Autonomous Mobile Robot Design

Topic: Particle Filter for Localization

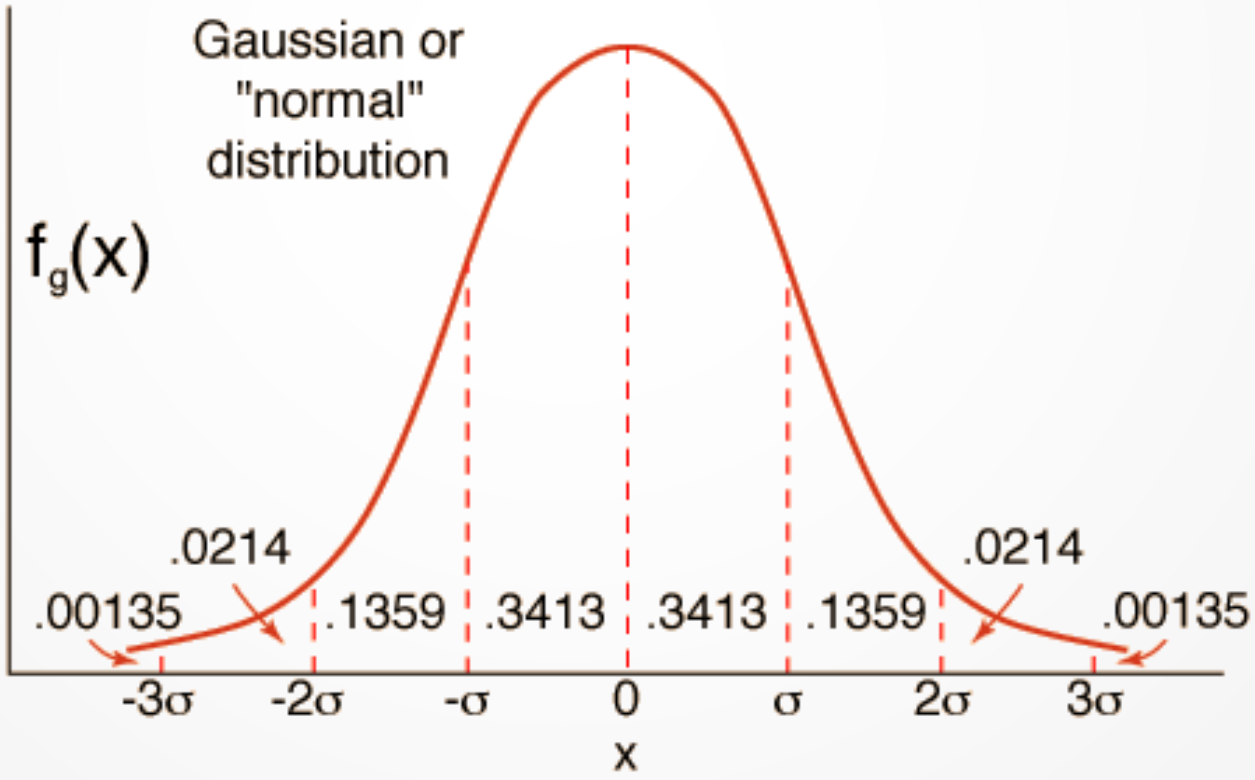
Dr. Kostas Alexis (CSE)

These slides relied on the lectures from C. Stachniss, and the book "Probabilistic Robotics" from Thurn et al.

Gaussian Filters

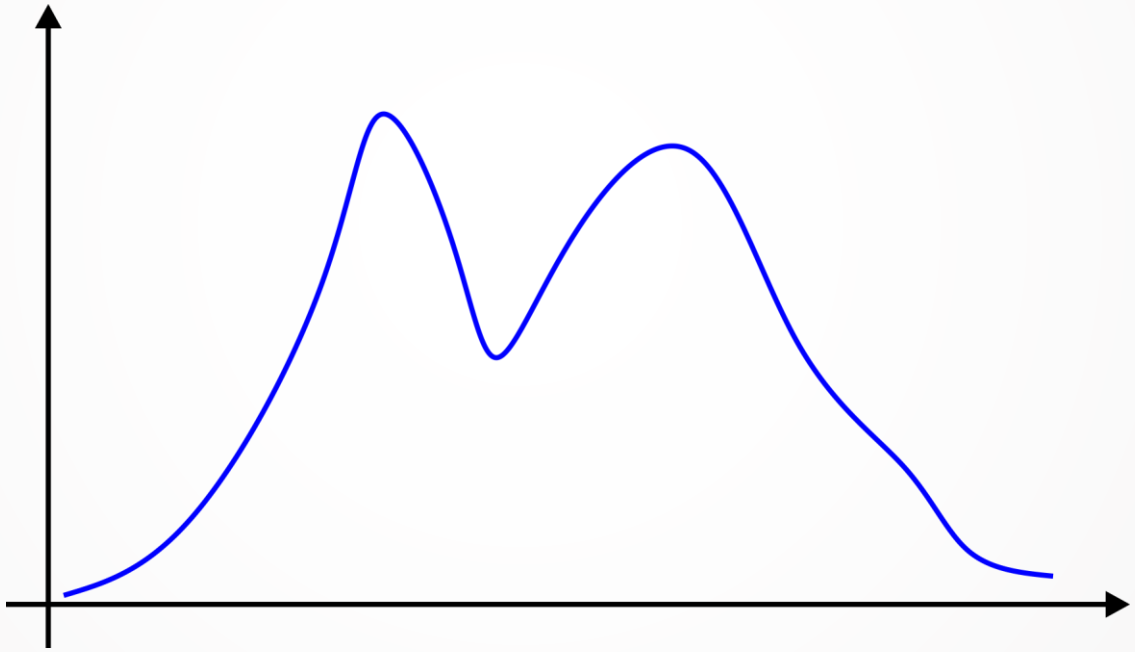
- The Kalman Filter (and its variants) can only model **Gaussian distributions**.

$$p(x) = det(2\pi\Sigma)^{-\frac{1}{2}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$$



