Particle Filters

Pieter Abbeel
UC Berkeley EECS

Many slides adapted from Thrun, Burgard and Fox, Probabilistic Robotics

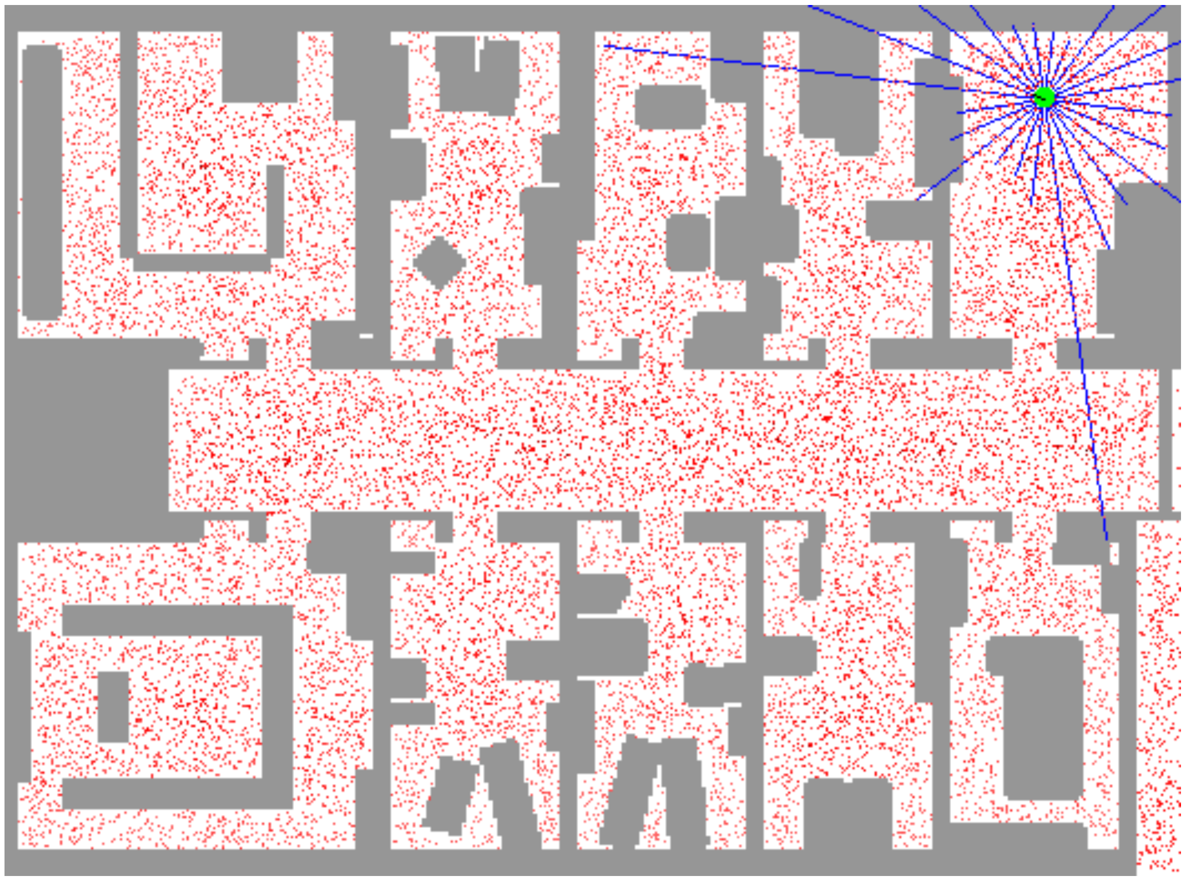
Motivation

- For continuous spaces: often no analytical formulas for Bayes filter updates

- Solution 1: Histogram Filters: (not studied in this course)
  - Partition the state space
  - Keep track of probability for each partition
  - Challenges:
    - What is the dynamics for the partitioned model?
    - What is the measurement model?
    - Often very fine resolution required to get reasonable results

- Solution 2: Particle Filters:
  - Represent belief by random samples
  - Can use actual dynamics and measurement models
  - Naturally allocates computational resources where required (~ adaptive resolution)
  - Aka Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter

Sample-based Localization (sonar)



Problem to be Solved

- Given a sample-based representation $S_t = \{x_t^1, x_t^2, \ldots, x_t^N\}$

  of $\mathrm{Bel}(x_t) = P(x_t \mid z_1, \ldots, z_t, u_1, \ldots, u_t)$

  Find a sample-based representation $S_{t+1} = \{x_{t+1}^1, x_{t+1}^2, \ldots, x_{t+1}^N\}$

  of $\mathrm{Bel}(x_{t+1}) = P(x_{t+1} \mid z_1, \ldots, z_t, \mathbf{z_{t+1}}, u_1, \ldots, \mathbf{u_{t+1}})$

Dynamics Update

- Given a sample-based representation $S_t = \{x_t^1, x_t^2, \ldots, x_t^N\}$

  of $\mathrm{Bel}(x_t) = P(x_t \mid z_1, \ldots, z_t, u_1, \ldots, u_t)$

  Find a sample-based representation

  of $P(x_{t+1} \mid z_1, \ldots, z_t, u_1, \ldots, \mathbf{u_{t+1}})$

- Solution:
  - For i=1, 2, …, N
    - Sample x^i_{t+1} from $P(X_{t+1} \mid X_t = x^i_t, u_{t+1})$

Observation Update

- Given a sample-based representation of $\{x_{t+1}^1, x_{t+1}^2, \ldots, x_{t+1}^N\}$

  $P(x_{t+1} \mid z_1, \ldots, z_t)$

  Find a sample-based representation of

  $P(x_{t+1} \mid z_1, \ldots, z_t, z_{t+1}) = C * P(x_{t+1} \mid z_1, \ldots, z_t) * P(z_{t+1} \mid x_{t+1})$

- Solution:
  - For i=1, 2, …, N
    - $w^{(i)}_{t+1} = w^{(i)}_t * P(z_{t+1} \mid X_{t+1} = x^{(i)}_{t+1})$
  - the distribution is represented by the weighted set of samples

$$\{< x_{t+1}^1, w_{t+1}^1 >, < x_{t+1}^2, w_{t+1}^2 >, \ldots, < x_{t+1}^N, w_{t+1}^N >\}$$

Sequential Importance Sampling (SIS) Particle Filter

- Sample $x^1_1, x^2_1, \dots, x^N_1$ from $P(X_1)$
- Set $w^i_1 = 1$ for all $i=1,\dots,N$
- For $t=1, 2, \dots$
  - Dynamics update:
    - For $i=1, 2, \dots, N$
      - Sample x^i_{t+1} from $P(X_{t+1} \mid X_t = x^i_t , u_{t+1})$
  - Observation update:
    - For $i=1, 2, \dots, N$
      - $w^i_{t+1} = w^i_t * P(z_{t+1} \mid X_{t+1} = x^i_{t+1})$
- At any time t, the distribution is represented by the weighted set of samples $\{ <x^i_t, w^i_t> ; i=1,\dots,N\}$

SIS particle filter major issue

- The resulting samples are only weighted by the evidence
- The samples themselves are never affected by the evidence

→ Fails to concentrate particles/computation in the high probability areas of the distribution $P(x_t \mid z_1, \ldots, z_t)$

Sequential Importance Resampling (SIR)

- At any time t, the distribution is represented by the weighted set of samples

  $\{ <x^i_t, w^i_t> ; i=1,\ldots,N\}$

→ Sample N times from the set of particles

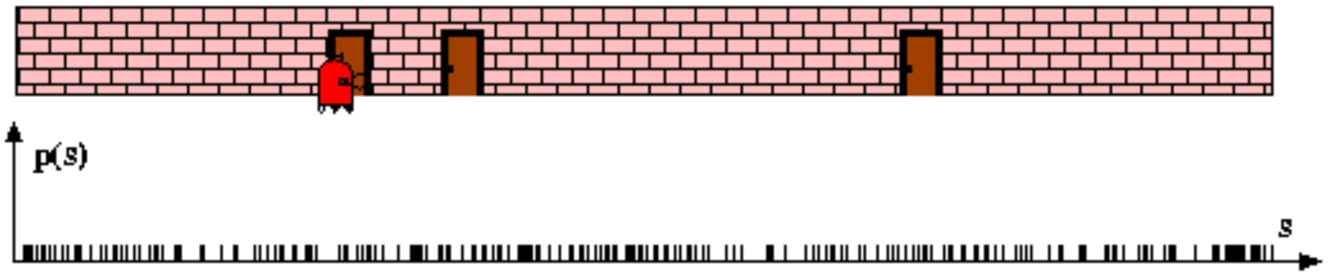
→ The probability of drawing each particle is given by its importance weight

→ More particles/computation focused on the parts of the state space with high probability mass

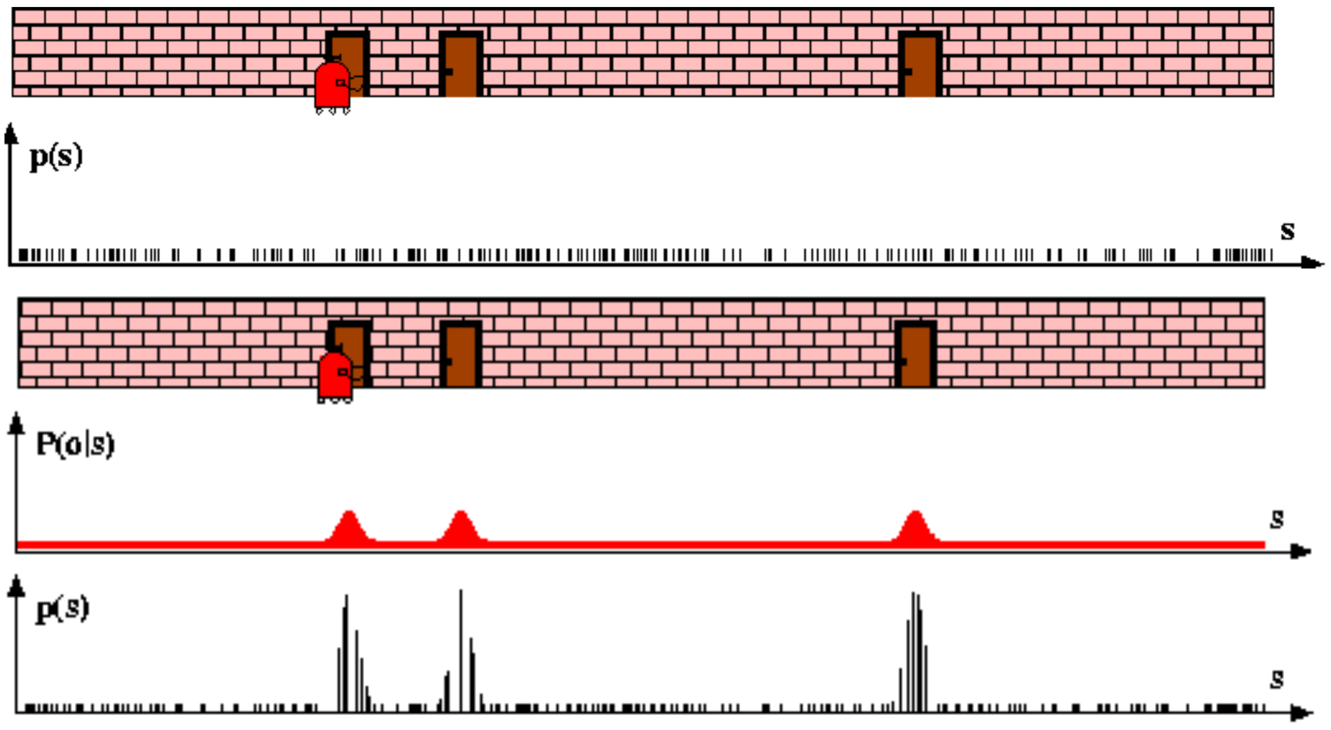
Sequential Importance Resampling (SIR) Particle Filter

```
1.  Algorithm **particle_filter**( $S_{t-1}$, $u_t$ , $z_t$):
2.  $S_t = \emptyset, \quad \eta = 0$
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 $p(x_t \mid x_{t-1}, u_t)$ 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.      $S_t = S_t \cup \{< x_t^i, w_t^i >\}$                   Insert
9.  For $i = 1 \ldots n$
10.     $w_t^i = w_t^i / \eta$                                   Normalize weights
11. Return $S_t$
```

