CS 547: Sensing and Planning in Robotics

Gauray S. Sukhatme

Computer Science
Robotic Embedded Systems Laboratory
University of Southern California

gaurav@usc.edu
http://robotics.usc.edu/~gaurav

Bayes Filters: Framework

• Given:

Stream of observations z and action data u:

$$d_t = \{u_1, z_1, \dots, u_t, z_t\}$$

- Sensor model P(z|x).
- Action model P(x|u,x').
- Prior probability of the system state P(x).

• Wanted:

- Estimate of the state X of a dynamical system.
- The posterior of the state is also called Belief:

$$Bel(x_t) = P(x_t | u_1, z_1, ..., u_t, z_t)$$

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

```
Algorithm Bayes_filter( Bel(x),d ):
1.
2.
     \eta=0
3.
      If d is a perceptual data item z then
4.
         For all x do
             Bel'(x) = P(z \mid x)Bel(x)
5.
             \eta = \eta + Bel'(x)
6.
7.
   For all x do
             Bel'(x) = \eta^{-1}Bel'(x)
8.
      Else if d is an action data item u then
9.
10.
         For all x do
             Bel'(x) = \int P(x \mid u, x') Bel(x') dx'
11.
      Return Bel'(x)
12.
```

Discrete Bayes Filter Algorithm

```
Algorithm Discrete_Bayes_filter( Bel(x),d ):
2.
     \eta=0
3.
     If d is a perceptual data item z then
4.
        For all x do
            Bel'(x) = P(z \mid x)Bel(x)
5.
            \eta = \eta + Bel'(x)
6.
7. For all x do
            Bel'(x) = \eta^{-1}Bel'(x)
8.
9.
     Else if d is an action data item u then
10.
        For all x do
             Bel'(x) = \sum P(x \mid u, x') Bel(x')
11.
12.
     Return Bel'(x)
```

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)

Bayes Filter Prediction/ Correction

Prediction

$$\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$$

Correction

$$bel(x_t) = \eta \ p(z_t \mid x_t) \overline{bel}(x_t)$$

Gaussians

$$p(x) \sim N(\mu, \sigma^2)$$
:

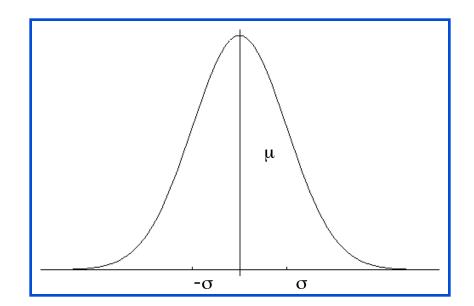
$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

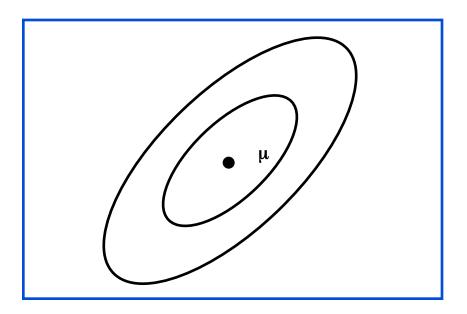
Univariate

$$p(\mathbf{x}) \sim \hat{I}(\hat{\mathbf{i}}, \hat{\mathbf{O}})$$
:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\mathbf{\acute{O}}|^{1/2}} e^{-\frac{1}{2} (\mathbf{x} - \mathbf{i})^t \mathbf{\acute{O}}^{-1} (\mathbf{x} - \mathbf{i})}$$

Multivariate





Properties of Gaussians

$$\begin{vmatrix} X_1 \sim N(\mu_1, \sigma_1^2) \\ X_2 \sim N(\mu_2, \sigma_2^2) \end{vmatrix} \Rightarrow p(X_1) \cdot p(X_2) \sim N \left(\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \mu_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \mu_2, \frac{1}{\sigma_1^{-2} + \sigma_2^{-2}} \right)$$

Multivariate Gaussians

$$\left. \begin{array}{l} X \sim N(\mu, \Sigma) \\ Y = AX + B \end{array} \right\} \quad \Rightarrow \quad Y \sim N(A\mu + B, A\Sigma A^{T})$$

$$\begin{vmatrix} X_1 \sim N(\mu_1, \Sigma_1) \\ X_2 \sim N(\mu_2, \Sigma_2) \end{vmatrix} \Rightarrow p(X_1) \cdot p(X_2) \sim N \left(\frac{\Sigma_2}{\Sigma_1 + \Sigma_2} \mu_1 + \frac{\Sigma_1}{\Sigma_1 + \Sigma_2} \mu_2, \frac{1}{\Sigma_1^{-1} + \Sigma_2^{-1}} \right)$$

 We stay in the "Gaussian world" as long as we start with Gaussians and perform only linear transformations.

Discrete Kalman Filter

Estimates the state x of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_{t} = A_{t} x_{t-1} + B_{t} u_{t} + \varepsilon_{t}$$

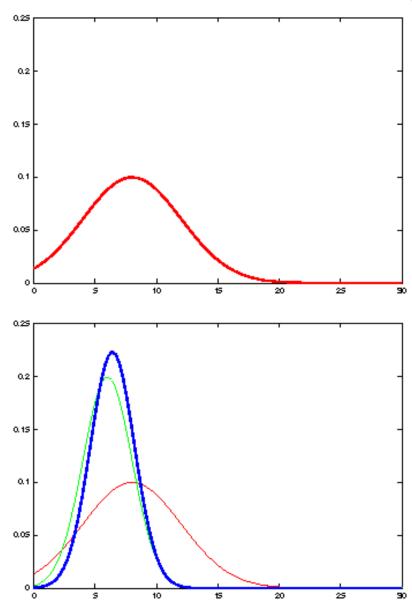
with a measurement

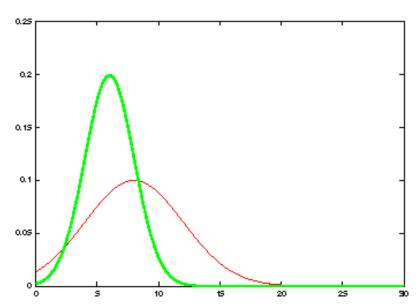
$$z_t = C_t x_t + \delta_t$$

Components of a Kalman Filter

- Matrix (nxn) that describes how the state evolves from t to t-1 without controls or noise.
- B_t Matrix (nxl) that describes how the control u_t changes the state from t to t-1.
- C_t Matrix (kxn) that describes how to map the state x_t to an observation z_t .
- Random variables representing the process and measurement noise that are assumed to be independent and normally distributed with covariance R_t and Q_t respectively.

Kalman Filter Updates in 1D

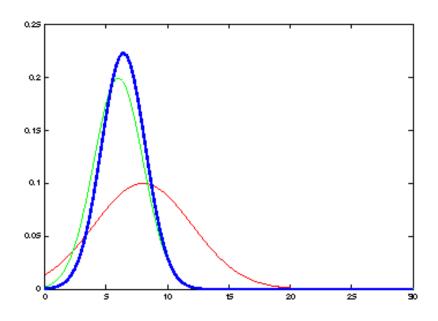




Kalman Filter Updates in 1D

$$bel(x_t) = \begin{cases} \mu_t = \overline{\mu}_t + K_t(z_t - \overline{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\overline{\sigma}_t^2 \end{cases} \quad \text{with} \quad K_t = \frac{\overline{\sigma}_t^2}{\overline{\sigma}_t^2 + \overline{\sigma}_{obs,t}^2}$$

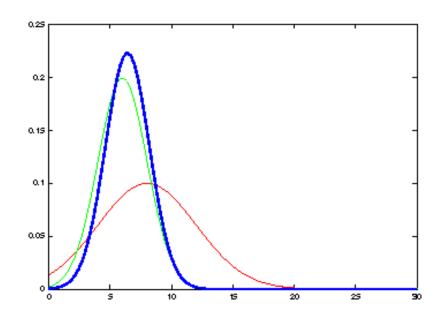
$$bel(x_t) = \begin{cases} \mu_t = \overline{\mu}_t + K_t(z_t - C_t \overline{\mu}_t) \\ \Sigma_t = (I - K_t C_t) \overline{\Sigma}_t \end{cases} \quad \text{with} \quad K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$$