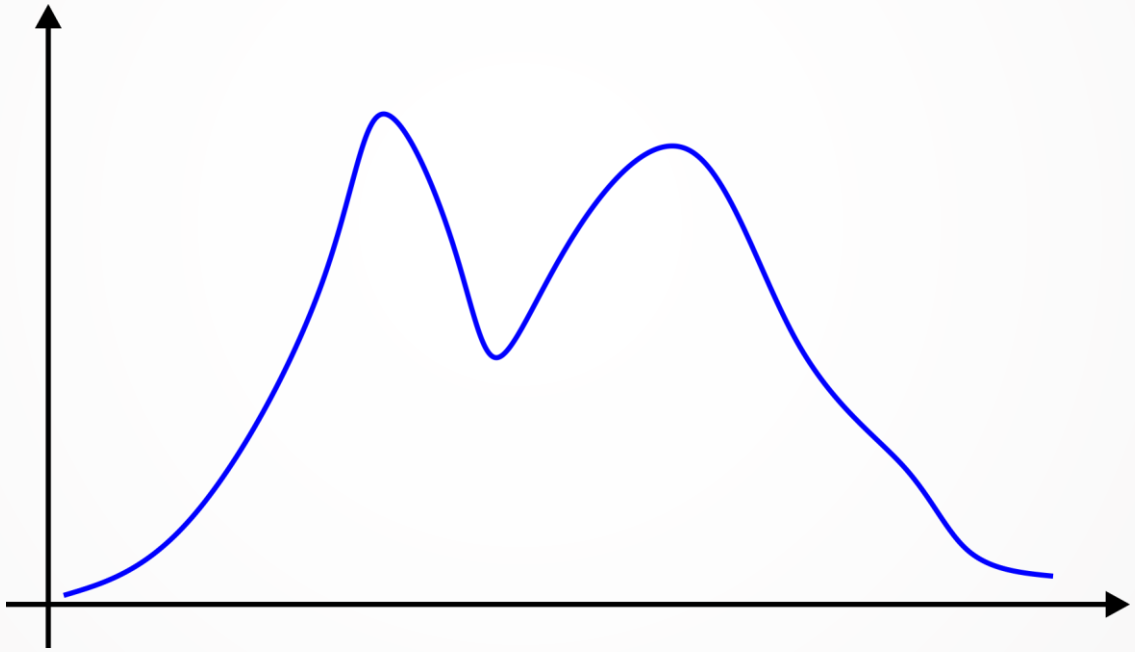
Motivation

- **Goal:** enable us to deal with arbitrary distributions



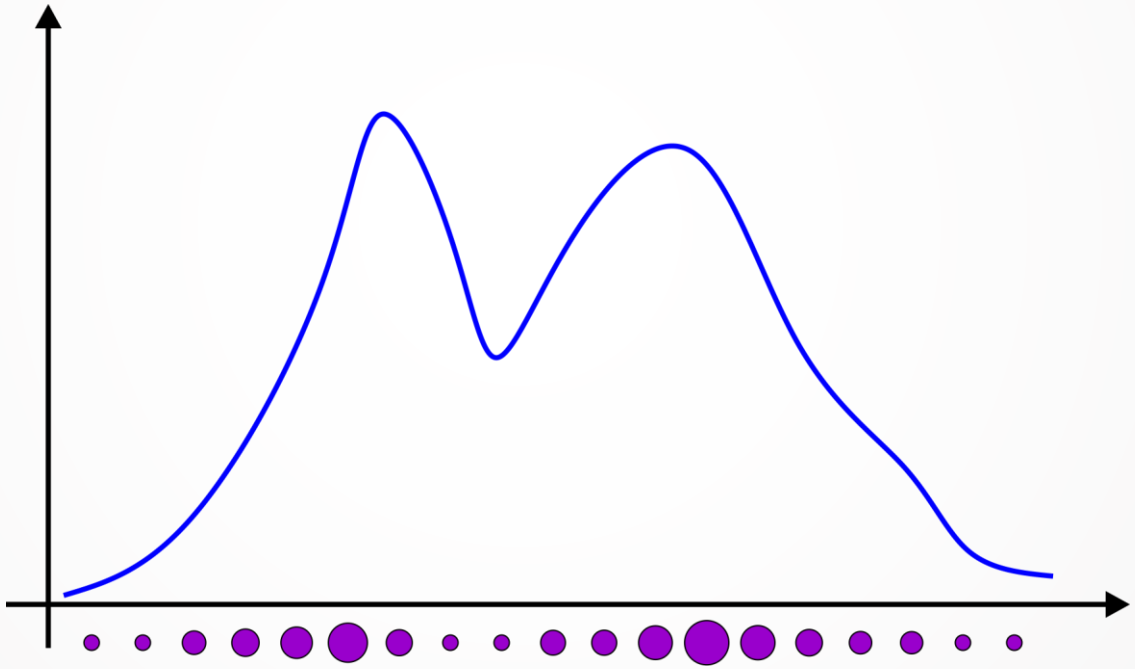
Key idea: Samples

- Use **multiple samples** to represent arbitrary distributions



Key idea: Samples

- Use **multiple samples** to represent arbitrary distributions



Particle Set

- Set of **weighted samples**

$$X = \{\langle x^{[j]}, w^{[j]} \rangle\}_{j=1,\ldots,J}$$

state hypothesis **importance weight**

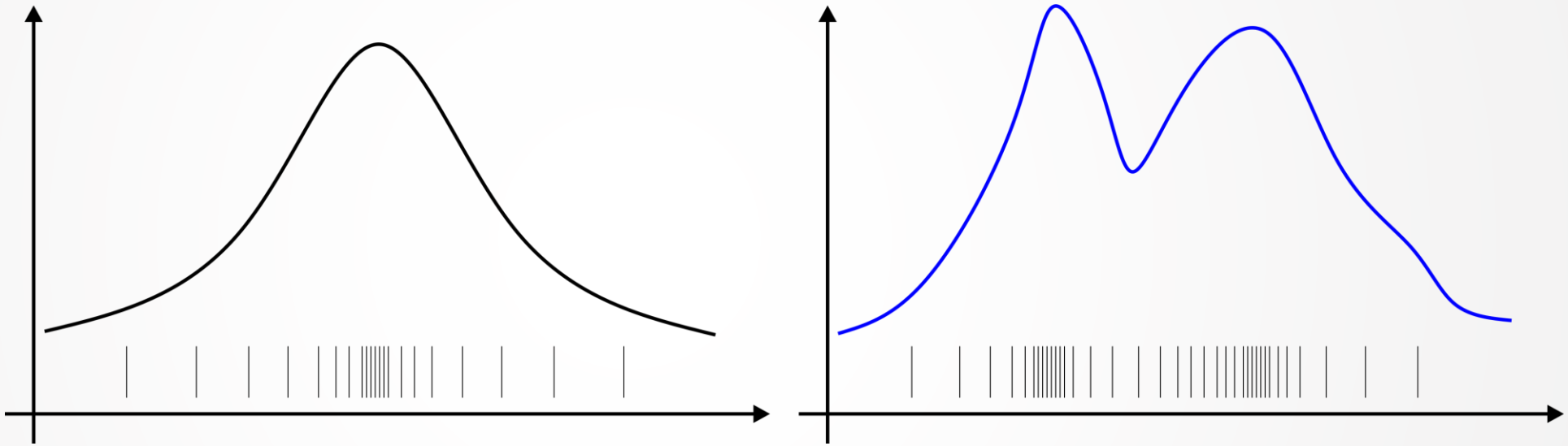
- The samples represent the **posterior**:

$$p(x) = \sum_{j=1}^{J} w^{[j]} \delta_{x^{[j]}}(x)$$

Delta-Dirac distribution

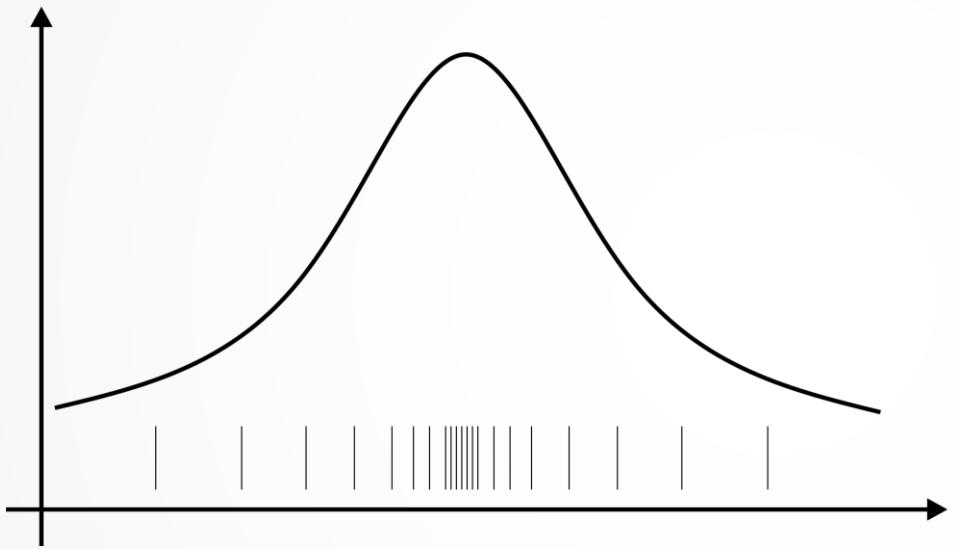
Particles for Approximation

- Particles for **function approximation**



- The more particles that fall into an interval, the higher its probability density
  - **How to obtain such samples?**

