CSE-571
Robotics

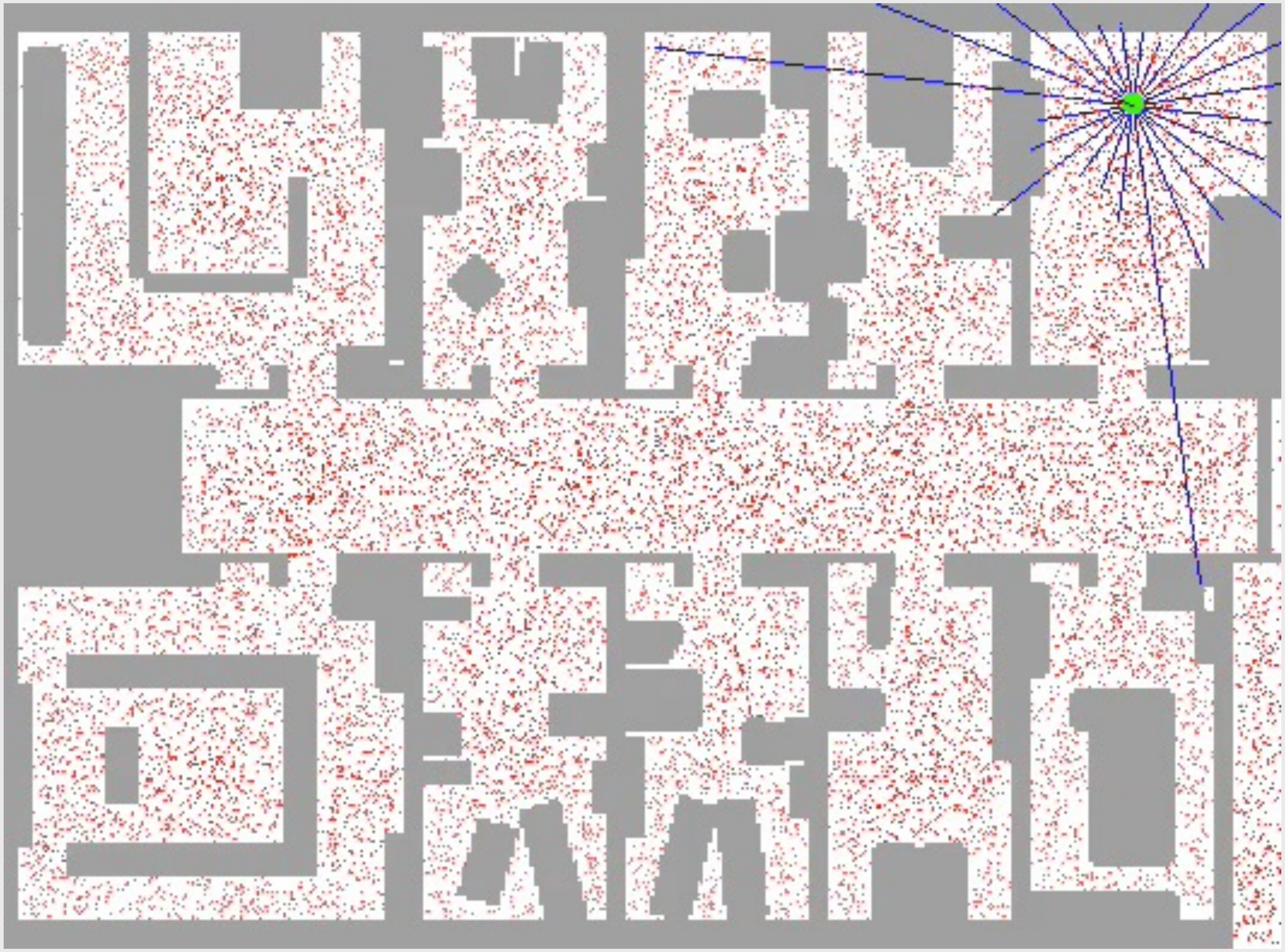
Bayes Filter Implementations

Particle filters

Motivation

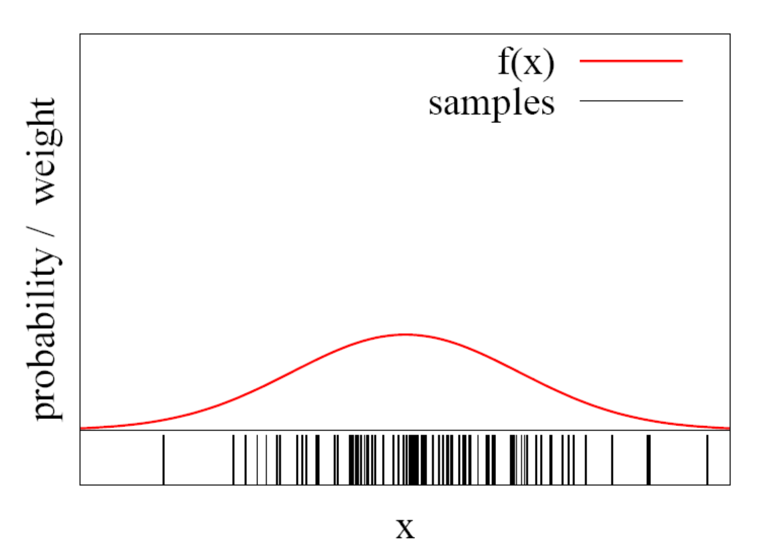
- So far, we discussed the
  - Kalman filter: Gaussian, linearization problems, multi-modal beliefs
- Particle filters are a way to **efficiently** represent **non-Gaussian distributions**
- Basic principle
  - Set of state hypotheses (“particles”)
  - Survival-of-the-fittest

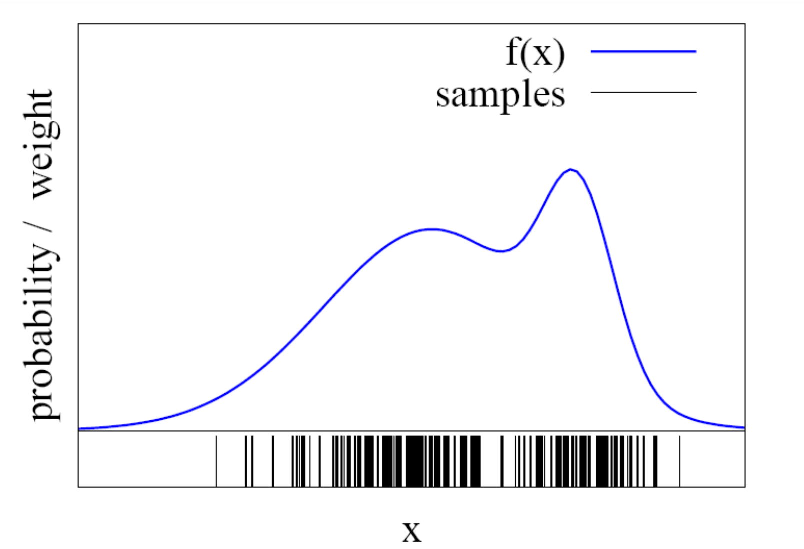
Sample-based Localization (sonar)



Density Approximation

- Particle sets can be used to approximate densities



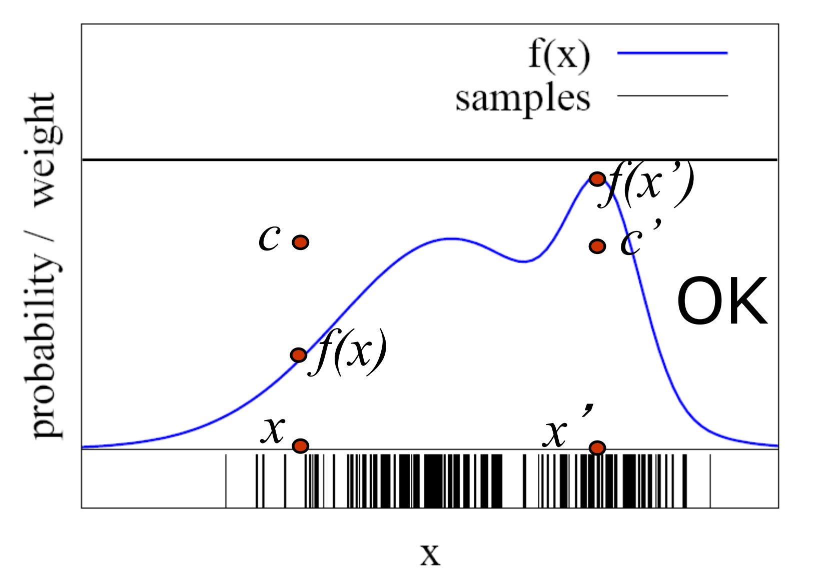


- The more particles fall into an interval, the higher the probability of that interval

- How to draw samples form a function/distribution?

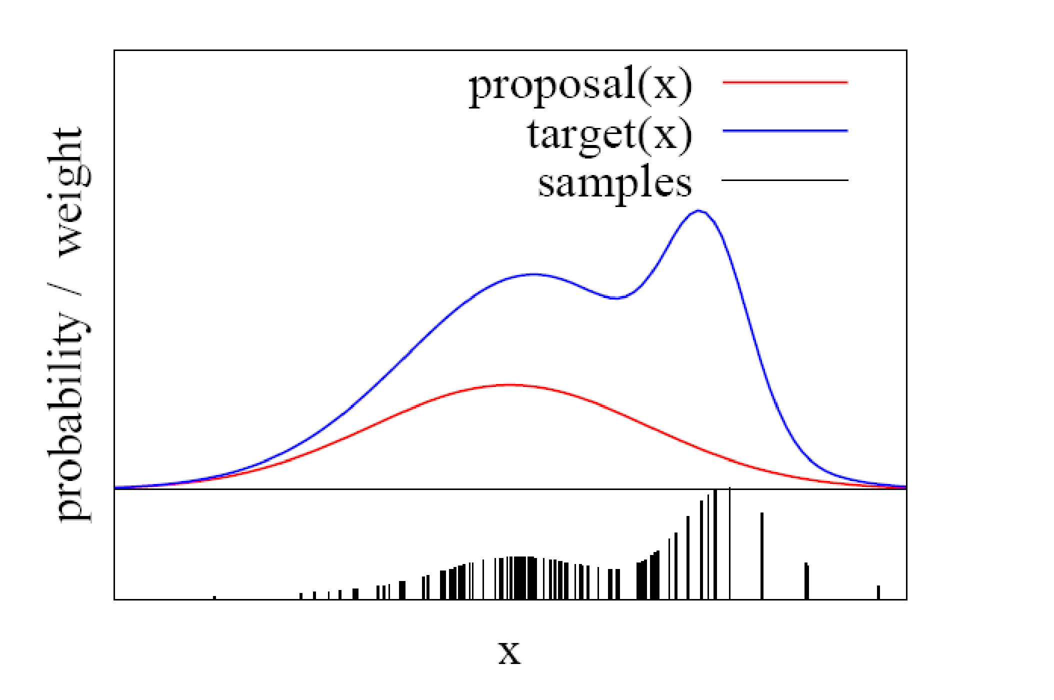
Rejection Sampling

- 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



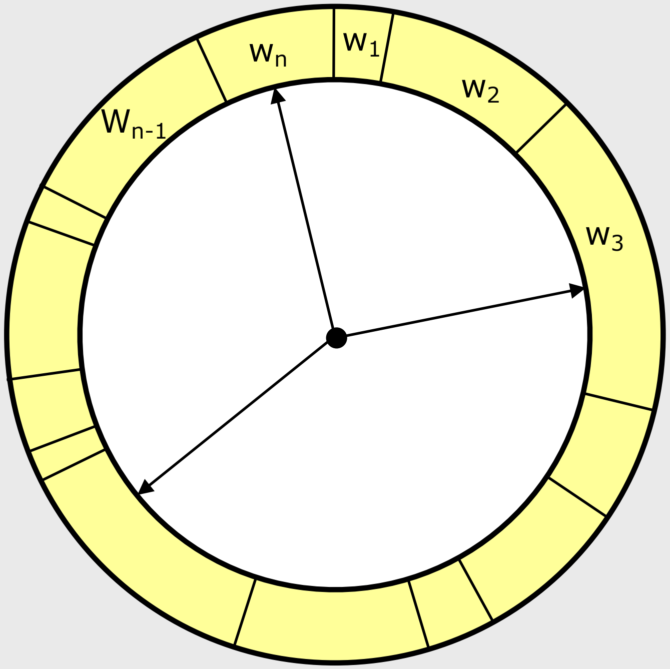
Resampling

- **Given**: Set S of weighted samples.

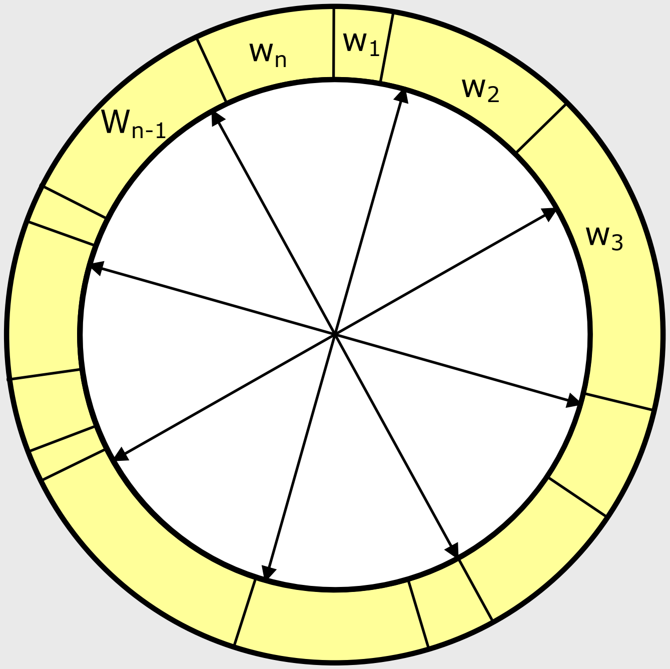
- **Wanted** : Random sample, where the probability of drawing x_i is given by w_i.

- Typically done n times with replacement to generate new sample set S'.

Resampling

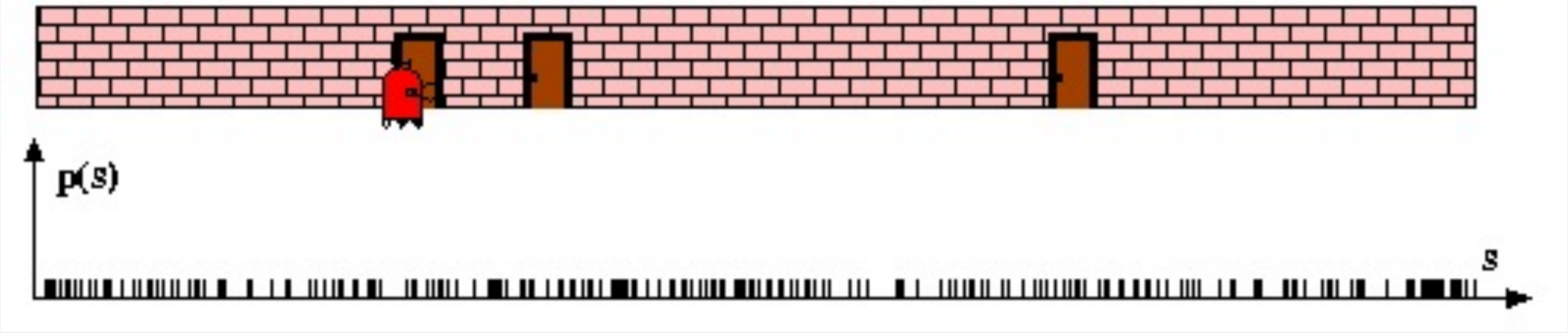


- Roulette wheel
- Binary search, n log n



- Stochastic universal sampling
- Systematic resampling
- Linear time complexity
- Easy to implement, low variance

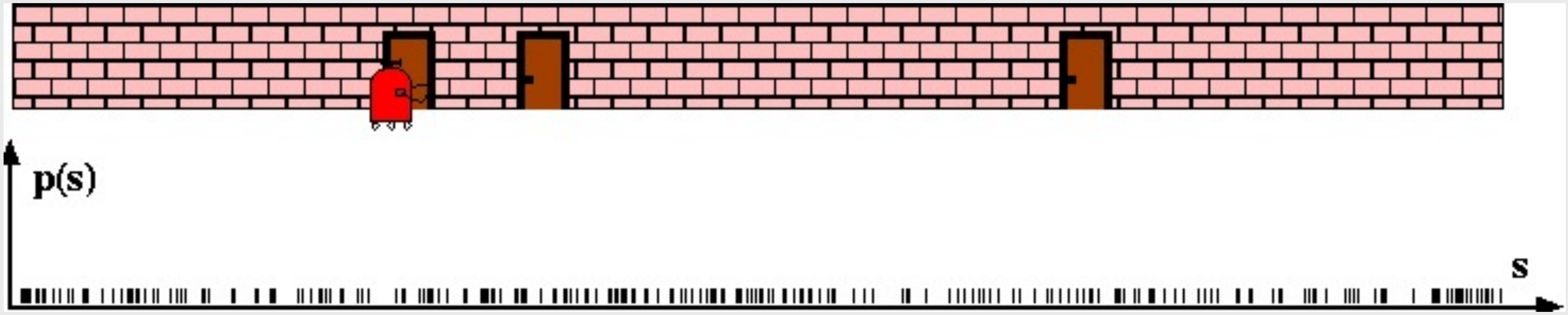
Particle Filters

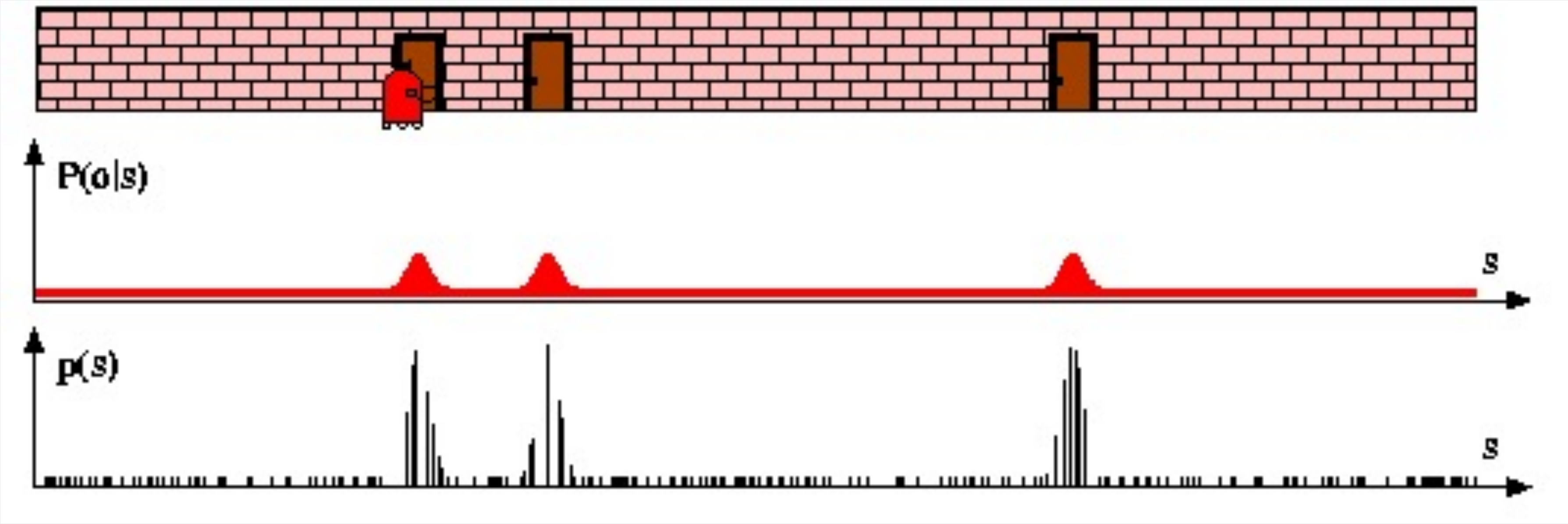


Sensor Information: Importance Sampling

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

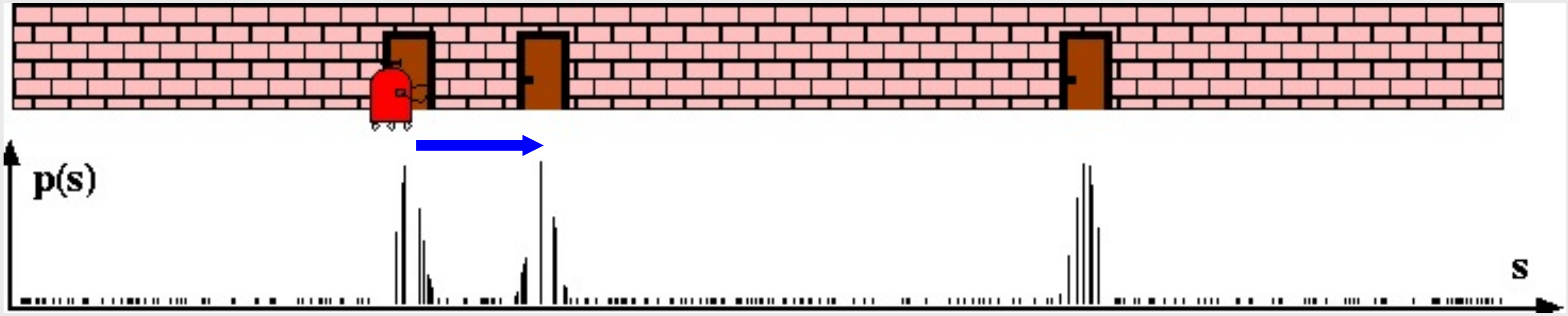
$$w \leftarrow \frac{\alpha\ p(z\,|\,x)\ Bel^{-}(x)}{Bel^{-}(x)} = \alpha\ p(z\,|\,x)$$