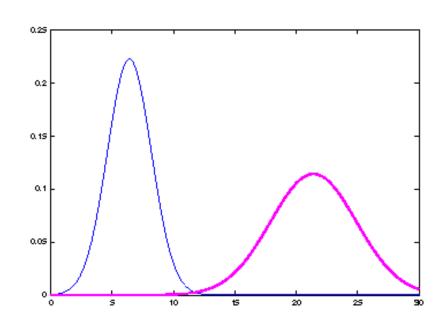


Kalman Filter Updates in 1D

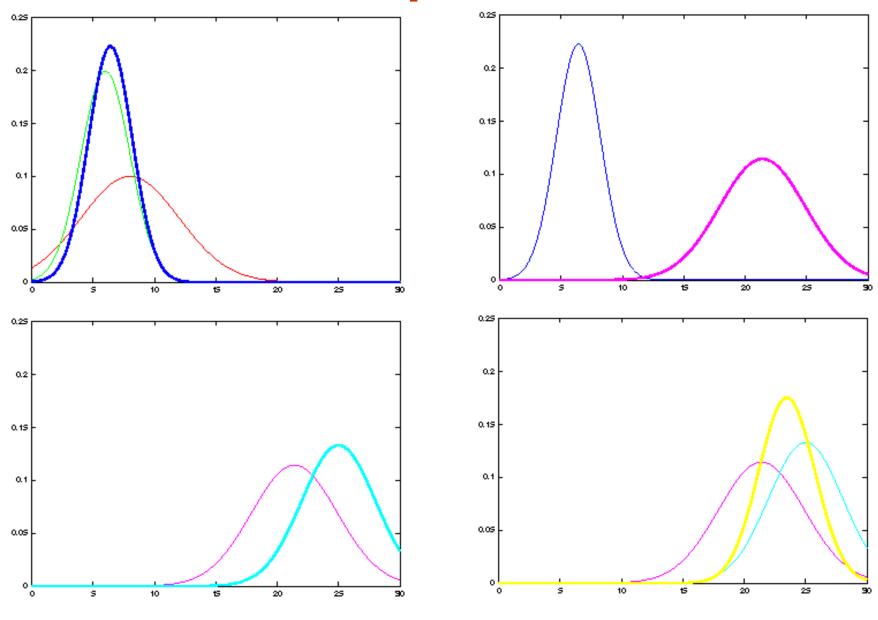
$$\overline{bel}(x_t) = \begin{cases} \overline{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \overline{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \overline{\mu}_t = A_t \mu_{t-1} + B_t \mu_t \\ \overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$





Kalman Filter Updates



Linear Gaussian Systems: Initialization

• Initial belief is normally distributed:

$$bel(x_0) = N(x_0; \mu_0, \Sigma_0)$$

Linear Gaussian Systems: Dynamics

 Dynamics are linear function of state and control plus additive noise:

$$x_{t} = A_{t}x_{t-1} + B_{t}u_{t} + \varepsilon_{t}$$

$$p(x_{t} | u_{t}, x_{t-1}) = N(x_{t}; A_{t}x_{t-1} + B_{t}u_{t}, R_{t})$$

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) \qquad bel(x_{t-1}) dx_{t-1}$$

$$\downarrow \qquad \qquad \qquad \downarrow$$

$$\sim N(x_t; A_t x_{t-1} + B_t u_t, R_t) \sim N(x_{t-1}; \mu_{t-1}, \Sigma_{t-1})$$

Linear Gaussian Systems: Dynamics

$$\overline{bel}(x_{t}) = \int p(x_{t} | u_{t}, x_{t-1}) \qquad bel(x_{t-1}) dx_{t-1}$$

$$\downarrow \qquad \qquad \downarrow \qquad \qquad$$

Linear Gaussian Systems: Observations

 Observations are linear function of state plus additive noise:

$$z_t = C_t x_t + \delta_t$$

$$p(z_t \mid x_t) = N(z_t; C_t x_t, Q_t)$$

$$bel(x_t) = \eta \quad p(z_t \mid x_t) \qquad \overline{bel}(x_t)$$

$$\downarrow \qquad \qquad \downarrow$$

$$\sim N(z_t; C_t x_t, Q_t) \qquad \sim N(x_t; \overline{\mu}_t, \overline{\Sigma}_t)$$

Linear Gaussian Systems: Observations

$$bel(x_{t}) = \eta \quad p(z_{t} \mid x_{t}) \qquad \overline{bel}(x_{t})$$

$$\sim N(z_{t}; C_{t}x_{t}, Q_{t}) \quad \sim N(x_{t}; \overline{\mu}_{t}, \overline{\Sigma}_{t})$$

$$\downarrow bel(x_{t}) = \eta \exp \left\{-\frac{1}{2}(z_{t} - C_{t}x_{t})^{T} Q_{t}^{-1}(z_{t} - C_{t}x_{t})\right\} \exp \left\{-\frac{1}{2}(x_{t} - \overline{\mu}_{t})^{T} \overline{\Sigma}_{t}^{-1}(x_{t} - \overline{\mu}_{t})\right\}$$

$$bel(x_{t}) = \begin{cases} \mu_{t} = \overline{\mu}_{t} + K_{t}(z_{t} - C_{t}\overline{\mu}_{t}) \\ \Sigma_{t} = (I - K_{t}C_{t})\overline{\Sigma}_{t} \end{cases} \quad \text{with} \quad K_{t} = \overline{\Sigma}_{t}C_{t}^{T}(C_{t}\overline{\Sigma}_{t}C_{t}^{T} + Q_{t})^{-1}$$

Kalman Filter Algorithm

- 1. Algorithm **Kalman_filter**(μ_{t-1} , Σ_{t-1} , u_t , z_t):
- 2. Prediction:

$$\mathbf{3.} \qquad \boldsymbol{\mu}_t = A_t \boldsymbol{\mu}_{t-1} + B_t \boldsymbol{u}_t$$

$$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

5. Correction:

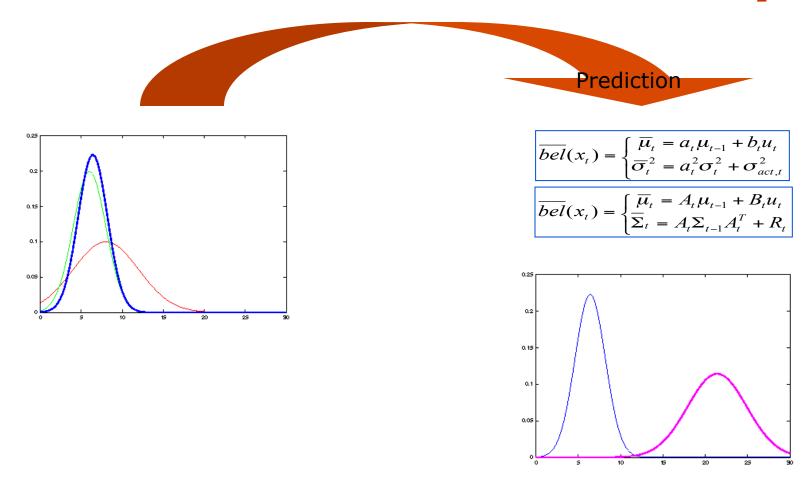
$$6. K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$$

7.
$$\mu_t = \overline{\mu}_t + K_t(z_t - C_t \overline{\mu}_t)$$

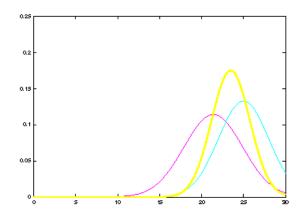
$$\mathbf{8.} \qquad \boldsymbol{\Sigma}_t = (I - K_t C_t) \overline{\boldsymbol{\Sigma}}_t$$

9. Return μ_t , Σ_t

The Prediction-Correction-Cycle

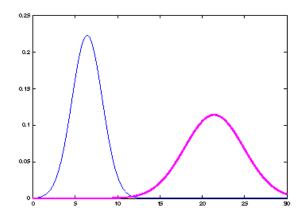


The Prediction-Correction-Cycle



$$bel(x_t) = \begin{cases} \mu_t = \overline{\mu}_t + K_t(z_t - \overline{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\overline{\sigma}_t^2 \end{cases}, K_t = \frac{\overline{\sigma}_t^2}{\overline{\sigma}_t^2 + \overline{\sigma}_{obs,t}^2}$$

$$bel(x_t) = \begin{cases} \mu_t = \overline{\mu}_t + K_t(z_t - C_t \overline{\mu}_t) \\ \Sigma_t = (I - K_t C_t) \overline{\Sigma}_t \end{cases}, K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$$



Correction

Slide courtesy of S. Thrun, D. Fox and W. Burgard

The Prediction-Correction-Cycle



$$bel(x_t) = \begin{cases} \mu_t = \overline{\mu}_t + K_t(z_t - \overline{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\overline{\sigma}_t^2 \end{cases}, K_t = \frac{\overline{\sigma}_t^2}{\overline{\sigma}_t^2 + \overline{\sigma}_{obs,t}^2}$$

$$bel(x_t) = \begin{cases} \mu_t = \overline{\mu}_t + K_t(z_t - C_t \overline{\mu}_t) \\ \Sigma_t = (I - K_t C_t) \overline{\Sigma}_t \end{cases}, K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$$

$$\overline{bel}(x_t) = \begin{cases} \overline{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \overline{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \overline{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \overline{\mu}_t = A_t \mu_{t-1} + B_t \mu_t \\ \overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$



Kalman Filter Summary

 Highly efficient: Polynomial in measurement dimensionality k and state dimensionality *n*:

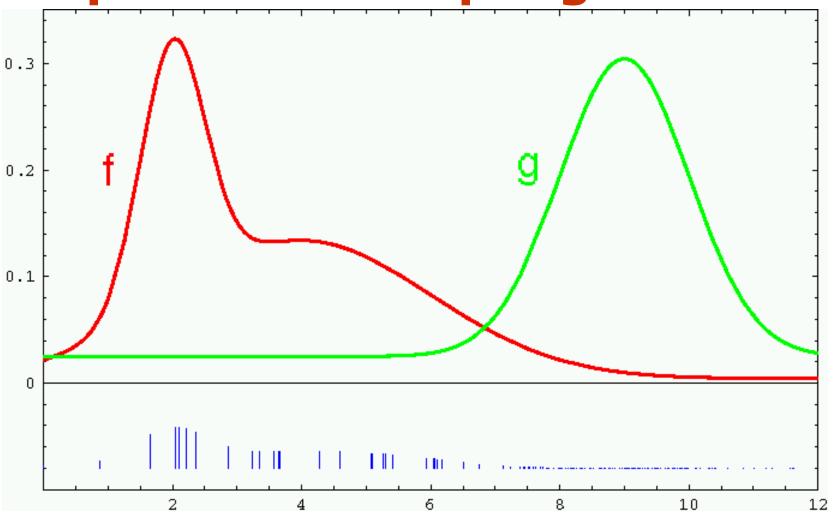
 $O(k^{2.376} + n^2)$

- Optimal for linear Gaussian systems!
- Most robotics systems are nonlinear!

Particle Filters

- Represent belief by random samples
- Estimation of non-Gaussian, nonlinear processes
- Monte Carlo filter, Survival of the fittest,
 Condensation, Bootstrap filter, Particle filter
- Filtering: [Rubin, 88], [Gordon et al., 93], [Kitagawa 96]
- Computer vision: [Isard and Blake 96, 98]
- Dynamic Bayesian Networks: [Kanazawa et al., 95]d

Importance Sampling



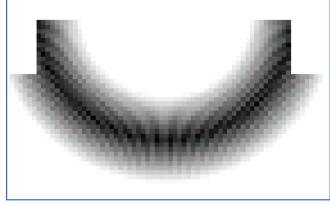
Weight samples: w = f/g

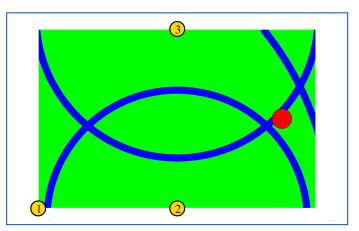
Importance Sampling with Resampling: Landmark Detection Example

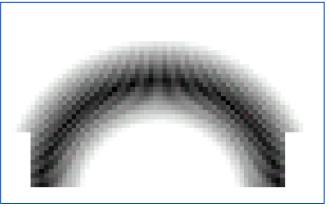


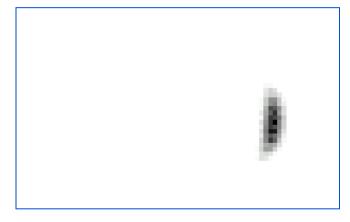
Distributions



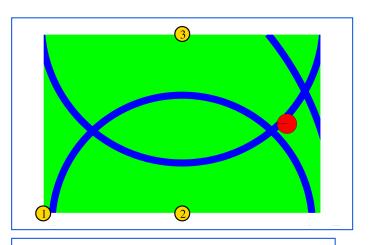






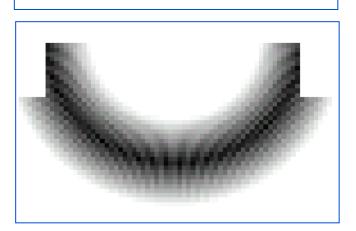


Distributions



Wanted: samples distributed according to

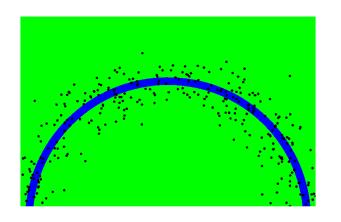
$$p(x | z_1, z_2, z_3)$$

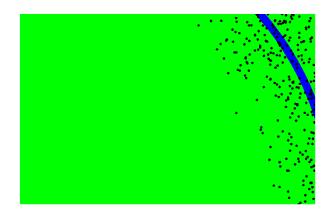


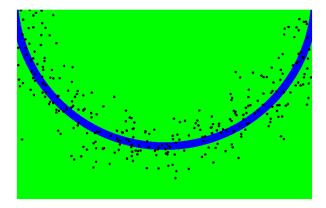


This is Easy!

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







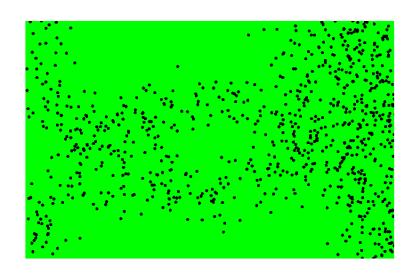
Importance Sampling with Resampling

Target distribution
$$f : p(x | z_1, z_2, ..., z_n) = \frac{\prod_{k} p(z_k | x) \quad p(x)}{p(z_1, z_2, ..., z_n)}$$

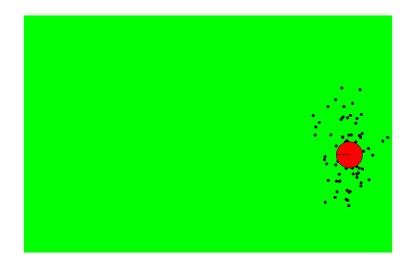
Sampling distribution
$$g: p(x | z_l) = \frac{p(z_l | x)p(x)}{p(z_l)}$$

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

Importance Sampling with Resampling

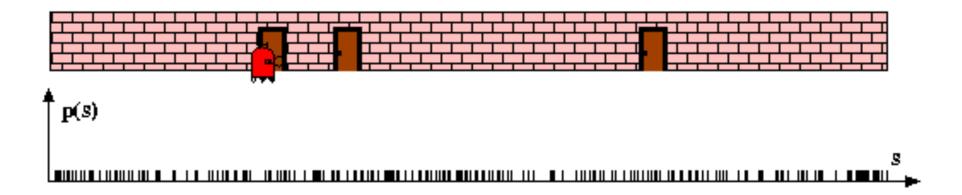


Weighted samples



After resampling

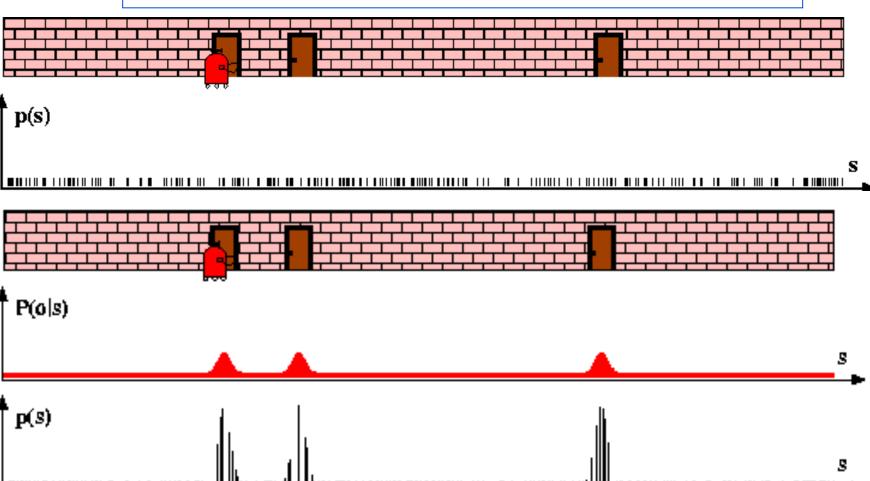
Particle Filters



Sensor Information: Importance Sampling

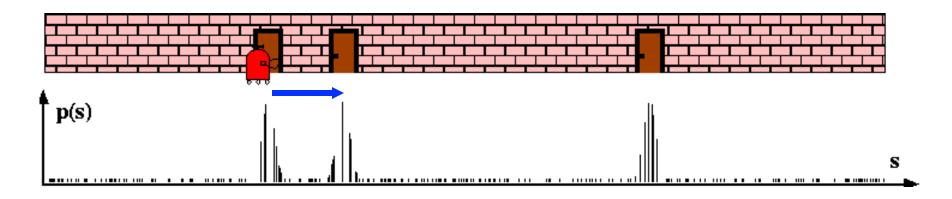
$$Bel(x) \leftarrow \alpha \ p(z \mid x) \ Bel^{-}(x)$$