Particles for Approximation

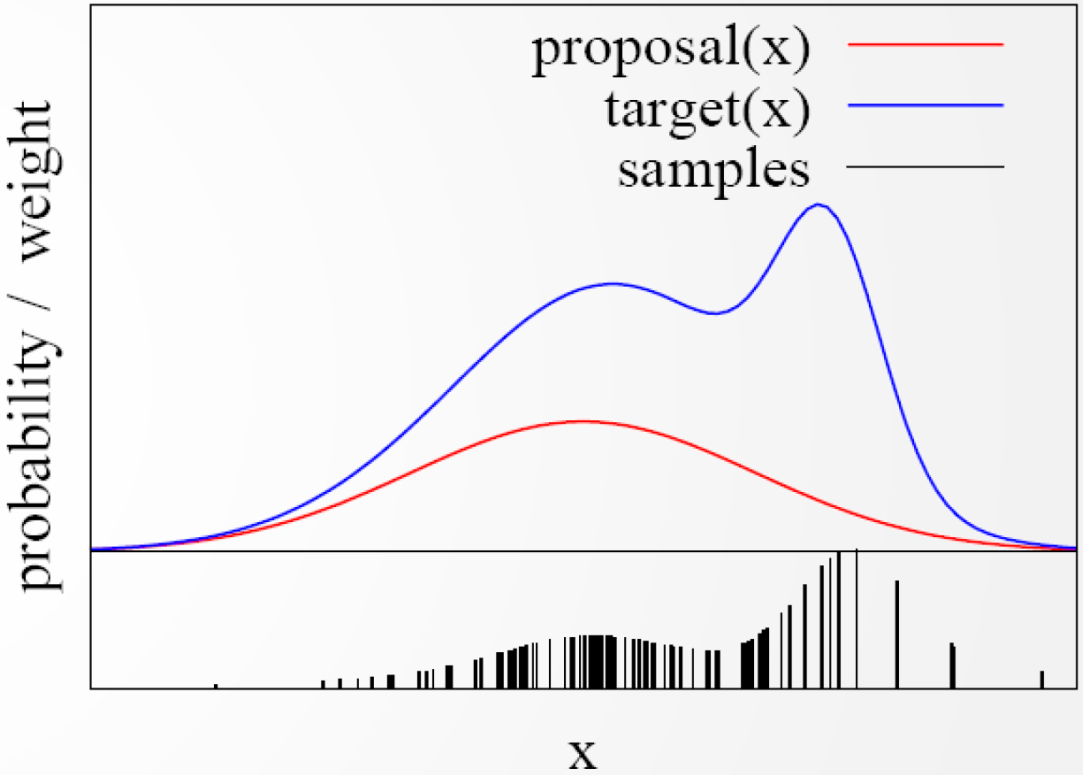


$$x \leftarrow \frac{1}{2}\sum_{i=1}^{12} rand(-\sigma, \sigma)$$

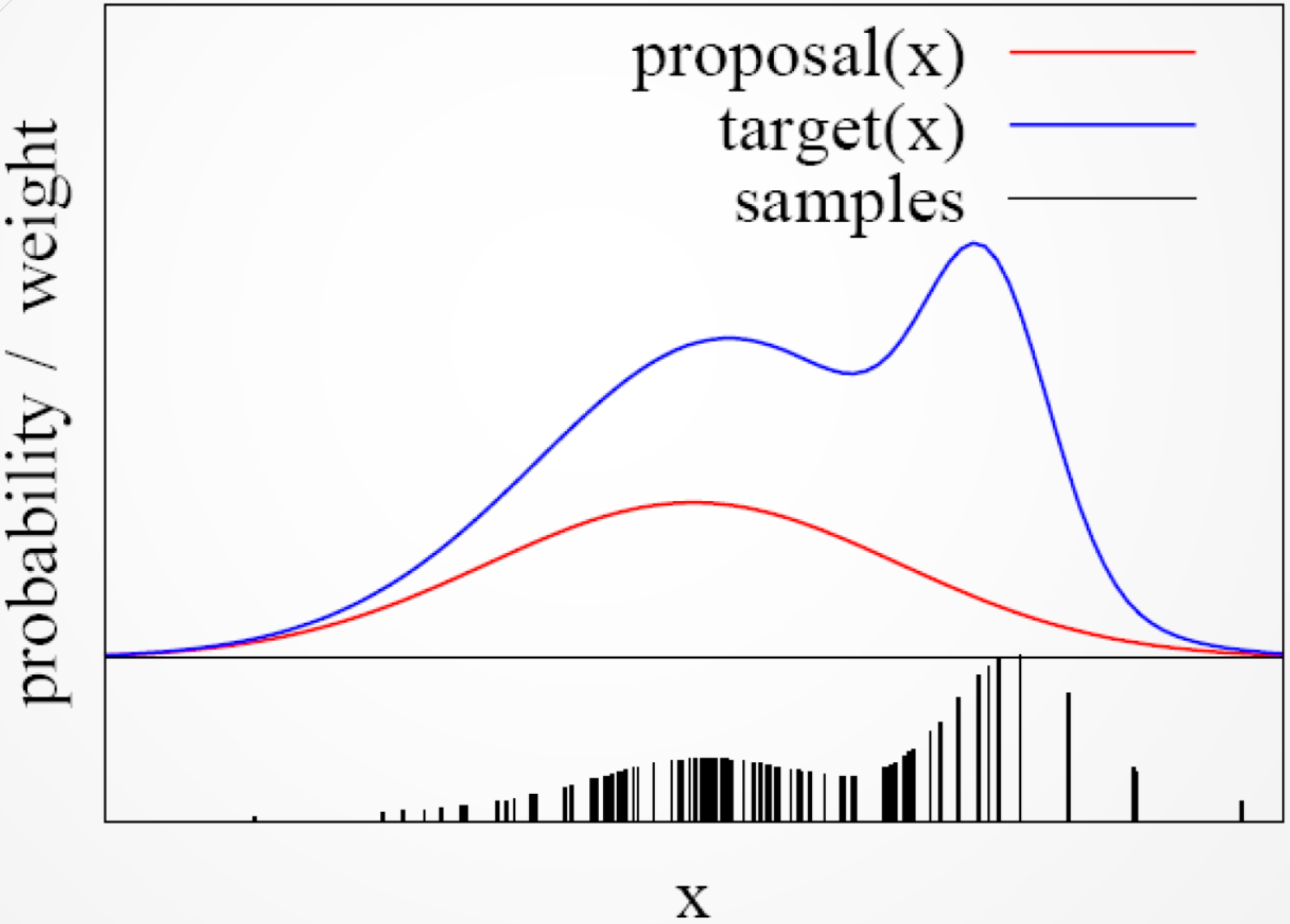
- For certain distributions (e.g. Gaussian) we may have closed-form solutions on how to take samples.
  - What about the other distributions?

Importance Sampling Principle

- We can use a different distribution g to generate samples from f
- Account for the "differences between g and f" using a weight $w = f/g$
- Target f
- Proposal g
- Pre-condition: $f(x) > 0 \rightarrow g(x) > 0$



Importance Sampling Principle



Particle Filter

- Recursive Bayes Filter
- Non-parametric approach
- Models the distribution by samples
- **Prediction:** draw from the "proposal" distribution
- **Correction:** weighting by the ratio of the "target" and the "proposal" distributions

- The more samples we use, the better the estimate can get

Particle Filter Algorithm

- **Sample the particles** using the proposal distribution

$$x_t^{[i]} \sim \pi(x_t|...)$$

- Compute the **importance weights**

$$w_t^{[i]} = \frac{target(x_t^{[i]})}{proposal(x_t^{[i]})}$$

- **Resampling:** "Replace unlikely samples by more likely ones"

Particle Filter Algorithm

- **Particle_Filter**(X_{t-1}, u_t, z_t)
  - $\bar{X}_t = X_t = 0$
  - **for** m = 1 to M do:
    - **Sample** $x_t^{[m]} \sim \pi(x_t)$
    - $w_t^{[m]} = \frac{p(x_t^{[m]})}{\pi(x_t^{[m]})} \bar{X}_t = \bar{X}_t + \left\langle x_t^{[m]}, w_t^{[m]}) \right\rangle$
    - $\bar{X}_t = \bar{X}_t + \left\langle x_t^{[m]}, w_t^{[m]}) \right\rangle$
  - **endfor**
  - **for** m = 1 to M do:
    - **Draw** I with probability proportional to $w_t^{[i]}$
    - **Add** $x_t^{[i]}$ to X_t
  - **endfor**
  - **return** X_t

