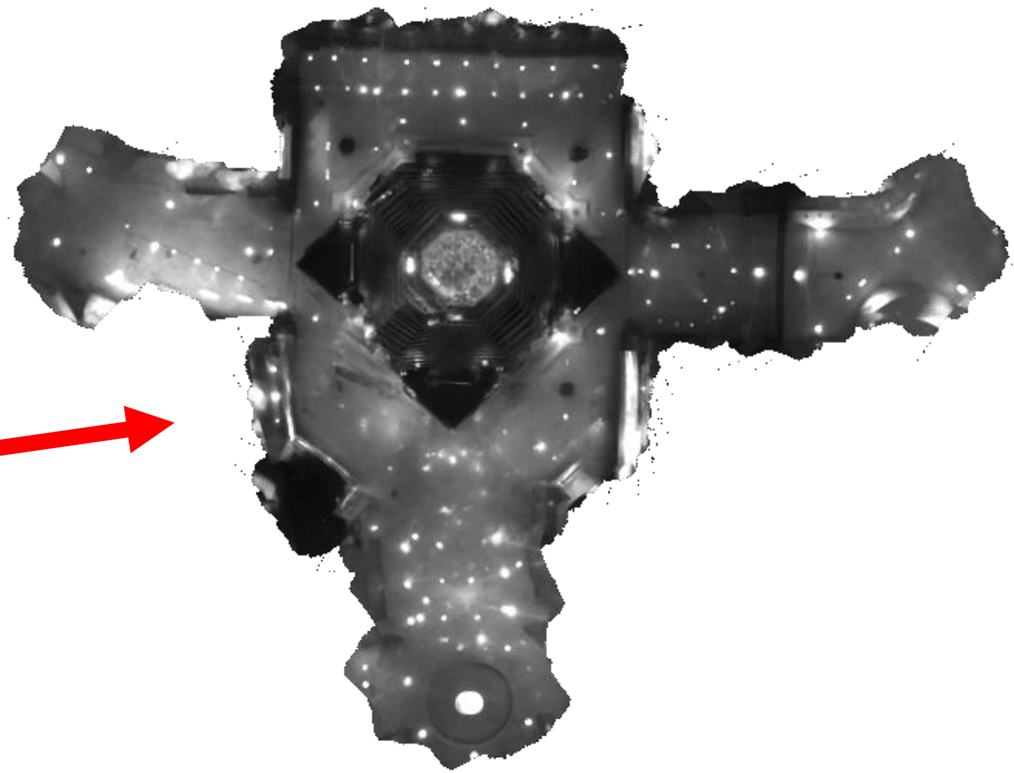




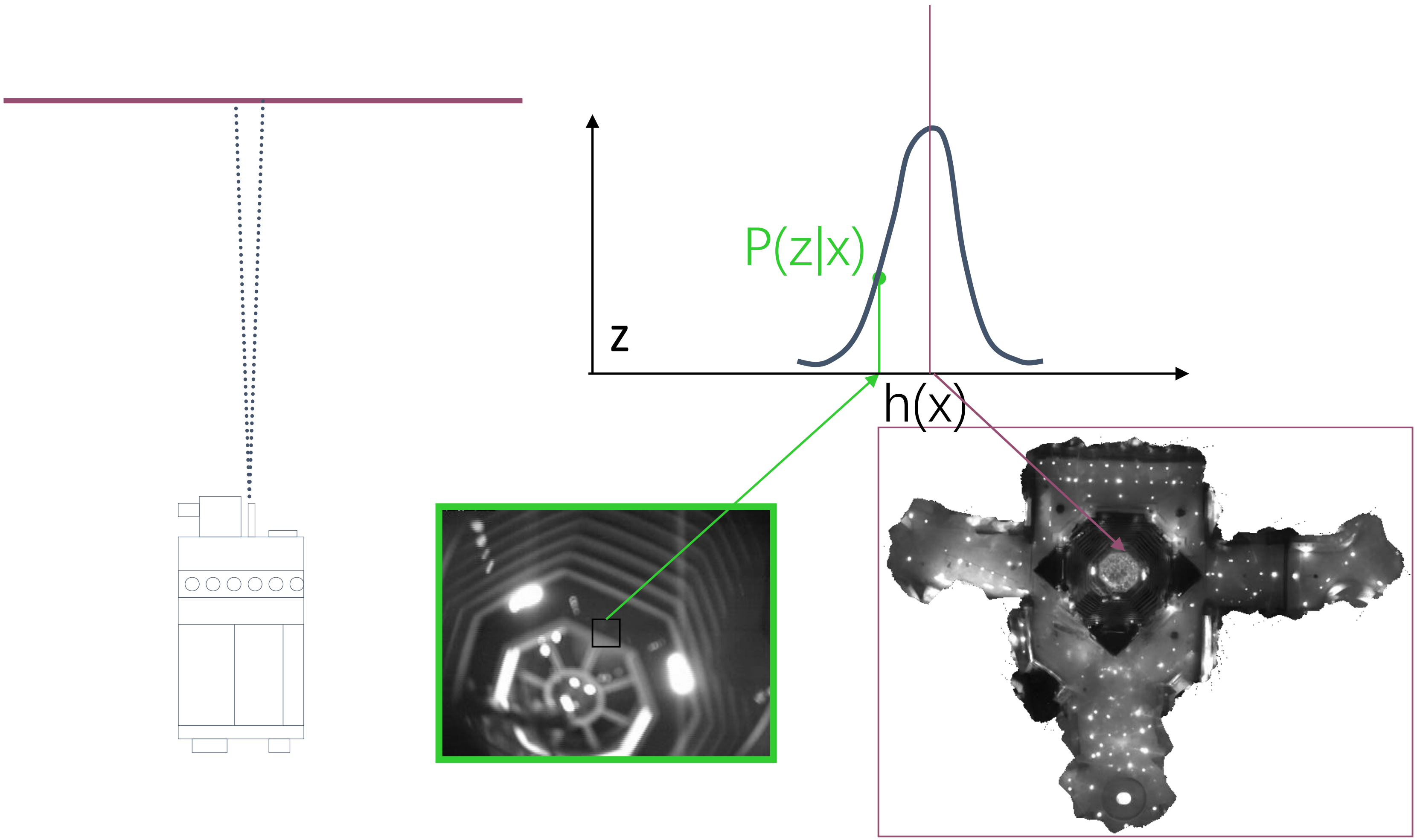




Ceiling Maps for Localization