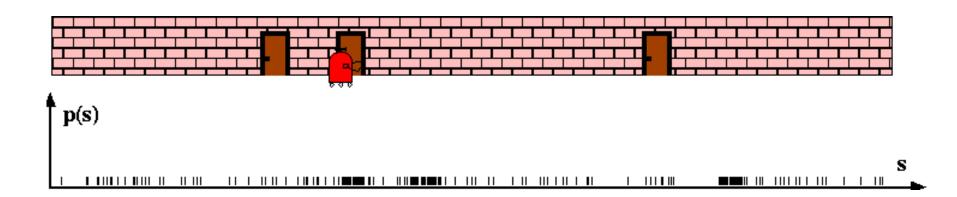
$$w \leftarrow \frac{\alpha \ p(z \mid x) \ Bel^{-}(x)}{Bel^{-}(x)} = \alpha \ p(z \mid x)$$



Robot Motion

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

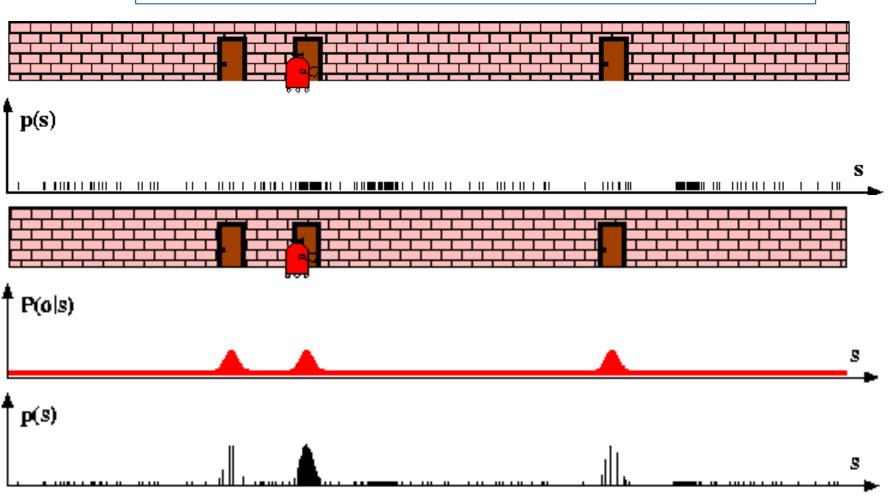




Sensor Information: Importance Sampling

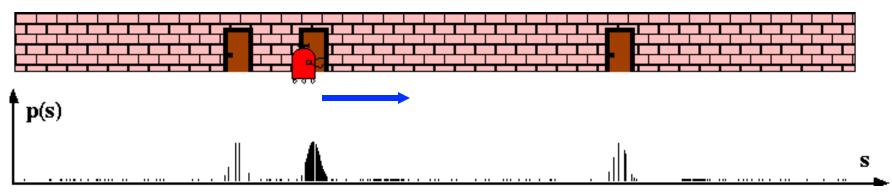
$$Bel(x) \leftarrow \alpha \ p(z \mid x) \ Bel^{-}(x)$$

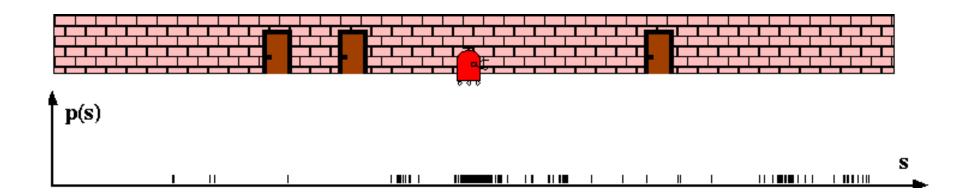
$$w \leftarrow \frac{\alpha \ p(z \mid x) \ Bel^{-}(x)}{Bel^{-}(x)} = \alpha \ p(z \mid x)$$



Robot Motion

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





Particle Filter Algorithm

- 1. Algorithm **particle_filter**(S_{t-1} , U_{t-1} Z_t):
- 2. $S_t = \emptyset$, $\eta = 0$
- 3. For i = 1...n

Generate new samples

- 4. Sample index j(i) from the discrete distribution given by w_{t-1}
- 5. Sample x_t^i from $p(x_t | x_{t-1}, u_{t-1})$ 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. $S_t = S_t \cup \{\langle x_t^i, w_t^i \rangle\}$

Insert

- 9. **For** i = 1...n
- 10. $w_t^i = w_t^i / \eta$

Normalize weights

Particle Filter Algorithm

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

$$\rightarrow \text{draw } x^{i}_{t-1} \text{ from } Bel(x_{t-1})$$

$$\rightarrow \text{draw } x^{i}_{t} \text{ from } p(x_{t} \mid x^{i}_{t-1}, u_{t-1})$$

$$\Rightarrow \text{Importance factor for } x^{i}_{t}$$

$$w^{i}_{t} = \frac{\text{target distribution}}{\text{proposal distribution}}$$

$$= \frac{\eta \ p(z_{t} \mid x_{t}) \ p(x_{t} \mid x_{t-1}, u_{t-1}) \ Bel(x_{t-1})}{p(x_{t} \mid x_{t-1}, u_{t-1}) \ Bel(x_{t-1})}$$

$$\propto p(z_{t} \mid x_{t})$$

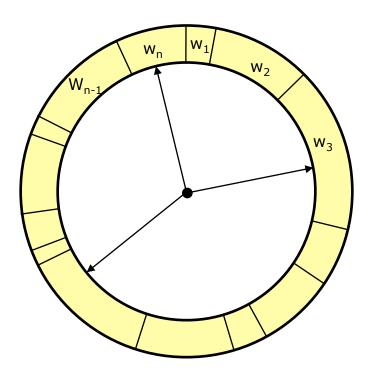
Resampling

Given: Set S of weighted samples.

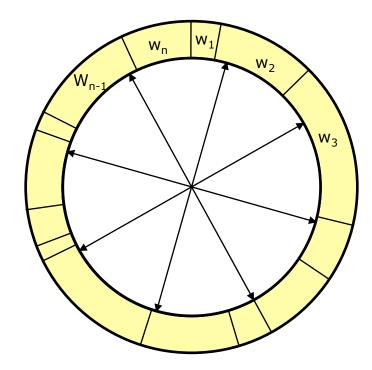
• Wanted: Random sample, where the probability of drawing x_i is given by w_i .

 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

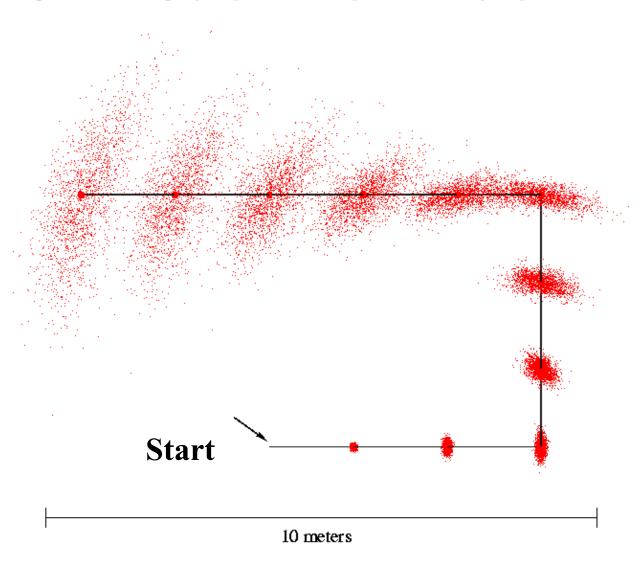
Resampling Algorithm

```
1. Algorithm systematic_resampling(S,n):
2. S' = \emptyset, c_1 = w^1
3. For i = 2...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...n
                    Draw samples ...
7. While (u_i > c_i) Skip until next threshold reached
8. i = i + 1

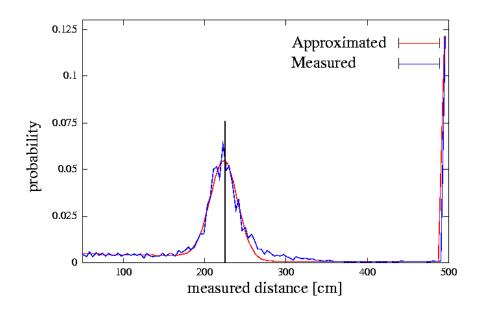
9. S' = S' \cup \{ < x^i, n^{-1} > \} Insert
10. u_{i+1} = u_i + n^{-1}
                         Increment threshold
```

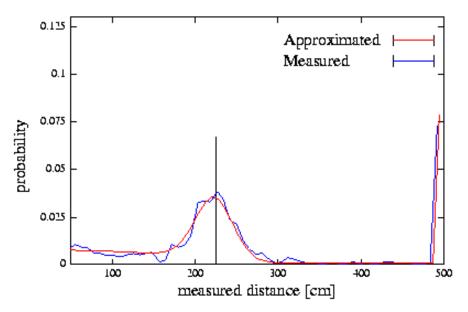
11. Return S'