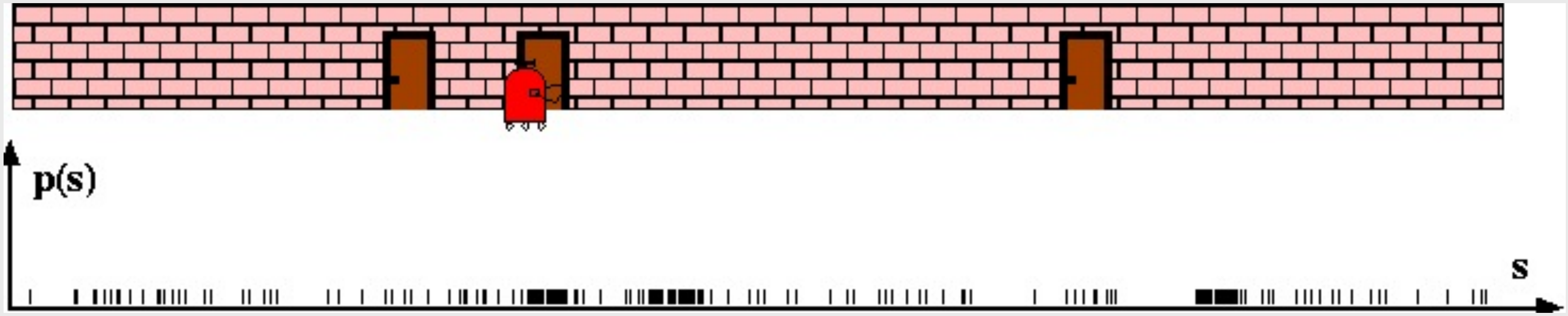




Robot Motion

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

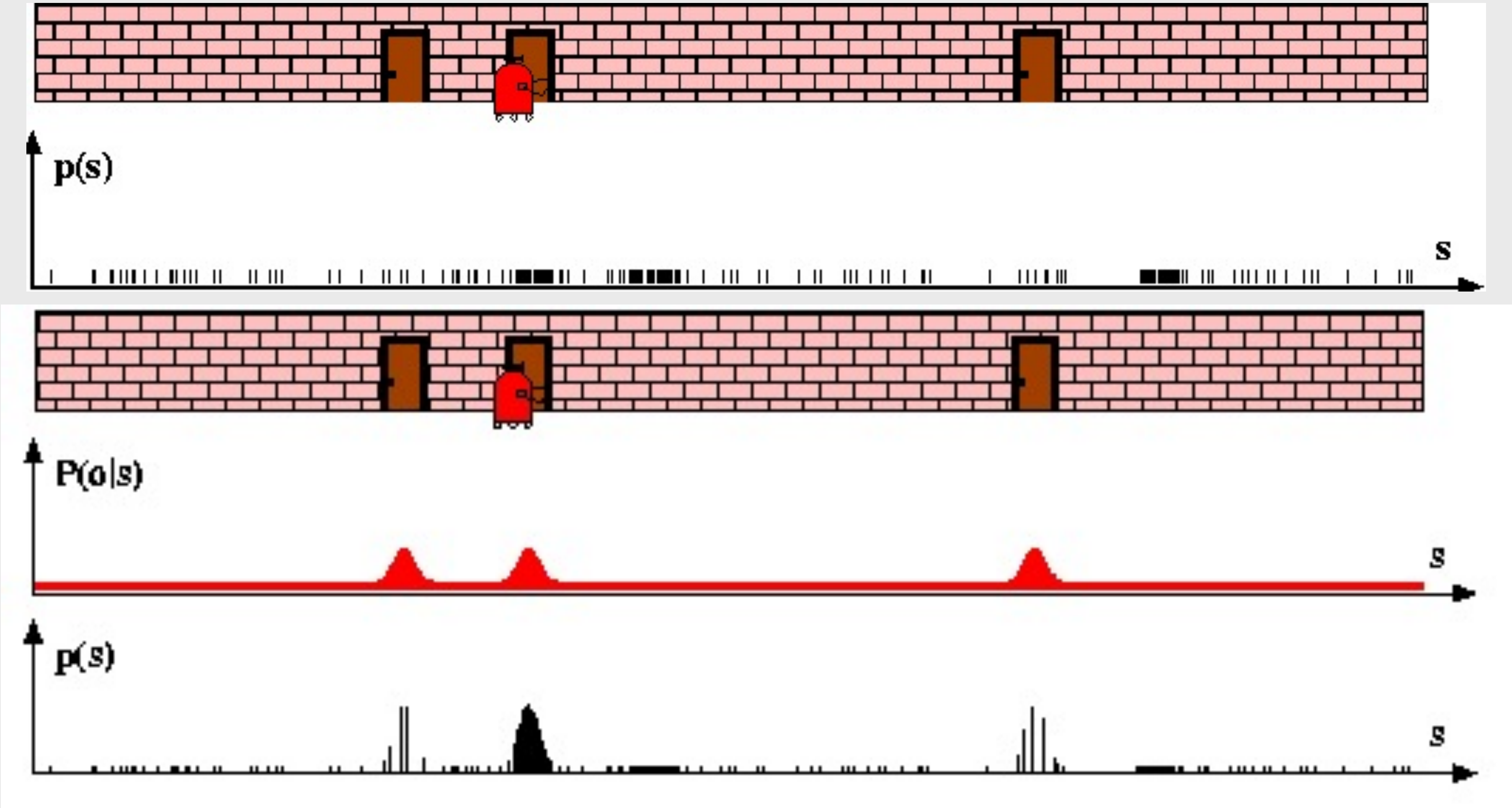




Sensor Information: Importance Sampling

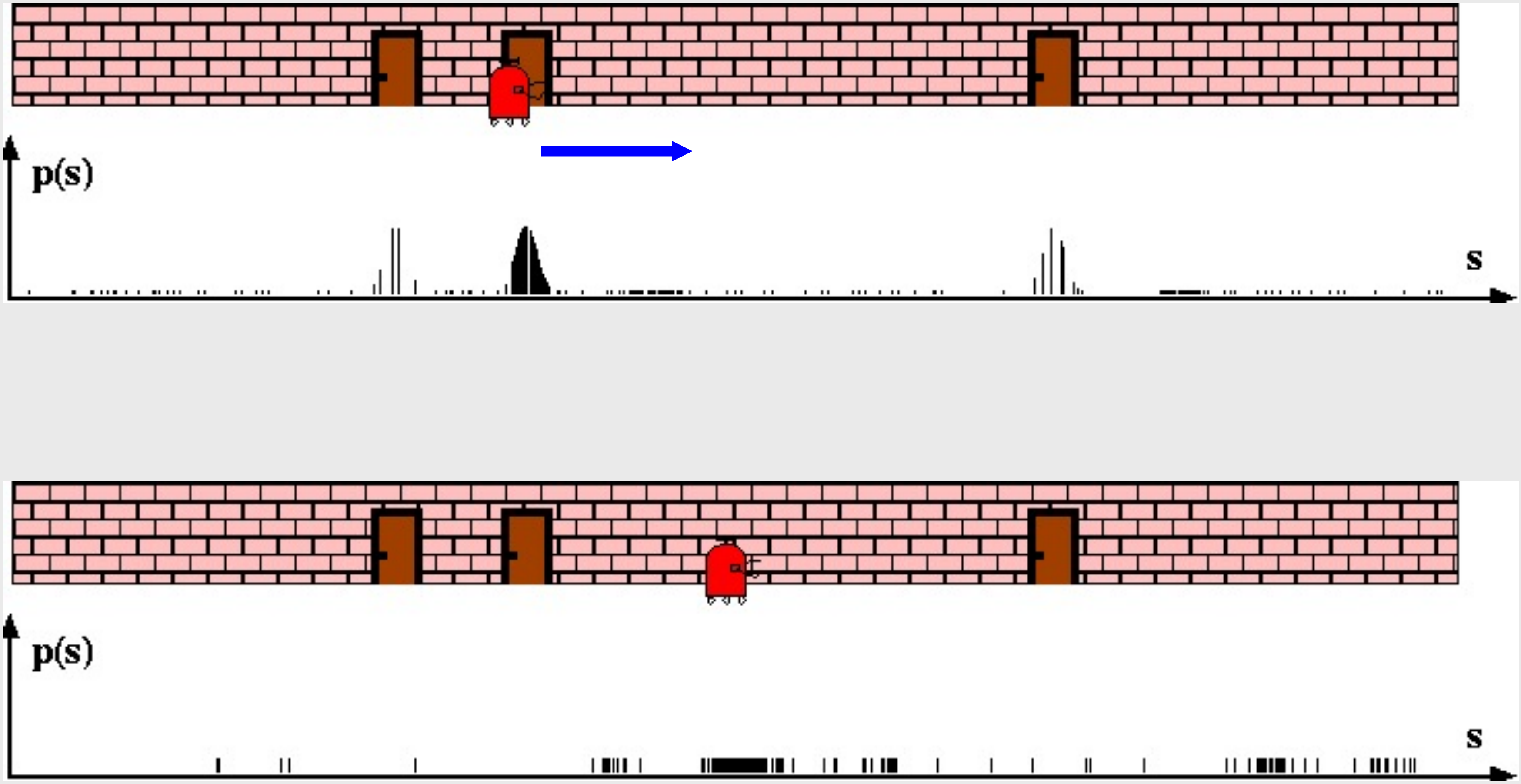
$$Bel(x) \leftarrow \alpha\, p(z\,|\,x)\, Bel^-(x)$$

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



Robot Motion

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



Particle Filter Algorithm

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

Particle Filter Algorithm

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

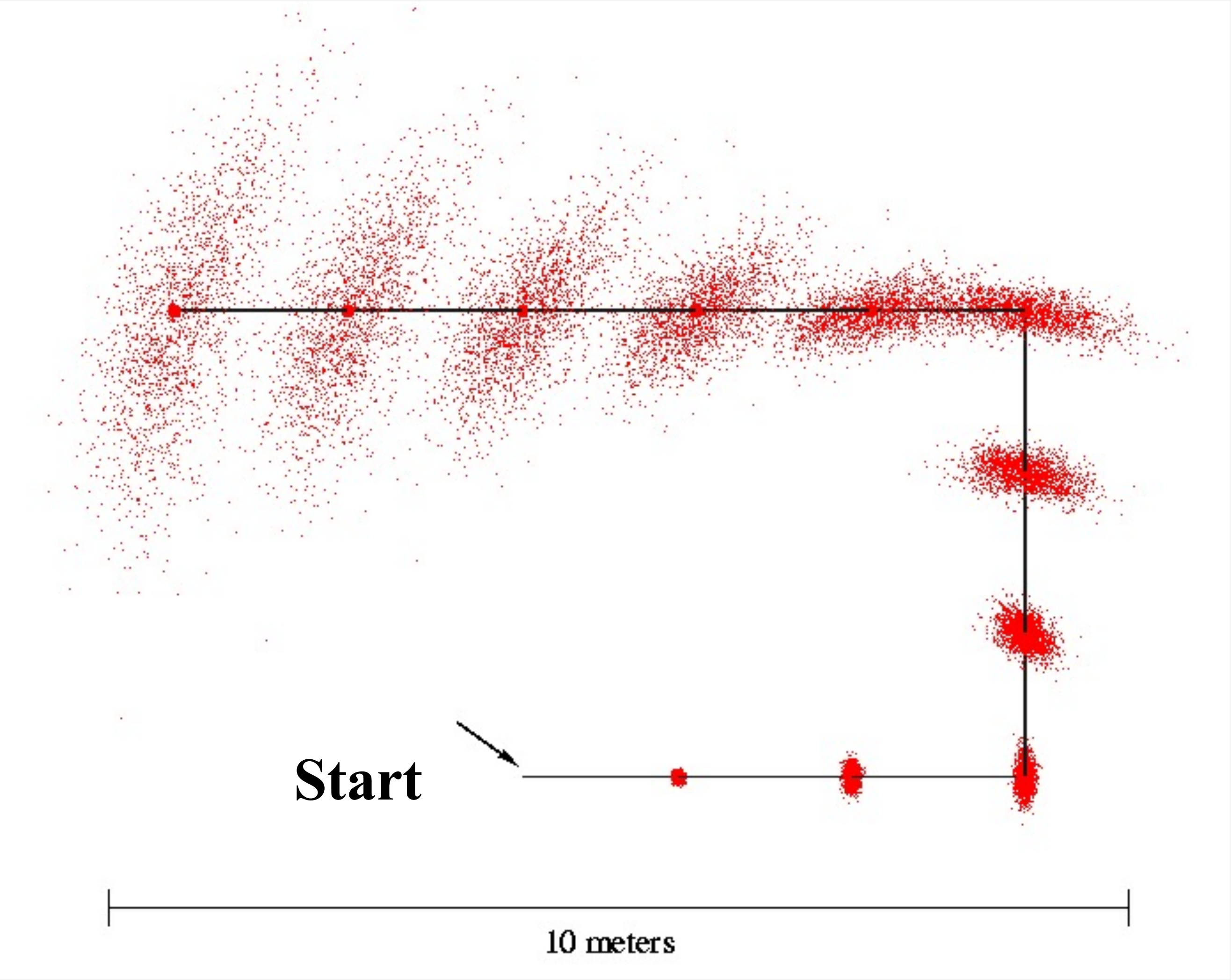
draw x^i_{t-1} from $Bel(\mathrm{x}_{t-1})$

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

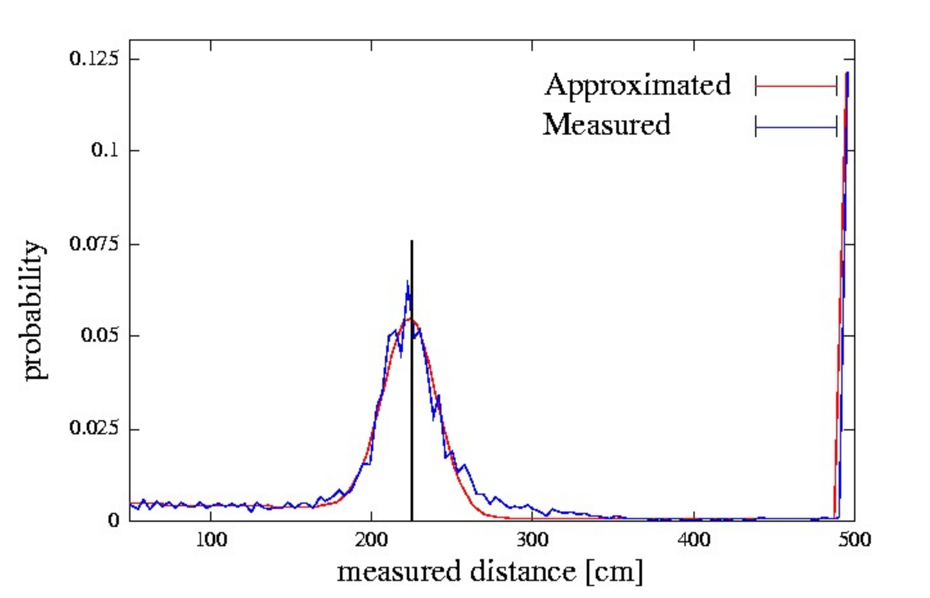
Importance factor for x^i_t:

$$\begin{aligned} w^i_t &= \frac{\text{target distribution}}{\text{proposal distribution}} \\ &= \frac{\eta\ p(z_t \mid x_t)\ p(x_t \mid x_{t-1}, u_{t-1})\ Bel\ (x_{t-1})}{p(x_t \mid x_{t-1}, u_{t-1})\ Bel\ (x_{t-1})} \\ &\propto p(z_t \mid x_t) \end{aligned}$$