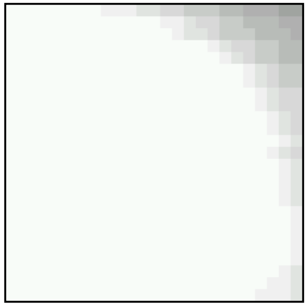


Vision-Based Localization

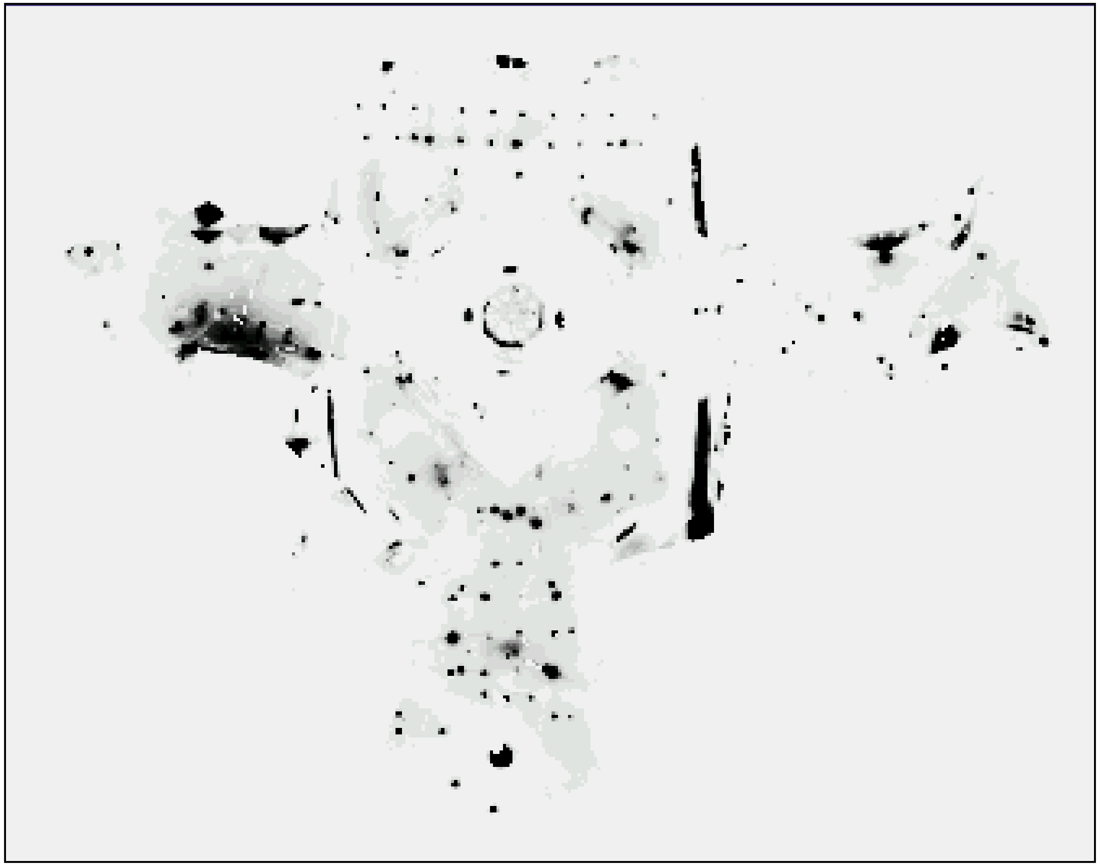


Under a Light

Measurement z:

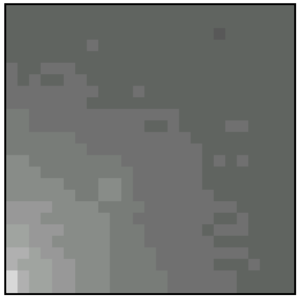


$P(z|x)$:

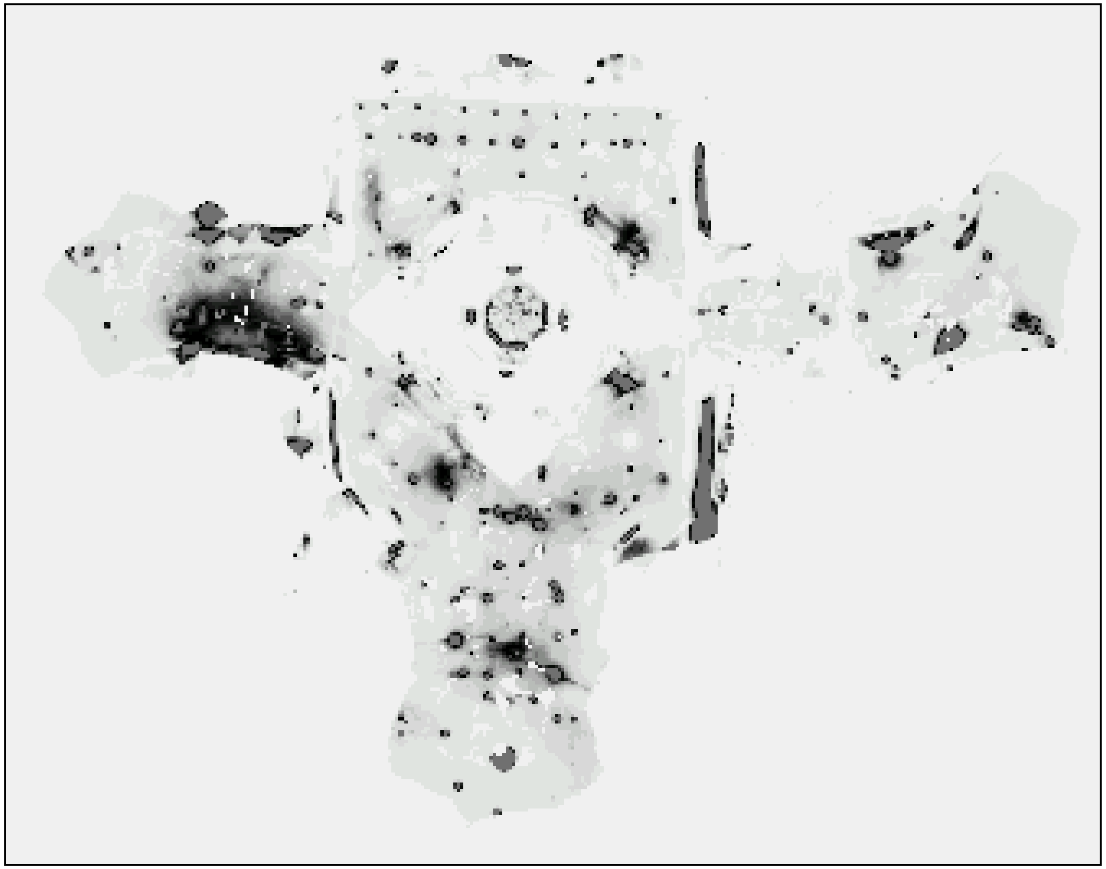


Next to a Light

Measurement z:



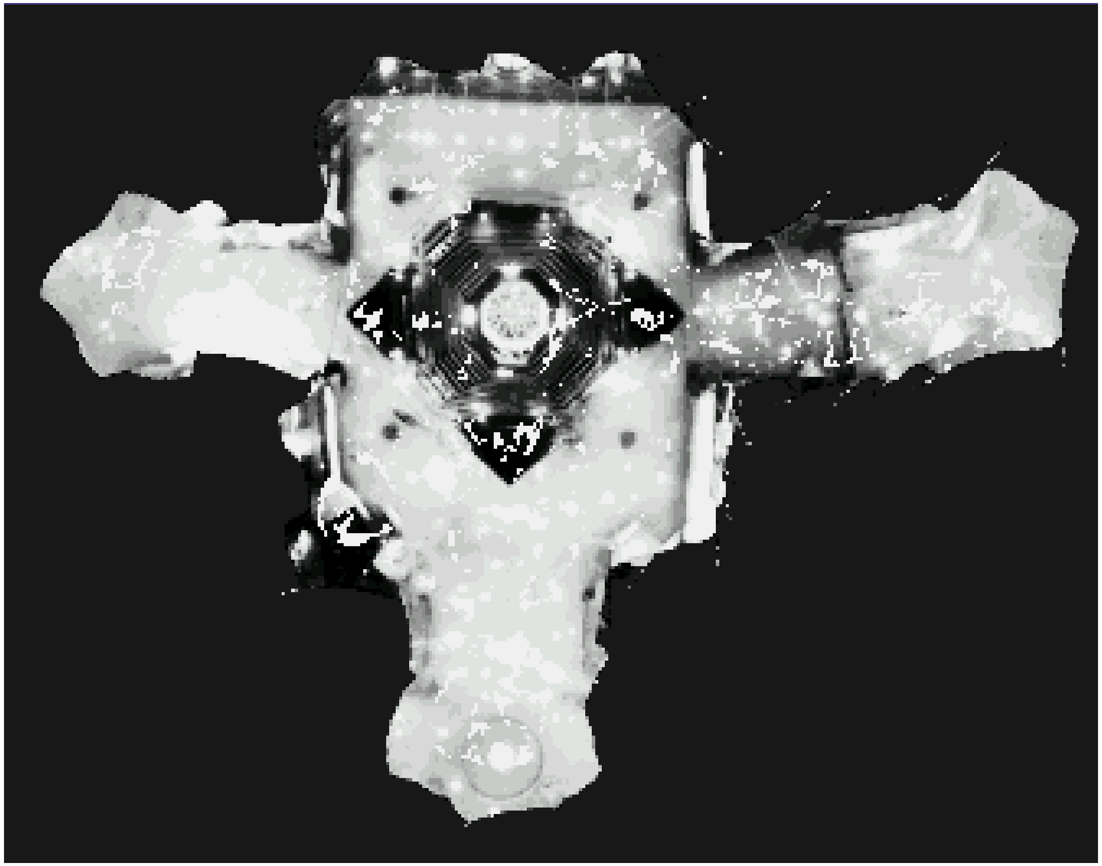
$P(z|x)$:



Elsewhere

Measurement z:

$P(z|x)$:

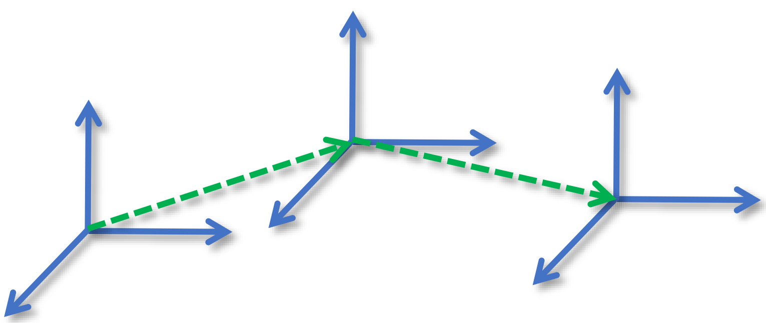


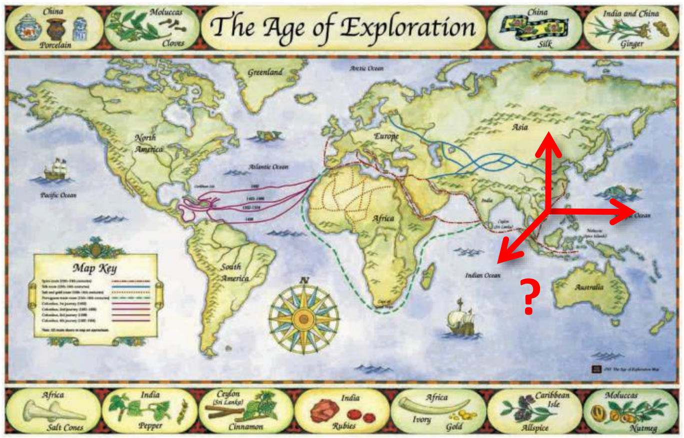
Courtesy:

Localization Problems (Recap L3)

- Pose tracking
  - the initial robot pose is known
  - the pose distribution is bounded, local precision for evaluation
- Global localization (GL)
  - estimate the pose without initial pose
  - with uniform distribution
  - **Kidnapped robot problem:** a variant of the GL problem

  - the robot might believe it knows where it is while it does not





Courtesy: Probabilistic Robotics Chap. 7.2

The Kidnapped Robot Problem

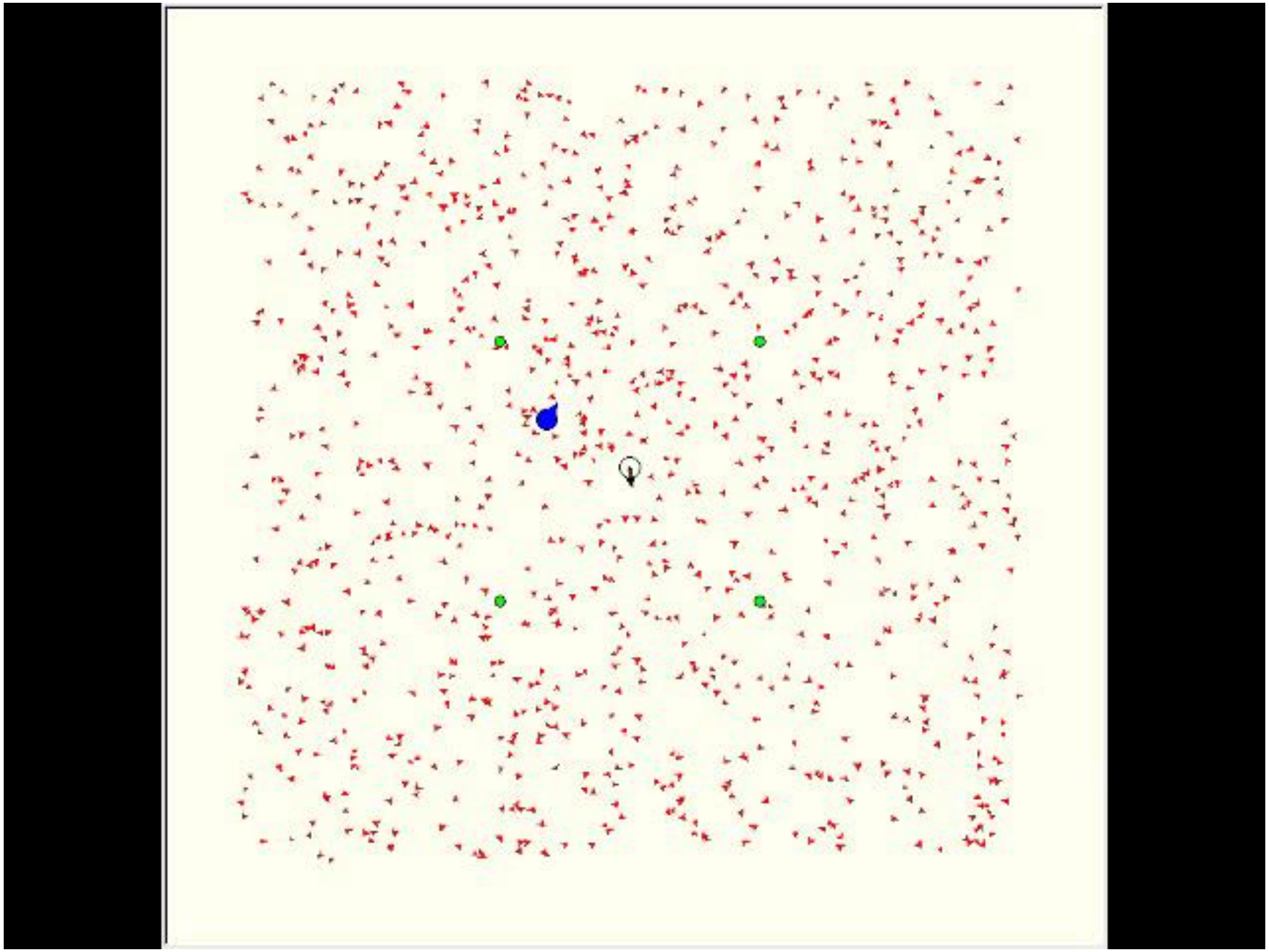
- Randomly insert samples (the robot can be teleported at any point in time)
- Insert random samples proportional to the average likelihood of the particles (the robot has been teleported with higher probability when the likelihood of its observations drops)