Motion Model Reminder



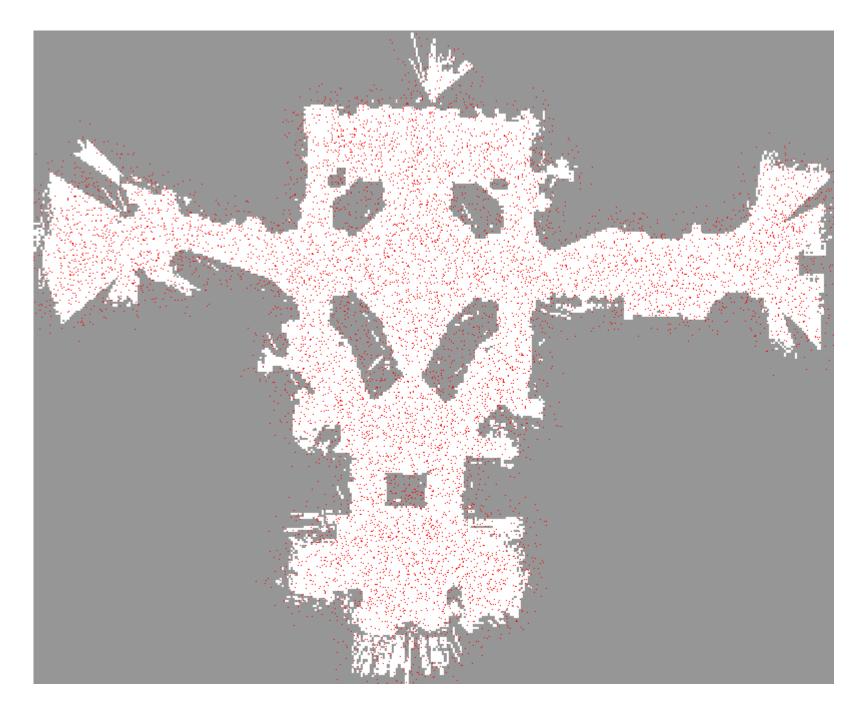
Proximity Sensor Model Reminder

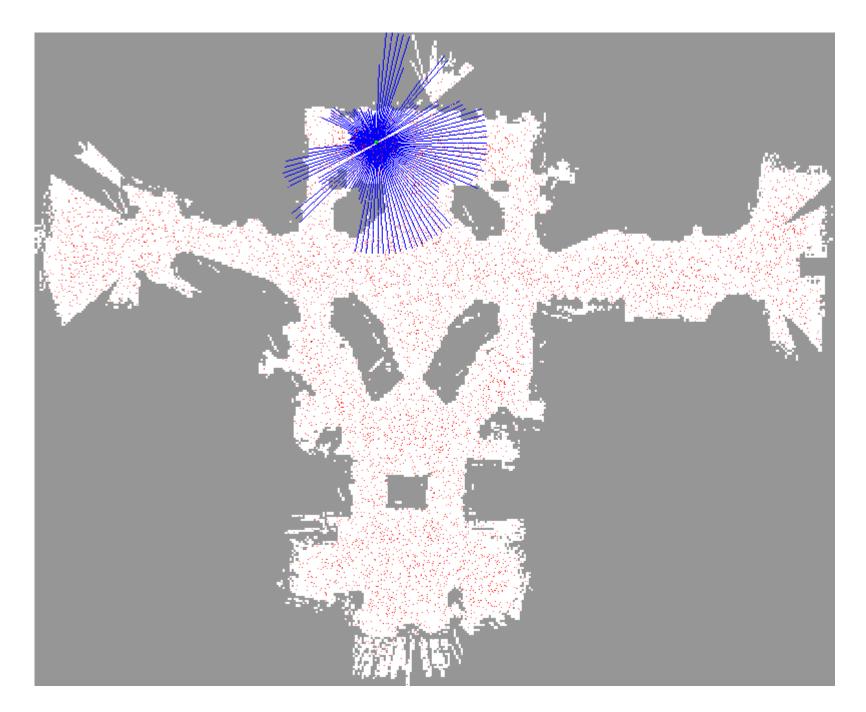


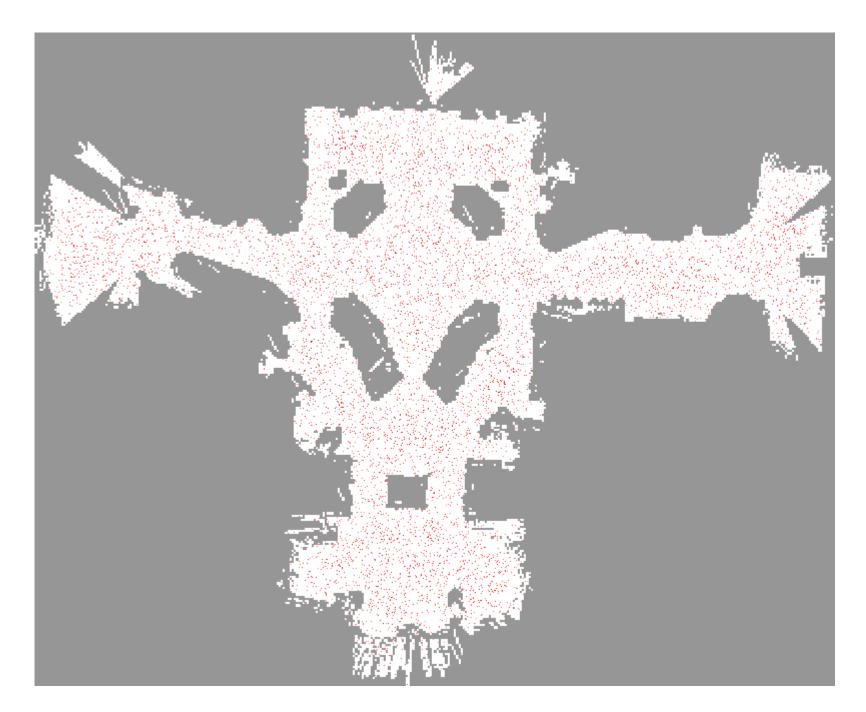


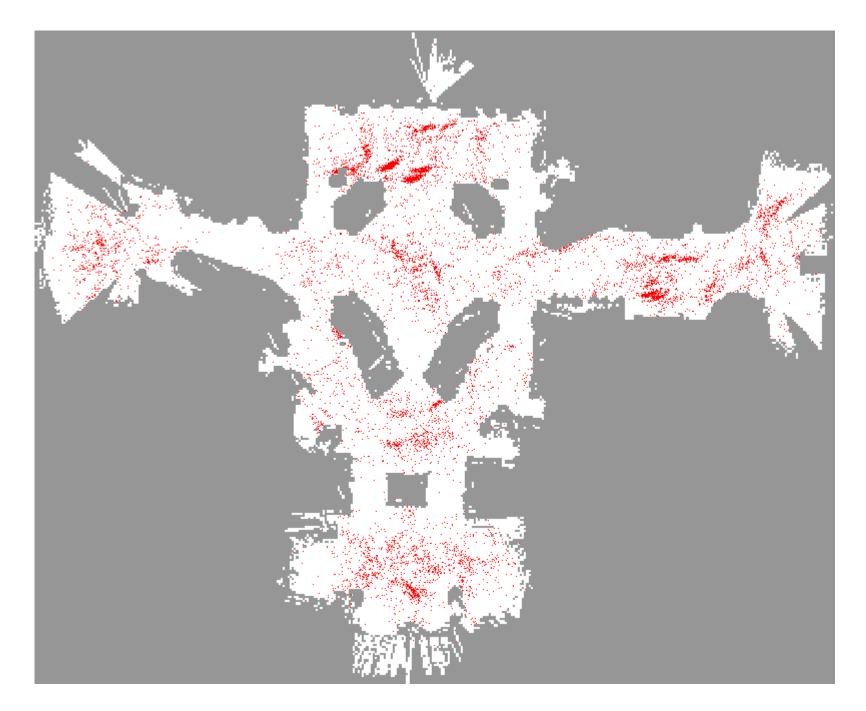
Laser sensor

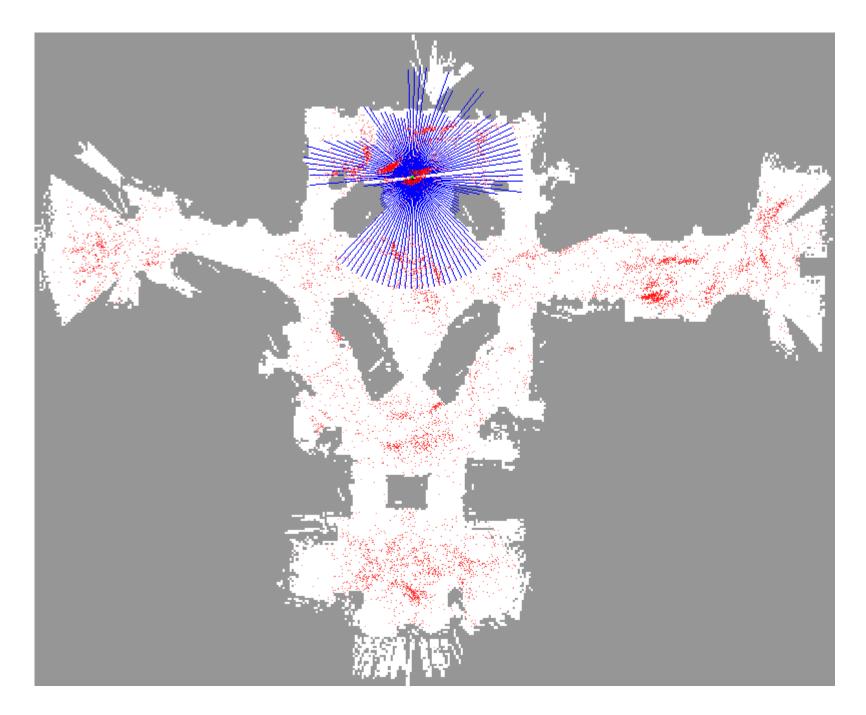