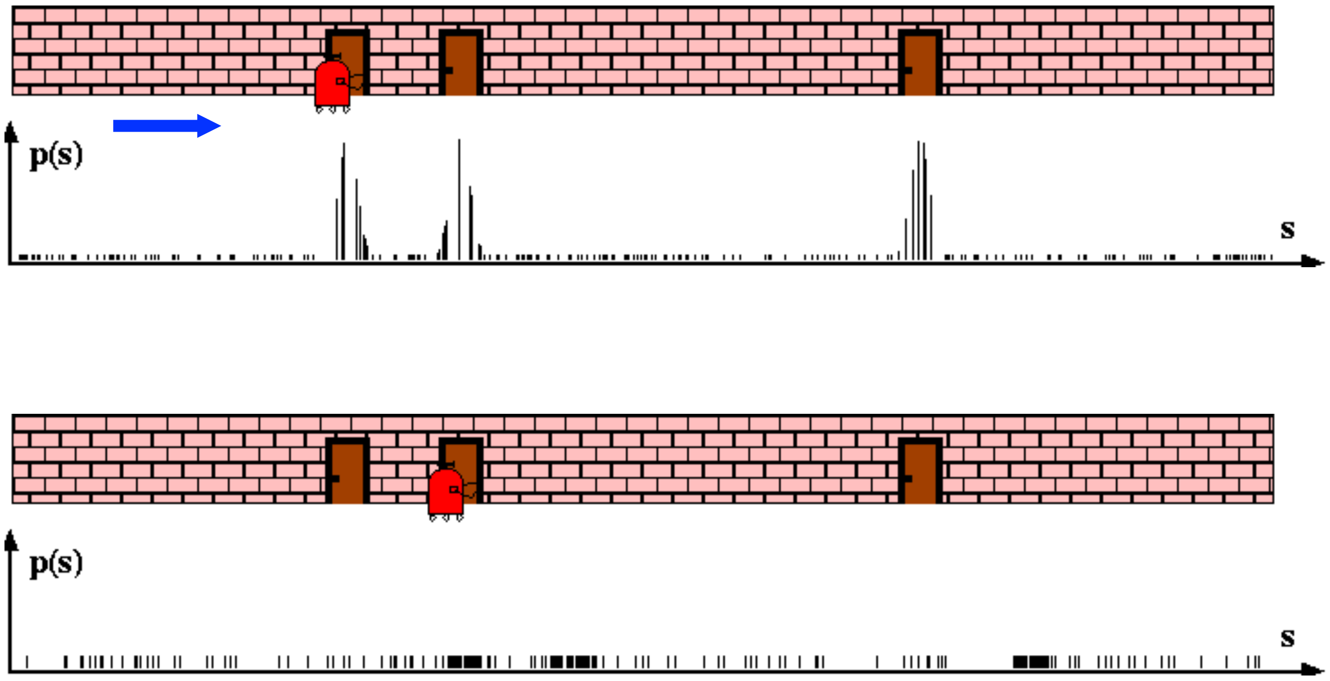
Particle Filters



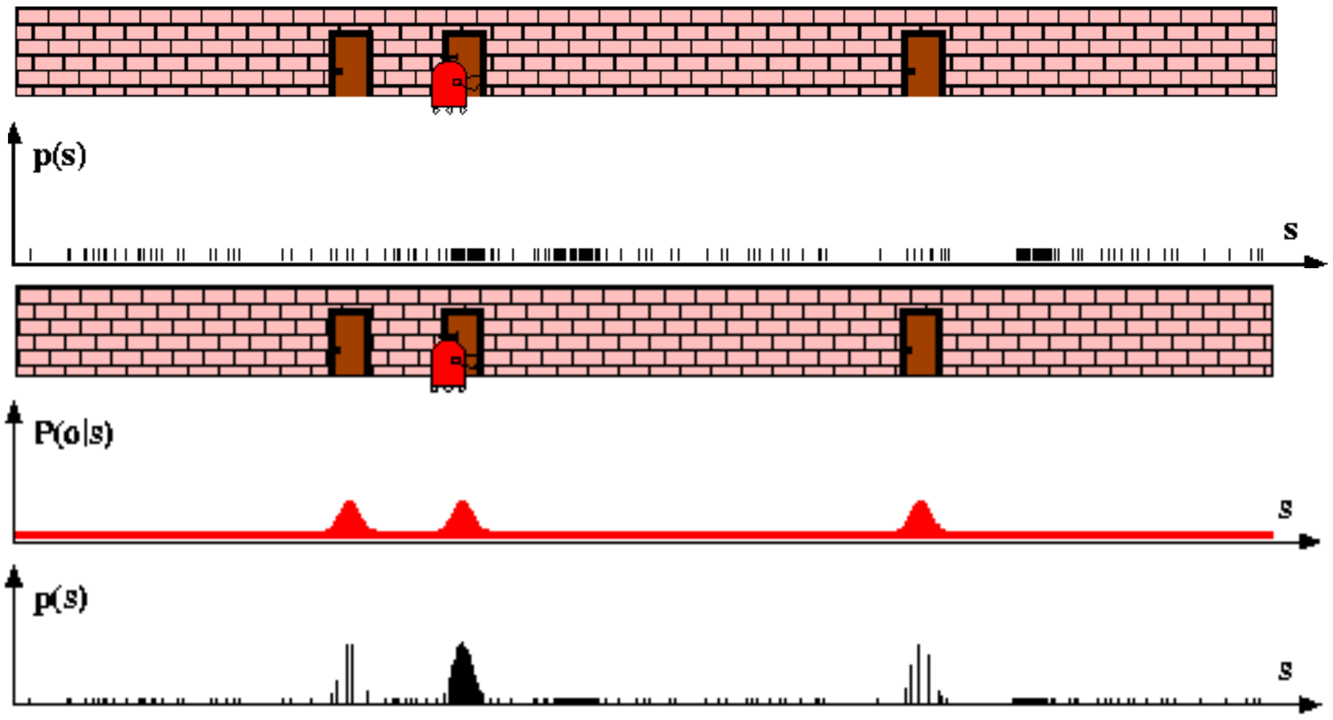
Sensor Information: Importance Sampling



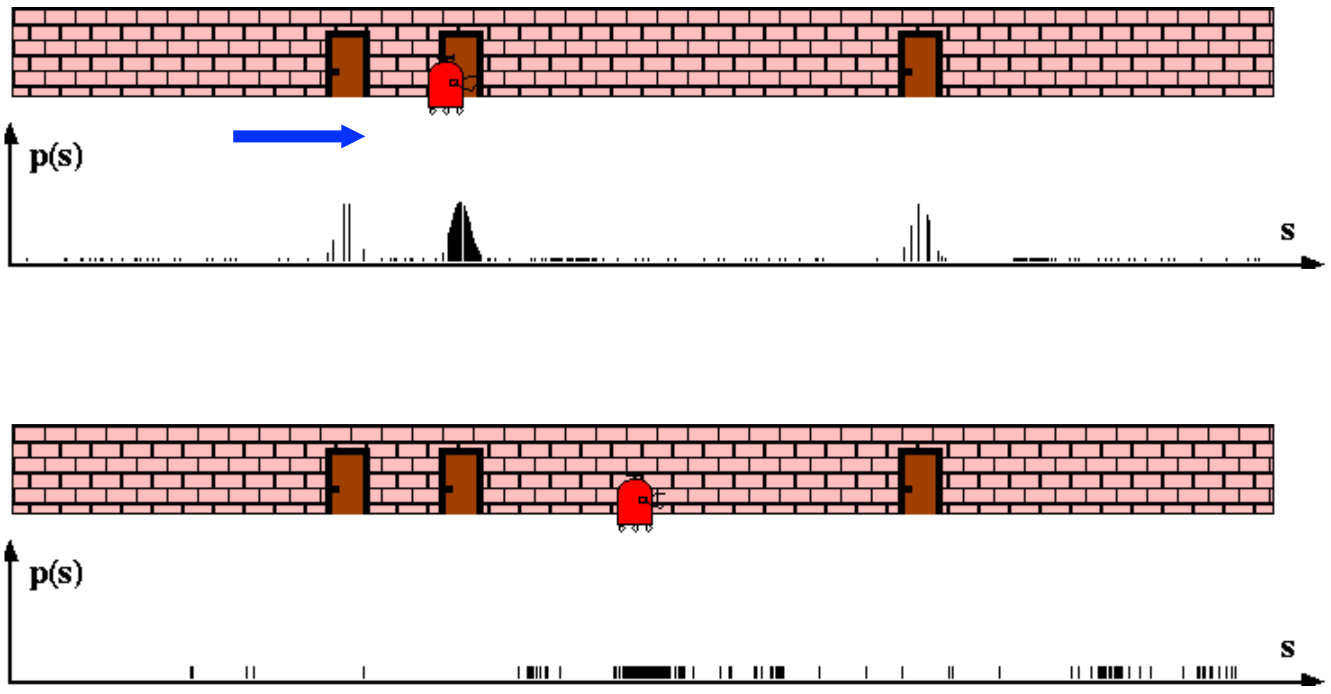
Robot Motion

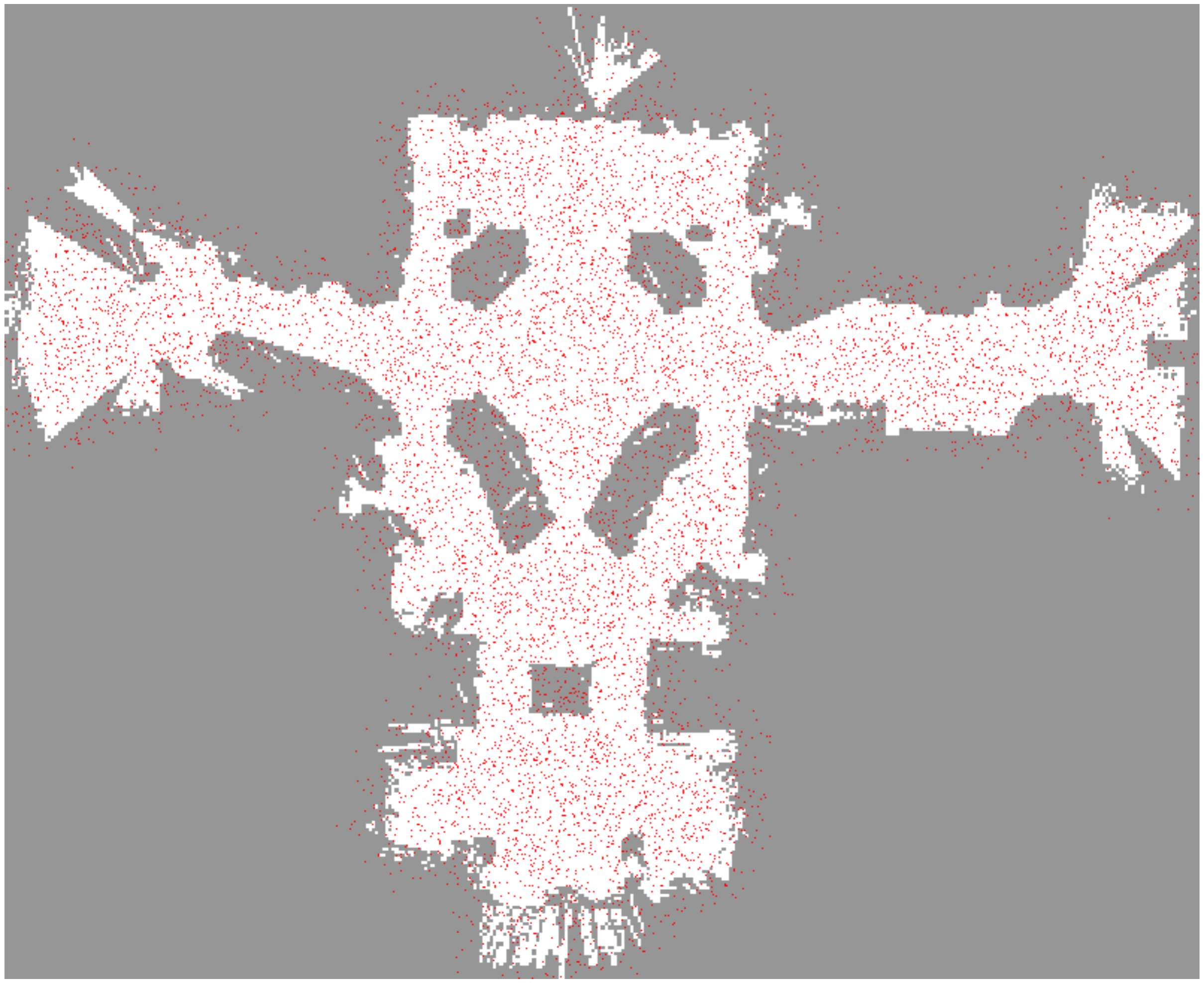


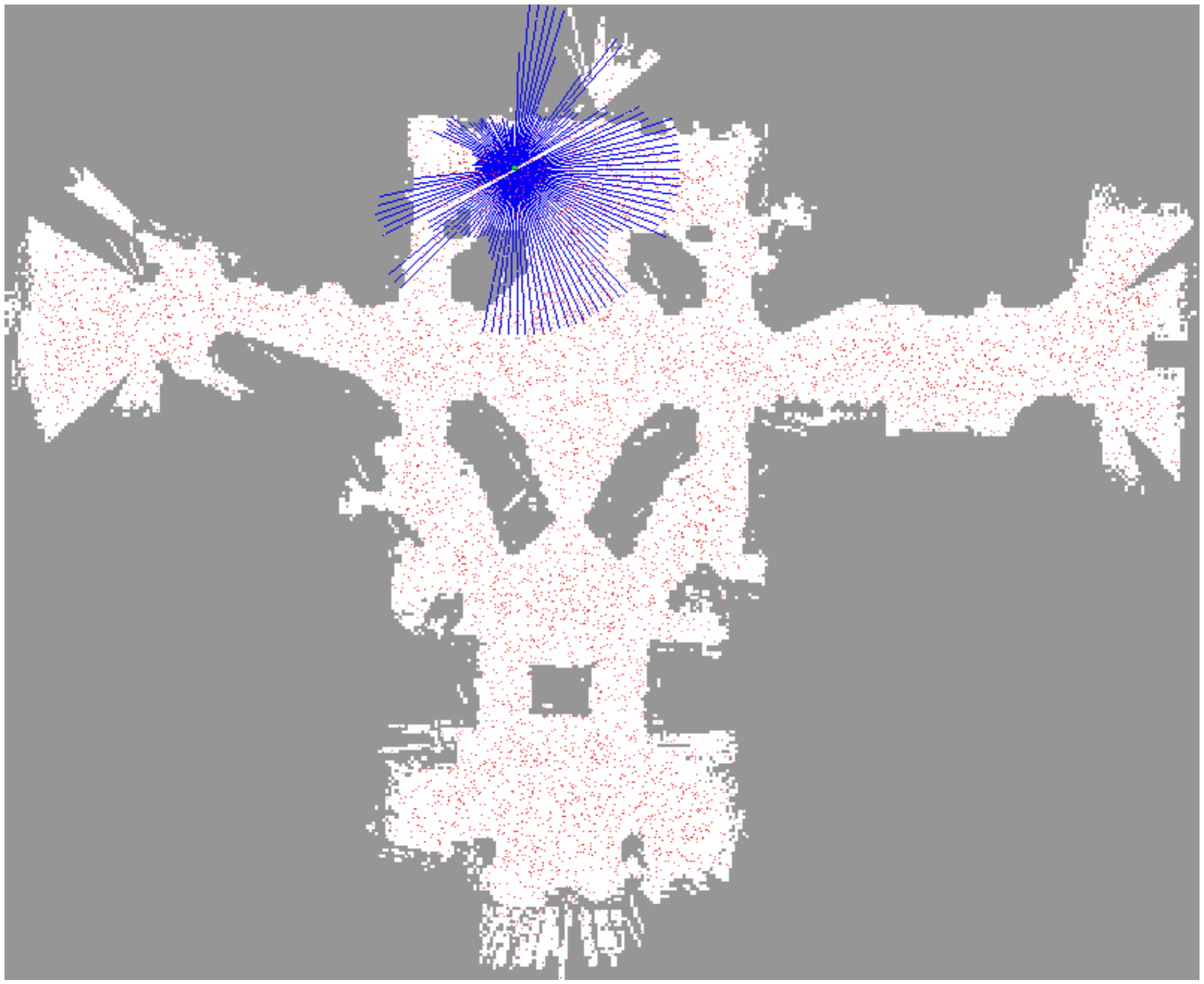
Sensor Information: Importance Sampling

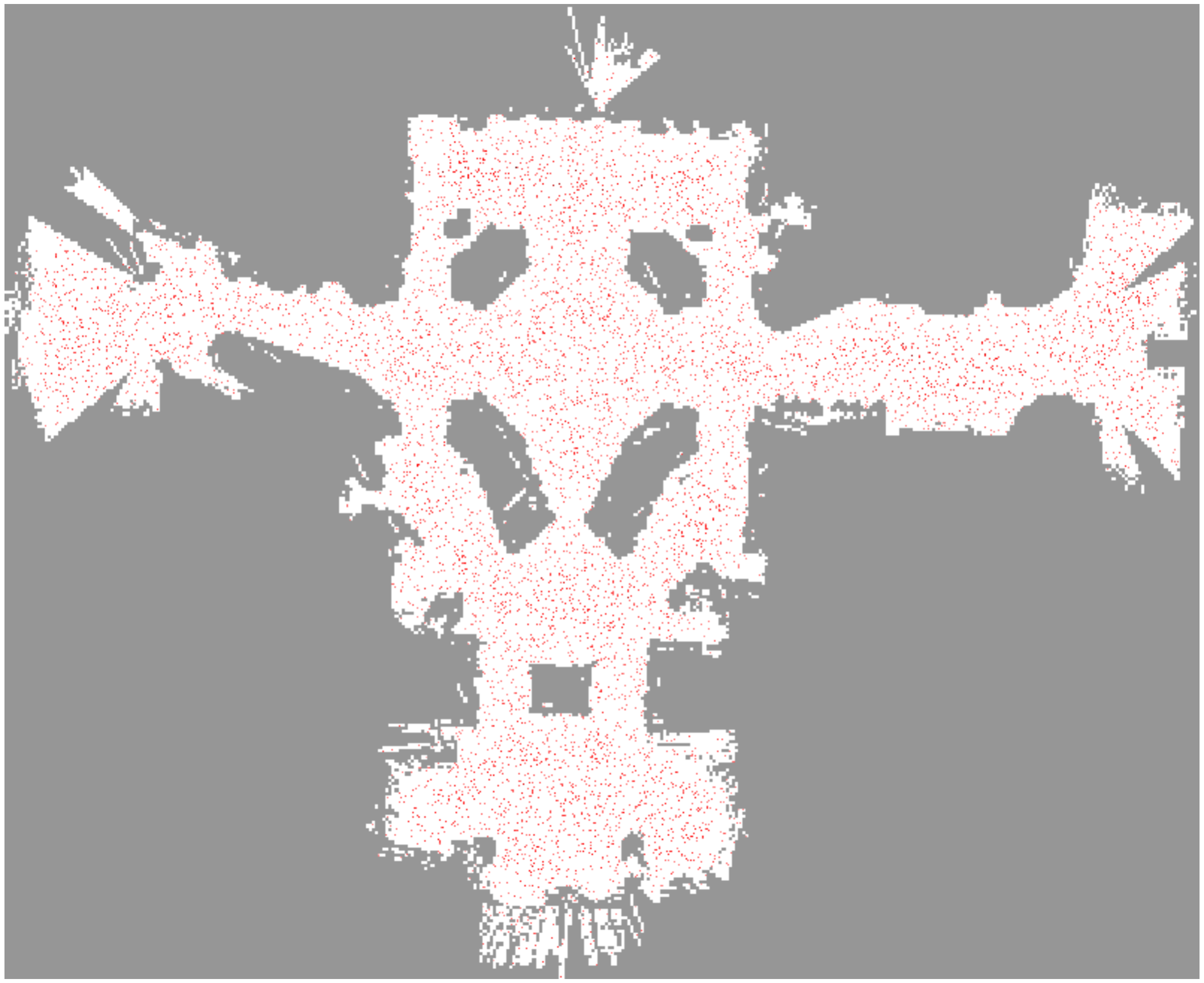


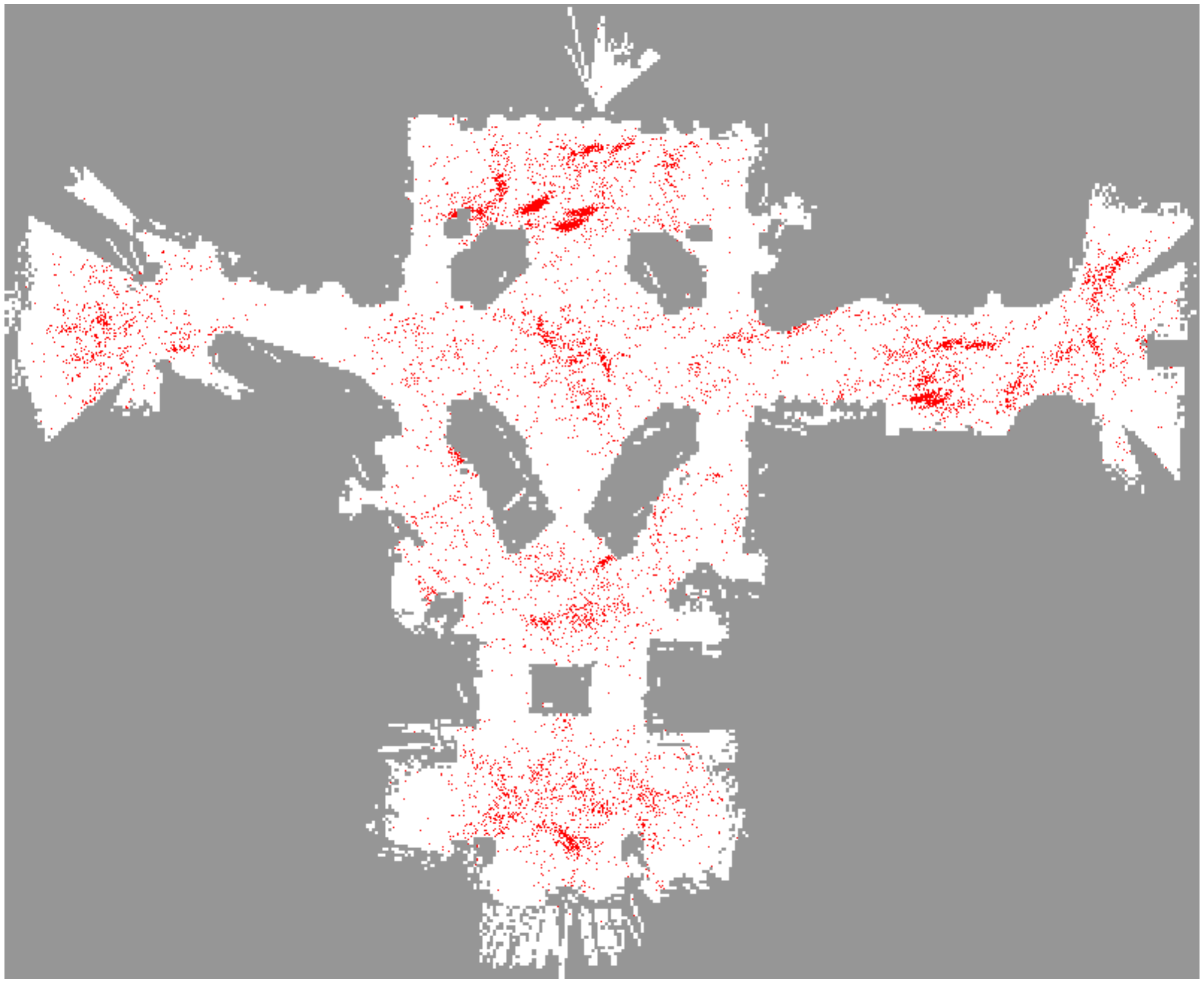
Robot Motion

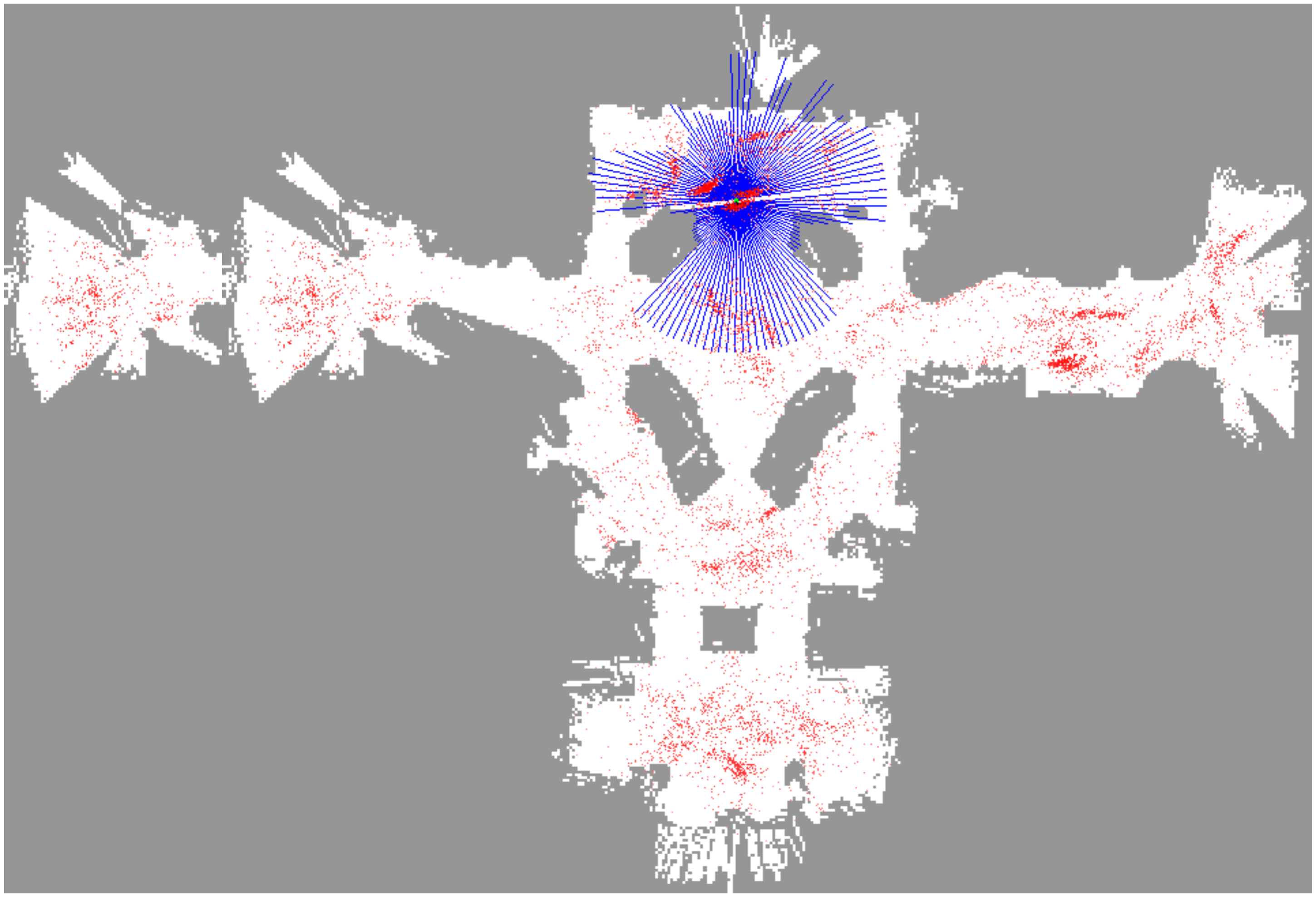


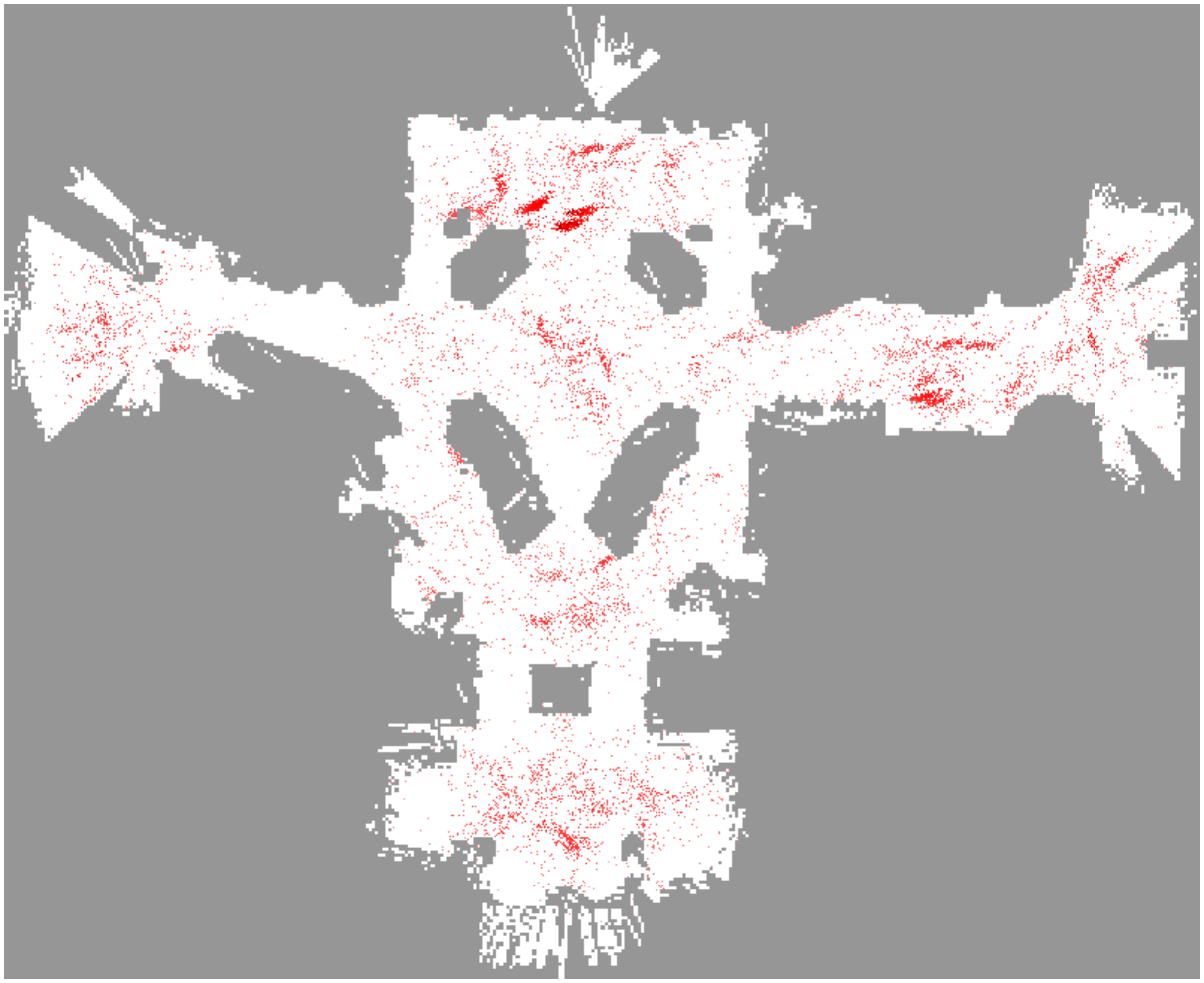


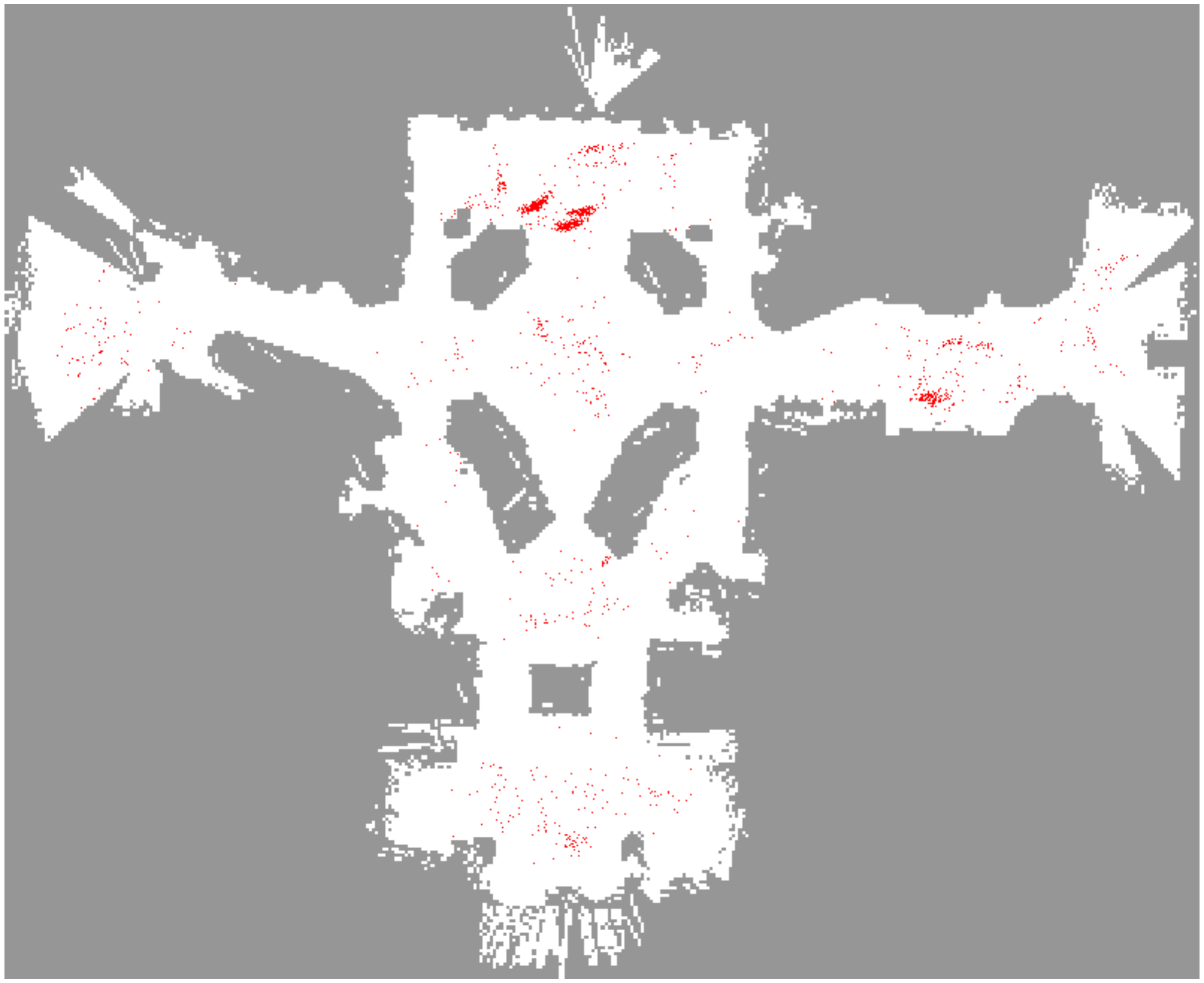


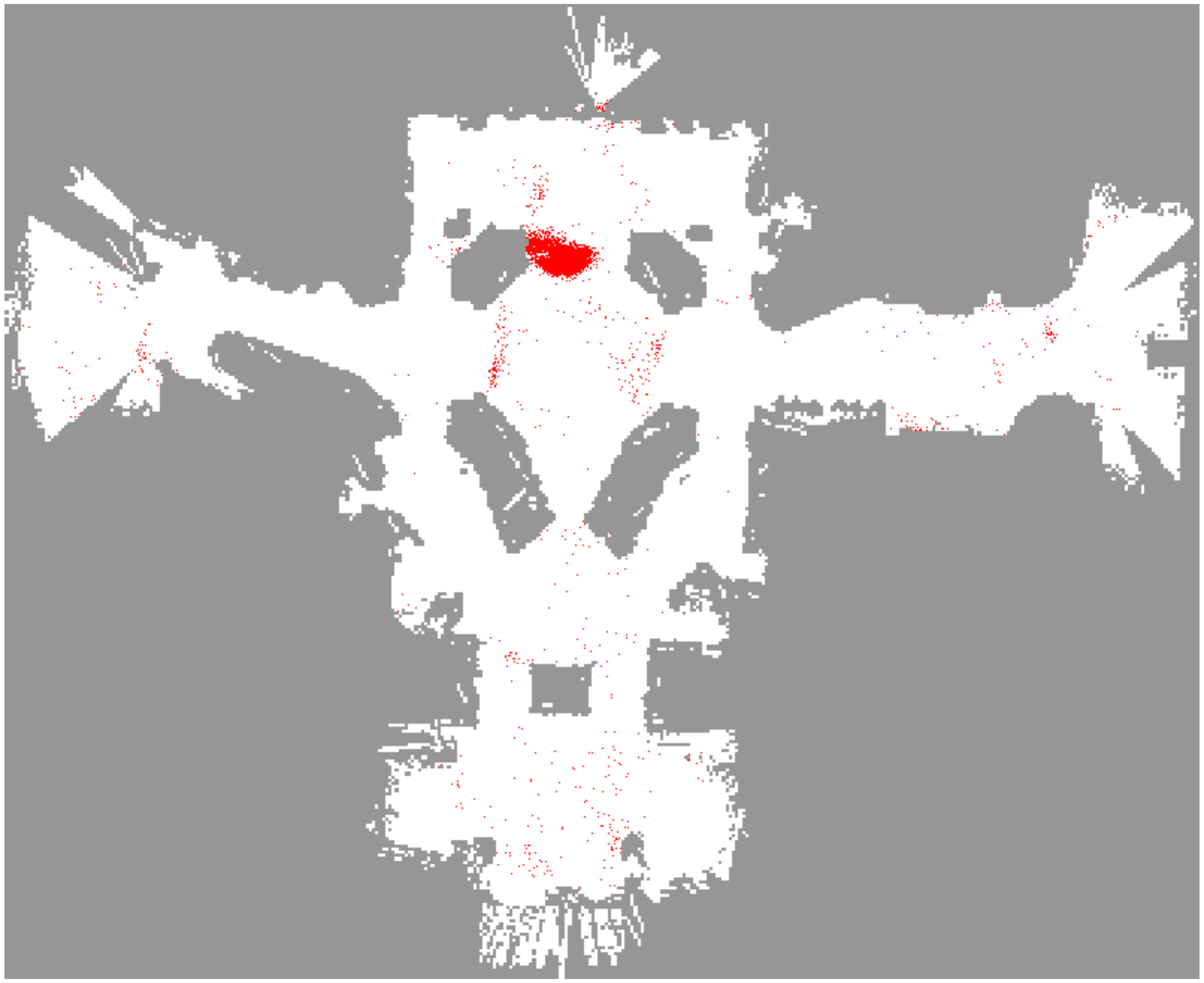


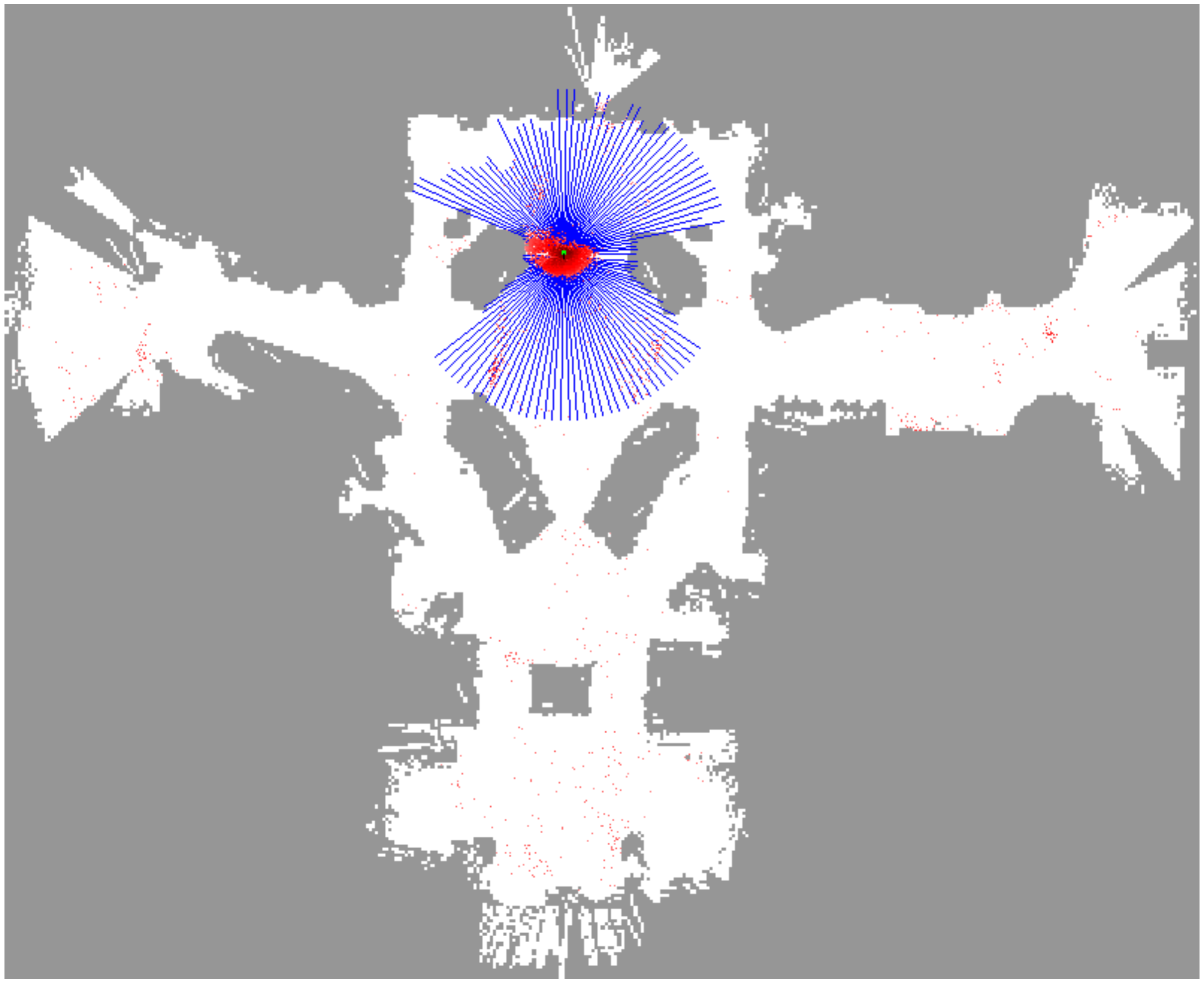


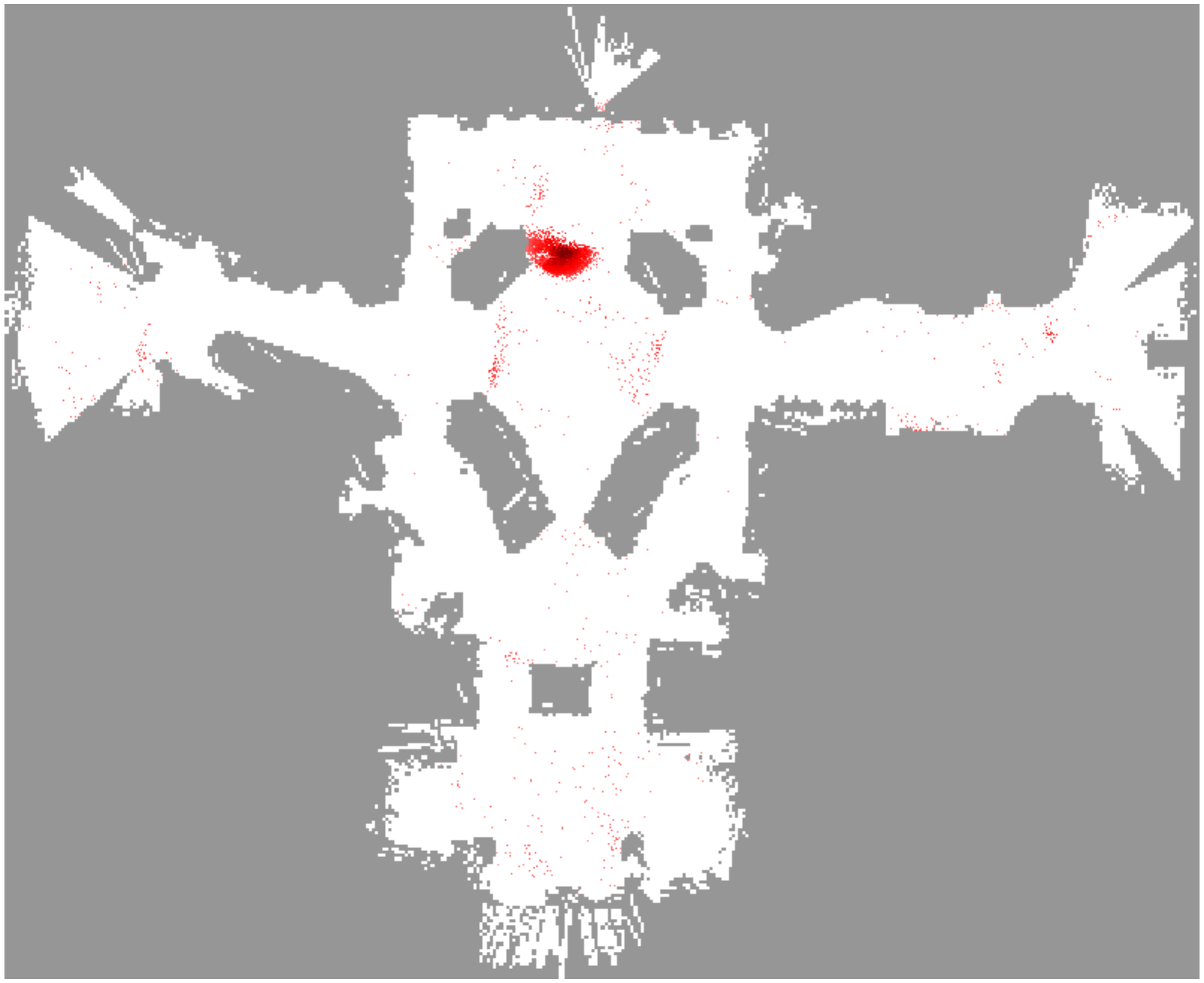


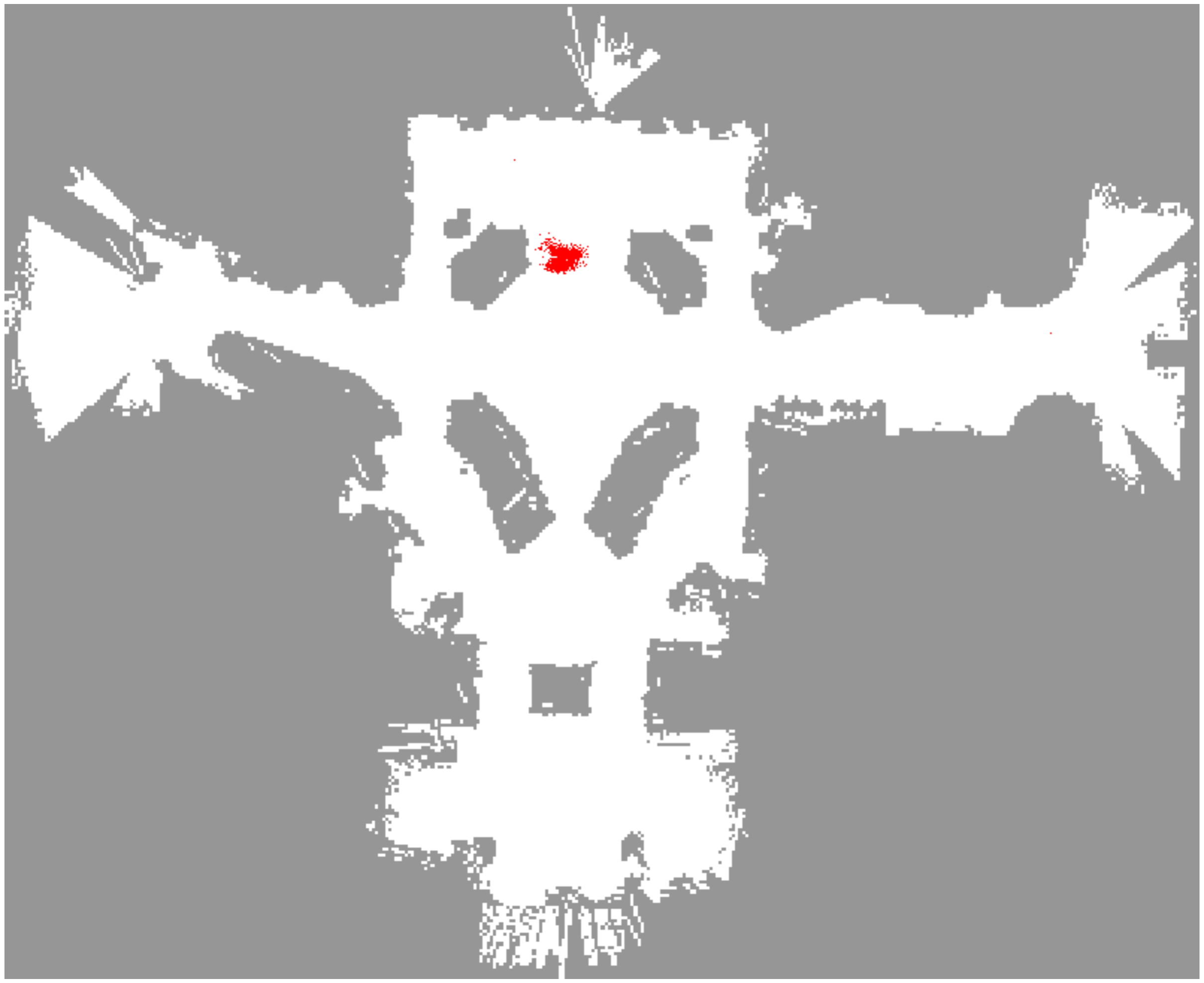


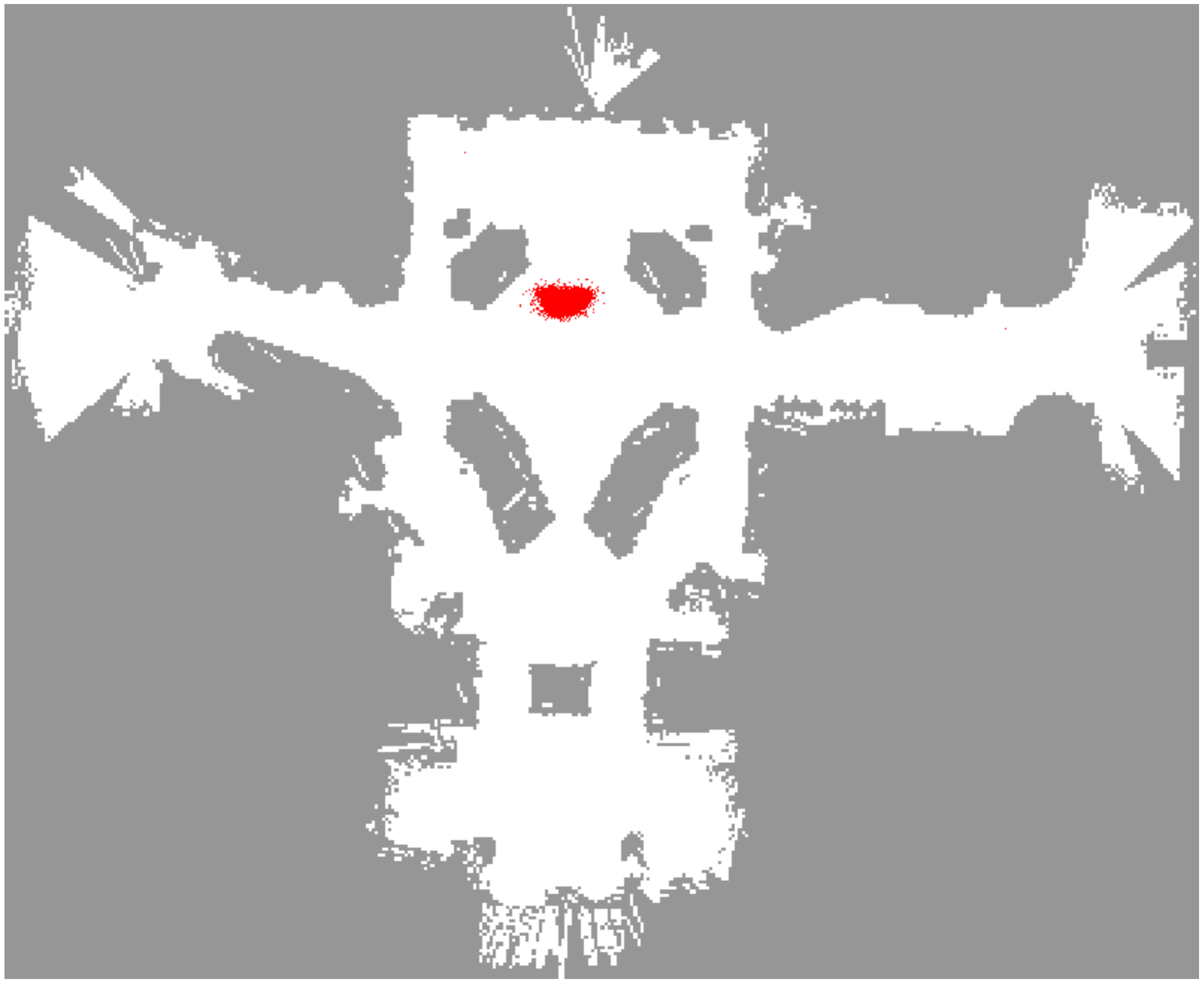


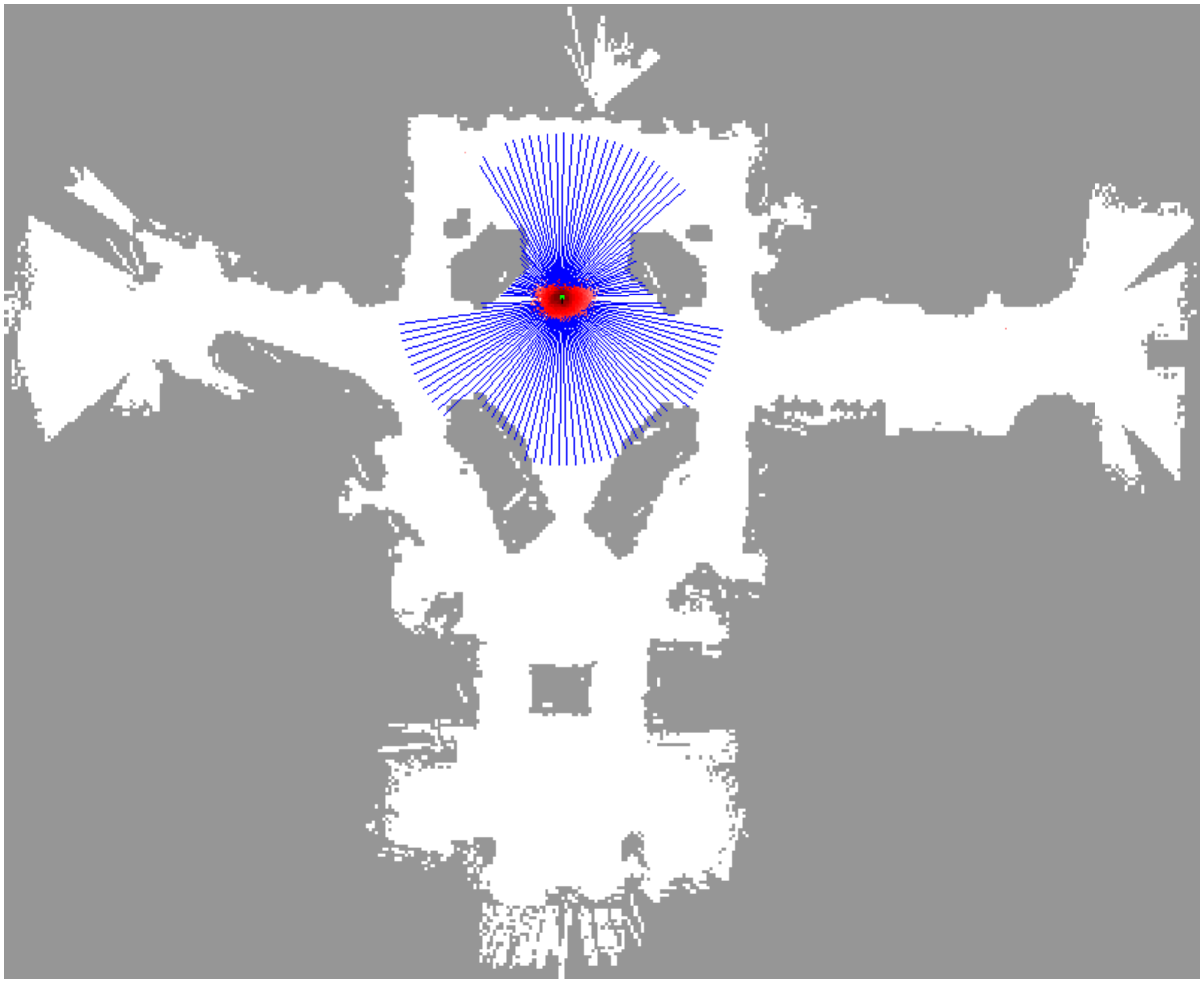


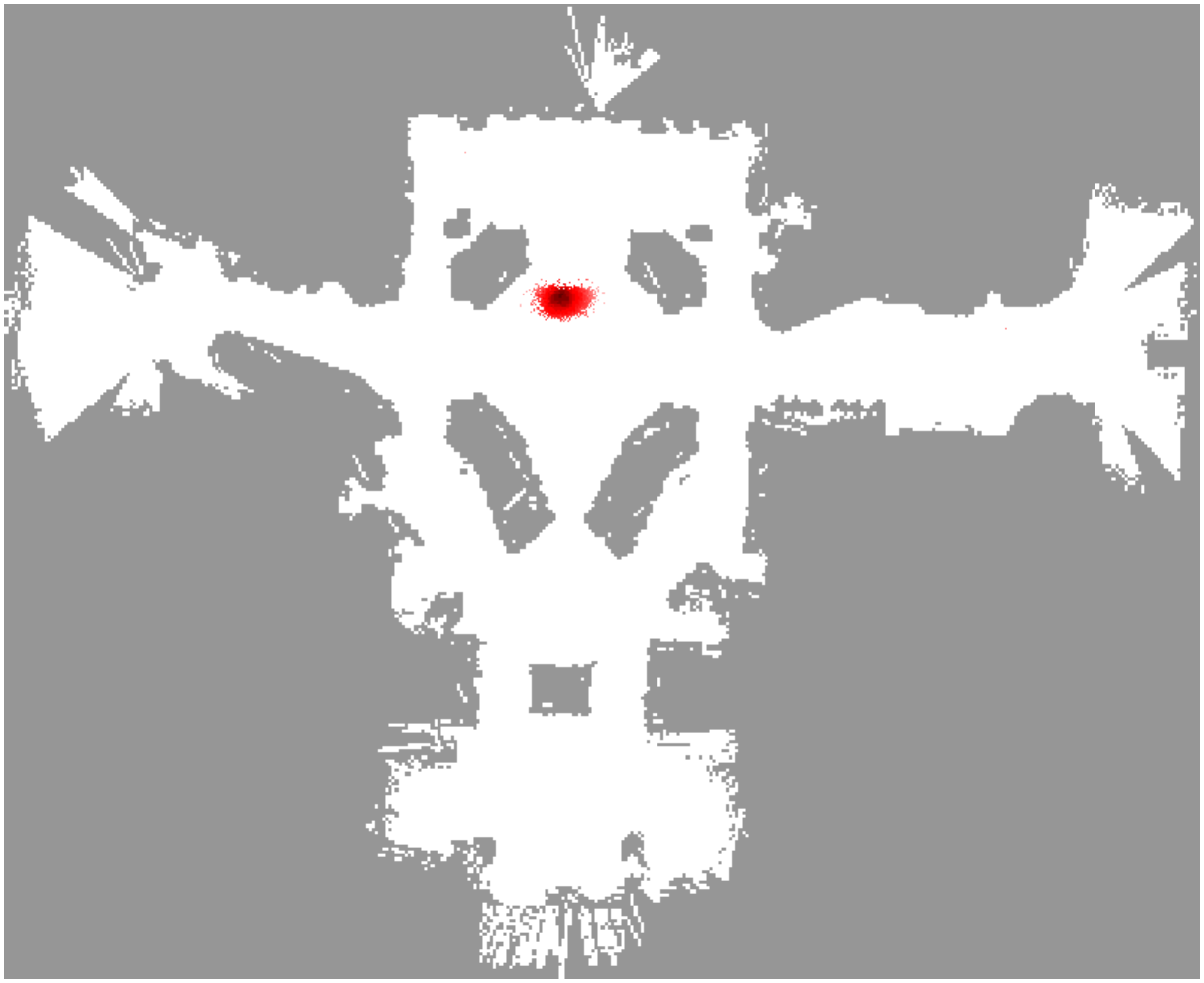


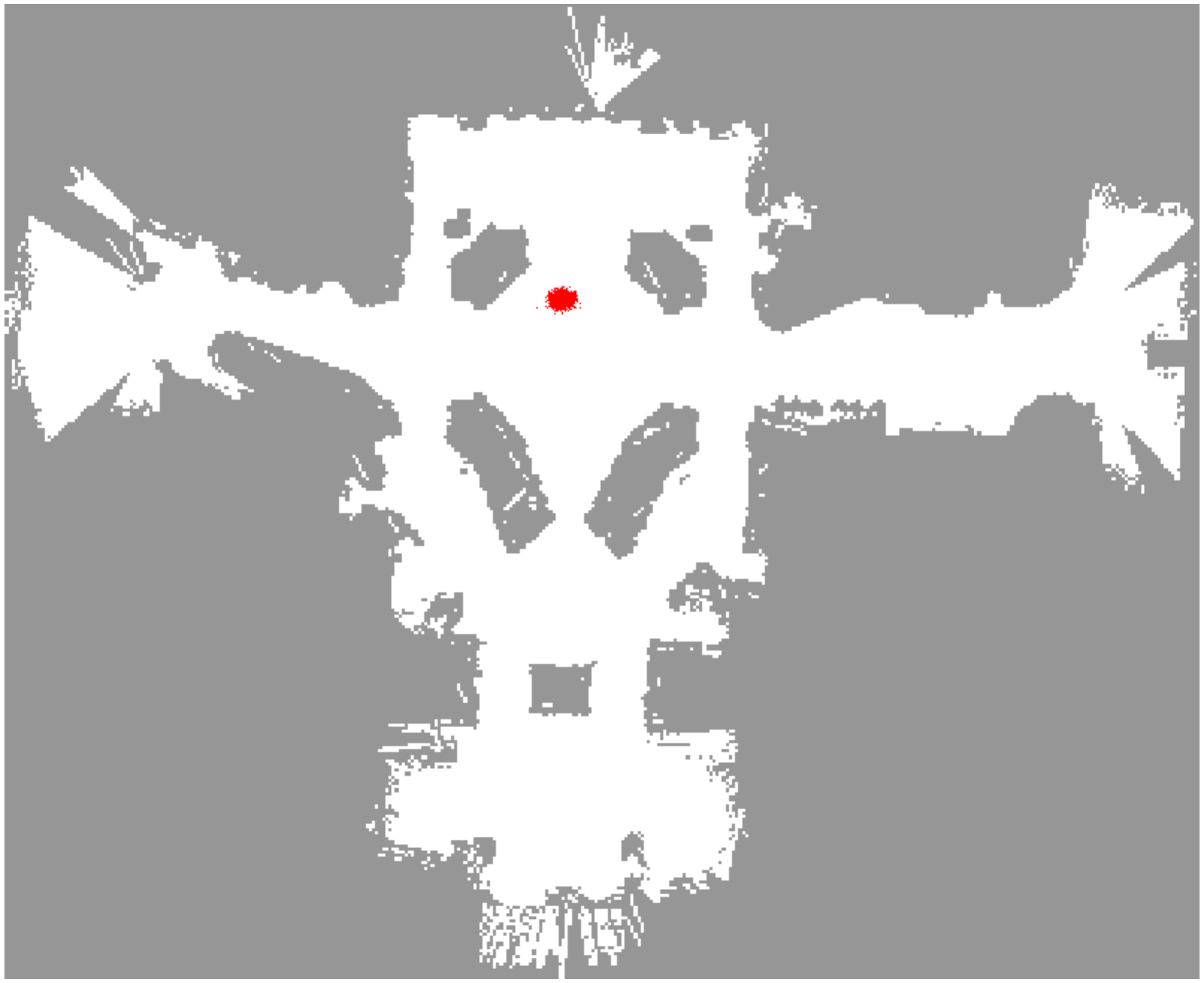