Monte Carlo Localization

- Each particle is a **pose hypothesis**
- **Proposal is the motion model**

$$x_t^{[i]} \sim p(x_t \mid x_{t-1}, u_t)$$

- **Correction via the observation model**

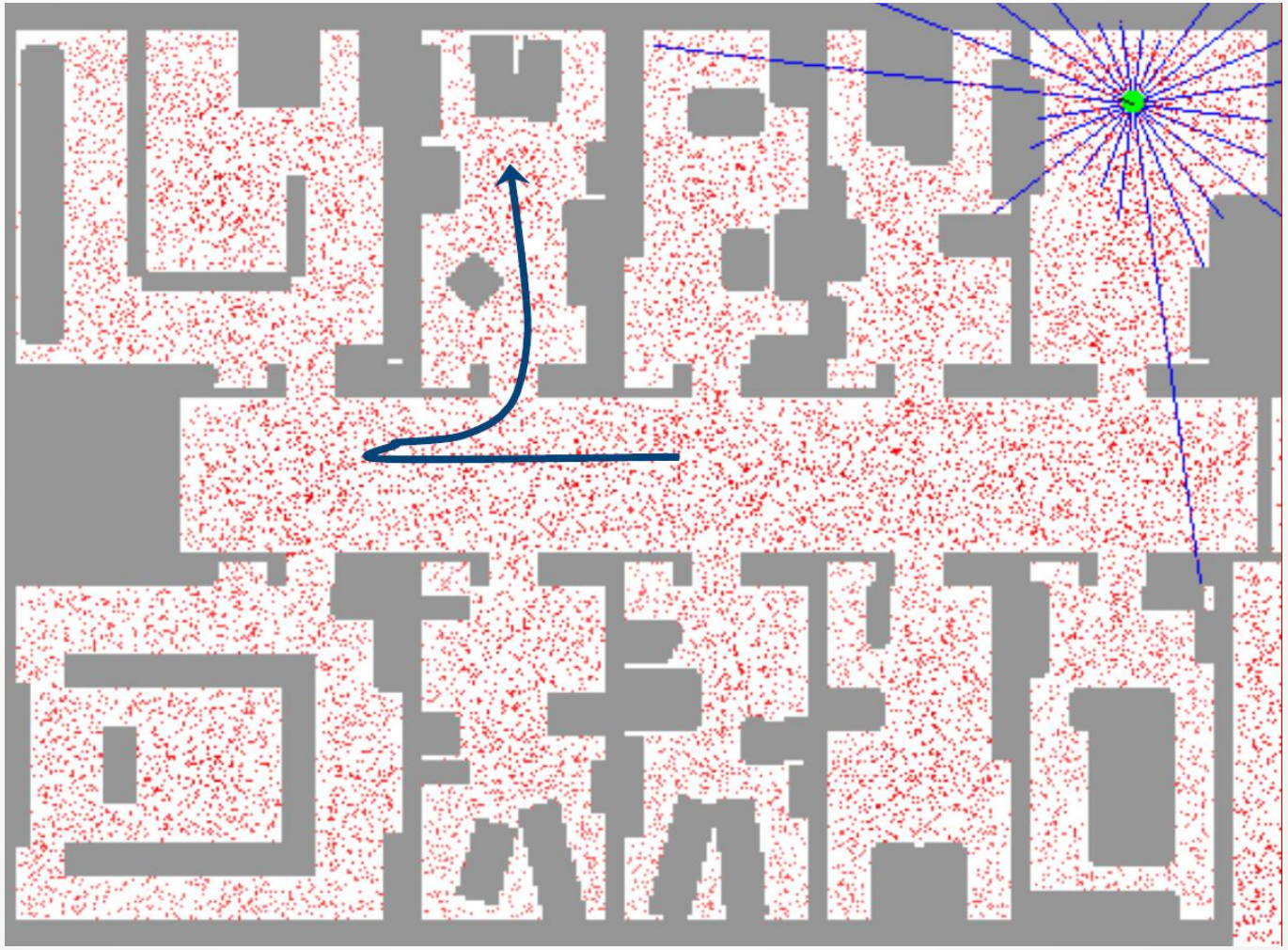
$$w_t^{[i]} = \frac{target}{proposal} \propto p(z_t \mid x_t, m)$$

Particle Filter Algorithm

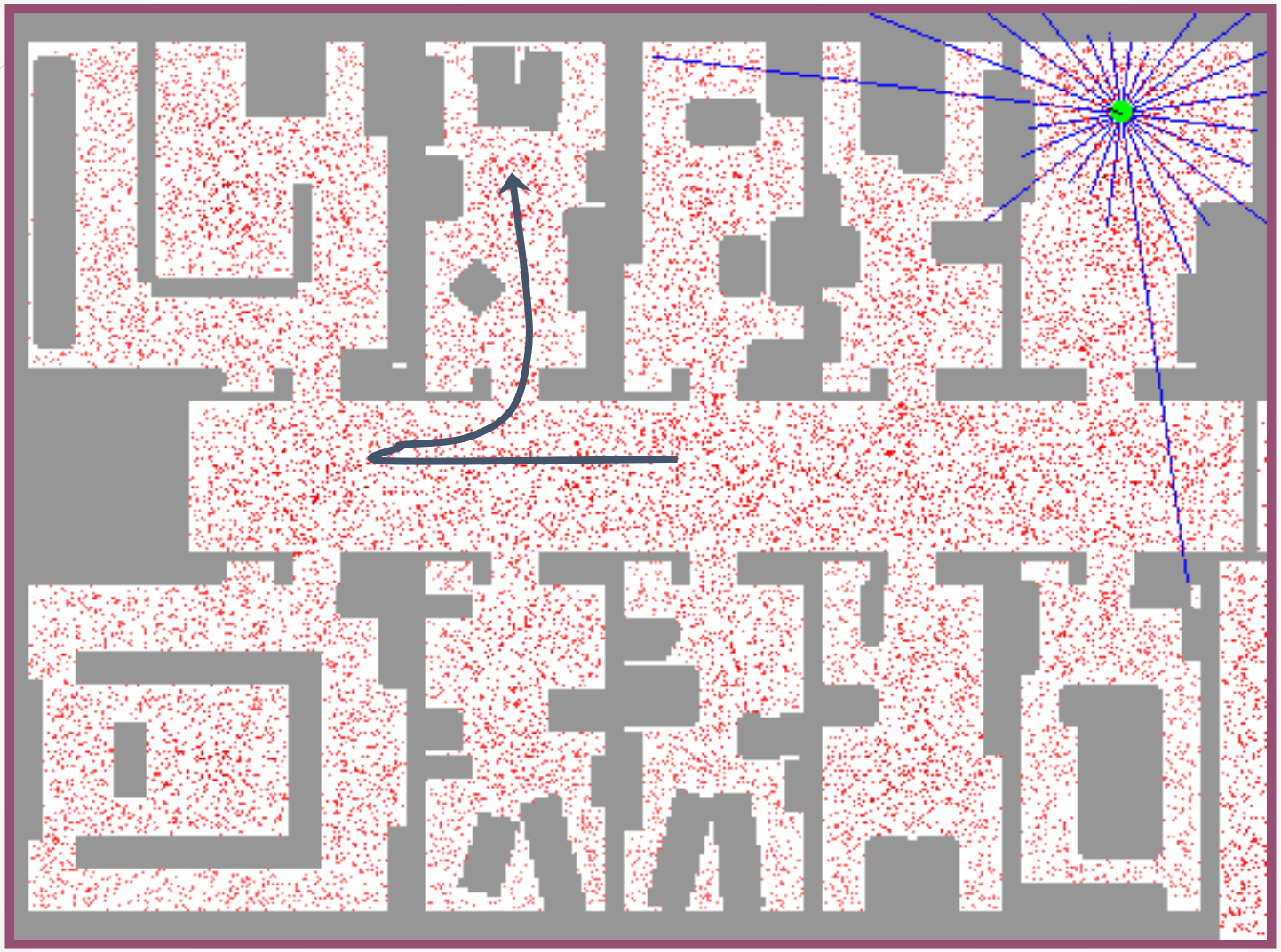
- **Particle_Filter**(X_{t-1}, u_t, z_t)
  - $\bar{X}_t = X_t = 0$
  - **for** m = 1 to M do:
    - **Sample** $\underline{x_t^{[m]} \sim p(x_t | u_t, x_{t-1}^{[m]})}$
    - $w_t^{[m]} = \underline{p(z_t | x_t^{[m]})}$
    - $\bar{X}_t = \bar{X}_t + \left\langle x_t^{[m]}, w_t^{[m]} \right\rangle$
  - **endfor**
  - **for** m = 1 to M do:
    - **Draw** I with probability proportional to $w_t^{[i]}$
    - **Add** $x_t^{[i]}$ to X_t
  - **endfor**
  - **return** X_t