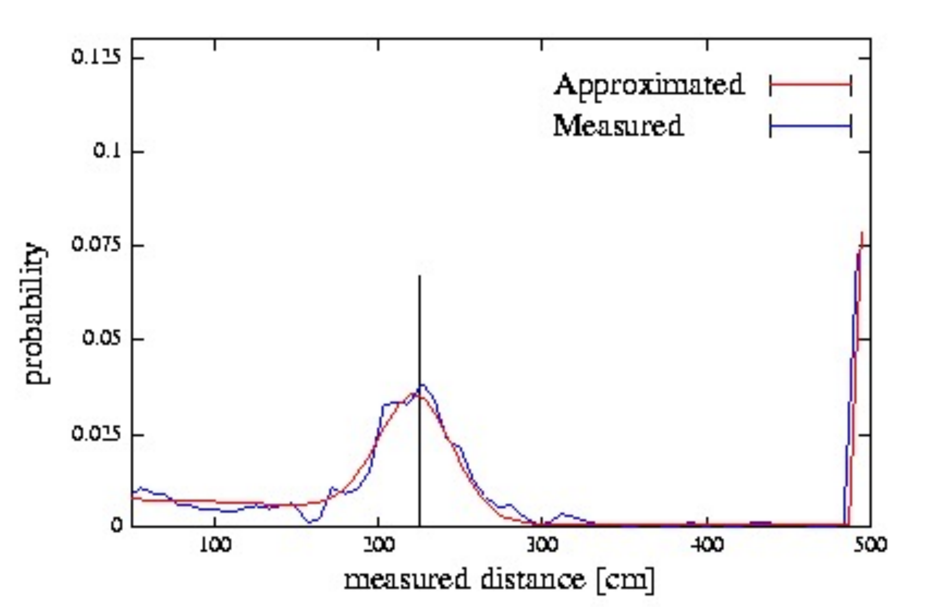
Motion Model Reminder



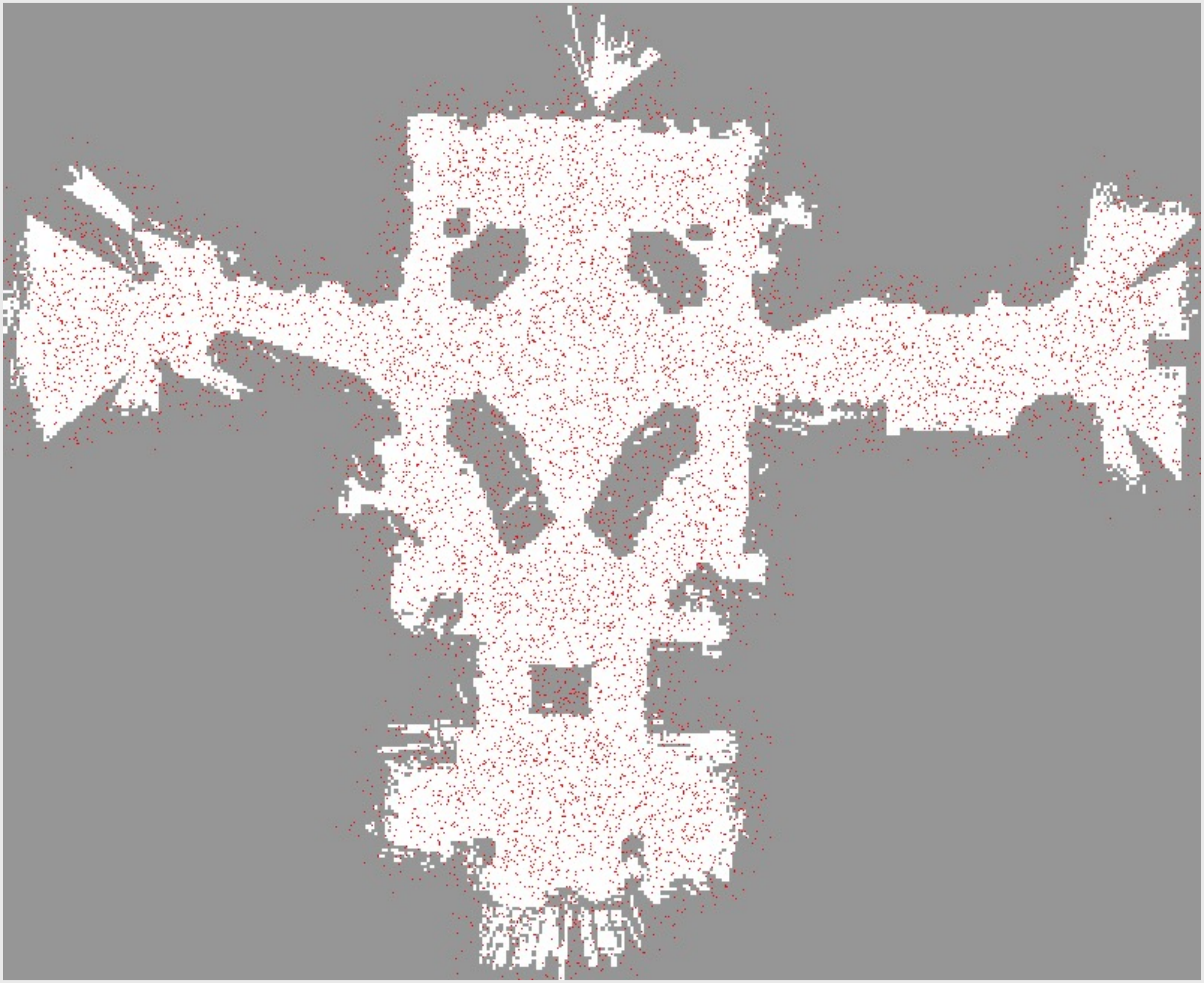
Proximity Sensor Model Reminder

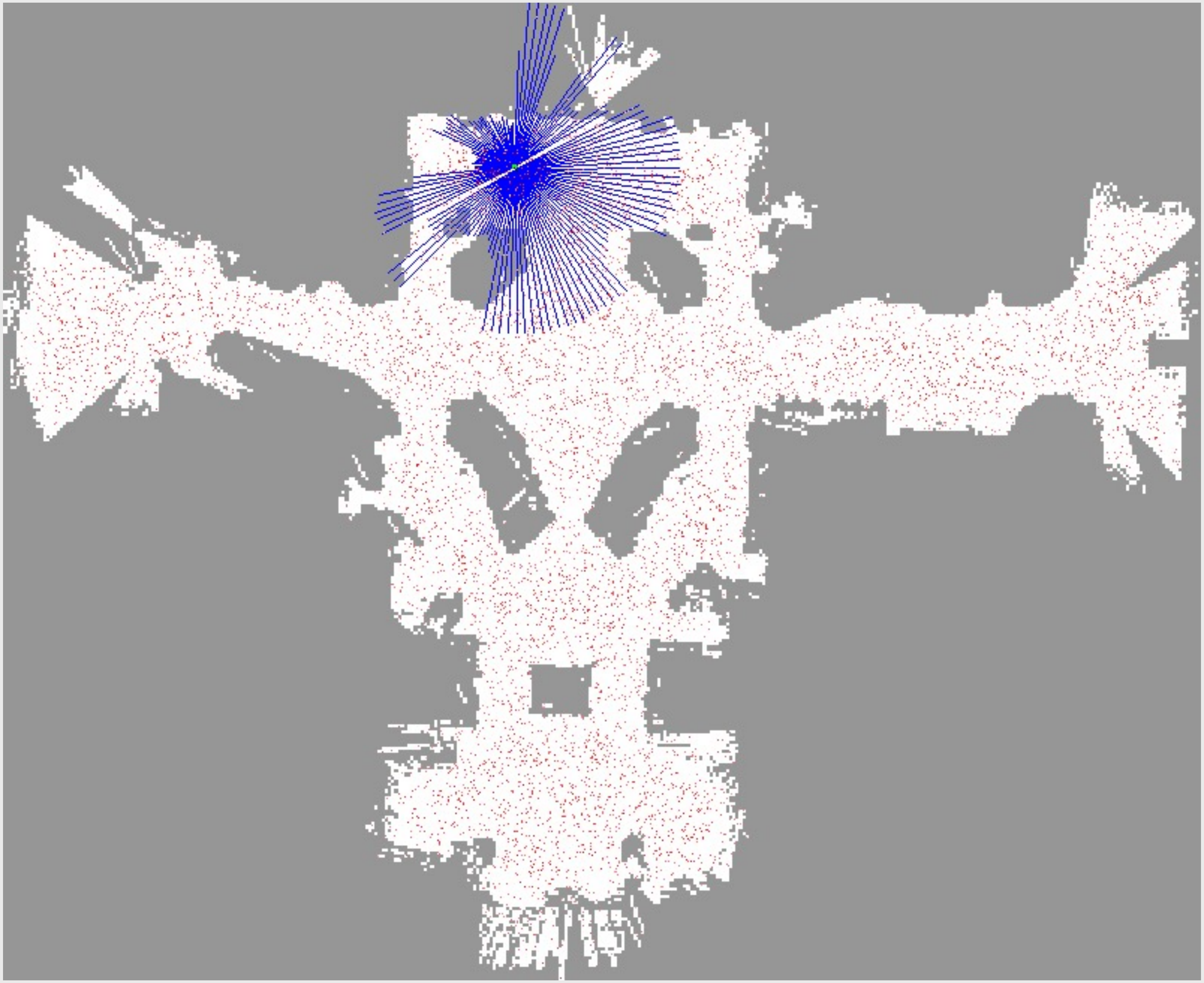


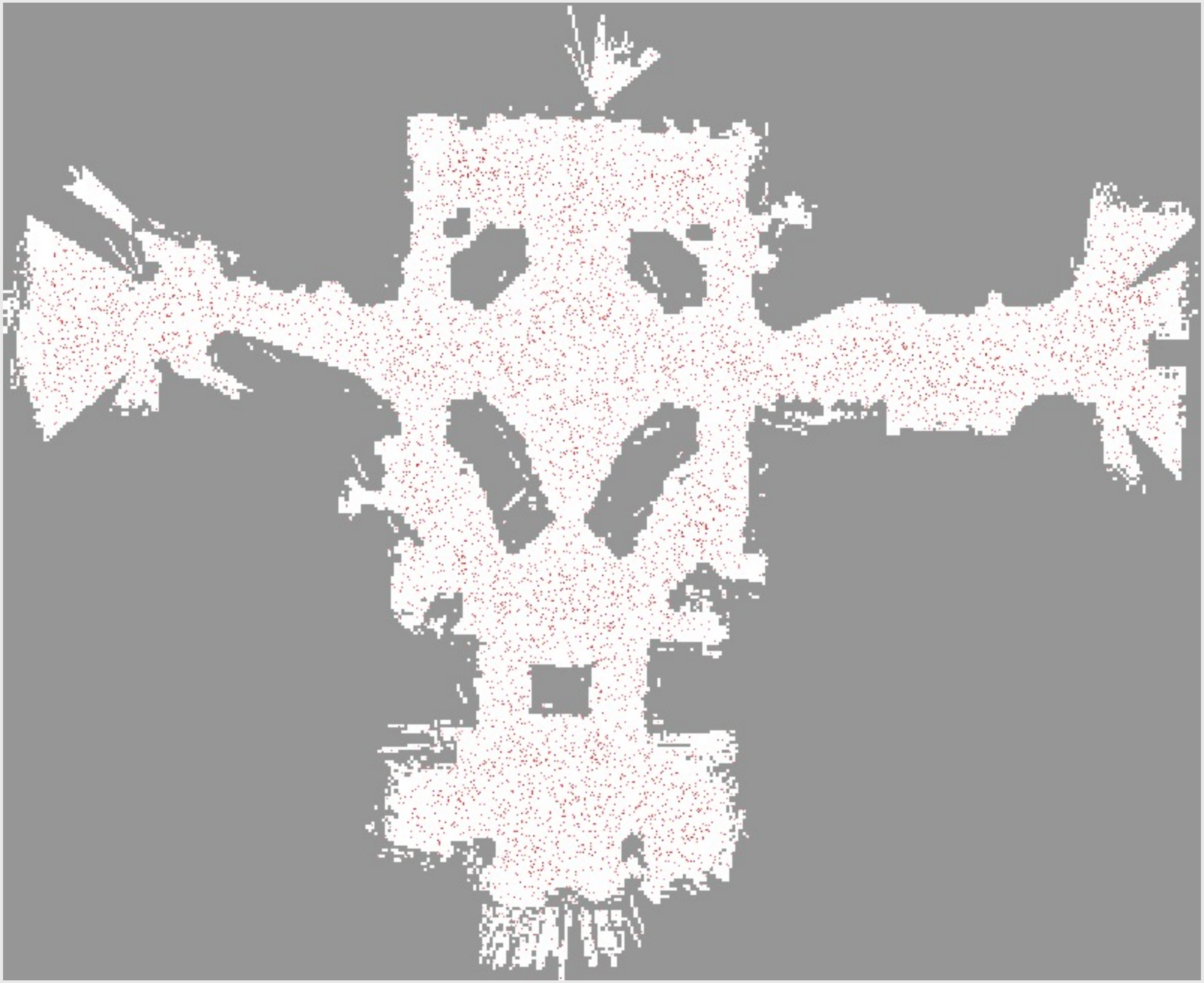
Laser sensor

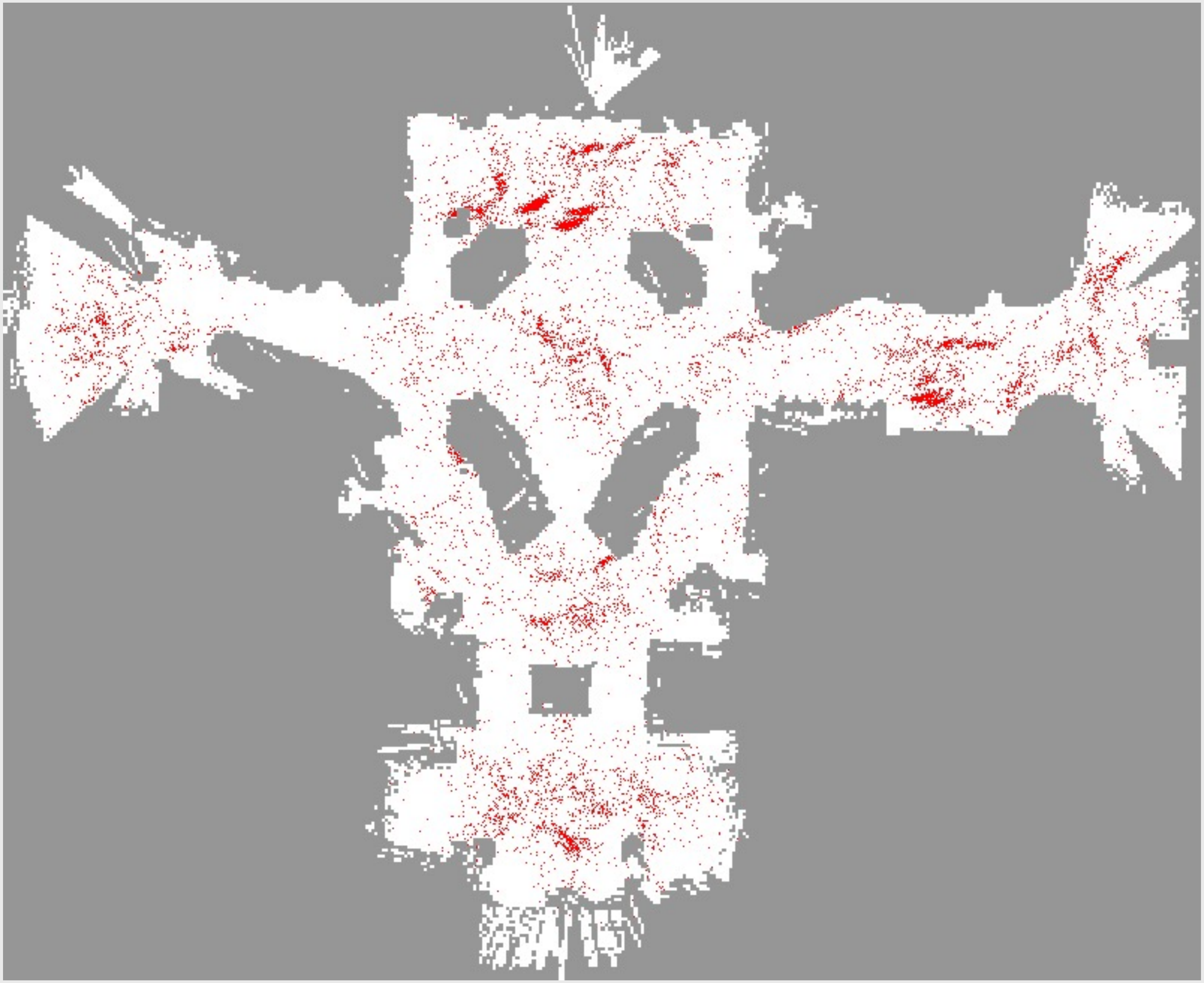


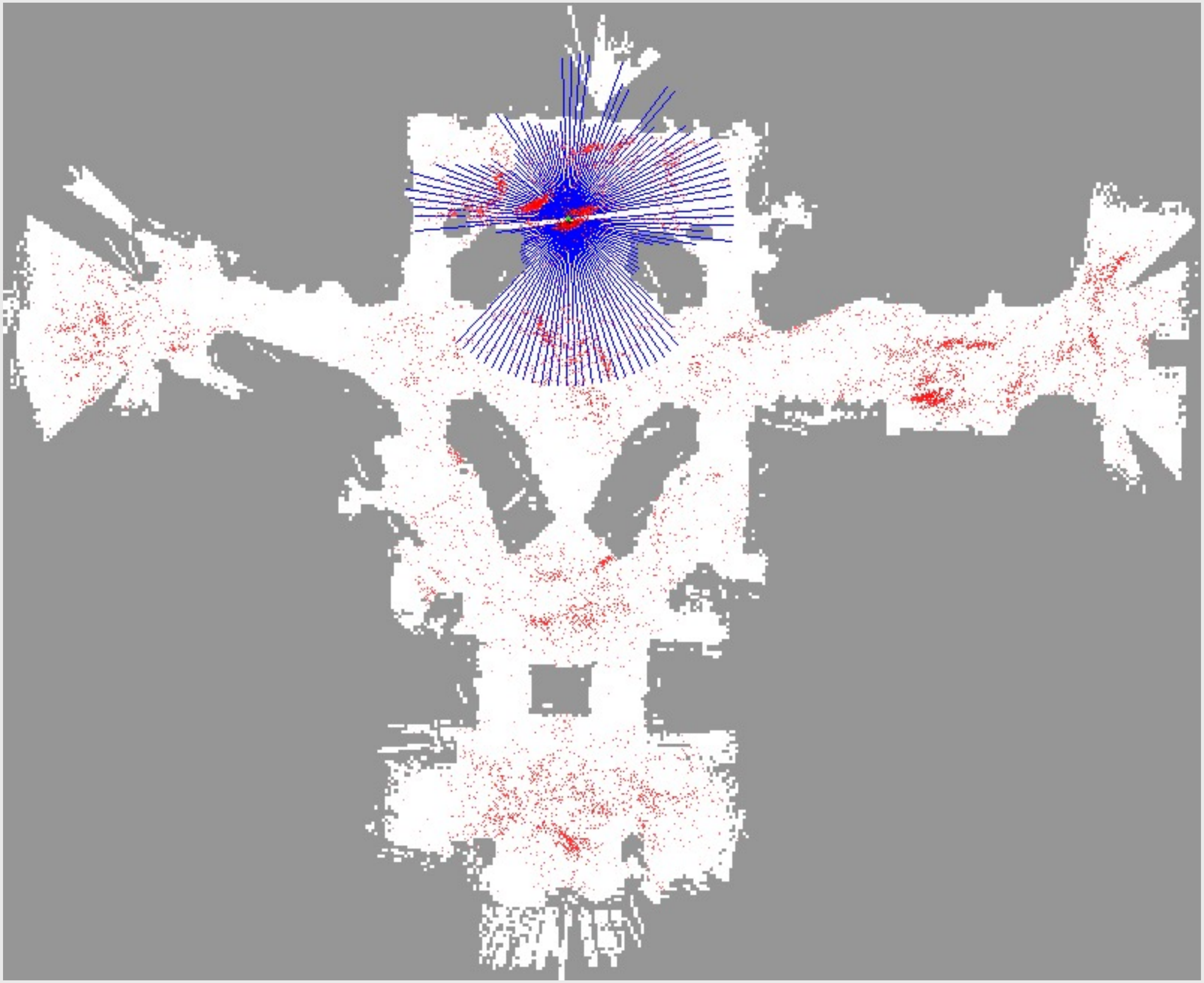
Sonar sensor

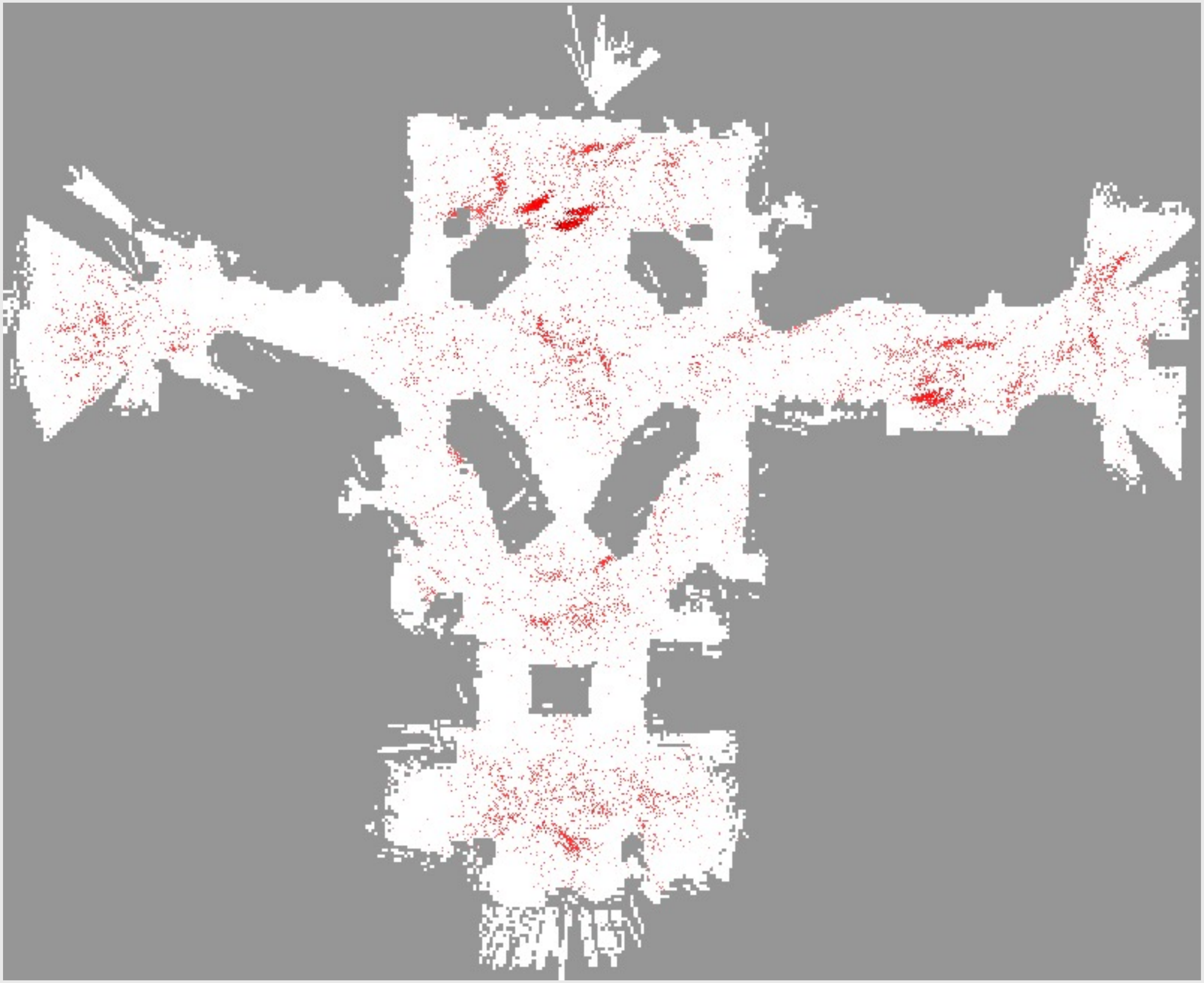