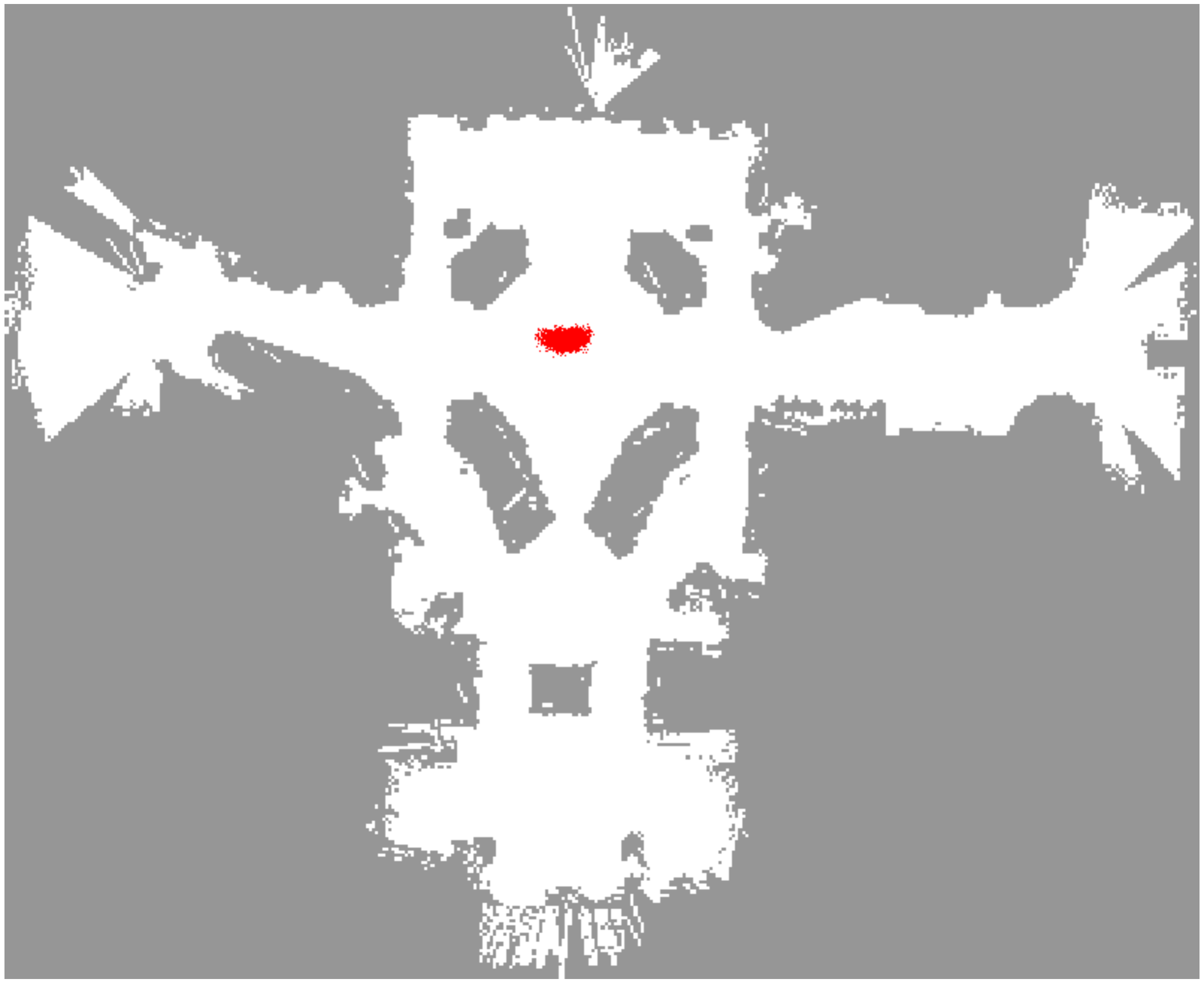


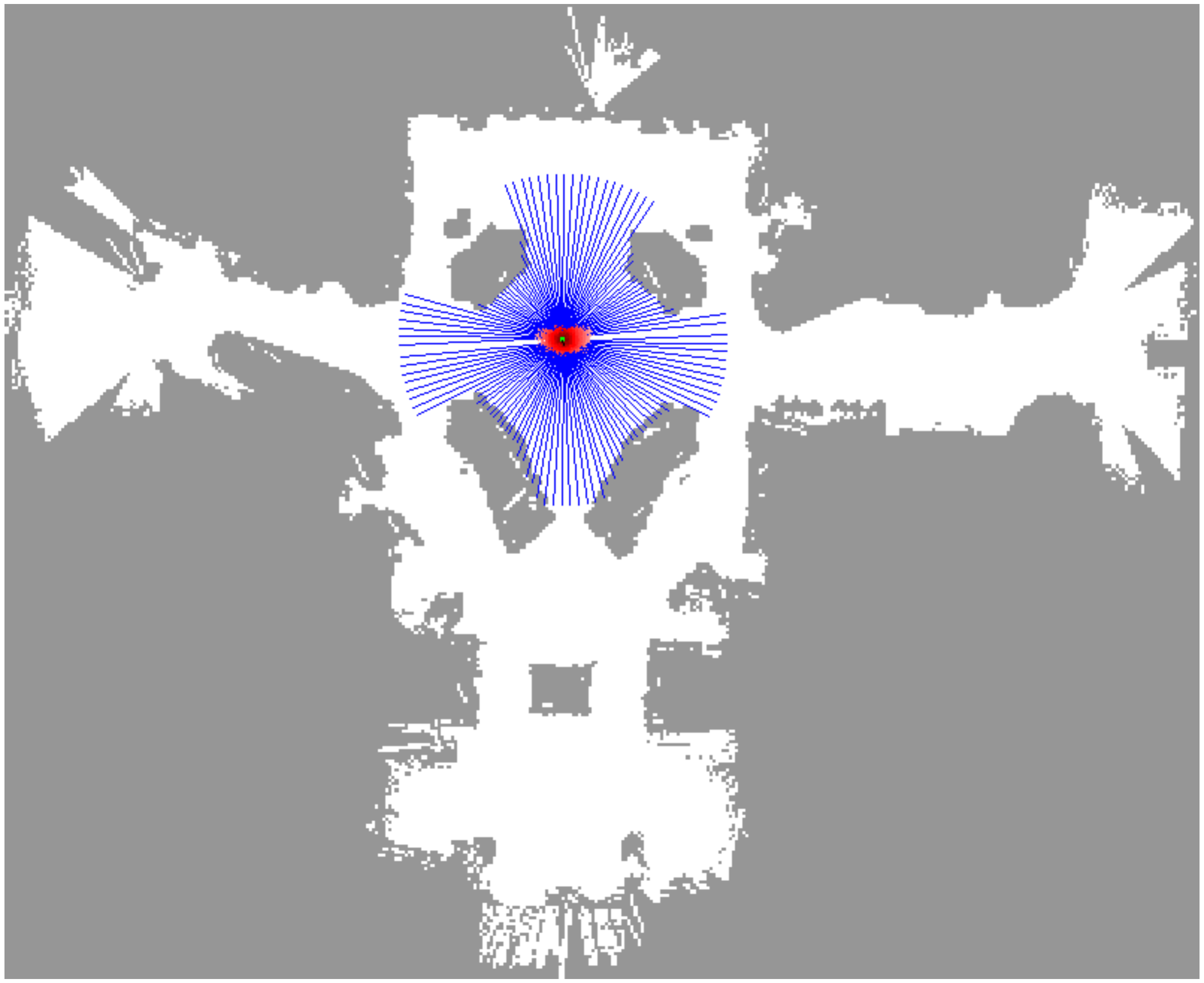


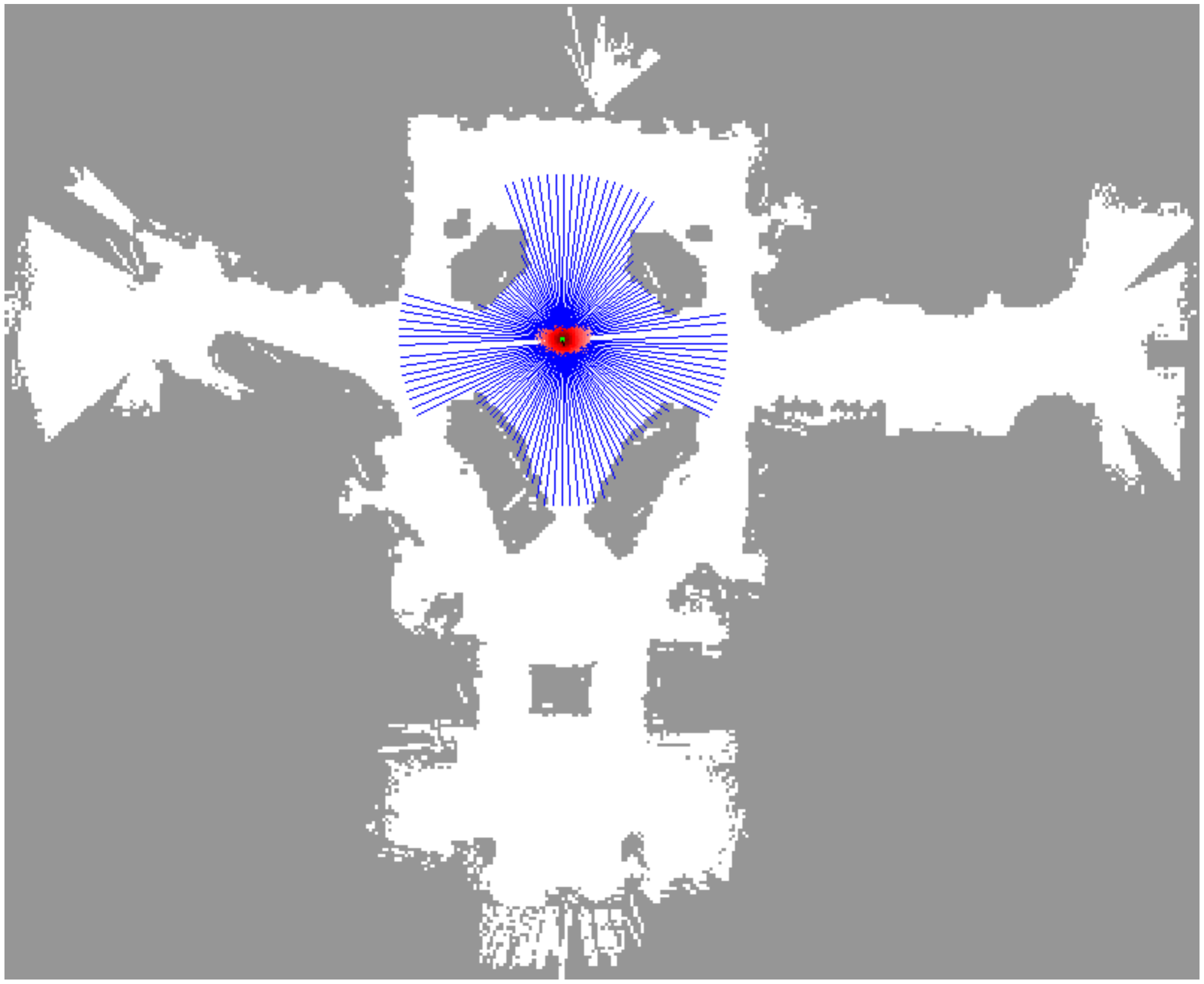












Noise Dominated by Motion Model

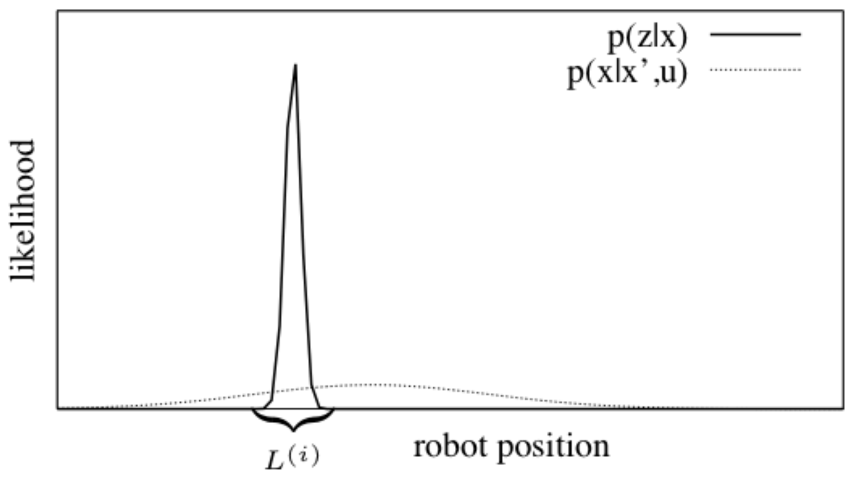


Fig. 1. Within the interval $L^{(i)}$ the product of both functions is dominated by the observation likelihood in case an accurate sensor is used. [Grisetti, Stachniss, Burgard, T-RO2006]

→ Most particles get (near) zero weights and are lost.

Importance Sampling

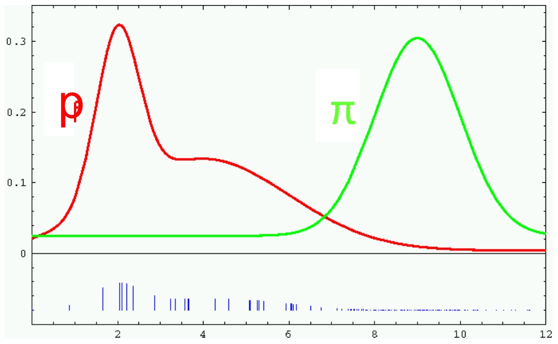
- Theoretical justification: for any function *f* we have:

$$\begin{aligned}
\mathrm{E}_{X\sim p}[f(X)] &= \int_x f(x)p(x)dx \\
&= \int_x f(x)p(x)\frac{\pi(x)}{\pi(x)}dx \quad \text{if} \pi(x)=0 \Rightarrow p(x)=0 \\
&= \int_x f(x)\frac{p(x)}{\pi(x)}\pi(x)dx \\
&= \mathrm{E}_{X\sim\pi}[\frac{p(X)}{\pi(X)}f(X)] \\
&\approx \frac{1}{m}\sum_{i=1}^{m}\frac{p(x^{(i)})}{\pi(x^{(i)})}f(x^{(i)}) \quad \text{with } x^{(i)}\sim\pi