Sonar sensor

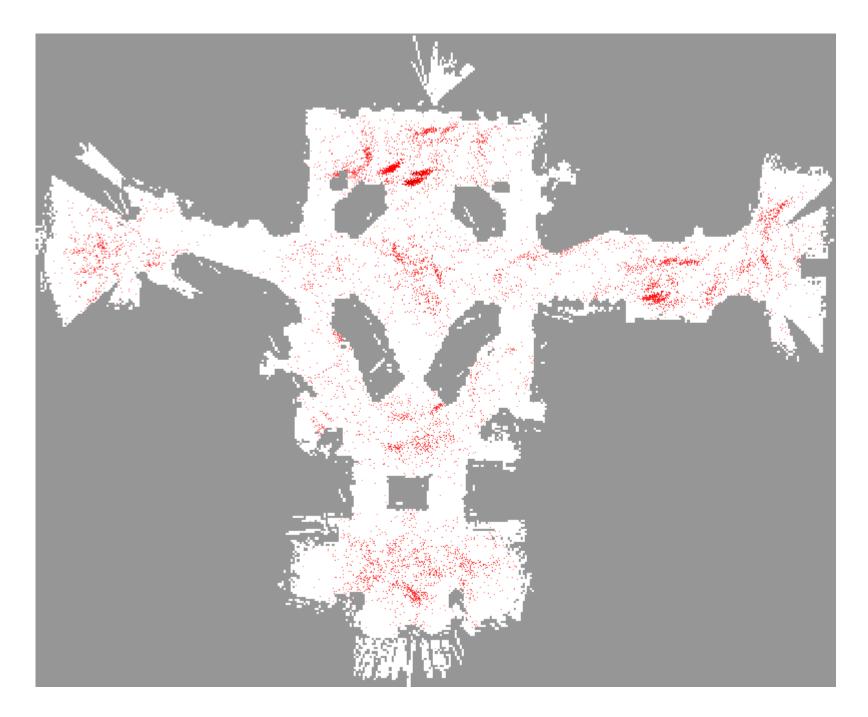


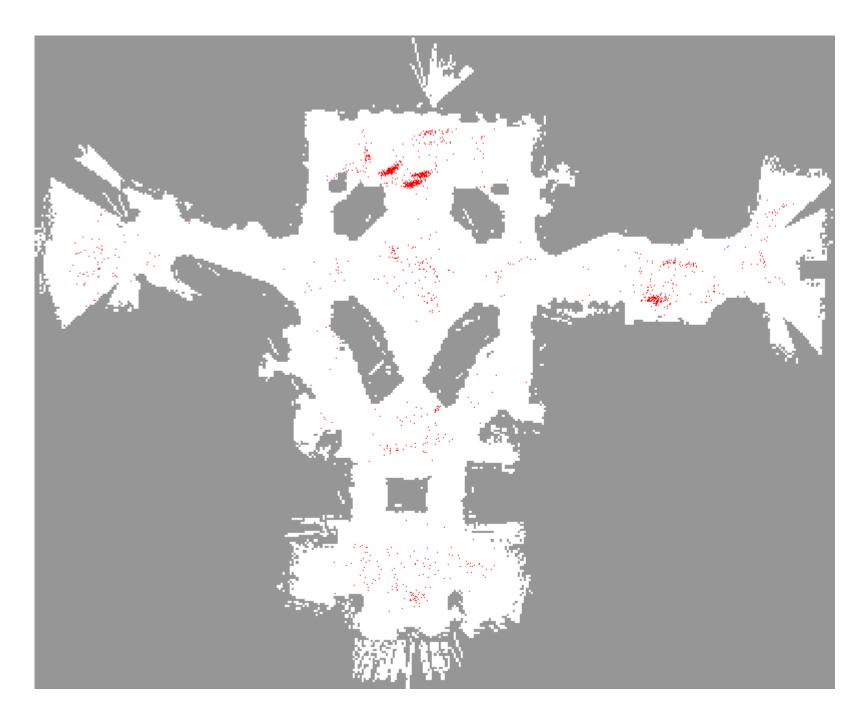


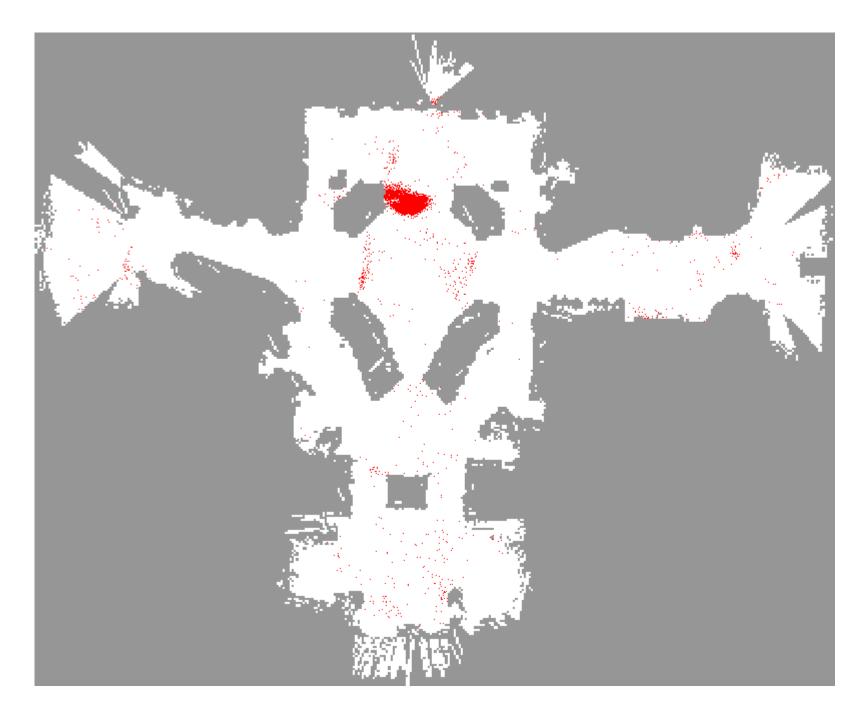


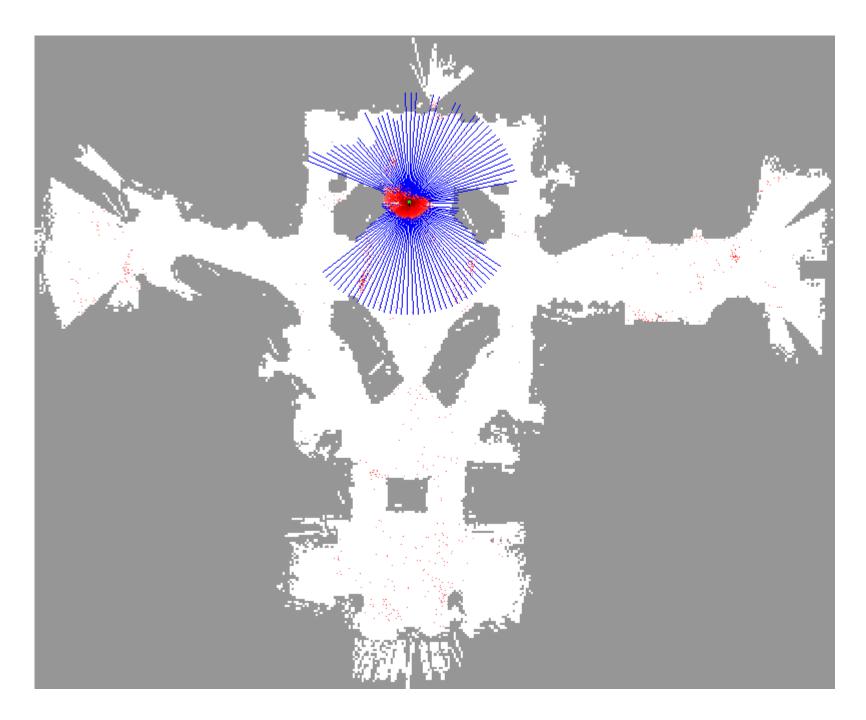


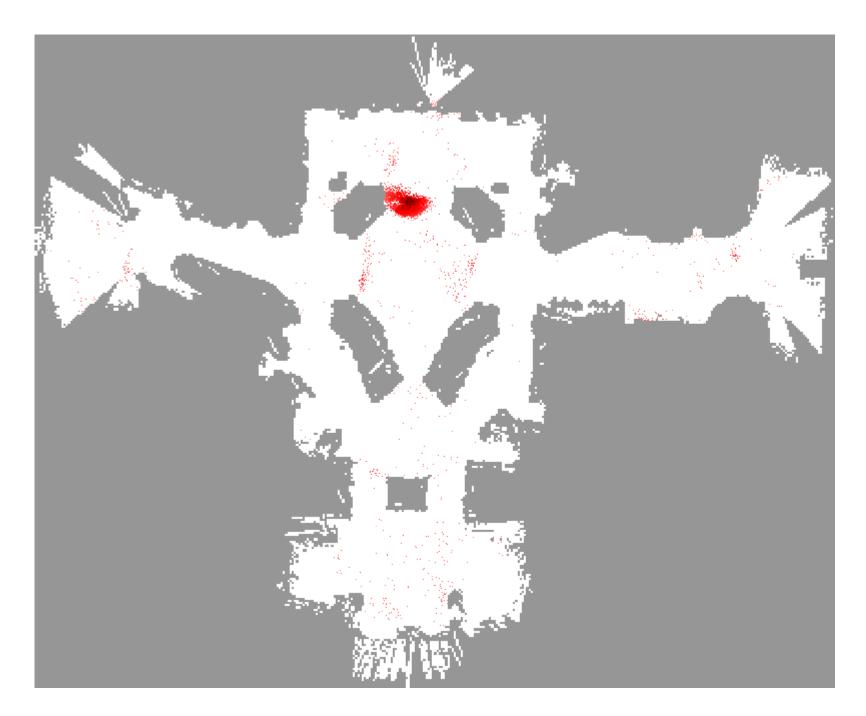


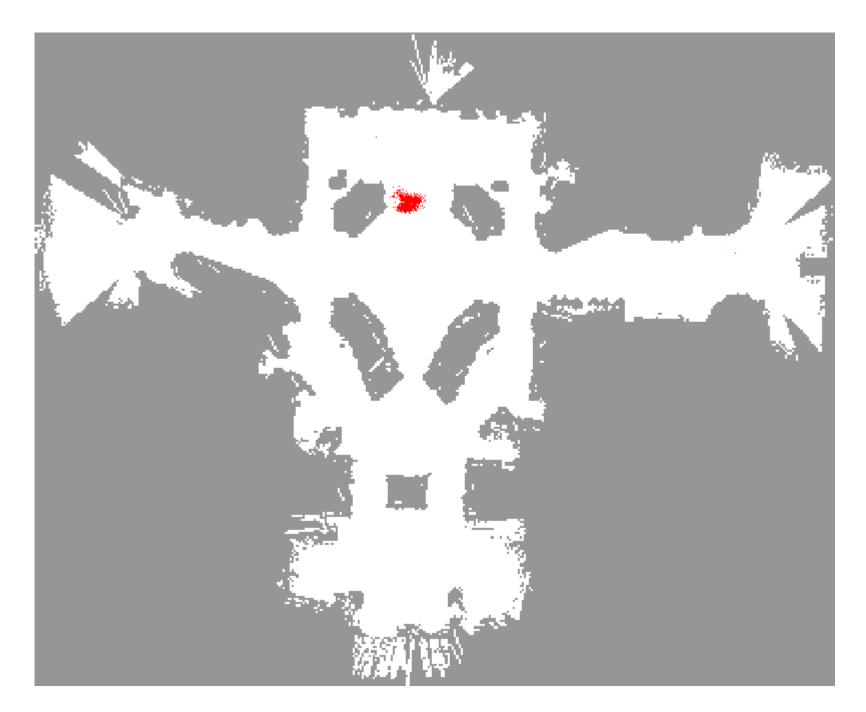