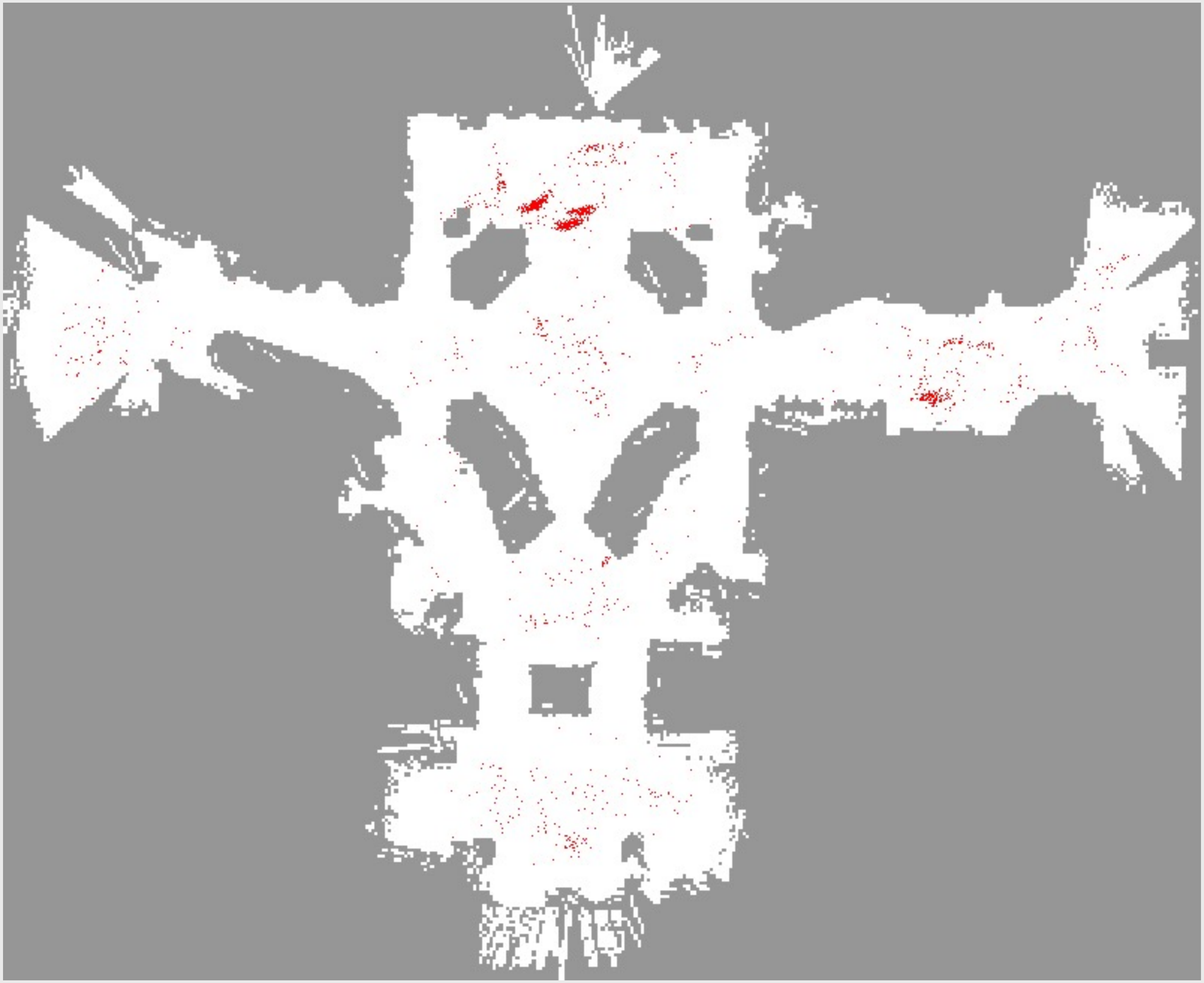


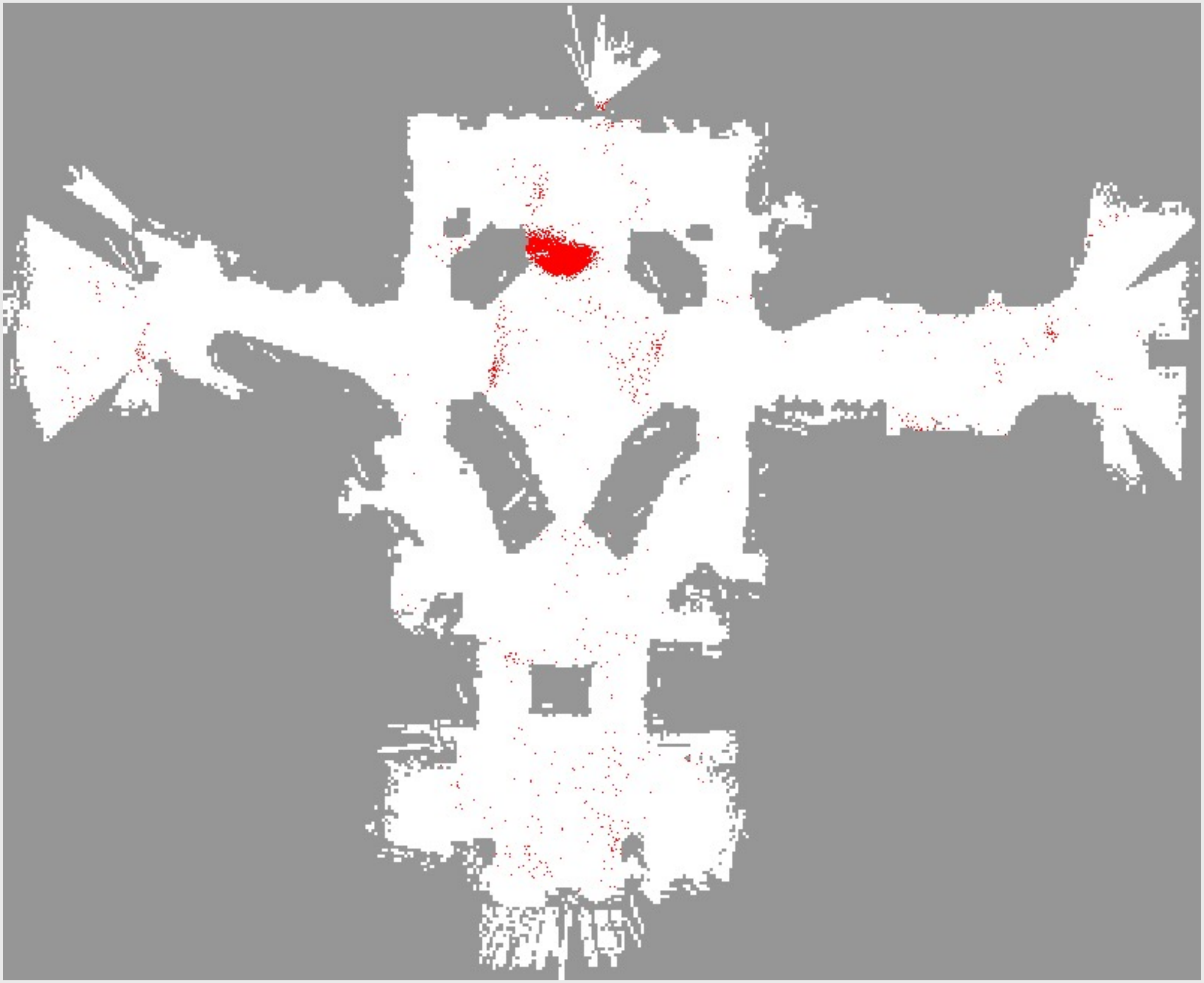


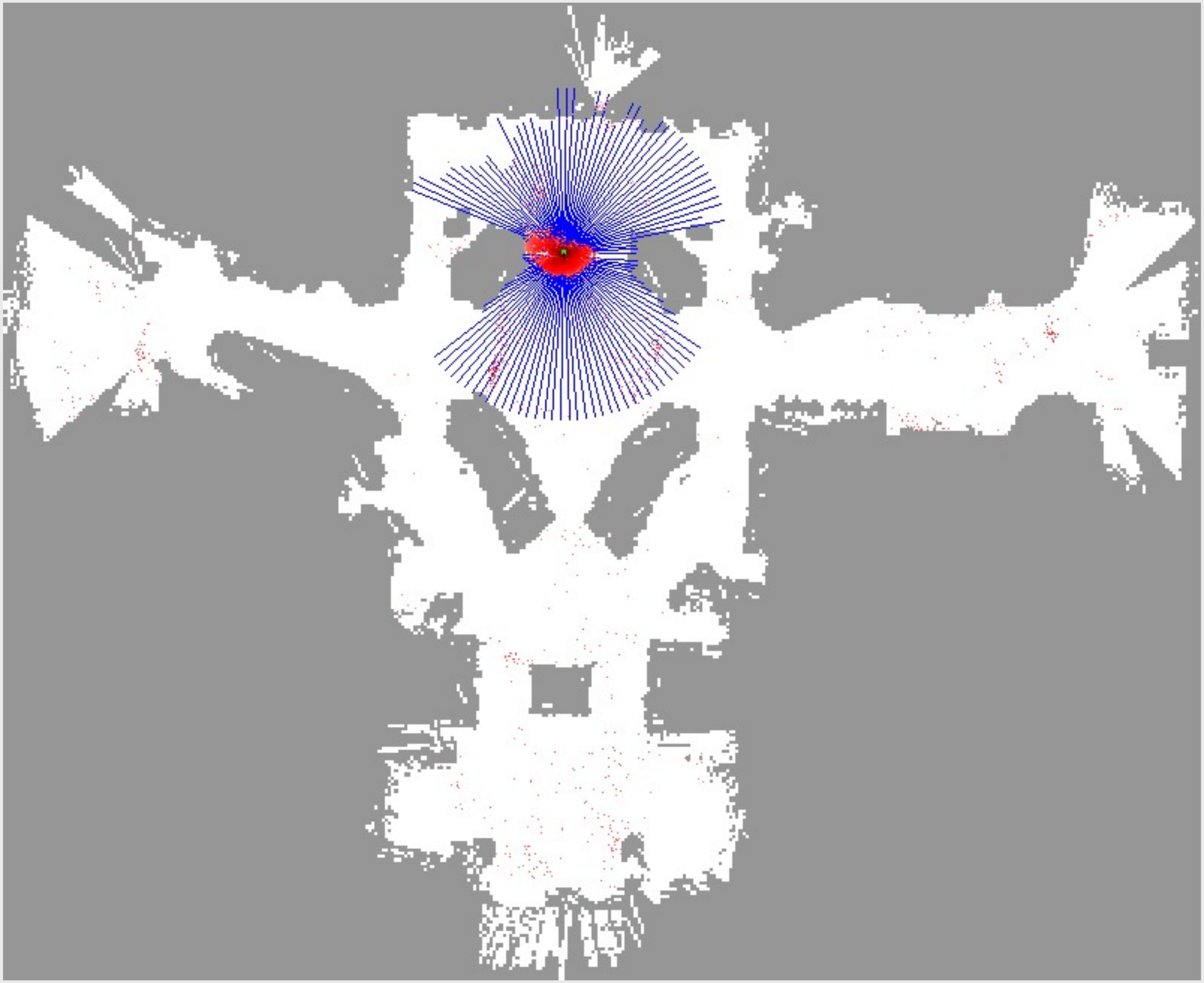


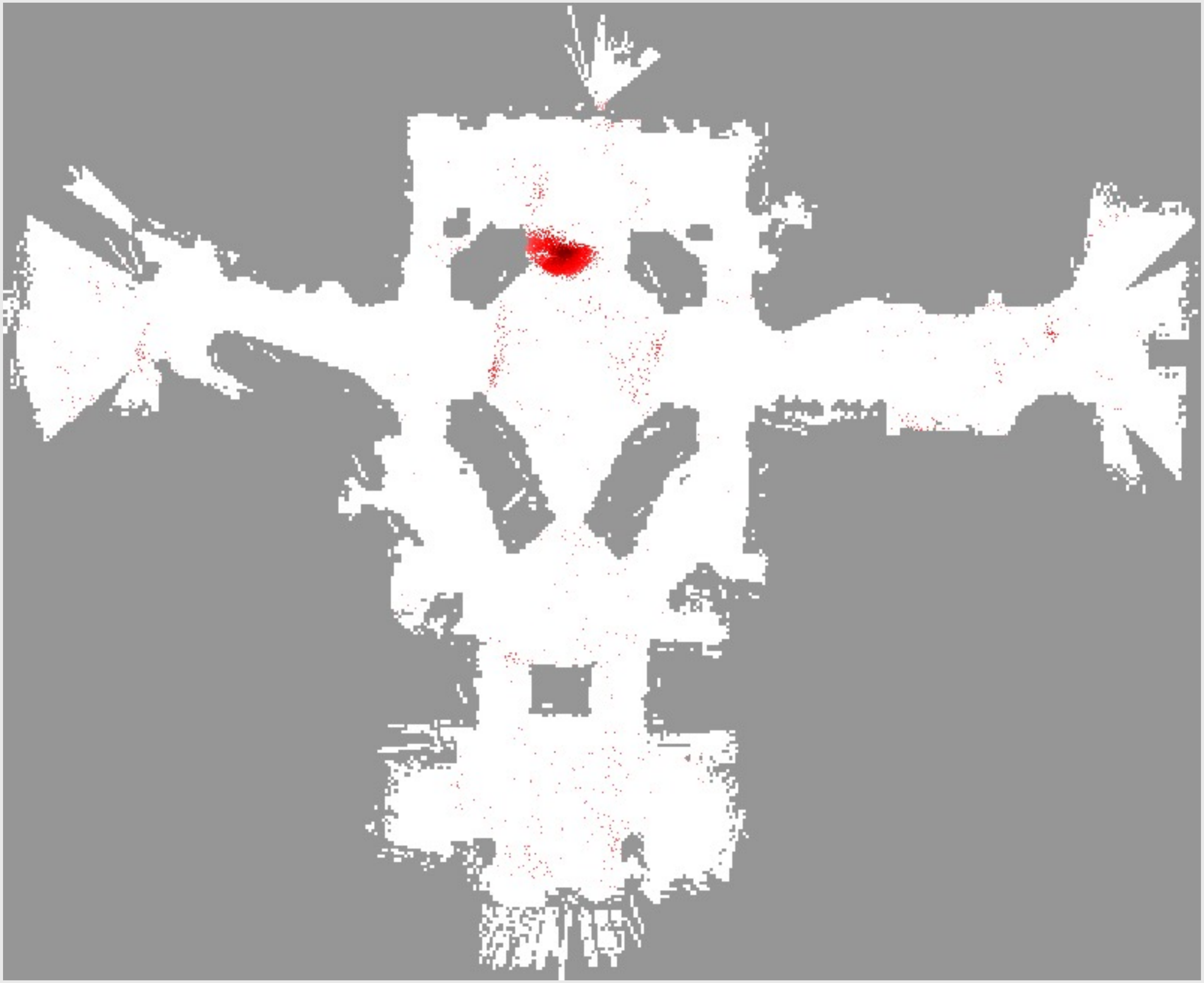


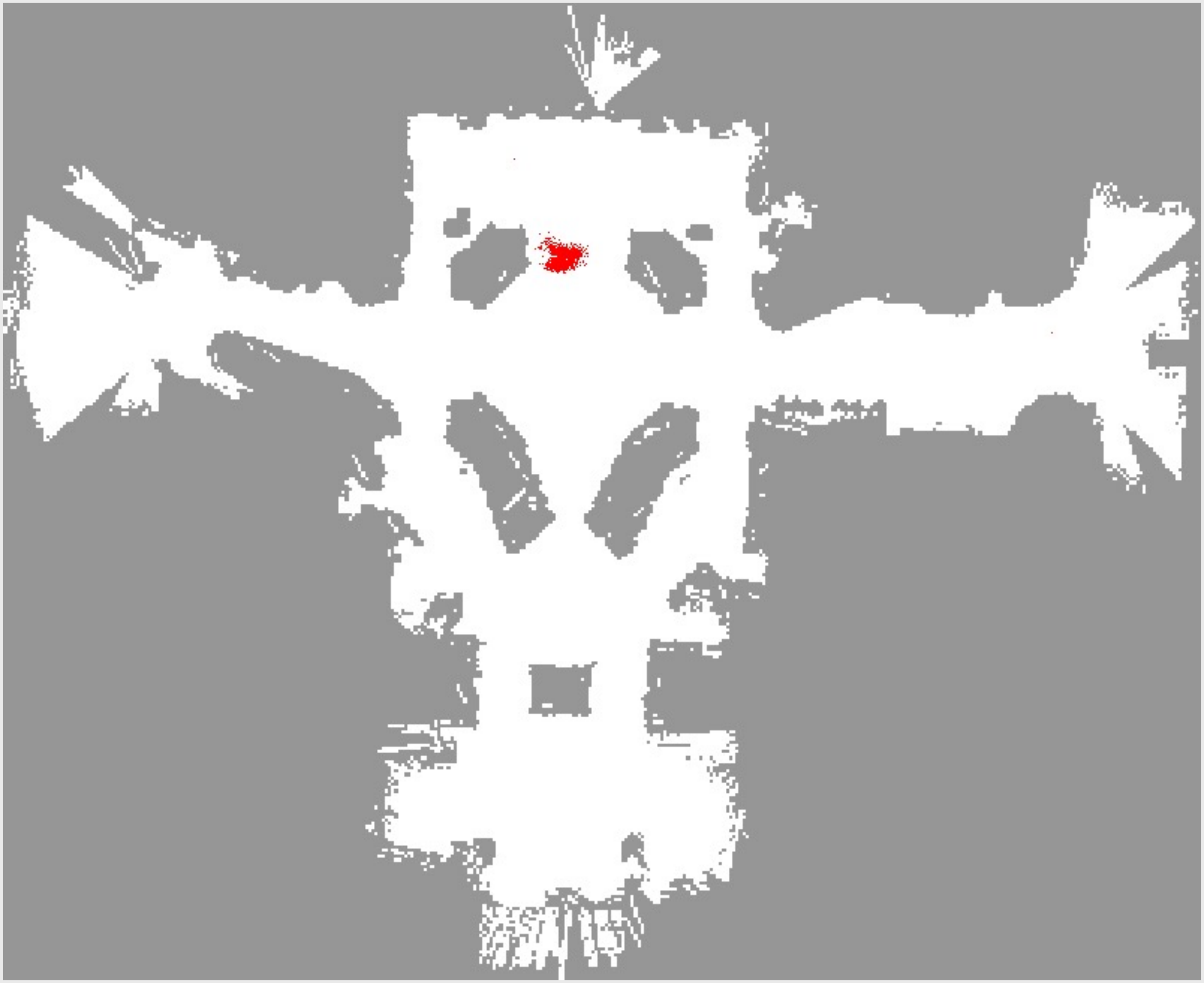


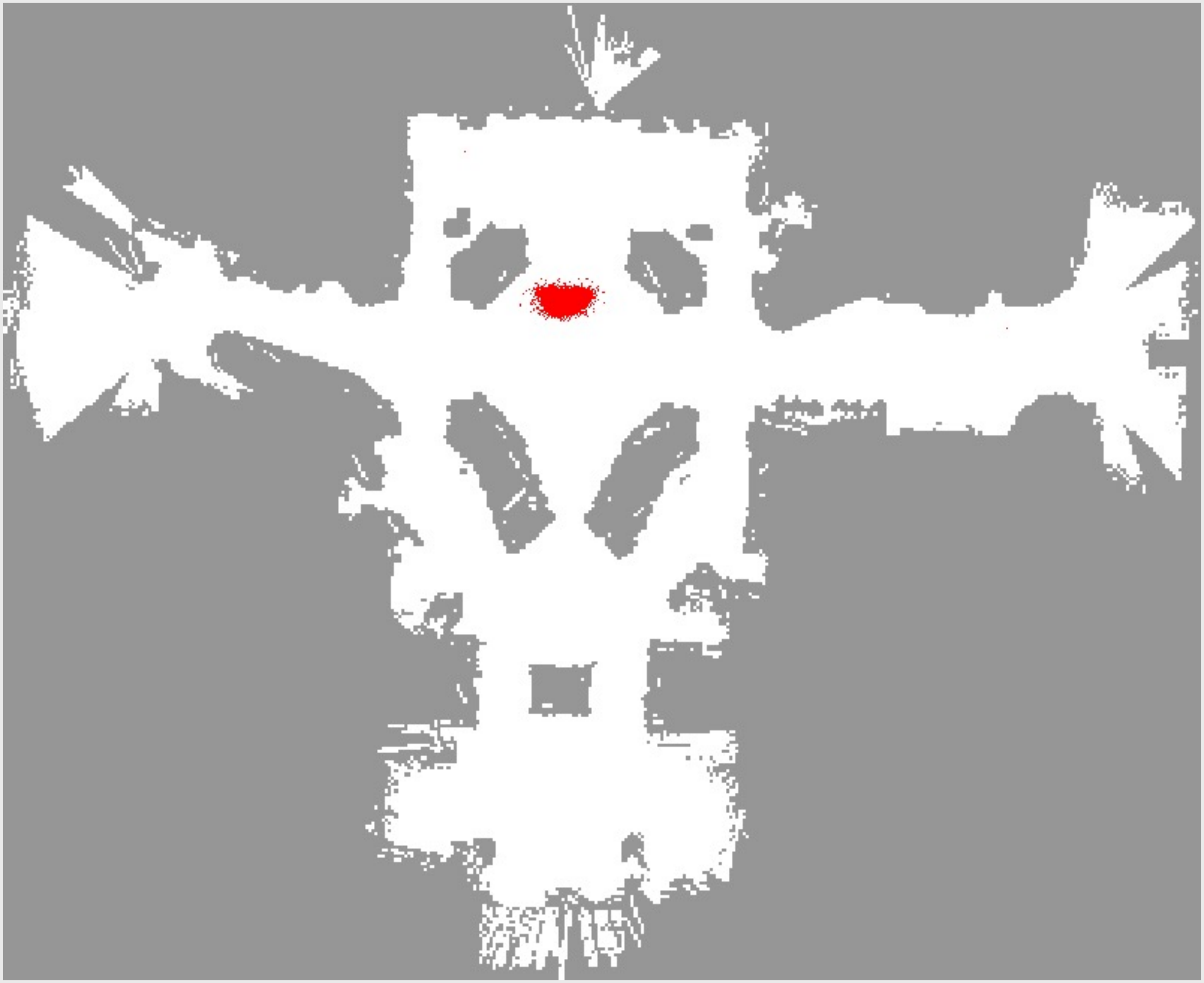


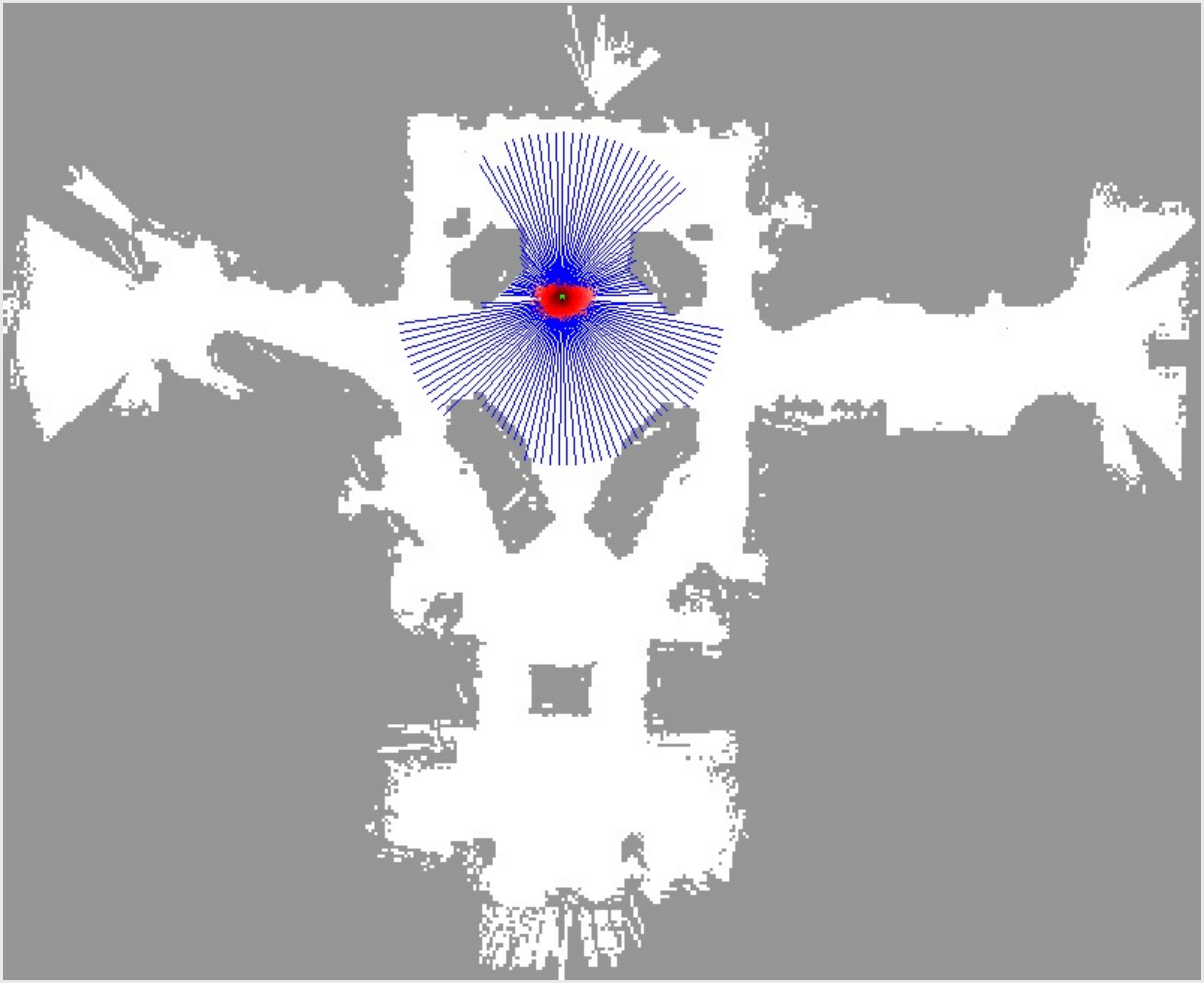


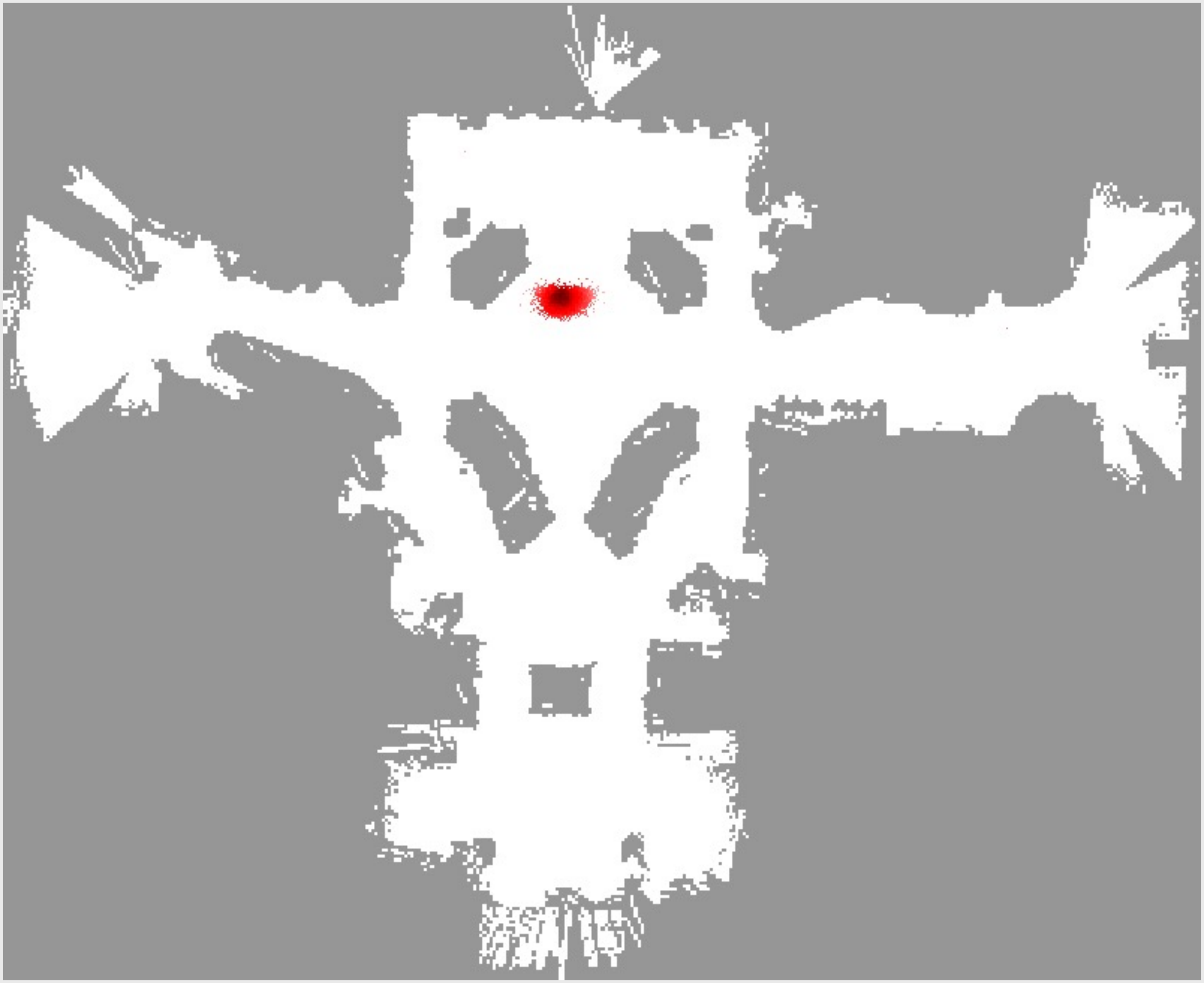