\end{aligned}$$

- *f* could be: whether a grid cell is occupied or not, whether the position of a robot is within 5cm of some (x,y), etc.

Importance Sampling

- Task: sample from density p(.)
- Solution:
  - sample from "proposal density" π(.)
  - Weight each sample $x^{(i)}$ by $p(x^{(i)}) / \pi(x^{(i)})$
- E.g.:



- Requirement: if π(x) = 0 then p(x) = 0.

Particle Filters Revisited

```
1.  Algorithm particle_filter( $S_{t-1}$, $u_t$ , $z_t$):
2.  $S_t = \emptyset, \quad \eta = 0$
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 $\boxed{\pi(x_t \mid x_{t-1}^{j(i)}, u_t, z_t)}$
6.          $\boxed{w_t^i = \frac{p(z_t \mid x_t^i)\, p(x_t^i \mid x_{t-1}^i, u_t)}{\pi(x_t^i \mid x_{t-1}^i, u_t, z_t)}}$          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
11. Return $S_t$
```

Optimal Sequential Proposal π(.)

- Optimal $\pi(x_t \mid x_{t-1}^i, u_t, z_t)$ = $p(x_t \mid x_{t-1}^i, u_t, z_t)$

→
$$w_t^i = \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i, u_t)}{\pi(x_t^i|x_{t-1}^i, u_t, z_t)}$$
$$= \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i, u_t)}{p(x_t^i|x_{t-1}^i, u_t, z_t)}$$

- Applying Bayes rule to the denominator gives:

$$p(x_t^i|x_{t-1}^i, u_t, z_t) = \frac{p(z_t|x_t^i, u_t, x_{t-1}^i)p(x_t^i|x_{t-1}^i, u_t)}{p(z_t|x_{t-1}^i, u_t)}$$

- Substitution and simplification gives

$$w_t^i = p(z_t|x_{t-1}, u_t) = \int p(z_t|x_t)p(x_t|x_{t-1}^i, u_t)dx_t$$

Optimal Sequential Proposal π(.)

- Optimal $\pi(x_t \mid x_{t-1}^i, u_t, z_t)$ = $p(x_t \mid x_{t-1}^i, u_t, z_t)$

  $\rightarrow$ $w_t^i = p(z_t | x_{t-1}, u_t) = \int p(z_t | x_t) p(x_t | x_{t-1}^i, u_t) dx_t$

- Challenges:
  - Typically difficult to sample from $p(x_t \mid x_{t-1}^i, u_t, z_t)$
  - Importance weight: typically expensive to compute integral