Application: Particle Filter for Localization

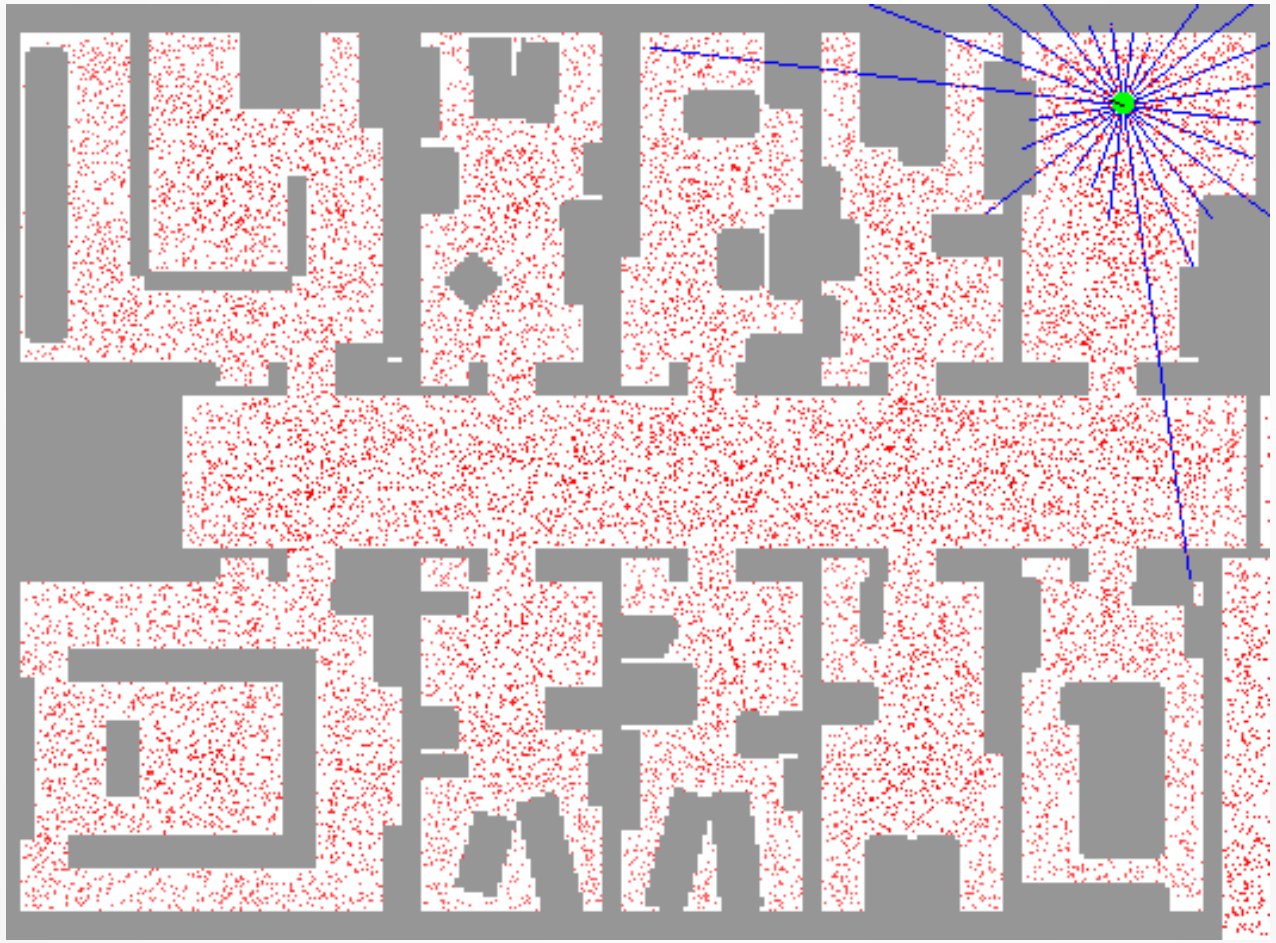
- Assumption of existence of a known map



Sample-based localization



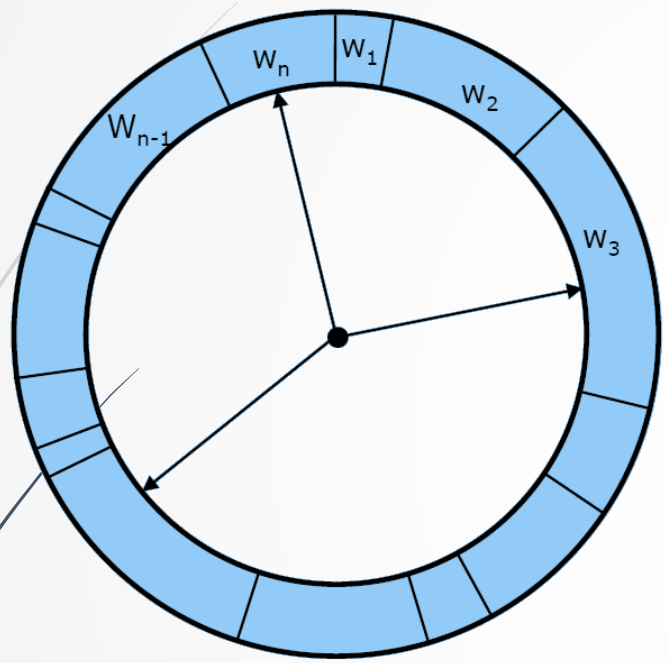
Monte Carlo Localization in Action



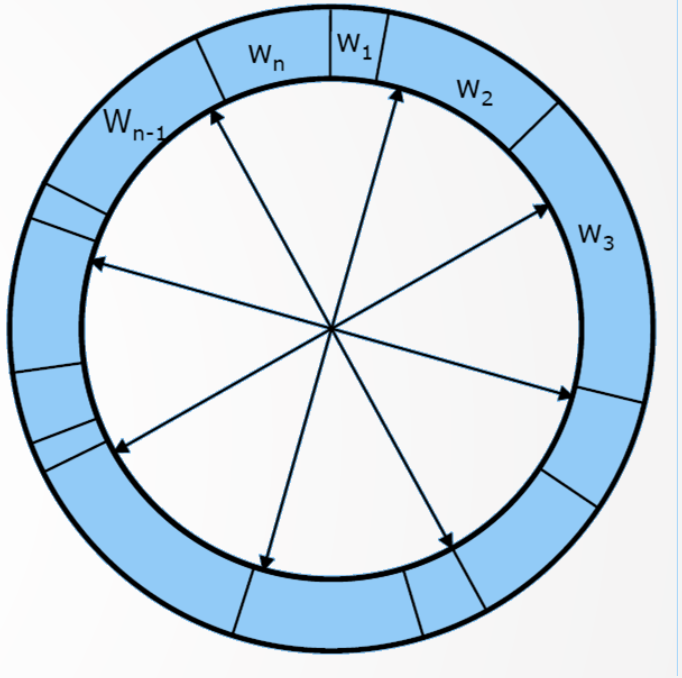
Resampling

- Draw sample i with probability $w_t^{[i]}$– Repeat J times.
- Informal: "replace unlikely samples by more likely ones"
- **Survival of the fittest**
- "Trick" to avoid that many samples cover unlikely states
- Needed as we have a limited number of samples

Resampling



- Roulette wheel
- Binary search
- O(nlogn)



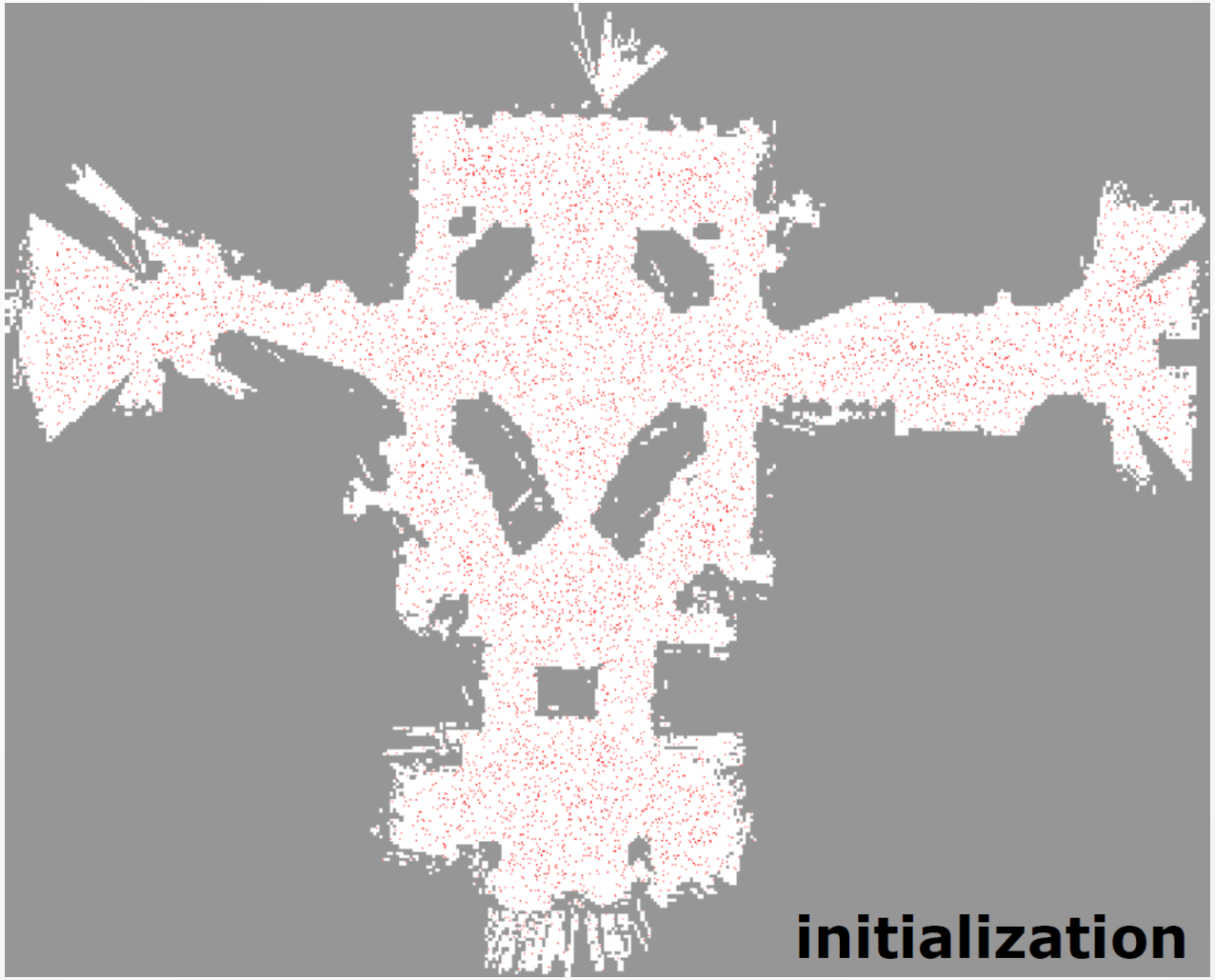
- Stochastic universal sampling
- Low variance
- O(n)