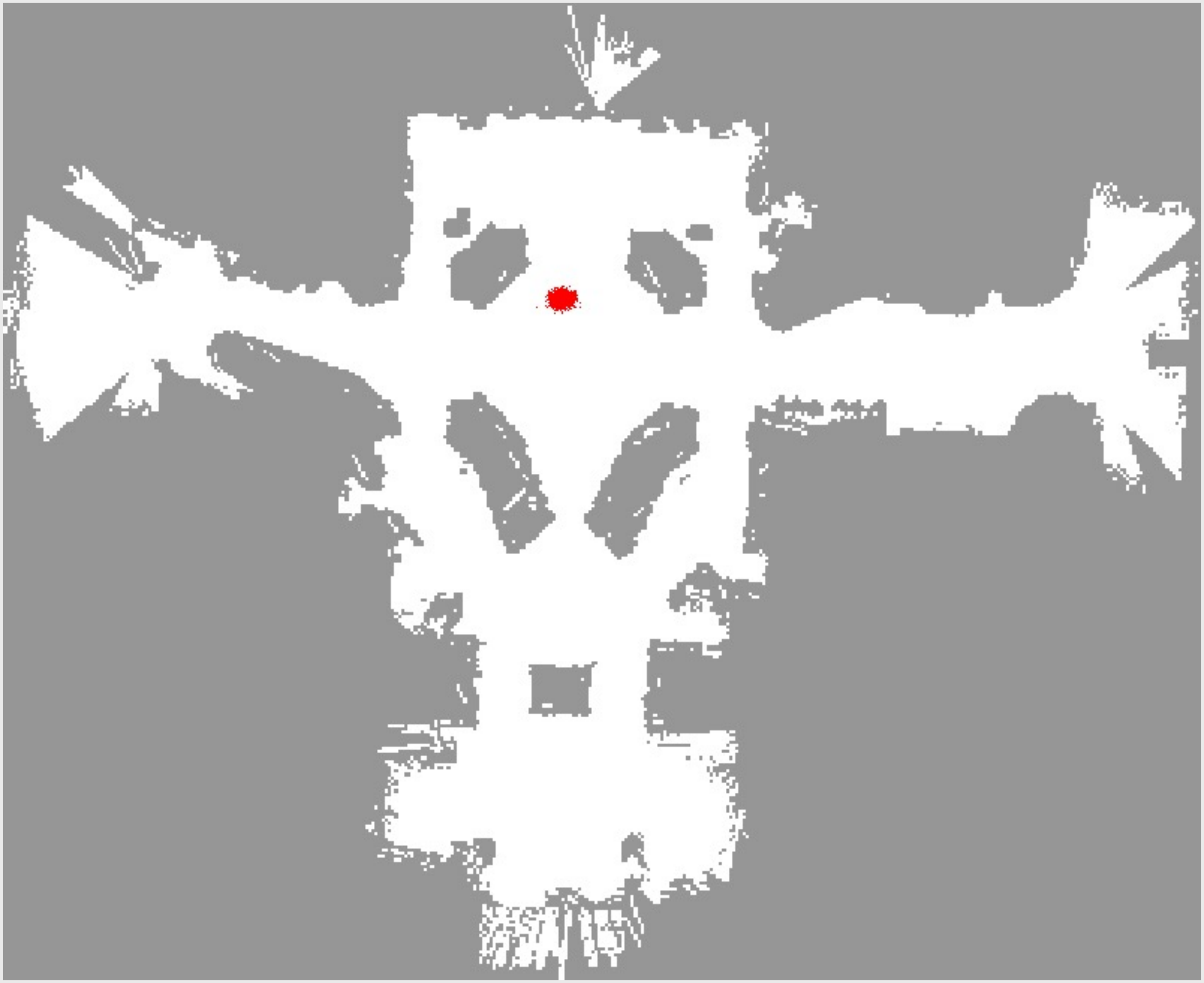


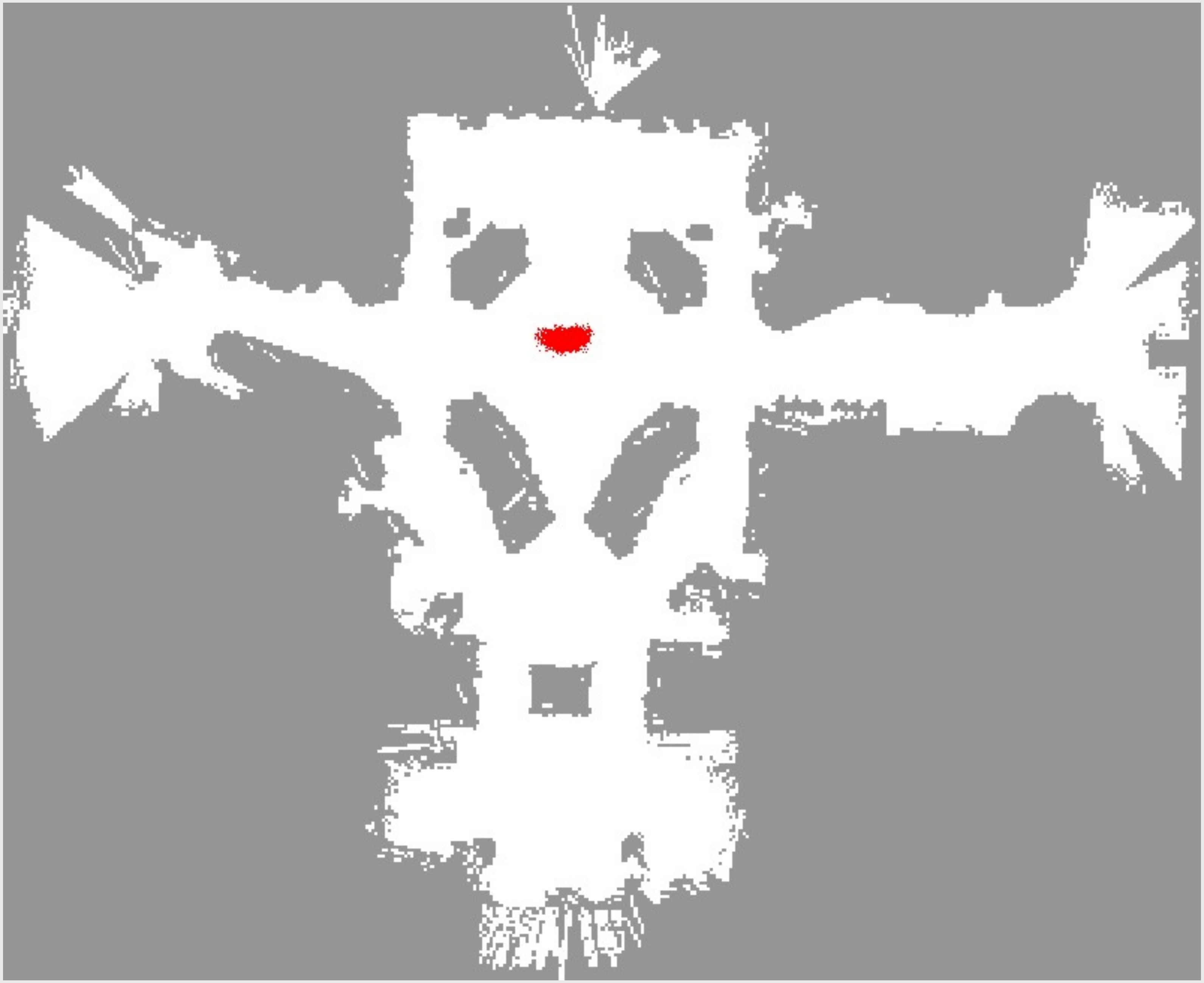


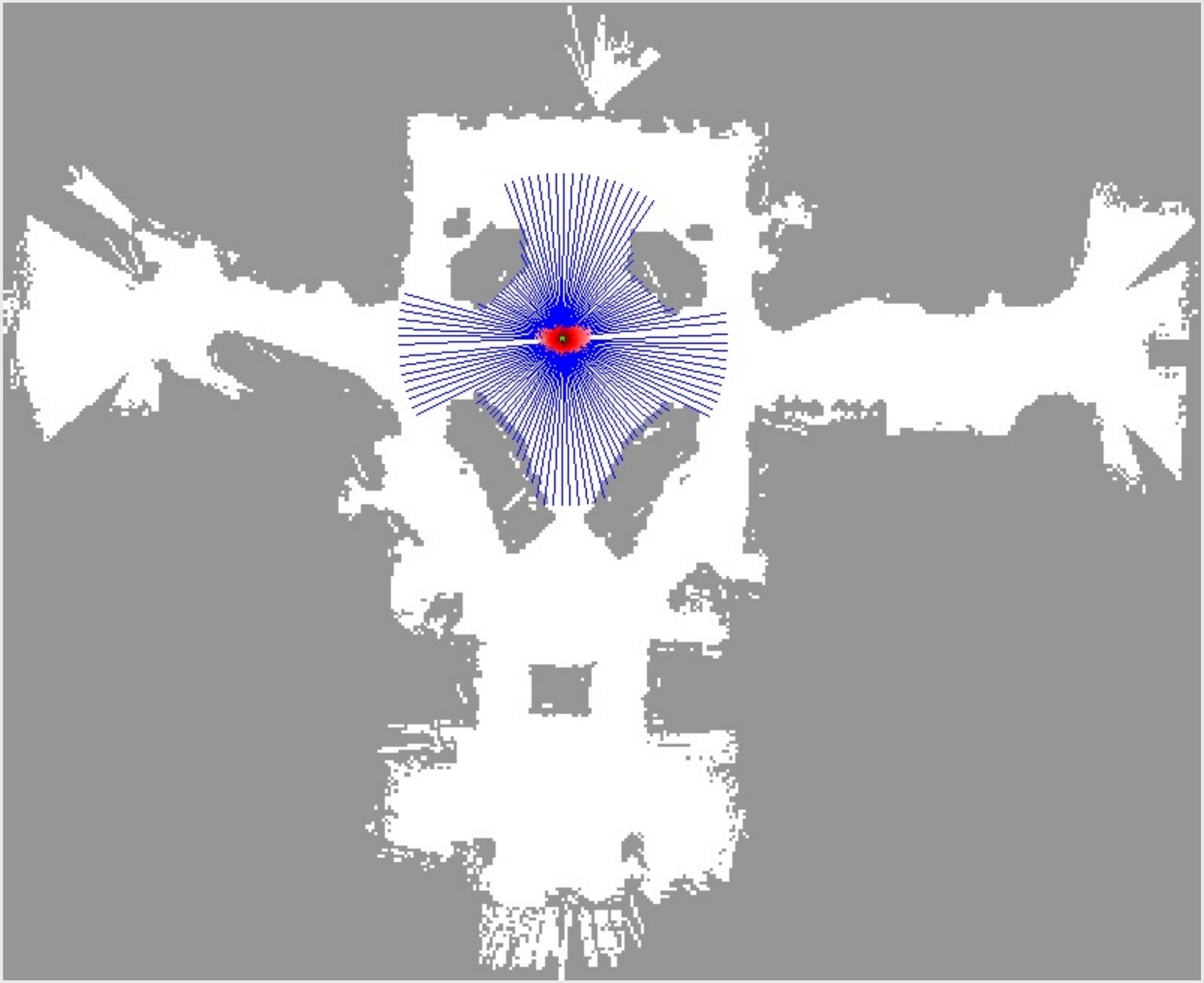


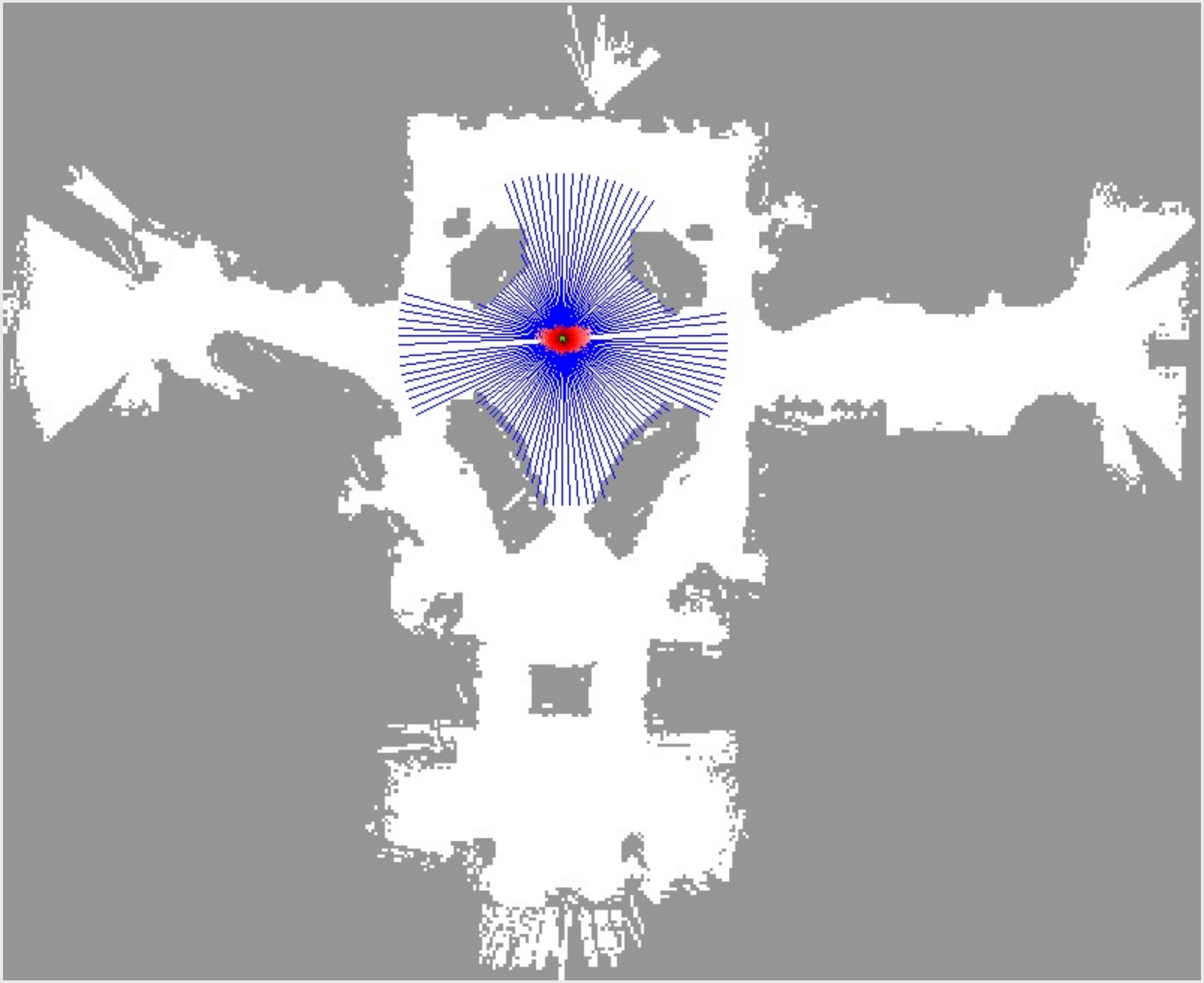




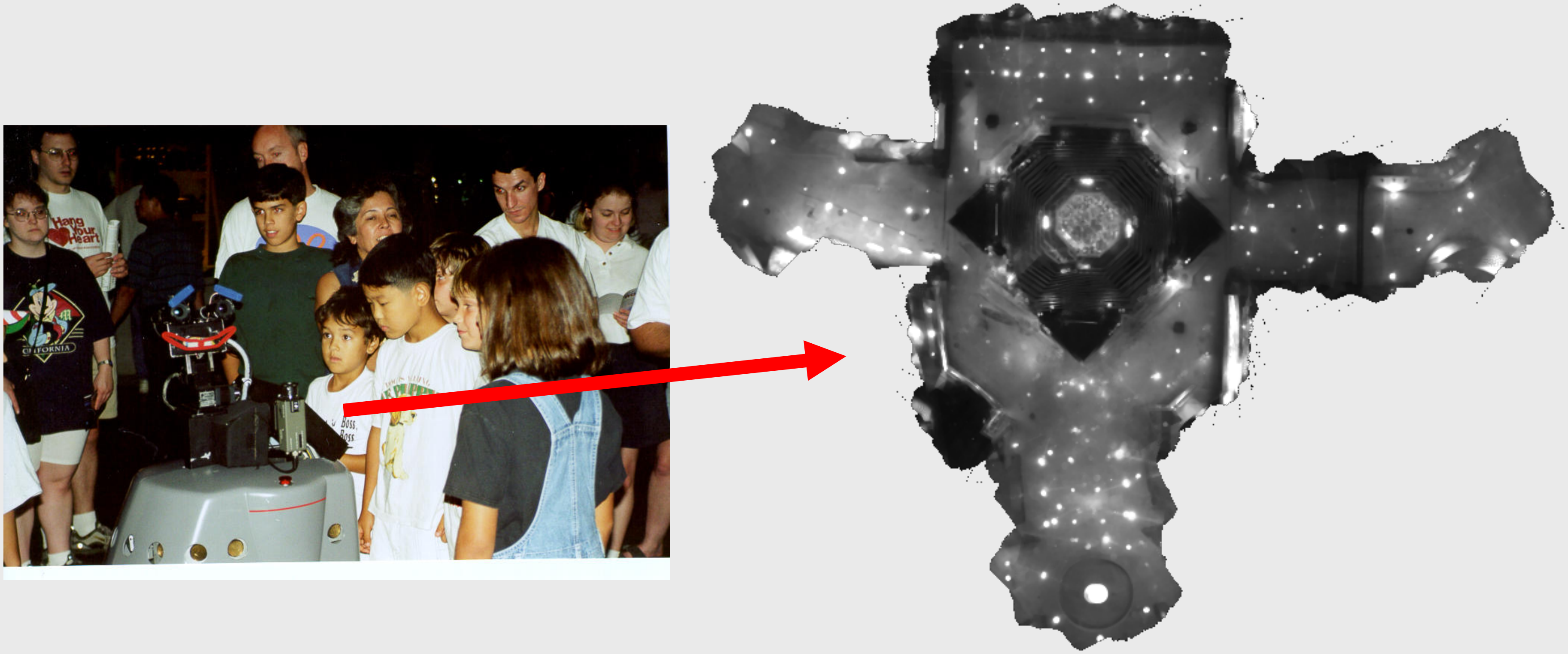






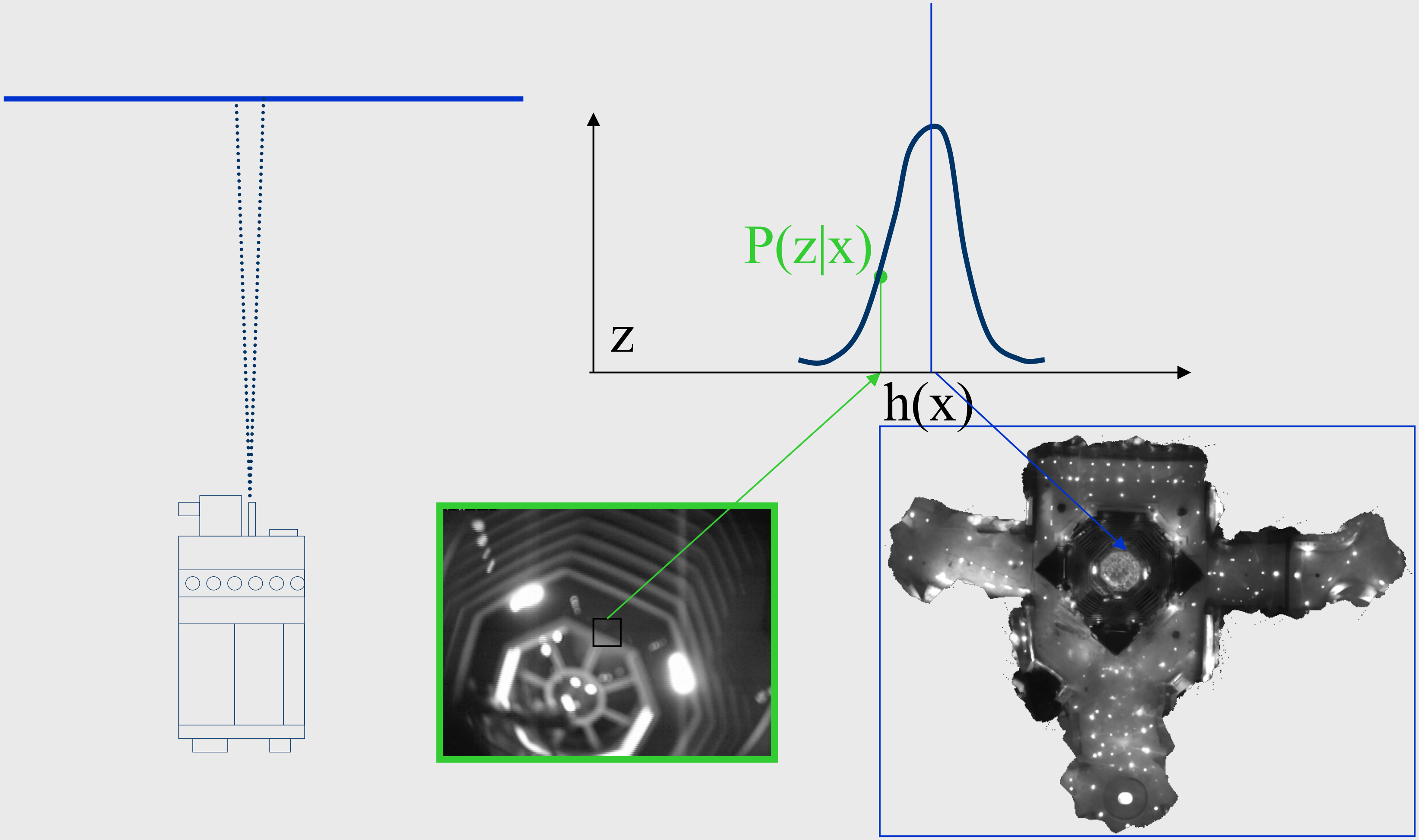


Using Ceiling Maps for Localization



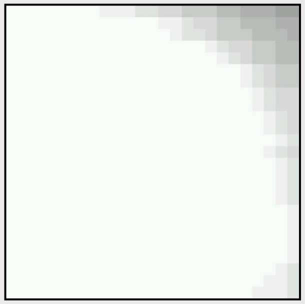
[Dellaert et al. 99]