Example 1: π(.) = Optimal Proposal Nonlinear Gaussian State Space Model

$$\pi(x_t | x_{t-1}^i, u_t, z_t) = p(x_t | x_{t-1}^i, u_t, z_t), \;\; w_t^i = p(z_t | x_{t-1}, u_t) = \int p(z_t | x_t) p(x_t | x_{t-1}^i, u_t) dx_t$$

- Nonlinear Gaussian State Space Model:

$$\begin{aligned} x_t &= f(x_{t-1}, u_t) + v_t, \quad v_t \sim \mathcal{N}(0, \Sigma_v) \\ z_t &= C x_t + w_t, \quad w_t \sim \mathcal{N}(0, \Sigma_w) \end{aligned}$$

- Then: $p(x_t | x_{t-1}^i, u_t, z_t) = \mathcal{N}(m_t, \Sigma)$ with $\begin{aligned} \Sigma &= \left(\Sigma_v^{-1} + C^\top \Sigma_w^{-1} C\right)^{-1} \\ m_t &= \Sigma \left(\Sigma_v^{-1} f(x_{t-1}, u_t) + C^\top \Sigma_w^{-1} z_t\right) \end{aligned}$

- And:

$$p(z_t | x_{t-1}, u_t) \propto$$
$$\exp\left(-\frac{1}{2}(z_t - Cf(x_{t-1}, u_t))^\top (\Sigma_v + C\Sigma_w C^\top)^{-1} (z_t - Cf(x_{t-1}, u_t))\right)$$

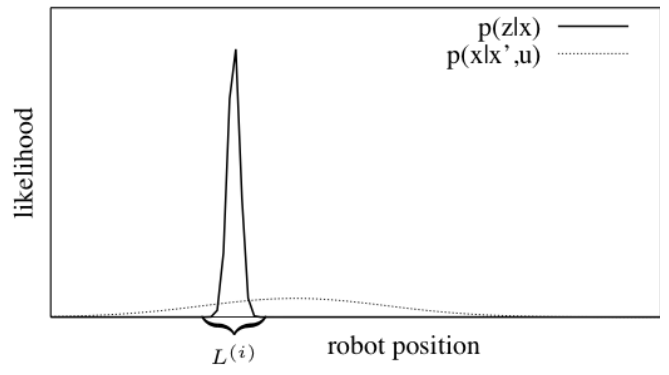
Example 2: π(.) = Motion Model

$$\pi(x_t|x_{t-1}^i, u_t, z_t) = p(x_t|x_{t-1}^i, u_t),$$

$$w_t^i = \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i, u_t)}{\pi(x_t^i|x_{t-1}^i, u_t, z_t)} = p(z_t|x_t^i)$$

→ the "standard" particle filter

Example 3: Approximating Optimal π for Localization



[Grisetti, Stachniss, Burgard, T-RO2006]

- One (not so desirable solution): use smoothed likelihood such that more particles retain a meaningful weight --- BUT information is lost
- Better: integrate latest observation z into proposal π

Example 3: Approximating Optimal π for Localization: Generating One Weighted Sample

Build Gaussian Approximation to Optimal Sequential Proposal

1. Initial guess $x_t'^i = f(x_{t-1}^i, u_t)$
2. Execute scan matching starting from the initial guess $\hat{x}_t^i$, resulting in pose estimate $x_t'^i$
3. Sample K points $\{x_1, \ldots, x_K\}$ in region around $\hat{x}_t^i$.
4. Proposal distribution is Gaussian