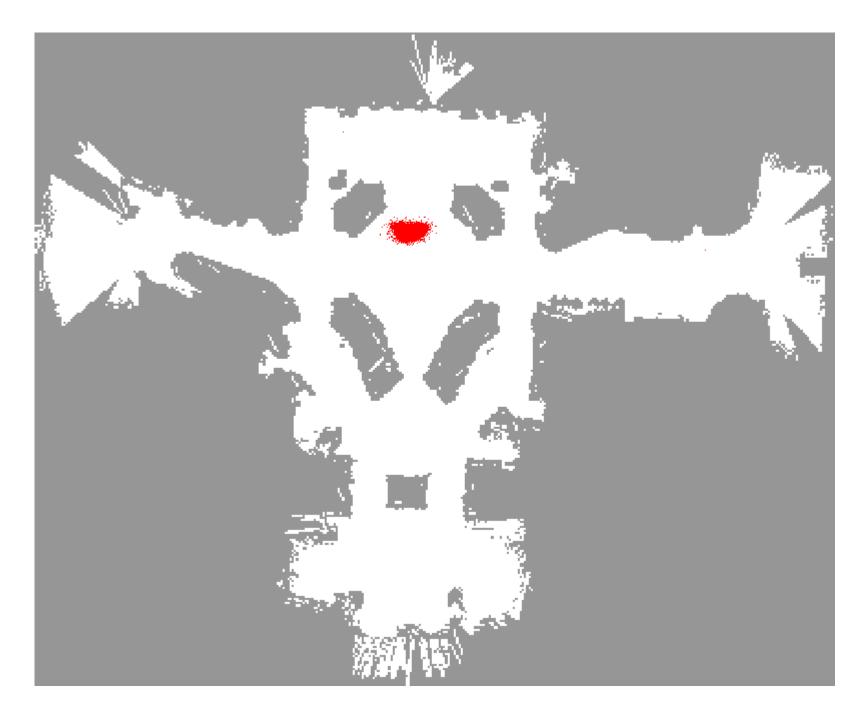


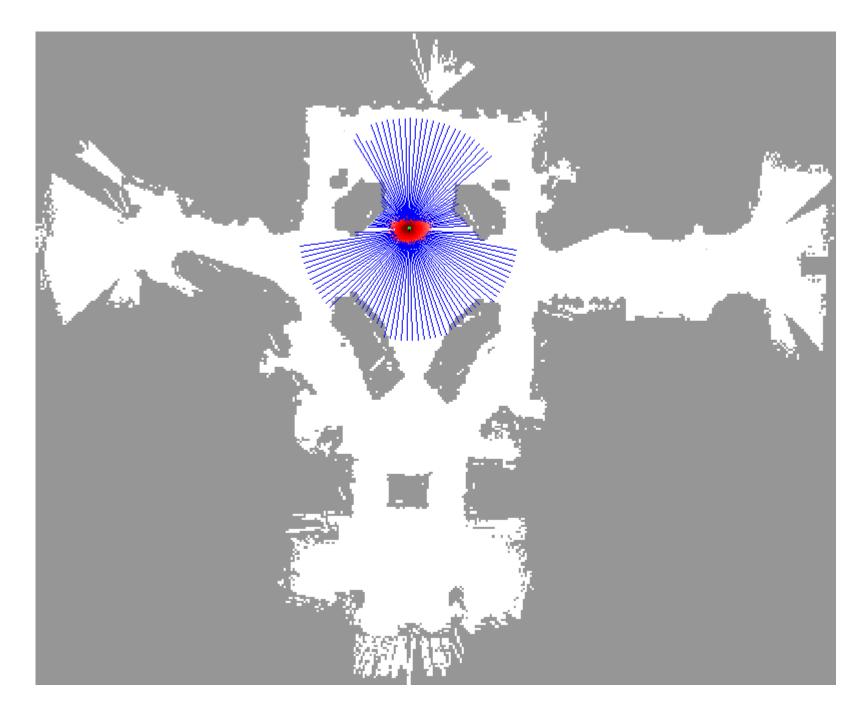


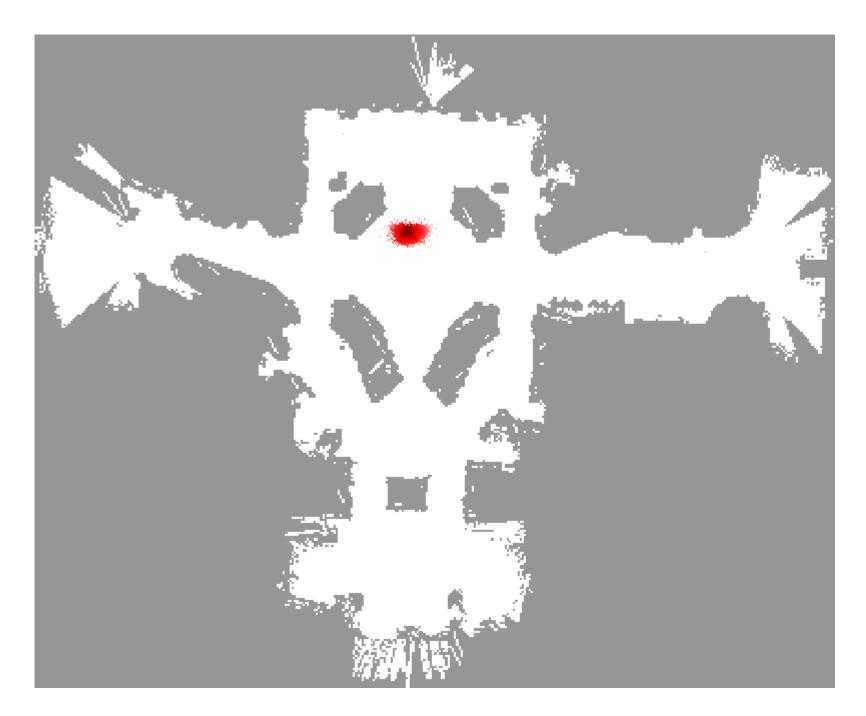


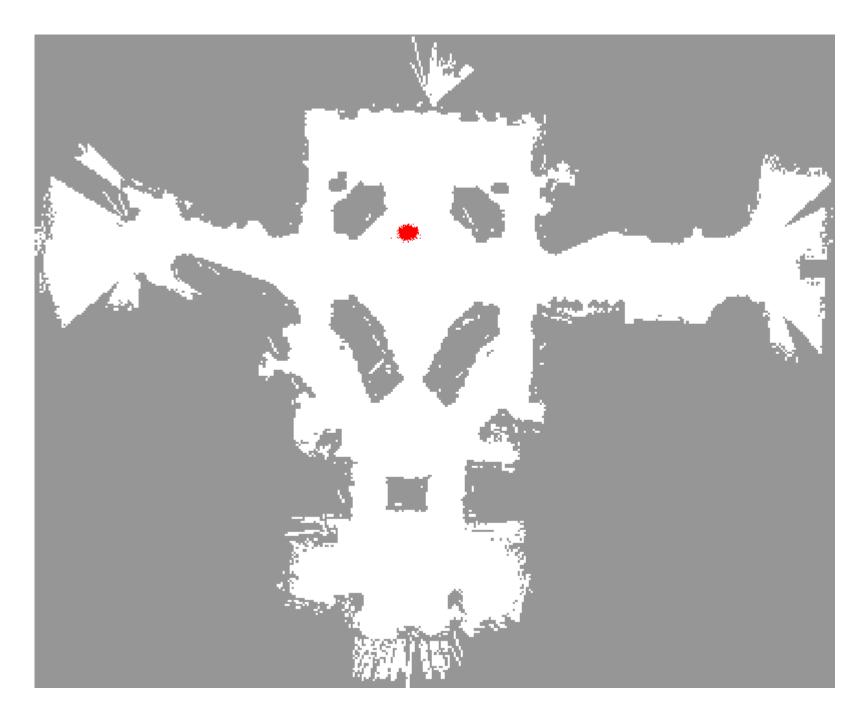


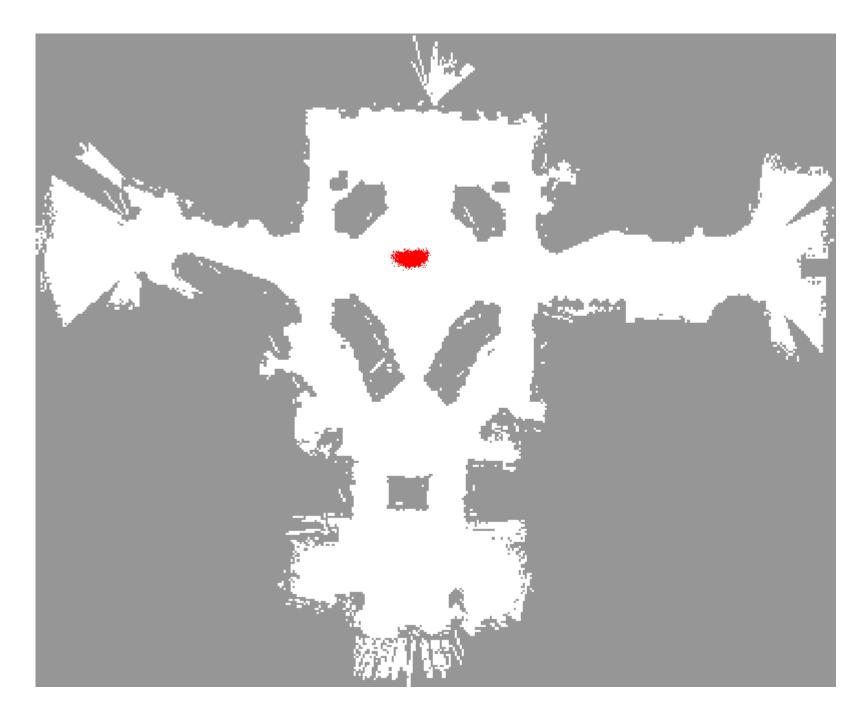


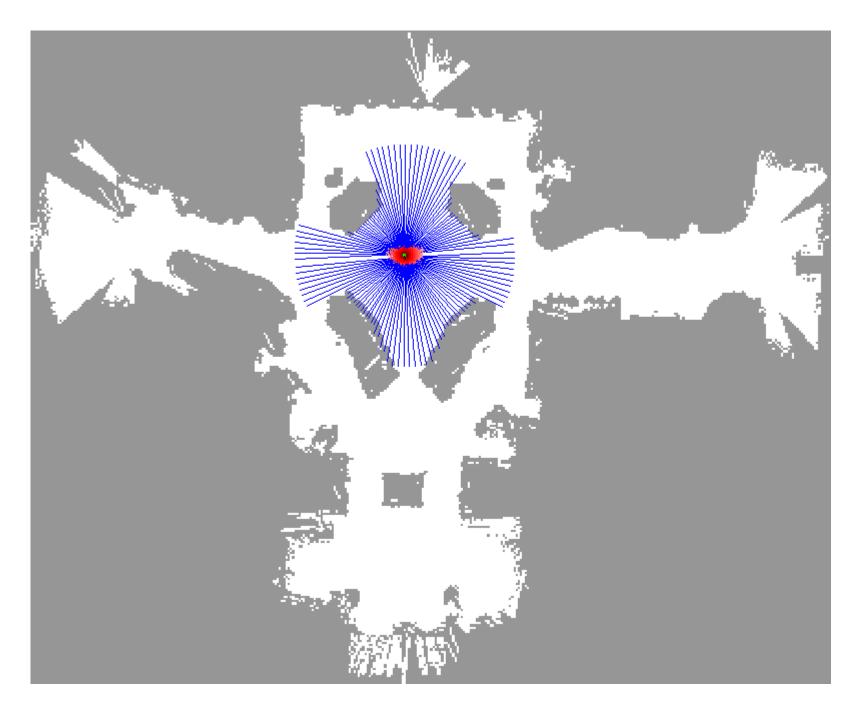