Vision-based Localization

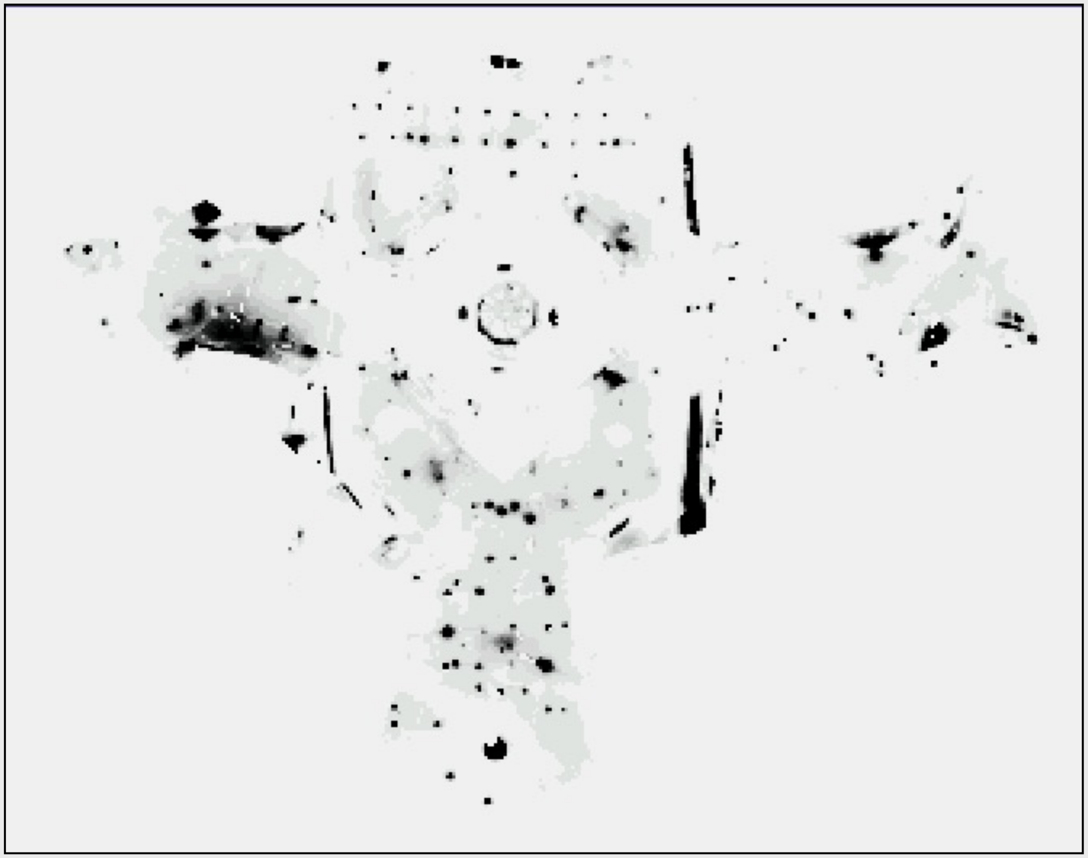


Under a Light

Measurement z:

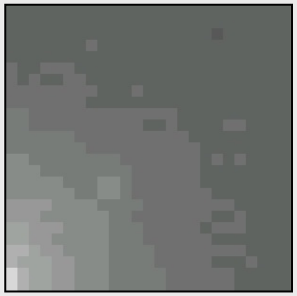


***P(z|x)*:**

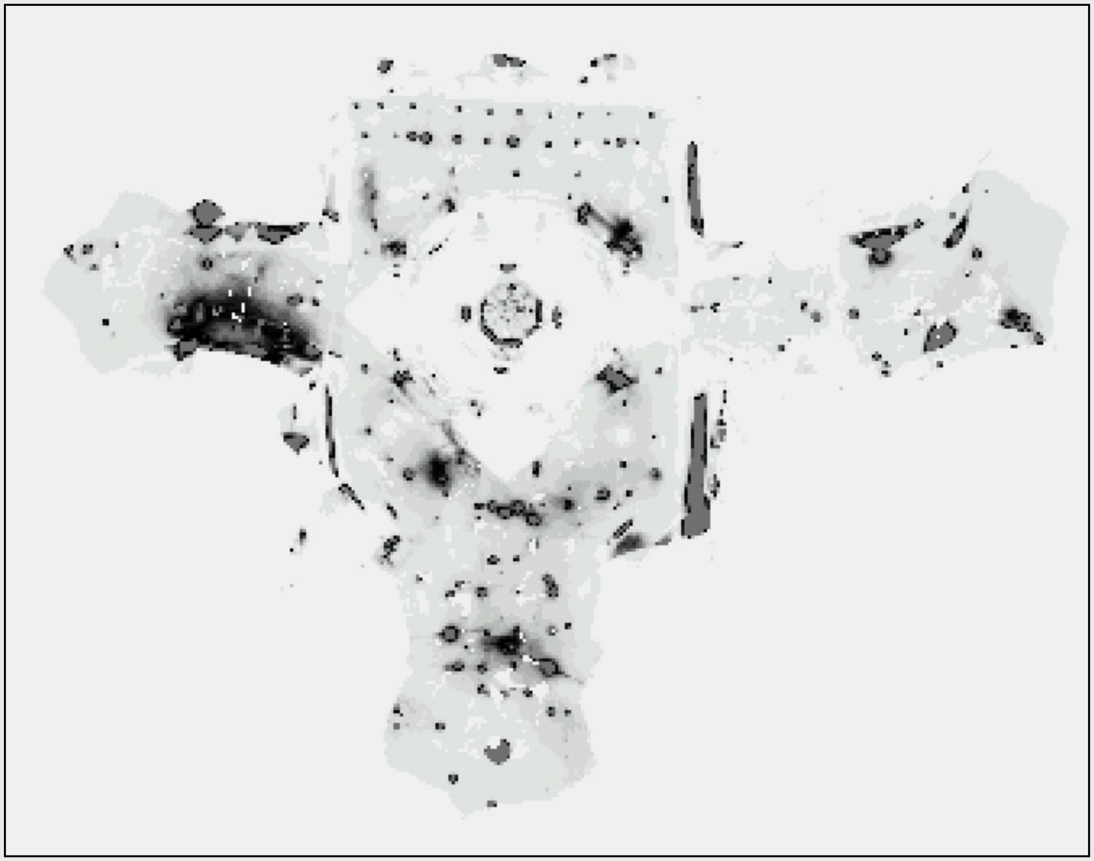


Next to a Light

Measurement z:



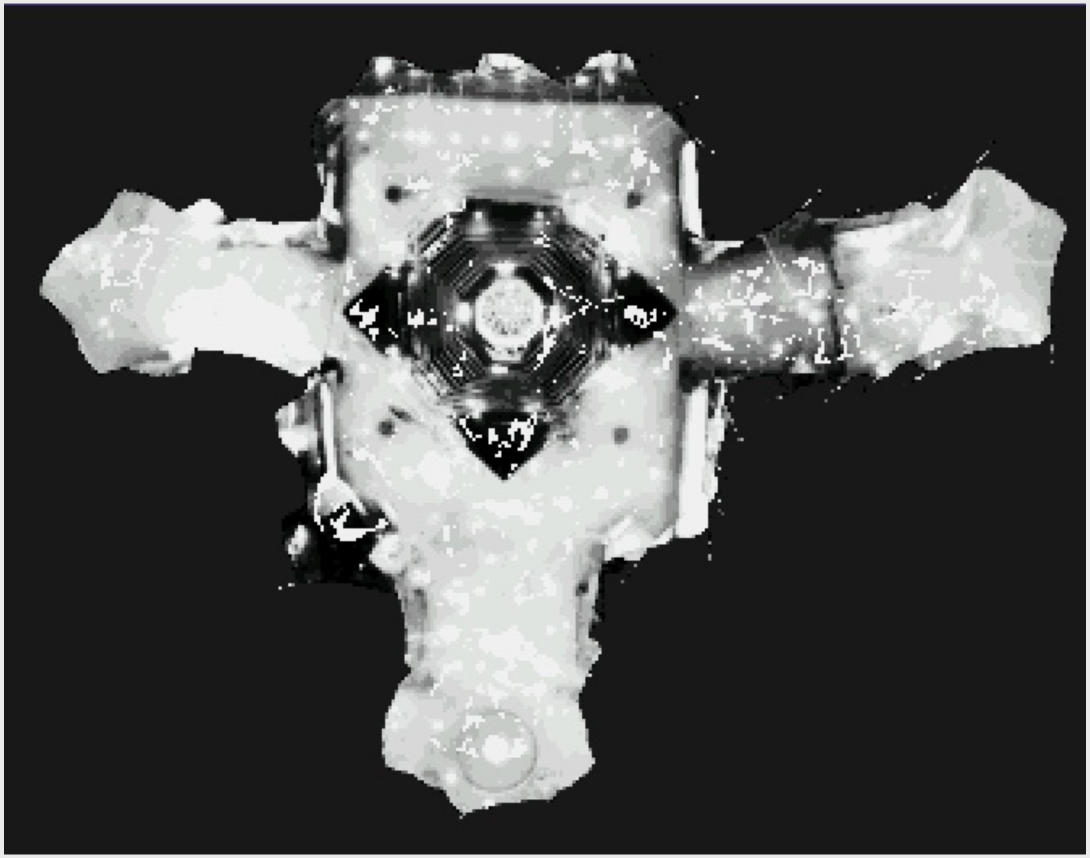
***P(z|x)*:**



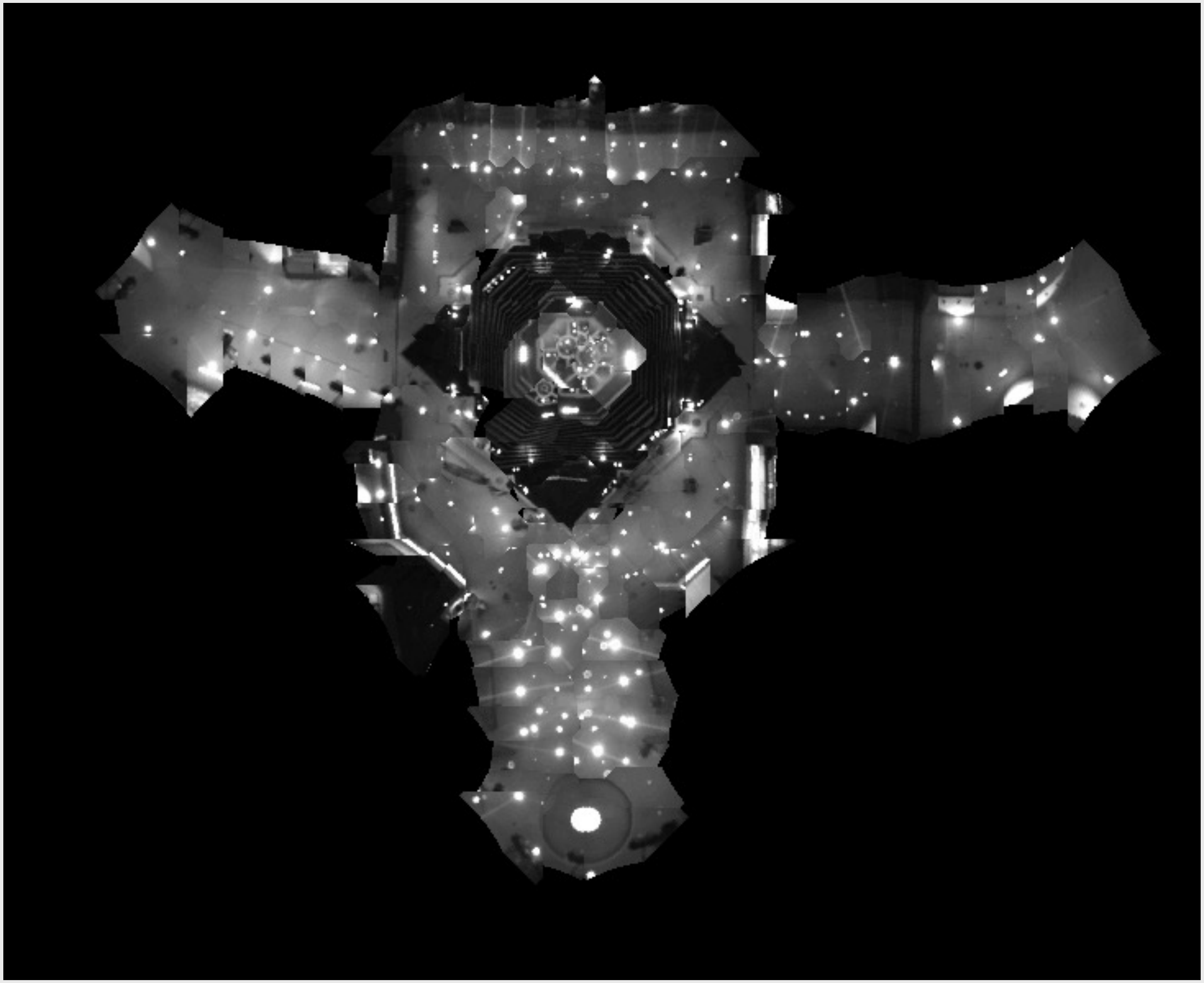
Elsewhere

Measurement z:

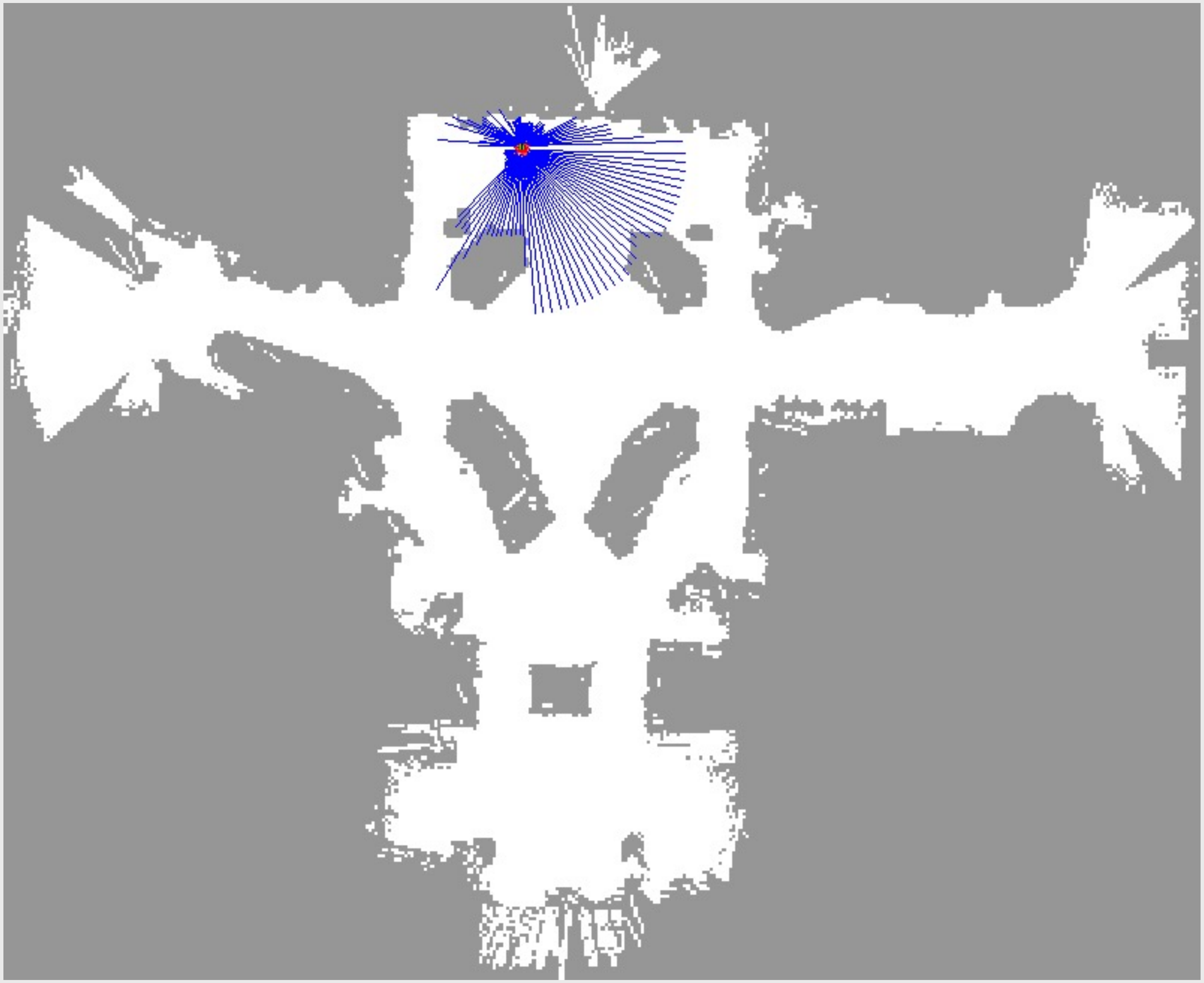
***P(z|x)*:**



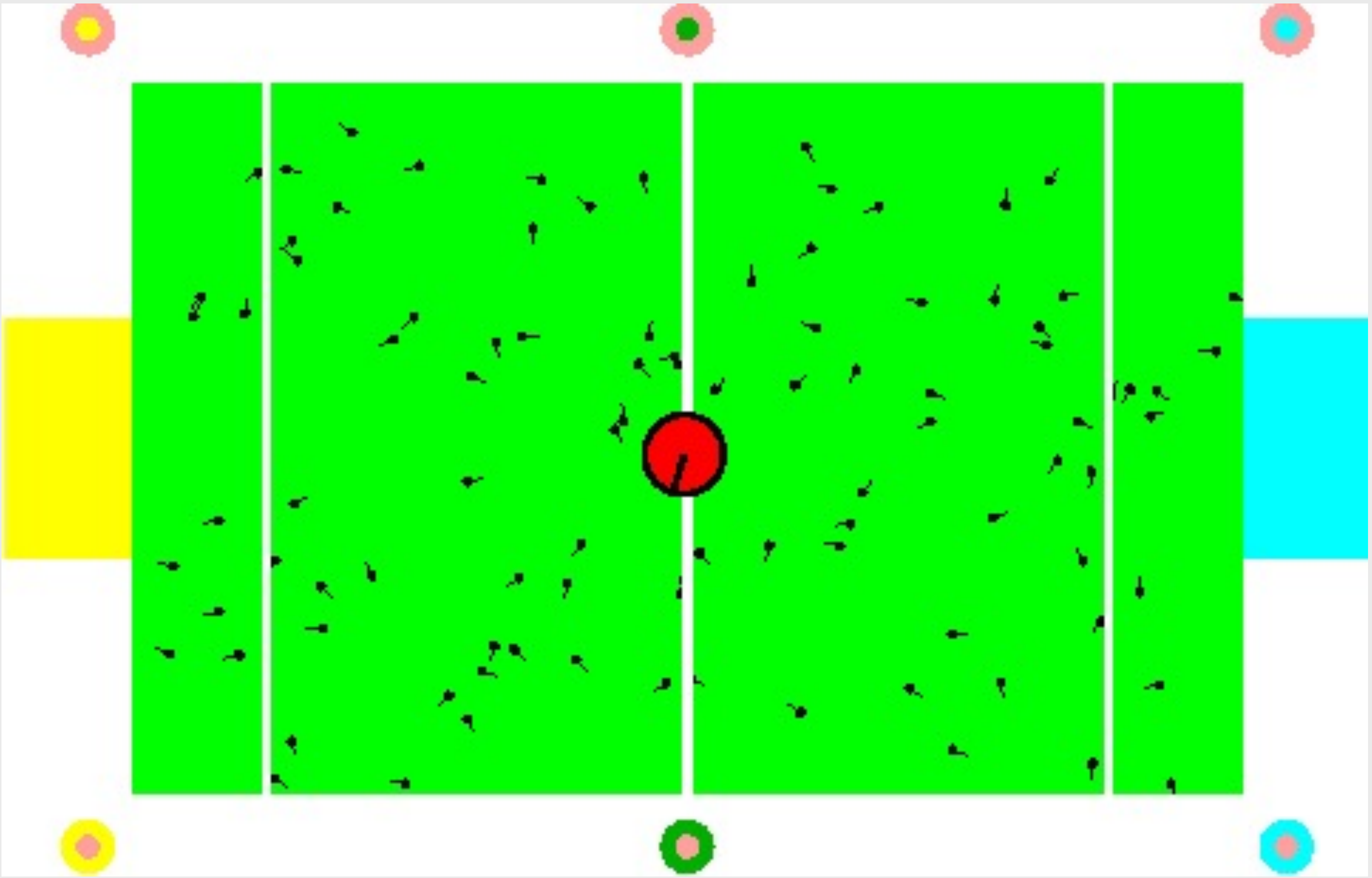
Global Localization Using Vision



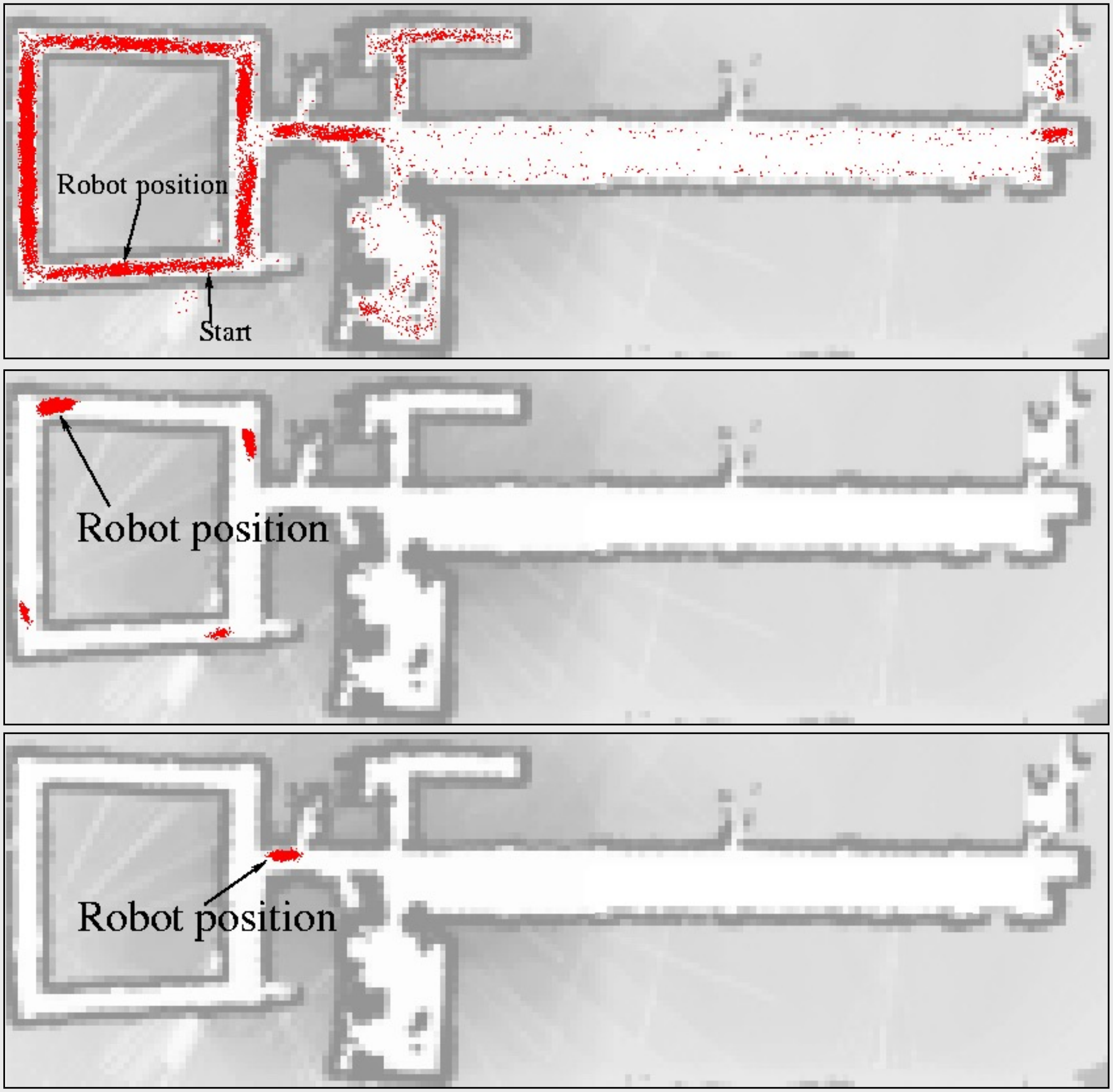
Recovery from Failure



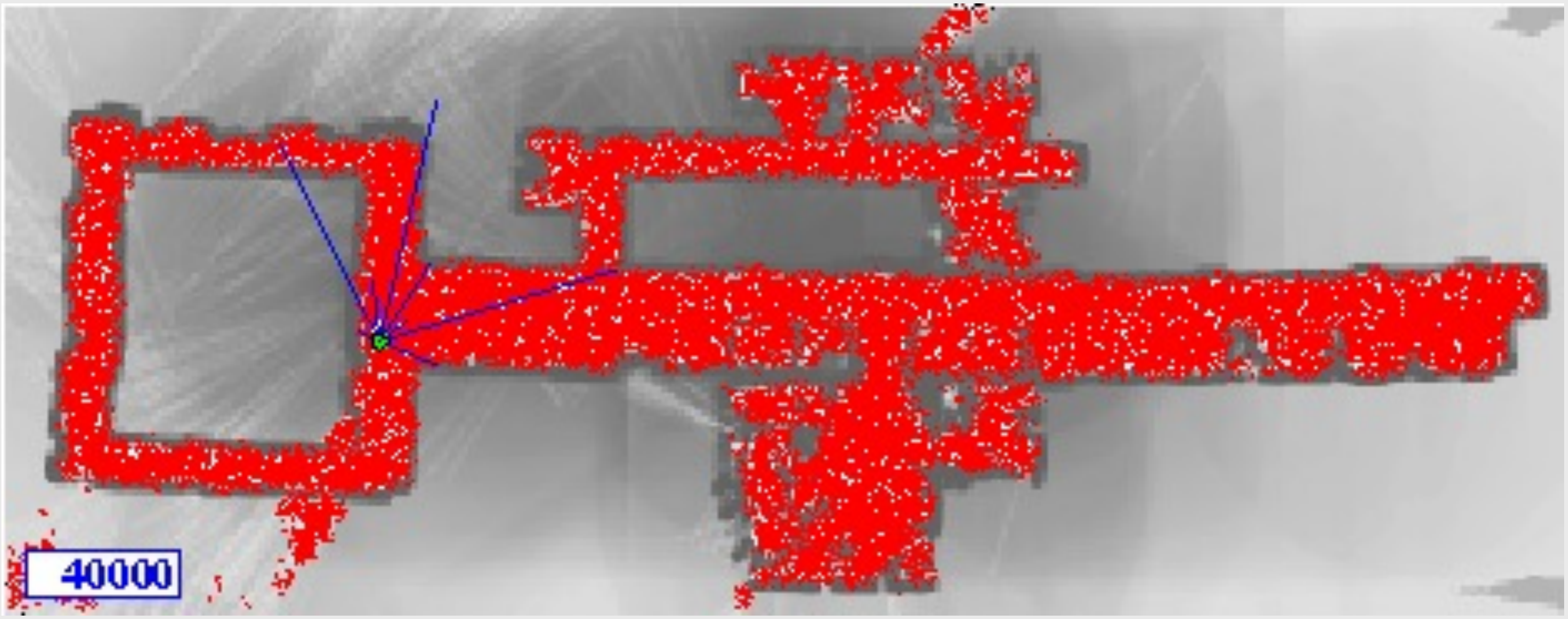
Localization for AIBO robots



Adaptive Sampling

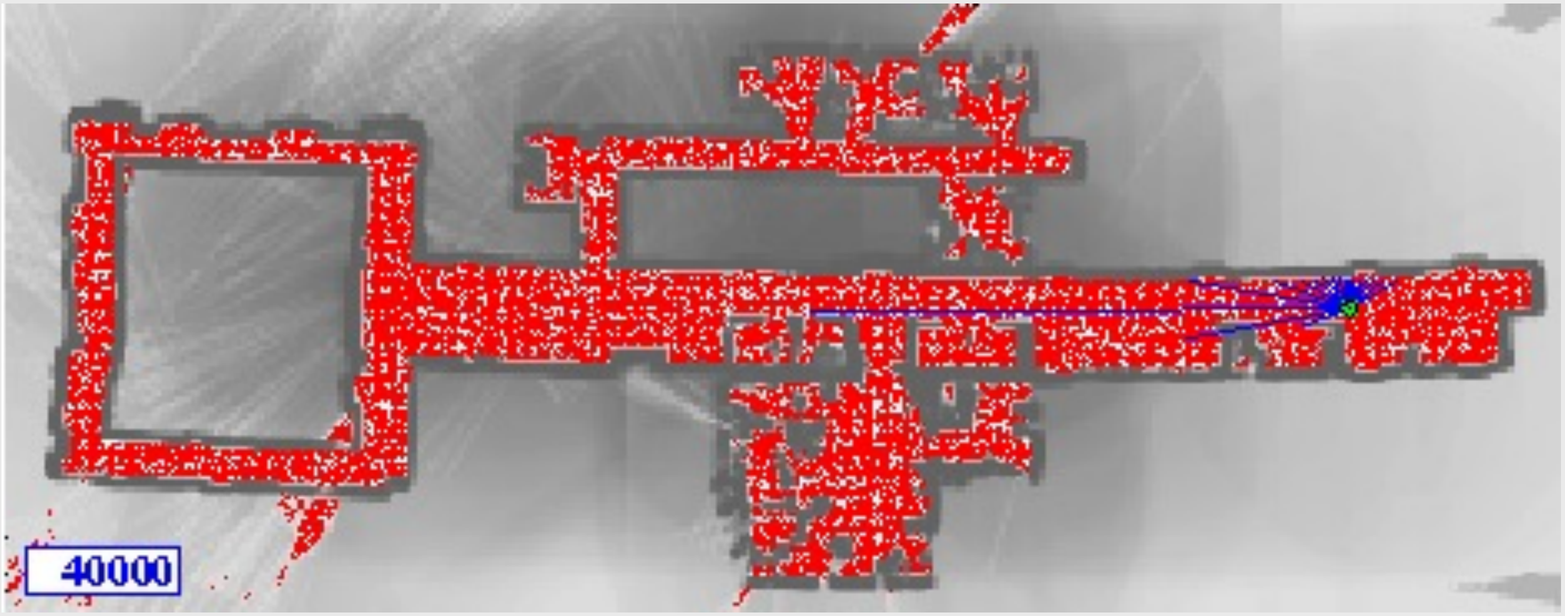


KLD-Sampling Sonar

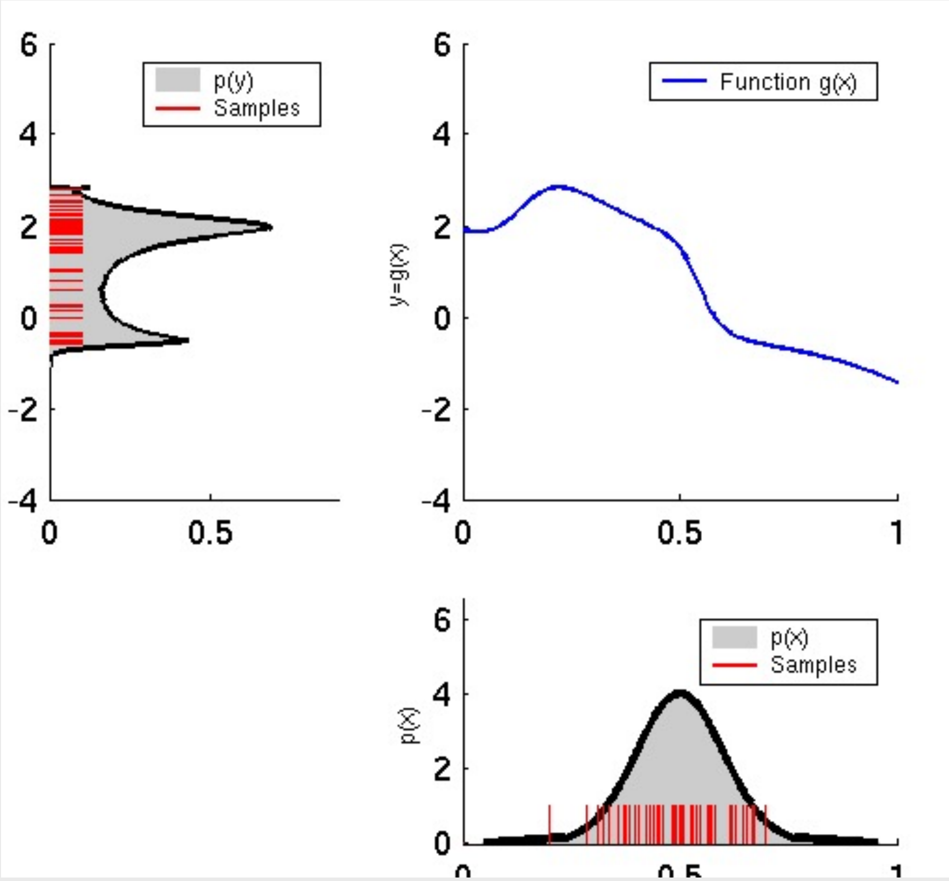


Adapt number of particles on the fly based on statistical approximation measure

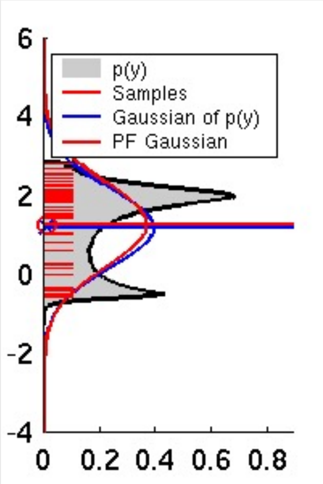
KLD-Sampling Laser

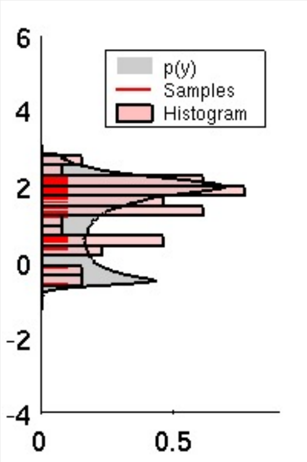


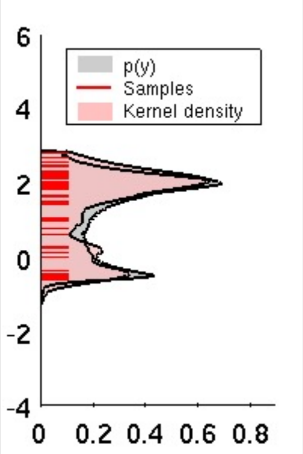
Particle Filter Projection

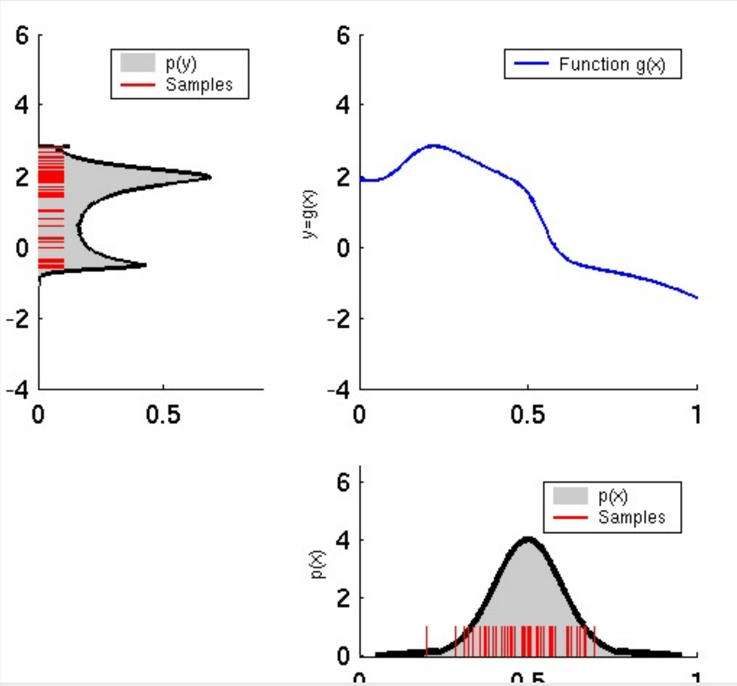


Density Extraction

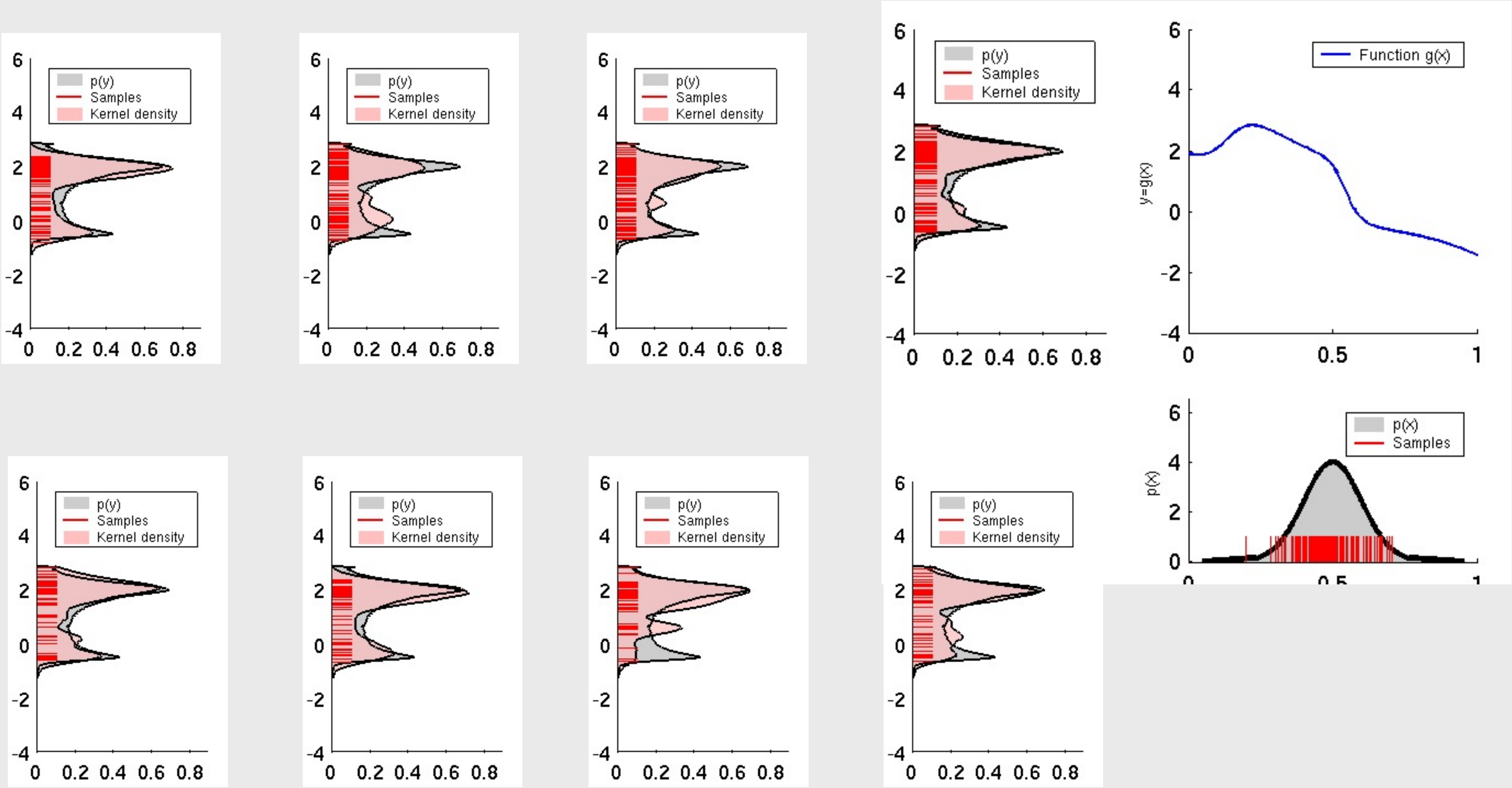








Sampling Variance

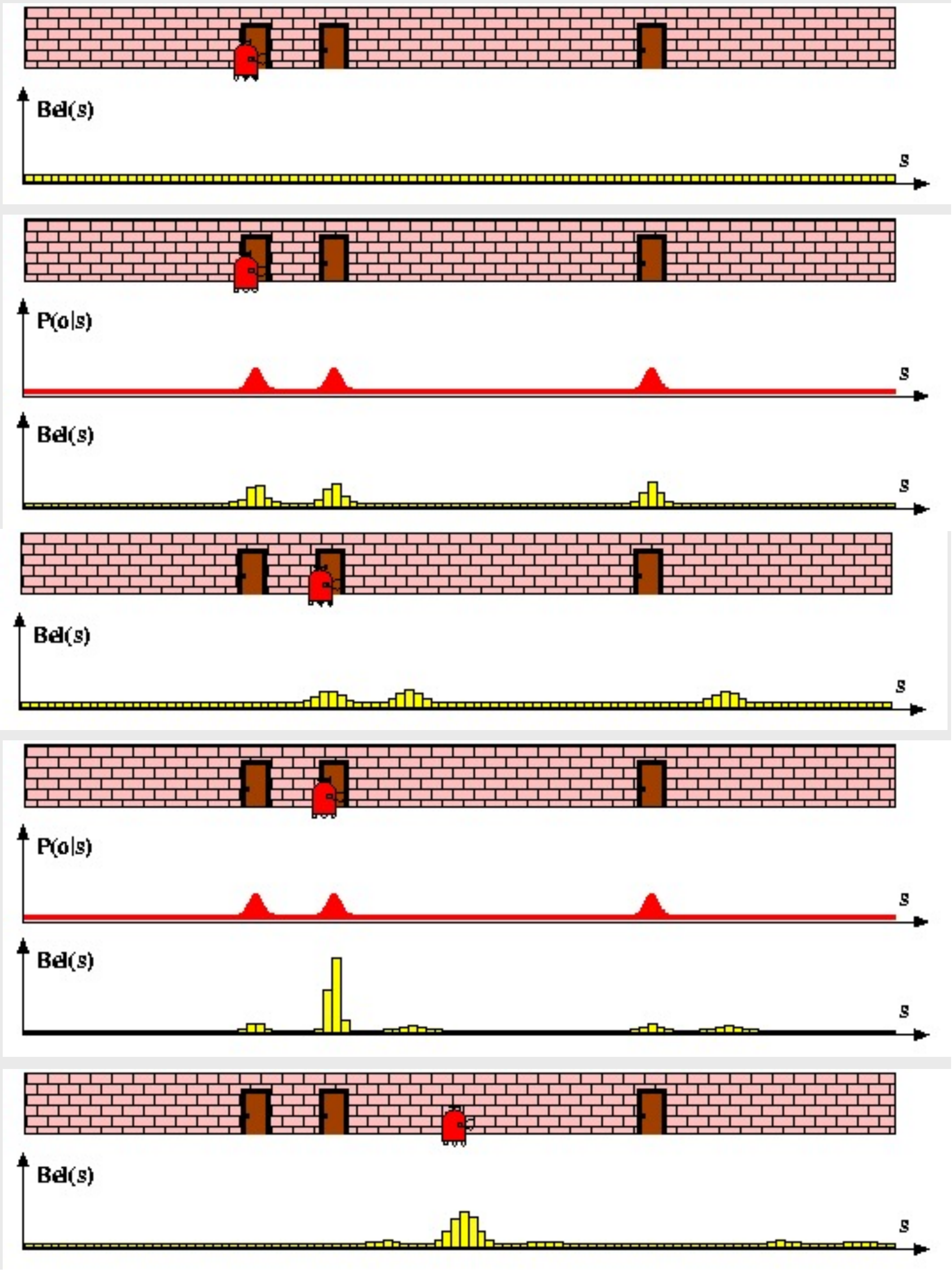


CSE-571
Robotics

Bayes Filter Implementations

Discrete filters

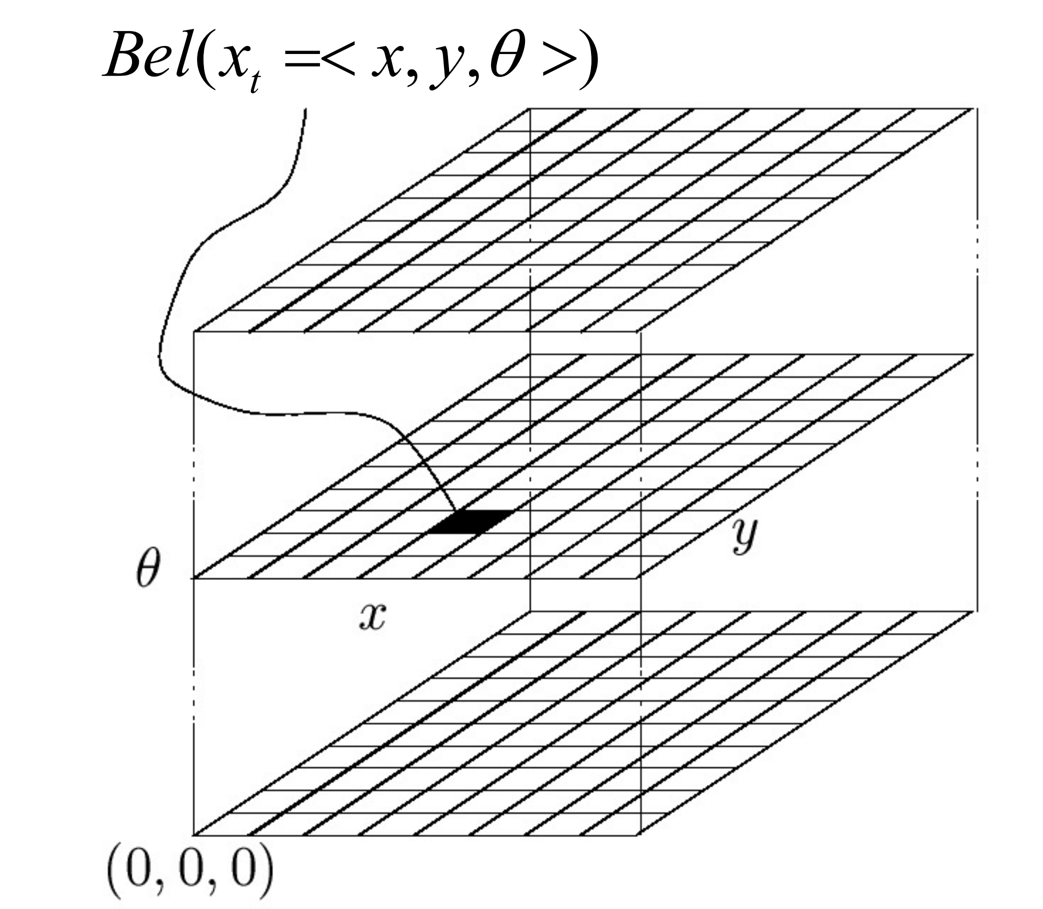