Example

- Kidnap the robot dog
- Global localization with Landmark Detection

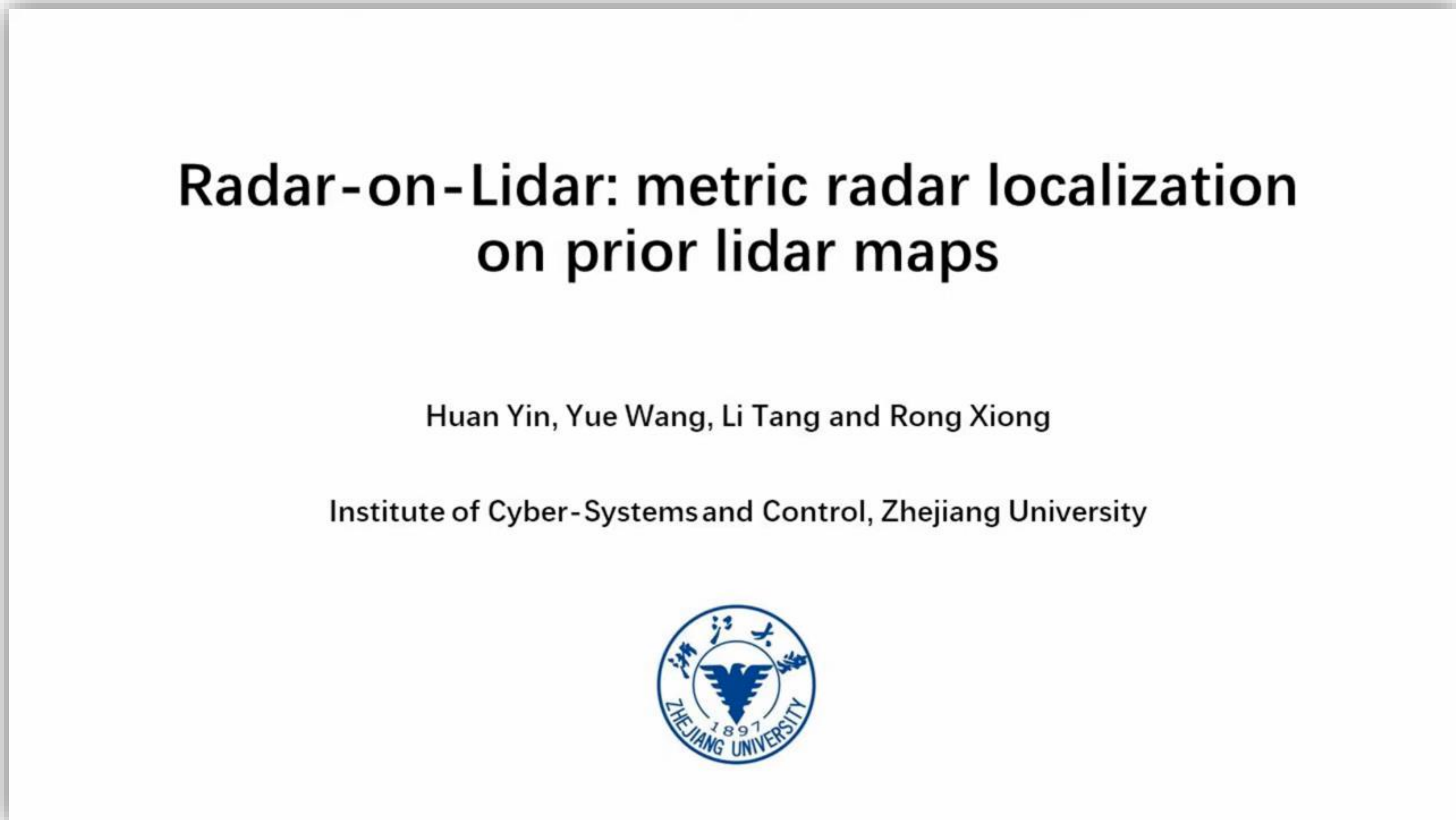
The Kidnapped Robot Problem

- Generate global samples randomly every step



Our Work using Particle Filter

- Huan Yin, Yue Wang, Li Tang, and Rong Xiong. "Radar-on-lidar: metric radar localization on prior lidar maps." In *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, Best Conference Paper