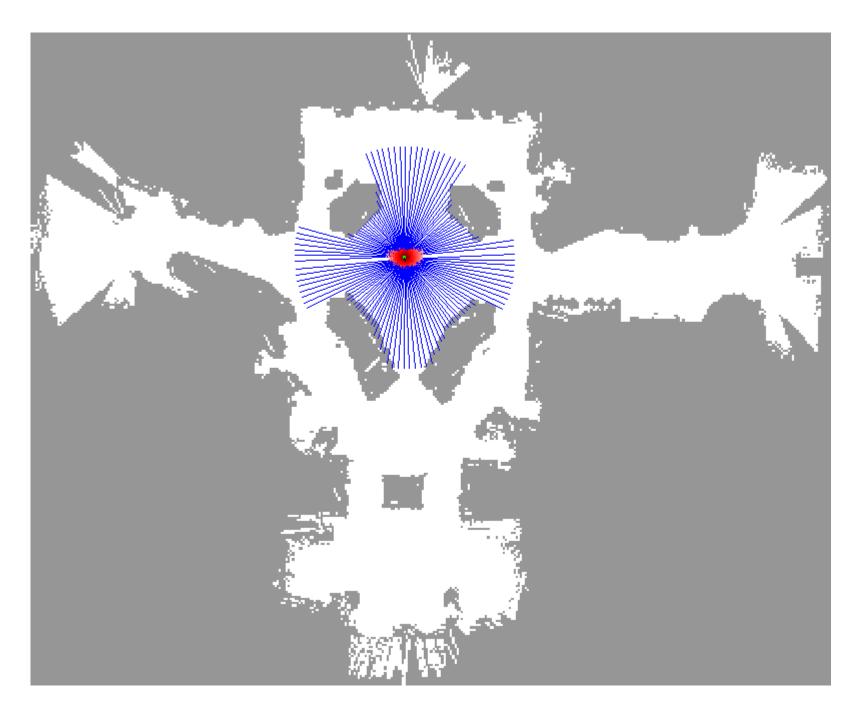












Summary

- Particle filters are an implementation 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 a re-sampling step, new particles are drawn with a probability proportional to the likelihood of the observation.