Particle Filter Algorithm

- **Low_Variance_Resampling** (X_t, W_t)
  - $\bar{X}_t = 0$
  - $r = rand(0; M^{-1})$
  - $c = w_t^{[1]}$
  - $i = 1$
  - **for** m = 1 to M do:
    - $U = r + (m-1)M^{-1}$
    - **while** $U > c$
      - $i = i + 1$
      - $c = c + w_t^{[i]}$
    - **endwhile**
  - **add** $x_t^{[i]}$ to $\bar{X}_t$
  - **endfor**
  - **return** $\bar{X}_t$

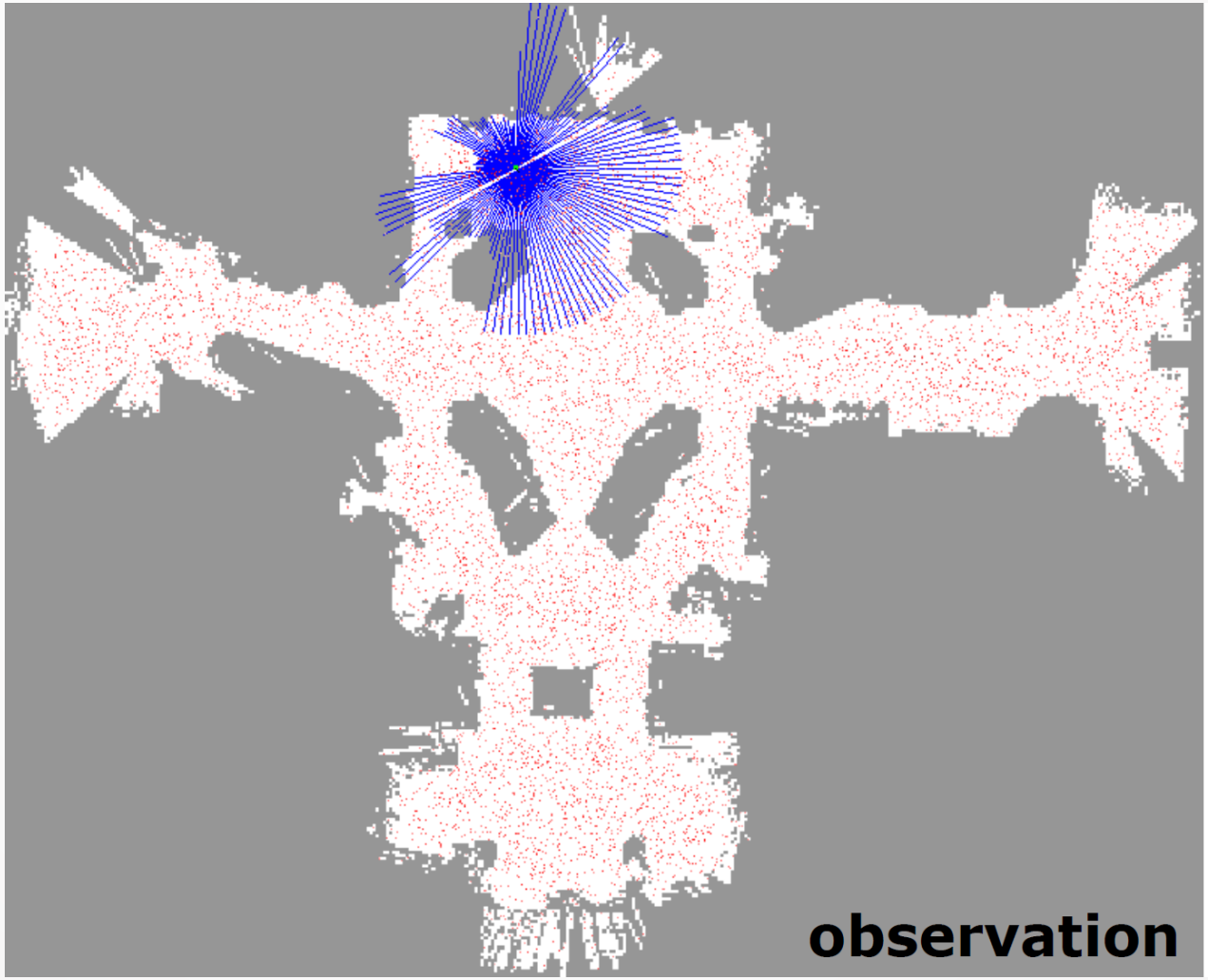
Example

- Assumption of existence of a known map



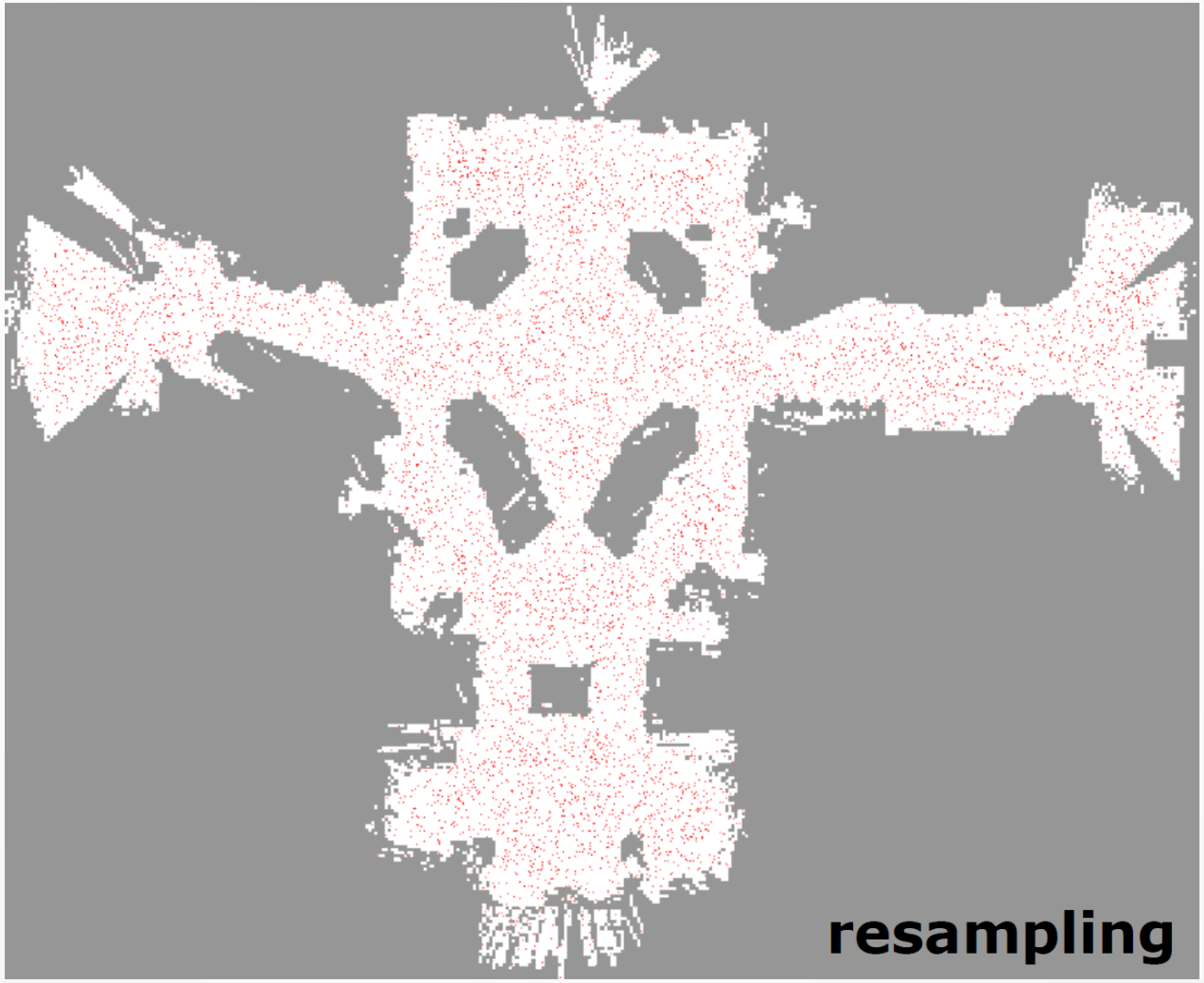
Example

- Assumption of existence of a known map



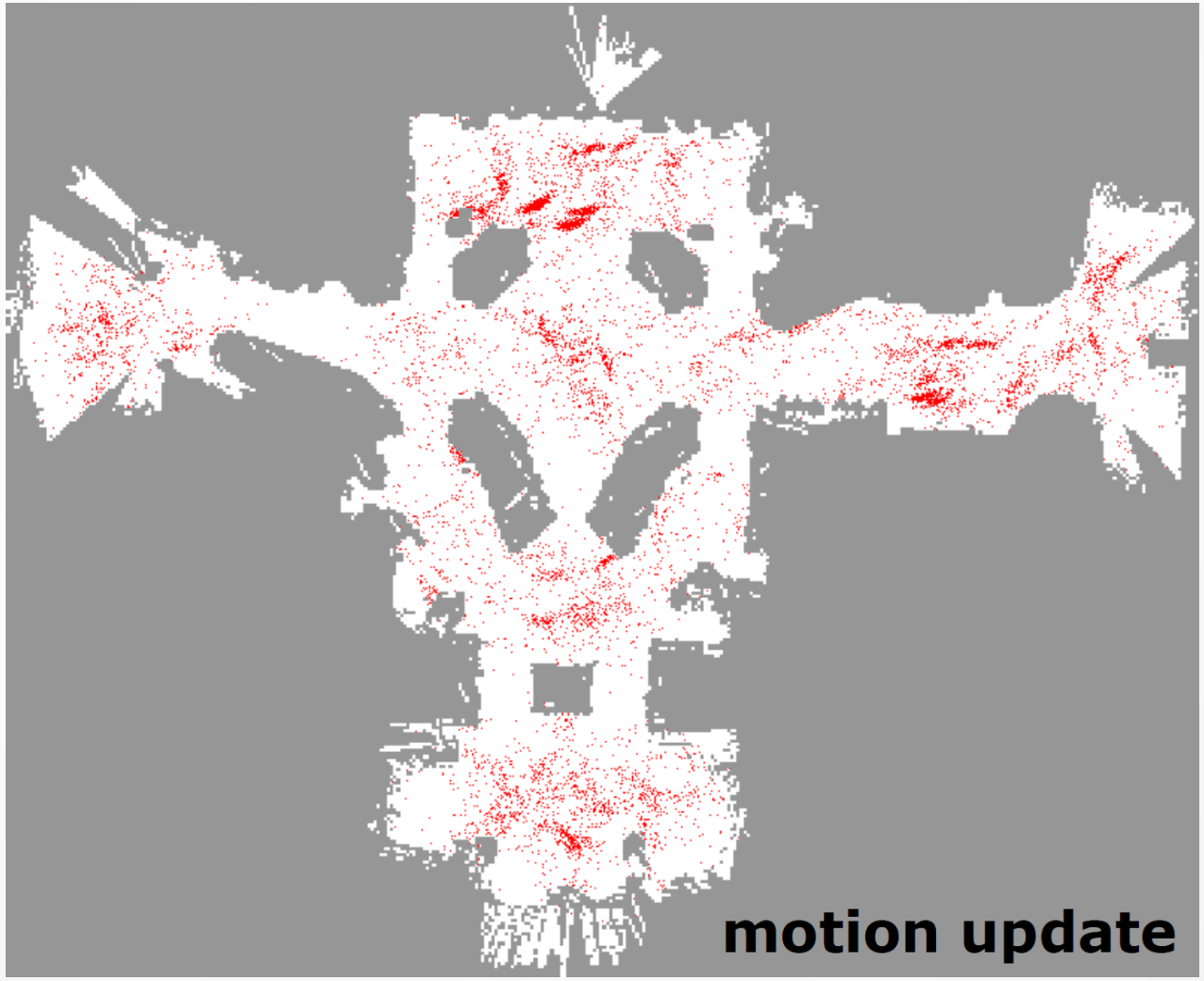
Example

- Assumption of existence of a known map