Visual Localization using PF

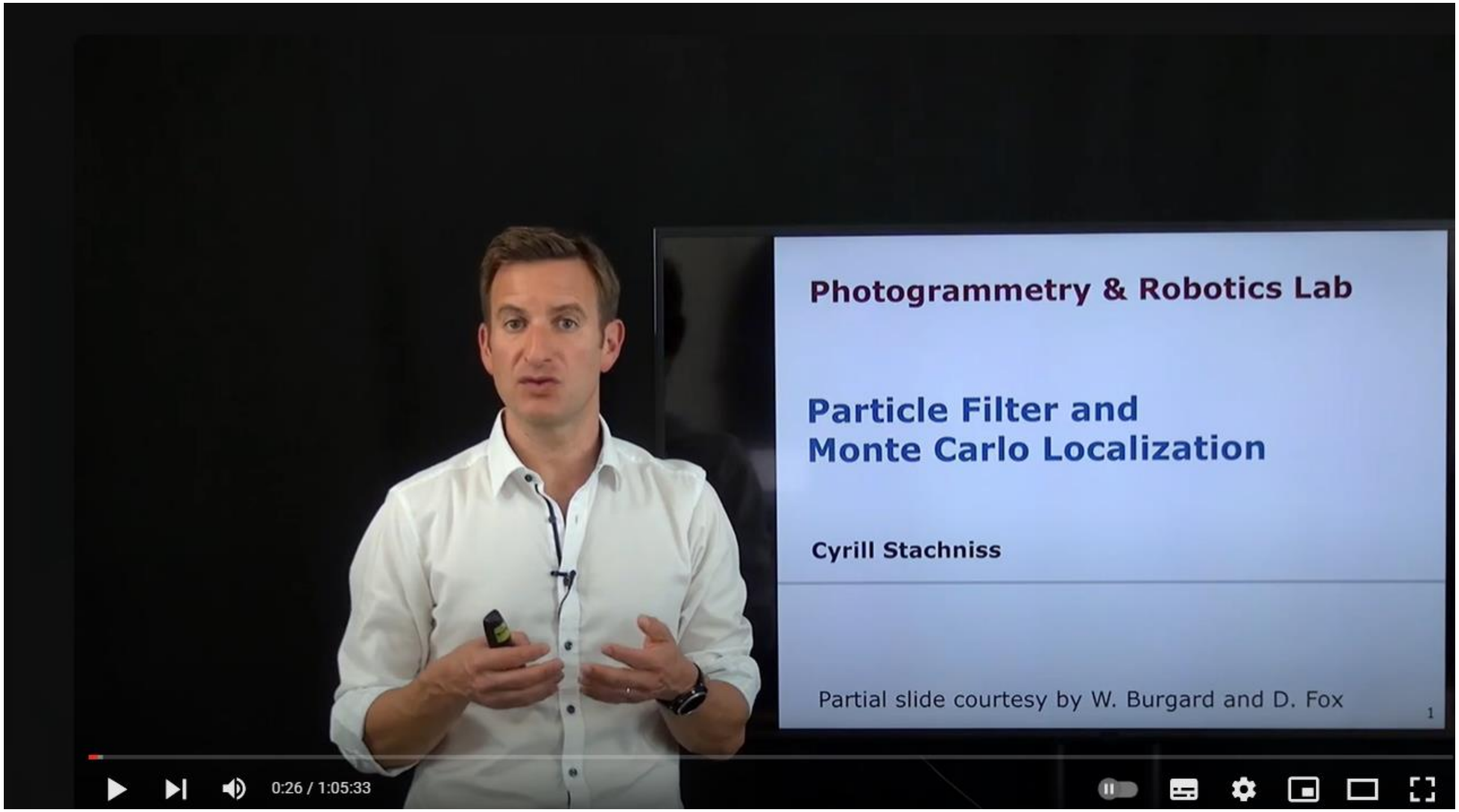
Courtesy: YouTube

Visual Tracking using PF

Courtesy: YouTube

Resources

- Probabilistic Robotics Chapter 4.3
- Prof. Cyrill Stachniss



Summary

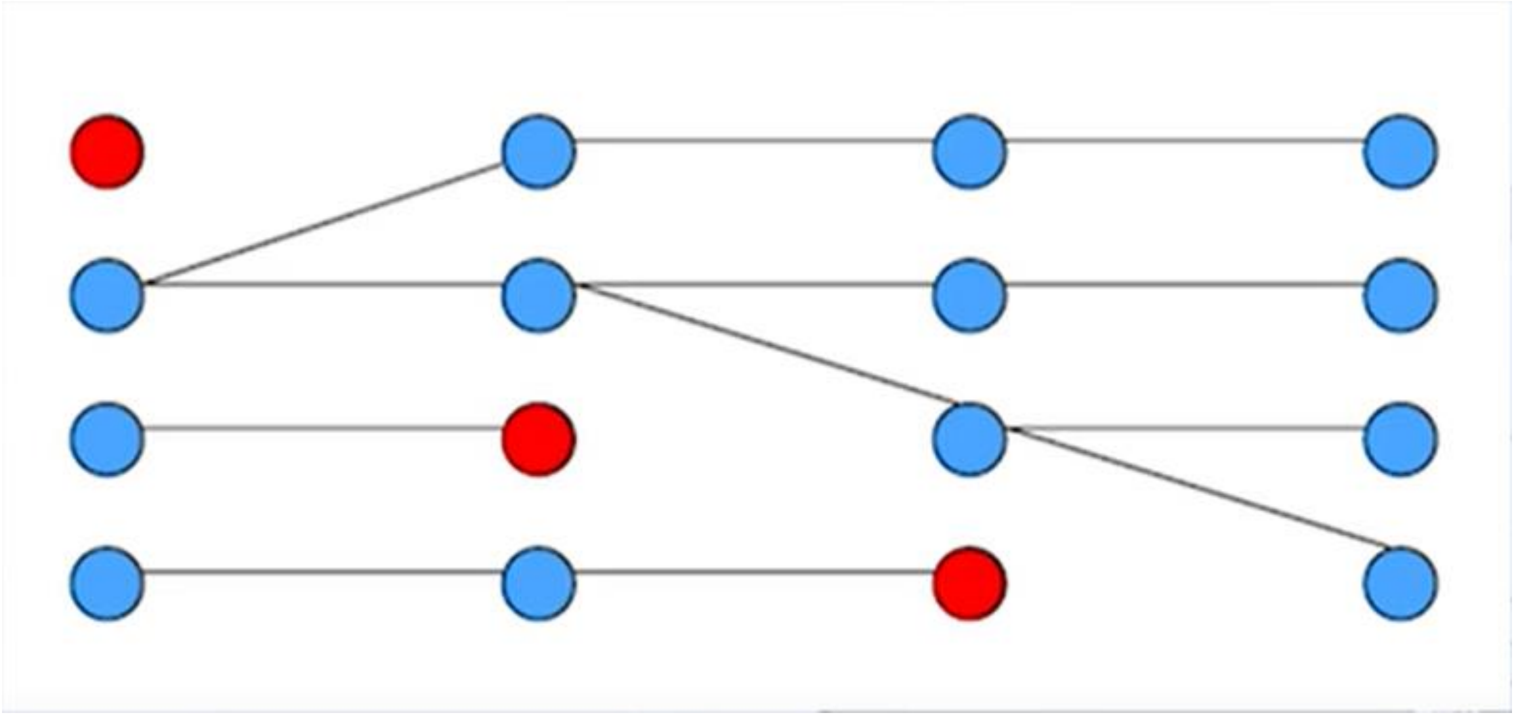
- Particle filters are implementations of recursive Bayesian filtering
- They represent the posterior by a set of weighted samples
- In the context of localization, the particles are propagated according to the motion model
- They are then weighted according to the likelihood of the observations
- In the re-sampling step, new particles are drawn with a probability proportional to the likelihood of the observation
- The art is the motion and measurement model

Summary

- Pros (Compared to KF family)
  - Easy to implement
  - Able to handle nonlinear systems without linearization
  - Able to represent arbitrary distribution
- Cons
  - Particle degeneracy problem
  - Need lots of particles to represent high dimensional state space, computational complexity increases significantly w.r.t state dimension
- Applications
  - Widely used for low dimensional problems: robot pose tracking, target tracking, etc.
  - Used for initialization of global localization to resolve the multi-modal issue, then switch to unimodal (e.g. Kalman Filter) methods
  - Used to be popular for SLAM, but not anymore

Particle Degeneracy problem

- Filter out 25% of particles each time
- After 4 operations, particles originate from the same hypothesis
- How to solve?



Courtesy: Rong Xiong

Next Lecture

- Gaussian Distribution
- Kalman Filter