Piecewise Constant



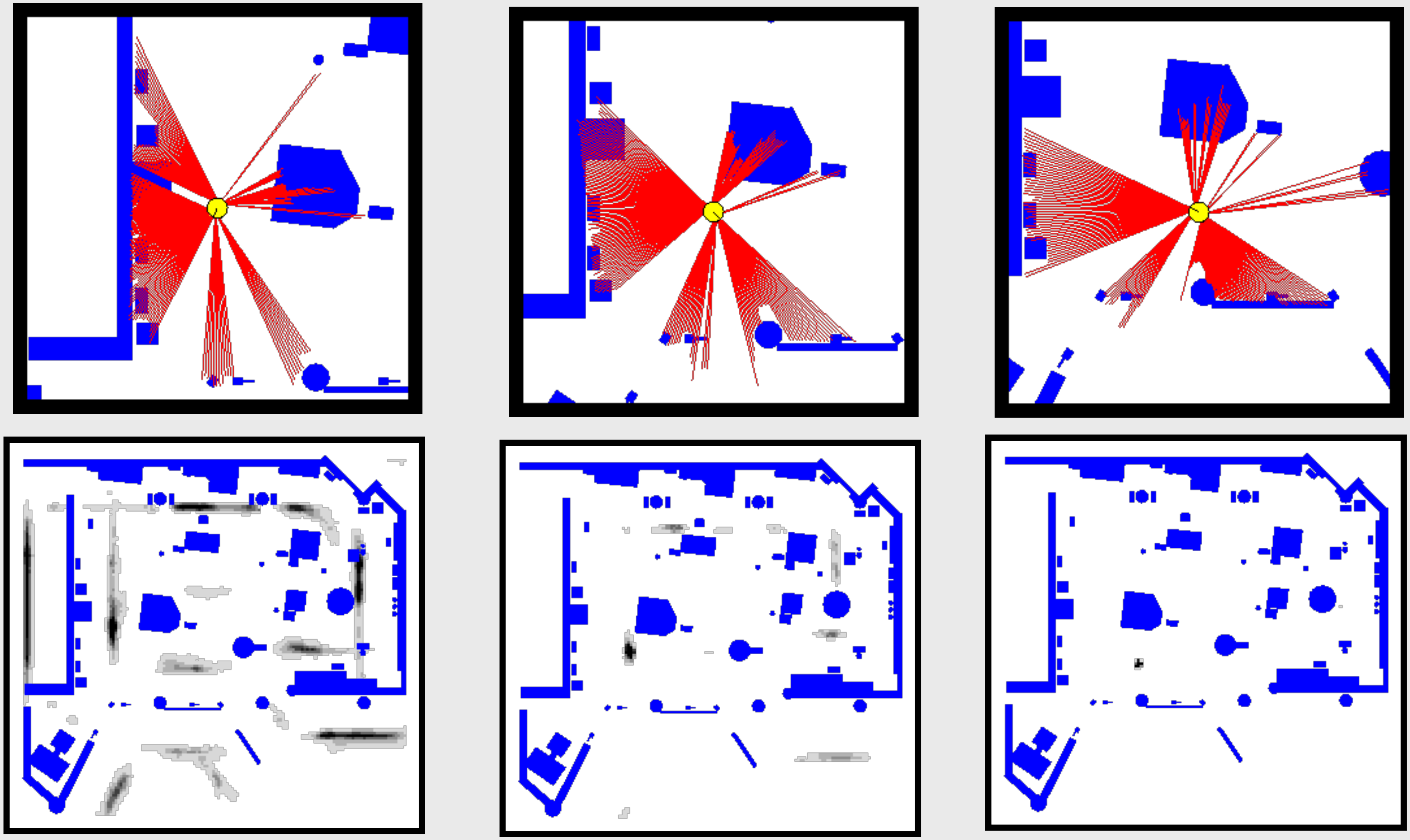
Discrete Bayes Filter Algorithm

```
Algorithm Discrete_Bayes_filter( Bel(x),d ):
$\eta$=0
If d is a perceptual data item z then
    For all x do
        $Bel'(x) = P(z \mid x) Bel(x)$
        $\eta = \eta + Bel'(x)$
    For all x do
        $Bel'(x) = \eta^{-1} Bel'(x)$
Else if d is an action data item u then
    For all x do
        $Bel'(x) = \sum_{x'} P(x \mid u, x')\, Bel(x')$
Return Bel'(x)
```

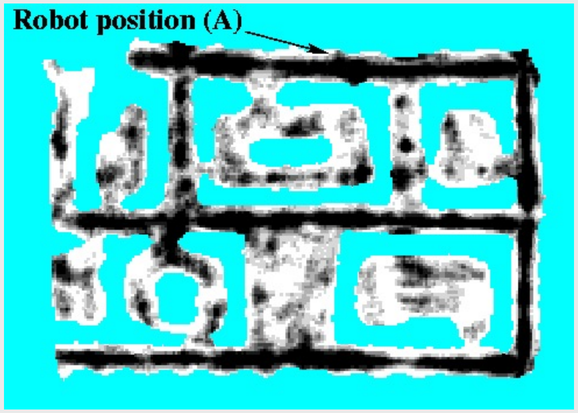
Piecewise Constant Representation

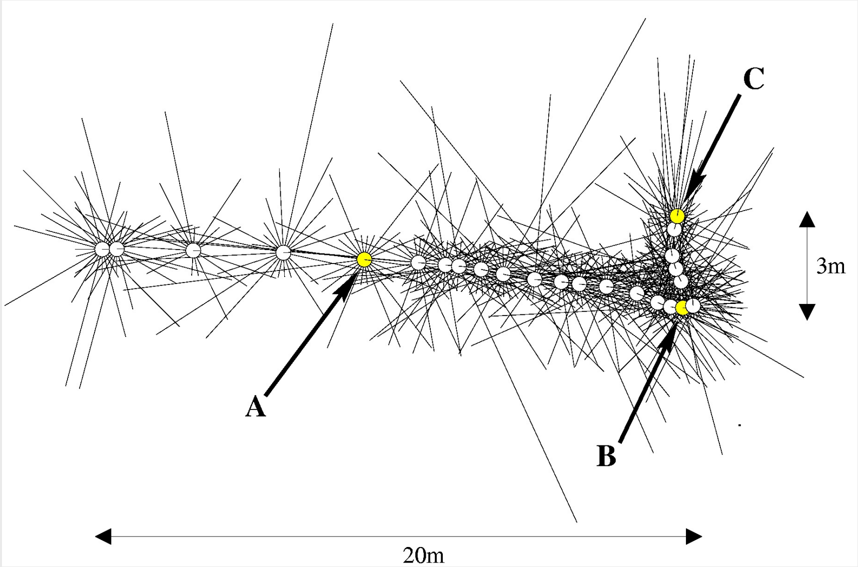


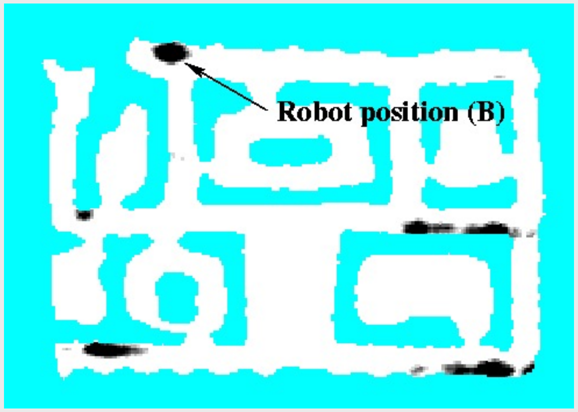
Grid-based Localization

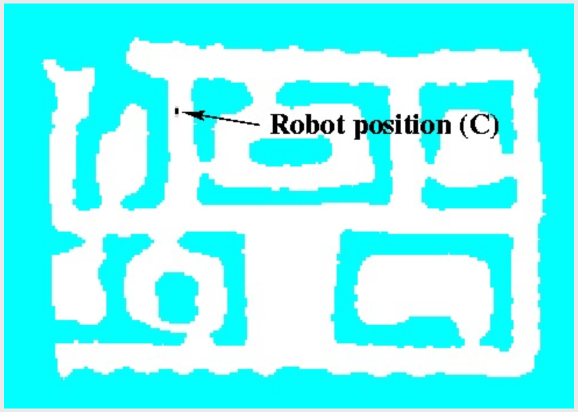


Sonars and Occupancy Grid Map



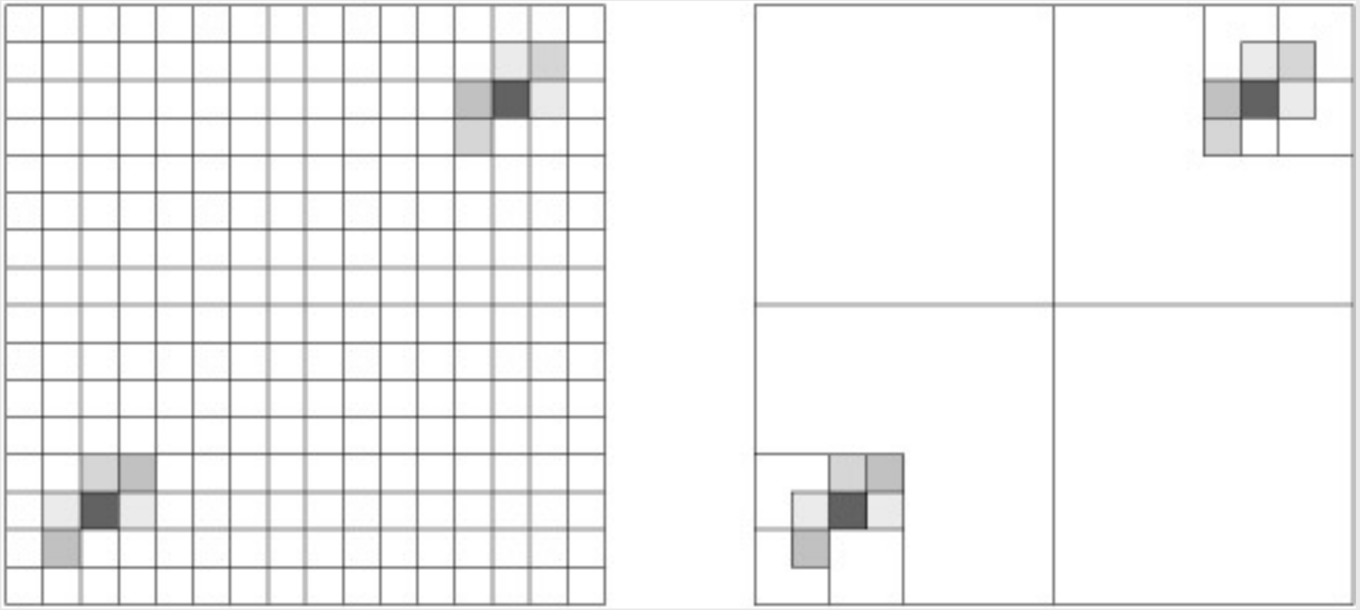






Tree-based Representation

Idea: Represent density using a variant of Octrees



Tree-based Representations

- Efficient in space and time
- Multi-resolution