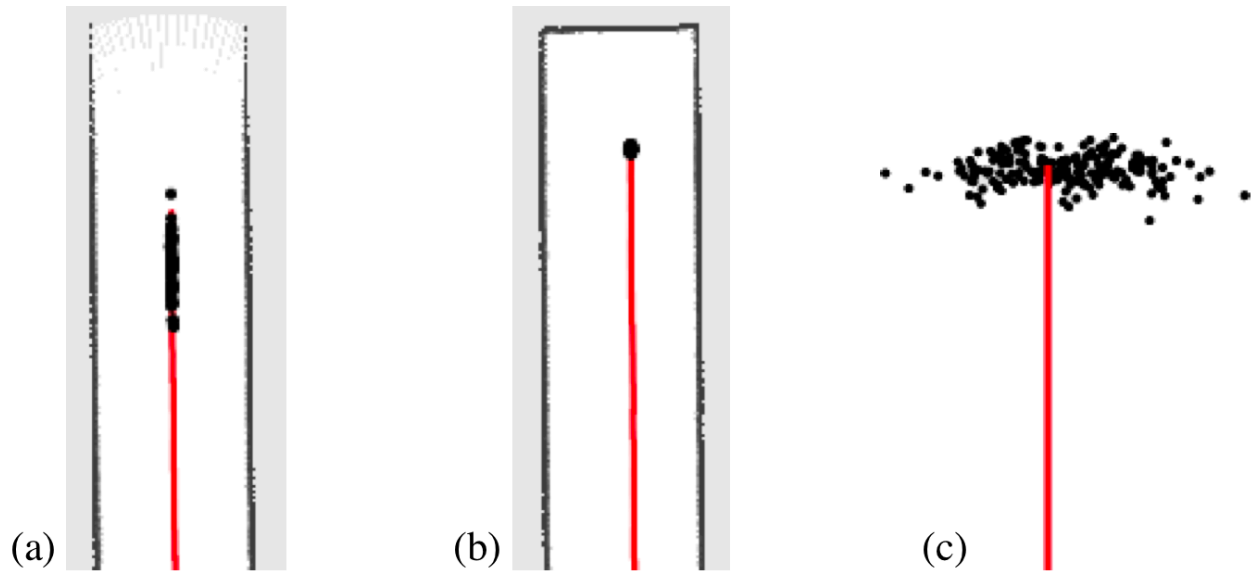
   with mean and covariance:

$$\begin{aligned} \mu_t^i &= \frac{1}{\eta^i} \sum_{j=1}^{K} x_j p(z_t | x_j, m) p(x_j | x_{t-1}^i, u_t) \\ \Sigma_t^i &= \frac{1}{\eta^i} \sum_{j=1}^{K} p(z_t | x_j, m) p(x_j | x_{t-1}^i, u_t)(x_j - \mu_t^i)(x_j - \mu_t^i)^\top \\ \eta^i &= \sum_{j=1}^{K} p(z_t | x_j, m) p(x_j | x_{t-1}^i, u_t) \end{aligned}$$

5. Sample from (approximately optimal) sequential proposal distribution.
6. Weight = $\int_{x'} p(z_t | x', m) p(x' | x_{t-1}^i, u_t) dx' \approx \eta^i$

Example 3: Example Particle Distributions

[Grisetti, Stachniss, Burgard, T-RO2006]



Particles generated from the approximately optimal proposal distribution. If using the standard motion model, in all three cases the particle set would have been similar to (c).

Resampling

- Consider running a particle filter for a system with deterministic dynamics and no sensors
- **Problem:**
  - While no information is obtained that favors one particle over another, due to resampling some particles will disappear and after running sufficiently long with very high probability all particles will have become identical.
  - On the surface it might look like the particle filter has uniquely determined the state.
- Resampling induces loss of diversity. The variance of the particles decreases, the variance of the particle set as an estimator of the true belief increases.

Resampling Solution I

- Effective sample size:

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (\tilde{w}_k^{(i)})^2}$$

Normalized weights

- Example:
  - All weights = 1/N → Effective sample size = N
  - All weights = 0, except for one weight = 1 → Effective sample size = 1

- Idea: resample only when effective sampling size is low

Resampling Solution I (ctd)

1. Importance Sampling

   - For $i = 1, \ldots, N$, sample $\tilde{x}_t^{(i)} \sim \pi(x_t | x_{0:t-1}^{(i)}, z_{0:t})$ and $\tilde{x}_{0:t}^{(i)} = \left(x_{0:t-1}^{(i)}, \tilde{x}_t^{(i)}\right)$.
   - For $i = 1, \ldots, N$, evaluate the importance weights up to a normalizing constant:

$$w_t^{*(i)} = w_{t-1}^{*(i)} \frac{p\left(z_t|\tilde{x}_t^{(i)}\right) p\left(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)}\right)}{\pi(\tilde{x}_t^{(i)}|\tilde{x}_{0:t-1}^{(i)}, z_{0:t})}$$

2. Resampling

   - For $i = 1, \ldots, N$, normalize the importance weights:

$$\tilde{w}_t^{(i)} = \frac{w^{*(i)}}{\sum_{j=1}^{N} w_t^{*(j)}}$$

   - $\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (\tilde{w}_t^{(i)})^2}$.

   If $\hat{N}_{\text{eff}} \geq N_{\text{thres}}$

   - $x_{0:t}^{(i)} = \tilde{x}_{0:t}^{(i)}$ for $i = 1, \ldots, N$.

   otherwise

   - For $i = 1, \ldots, N$, sample an index $j(i)$ distributed according to the discrete distribution with N elements satisfying $\Pr\{j(i) = l\} = \tilde{w}_t^{(i)}$ for $l = 1, \ldots, N$.
   - For $i = 1, \ldots, N$, $x_{0:t}^{j(i)}$ and $w_t^{(i)} = \frac{1}{N}$.

Resampling Solution II: Low Variance Sampling

- M = number of particles

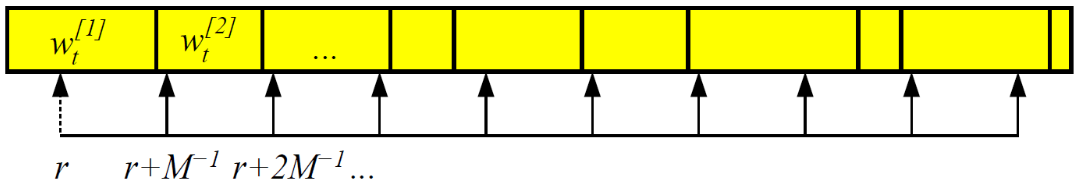


Figure 4.7 Principle of the low variance resampling procedure. We choose a random number r and then select those particles that correspond to $u = r + (m-1) \cdot M^{-1}$ where $m = 1, \ldots, M$.

  - r in [0, 1/M]

- Advantages:
  - More systematic coverage of space of samples
  - If all samples have same importance weight, no samples are lost
  - Lower computational complexity

Resampling Solution III

- Loss of diversity caused by resampling from a discrete distribution

- Solution: “regularization”
  - Consider the particles to represent a continuous density
  - Sample from the continuous density
  - E.g., given (1-D) particles $\{x^{(1)}, x^{(2)}, \ldots, x^{(K)}\}$

    sample from the density: $p(x) = \sum_{k=1}^{K} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-x^{(k)})^2}{\sigma^2}}$

Particle Deprivation

- = when there are no particles in the vicinity of the correct state
- Occurs as the result of the variance in random sampling. An unlucky series of random numbers can wipe out all particles near the true state. This has non-zero probability to happen at each time $\rightarrow$ will happen eventually.
- Popular solution: add a small number of randomly generated particles when resampling.
  - Advantages: reduces particle deprivation, simplicity.
  - Con: incorrect posterior estimate even in the limit of infinitely many particles.
  - Other benefit: initialization at time 0 might not have gotten anything near the true state, and not even near a state that over time could have evolved to be close to true state now; adding random samples will cut out particles that were not very consistent with past evidence anyway, and instead gives a new chance at getting close the true state.

Particle Deprivation: How Many Particles to Add?

- Simplest: Fixed number.
- Better way:
  - Monitor the probability of sensor measurements $p(z_t | z_{1:t-1}, u_{1:t}, m)$

which can be approximated by:

$$\frac{1}{N} \sum_{i=1}^{N} w_t^{(i)}$$

  - Average estimate over multiple time-steps and compare to typical values when having reasonable state estimates. If low, inject random particles.

```
1.  Algorithm Augmented_MCL($\mathcal{X}_{t-1}, u_t, z_t, m$):
2.    static $w_{\text{slow}}, w_{\text{fast}}$
3.    $w_{\text{avg}} = 0$
4.    $\bar{\mathcal{X}}_t = \mathcal{X}_t = \phi$
5.    for $i = 1$ to $N$ do
6.      $x_t^{(i)} = \text{sample\_motion\_model}(u_t, x_{t-1}^{(i)})$
7.      $w_t^{(i)} = \text{measurement\_model}(z_t, x_t^{(i)}, m)$
8.      $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{(i)}, w_t^{(i)} \rangle$
9.      $w_{\text{avg}} = w_{\text{avg}} + \frac{1}{N} w_t^{(i)}$
10.   endfor
11.   $w_{\text{slow}} = w_{\text{slow}} + \alpha_{\text{slow}}(w_{\text{avg}} - w_{\text{slow}})$
12.   $w_{\text{fast}} = w_{\text{fast}} + \alpha_{\text{fast}}(w_{\text{avg}} - w_{\text{fast}})$
13.   for $k = 1$ to $N$ do
14.     with probability $\max\{0.0, 1.0 - w_{\text{fast}}/w_{\text{slow}}\}$ do
15.       add random pose to $\mathcal{X}_t$
16.     else
17.       draw $i \in \{1, \ldots, N\}$ with probability $\propto w_t^{(i)}$
18.       add $x_t^{(i)}$ to $\mathcal{X}_t$
19.     endwith
20.   endfor
21.   return $\mathcal{X}_t$
```

Noise-free Sensors

- Consider a measurement obtained with a noise-free sensor, e.g., a noise-free laser-range finder---issue?
  - All particles would end up with weight zero, as it is very unlikely to have had a particle matching the measurement exactly.
- Solutions:
  - Artificially inflate amount of noise in sensors
  - Better proposal distribution (e.g., optimal sequential proposal)

Adapting Number of Particles: KLD-Sampling

- E.g., typically more particles need at the beginning of localization run
- Idea:
  - Partition the state-space
  - When sampling, keep track of number of bins occupied
  - Stop sampling when a threshold that depends on the number of occupied bins is reached
    - If all samples fall in a small number of bins $\rightarrow$ lower threshold

```
1.  Algorithm KLD_Sampling_MCL($\mathcal{X}_{t-1}, u_t, z_t, m, \epsilon, \delta$):
2.    $\mathcal{X}_t = \phi, M = 0, M_{\mathcal{X}} = 0, k = 0$
3.    for all $b \in H$ do
4.      $b =$ empty
5.    endfor
6.    do
7.      draw $i$ with probability $\propto w_{t-1}^{(i)}$
8.      $x_t^{(M)} =$ sample_motion_model($u_t, x_{t-1}^{(i)}$)
9.      $w_t^{(M)} =$ measurement_model($z_t, x_t^{(M)}, m$)
10.     $\mathcal{X}_t = \mathcal{X}_t + \langle (x_t^{(M)}, w_t^{(M)} \rangle$
11.     if $x_t^{(M)}$ falls into empty bin $b$ then
12.       $k = k + 1$
13.       $b =$ non-empty
14.       if $k > 1$ then
15.         $M_{\mathcal{X}} = \frac{k-1}{2\epsilon} \left( 1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} z_{1-\delta} \right)^3$
16.     endif
17.     $M = M + 1$
18.   while $M < M_{\mathcal{X}}$ or $M < M_{\mathcal{X}\,\min}$
19.   return $\mathcal{X}_t$
```

- z_{1-±}: the upper 1-± quantile of the standard normal distribution
- ± = 0.01 and ² = 0.05 works well in practice

KLD-sampling

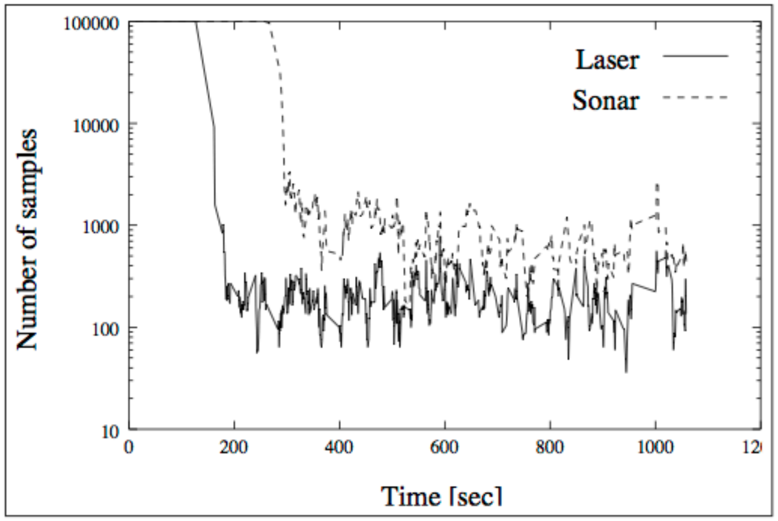


Figure 8.18 KLD-sampling: Typical evolution of number of samples for a global localization run, plotted against time (number of samples is shown on a log scale). The solid line shows the number of samples when using the robot's laser range-finder, the dashed graph is based on sonar sensor data.

KLD-sampling

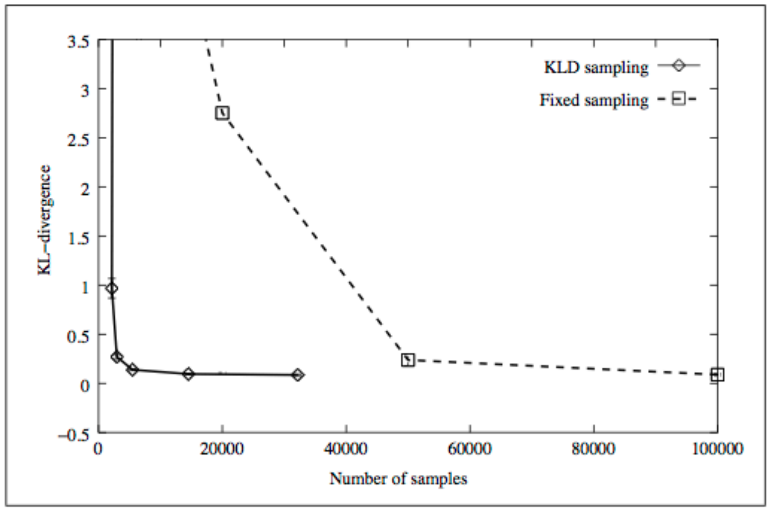


Figure 8.19 Comparison of KLD-sampling and MCL with fixed sample set sizes. The x-axis represents the average sample set size. The y-axis plots the KL-distance between the reference beliefs and the sample sets generated by the two approaches.