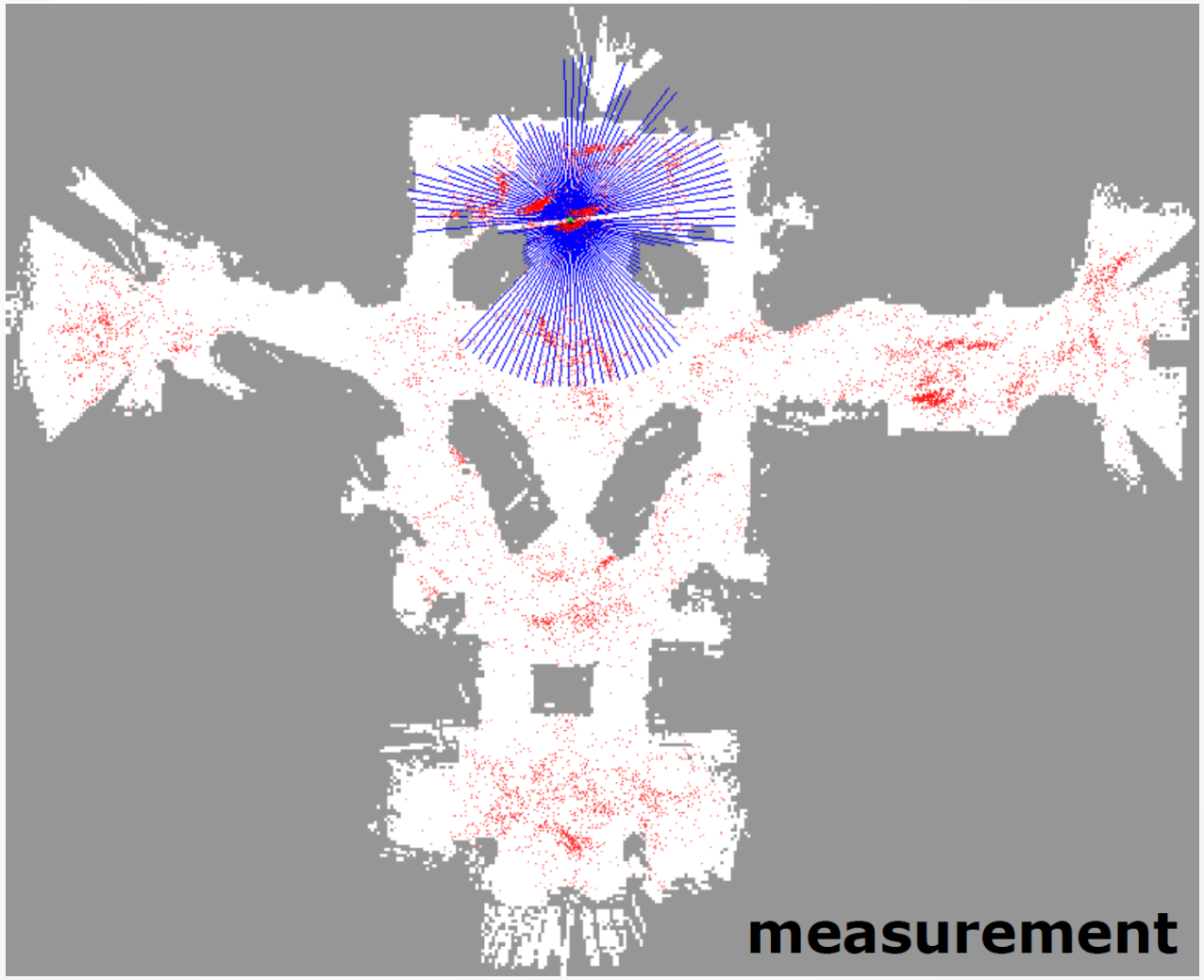
Example

- Assumption of existence of a known map



Example

- Assumption of existence of a known map

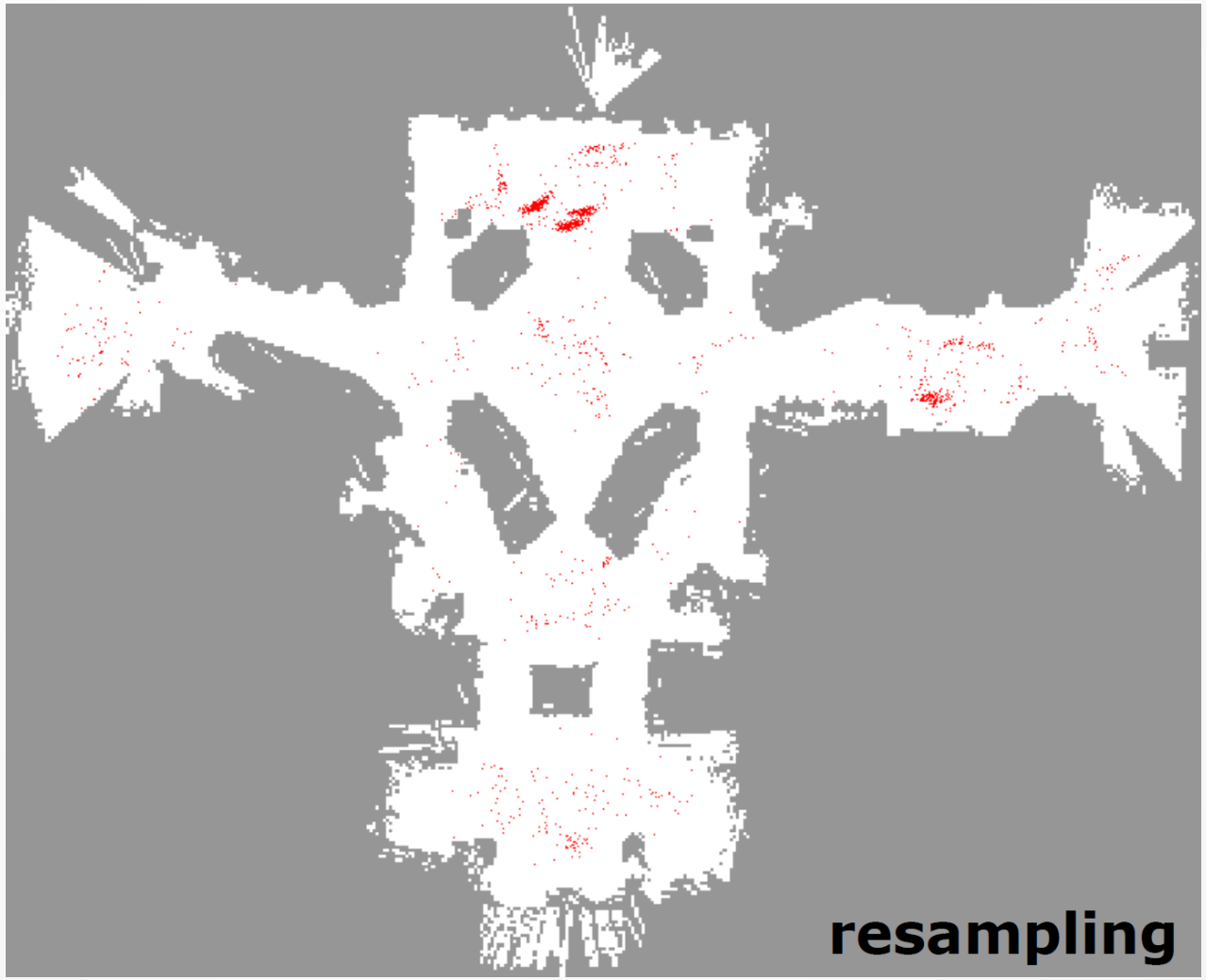


Example

- Assumption of existence of a known map

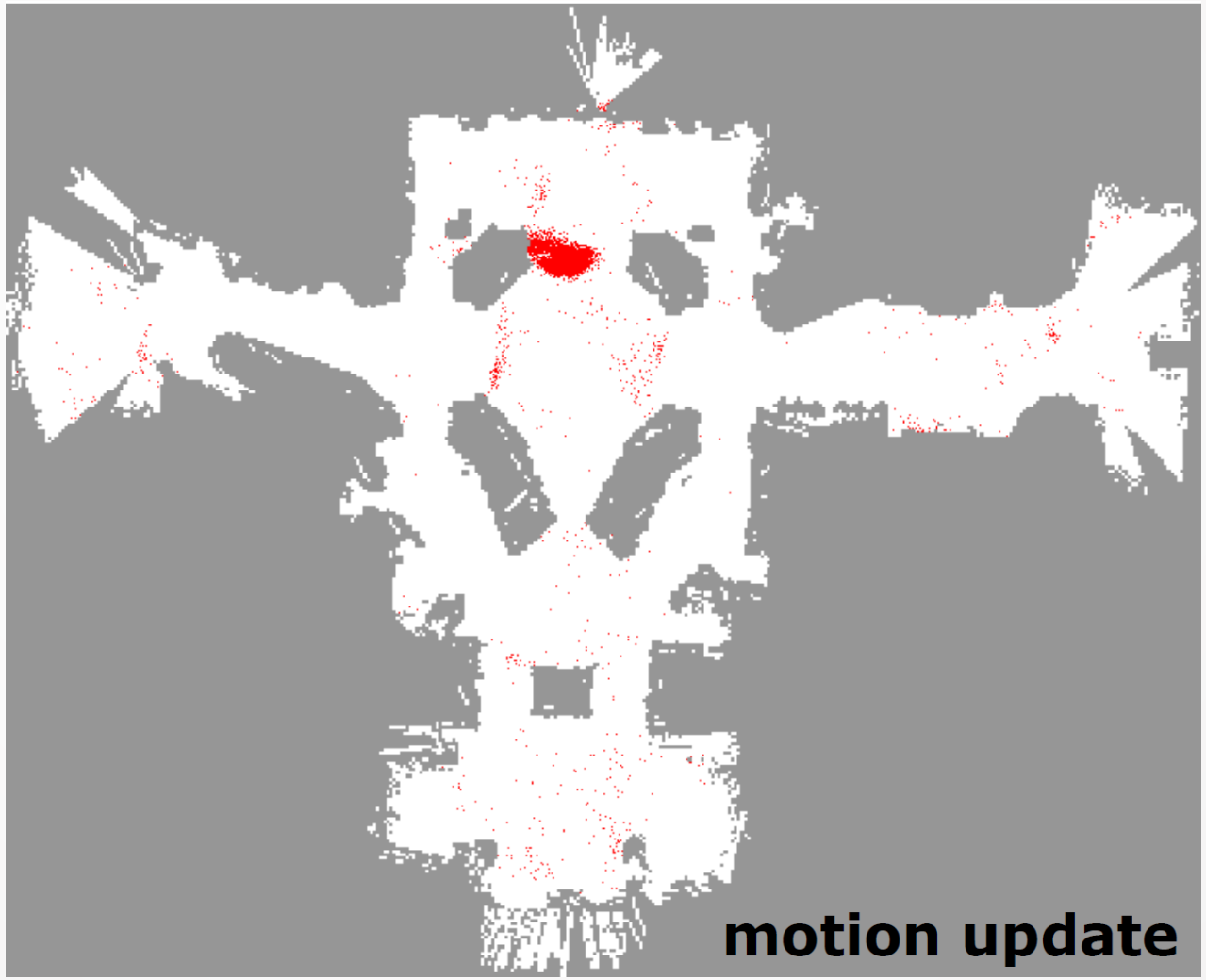
Example

- Assumption of existence of a known map



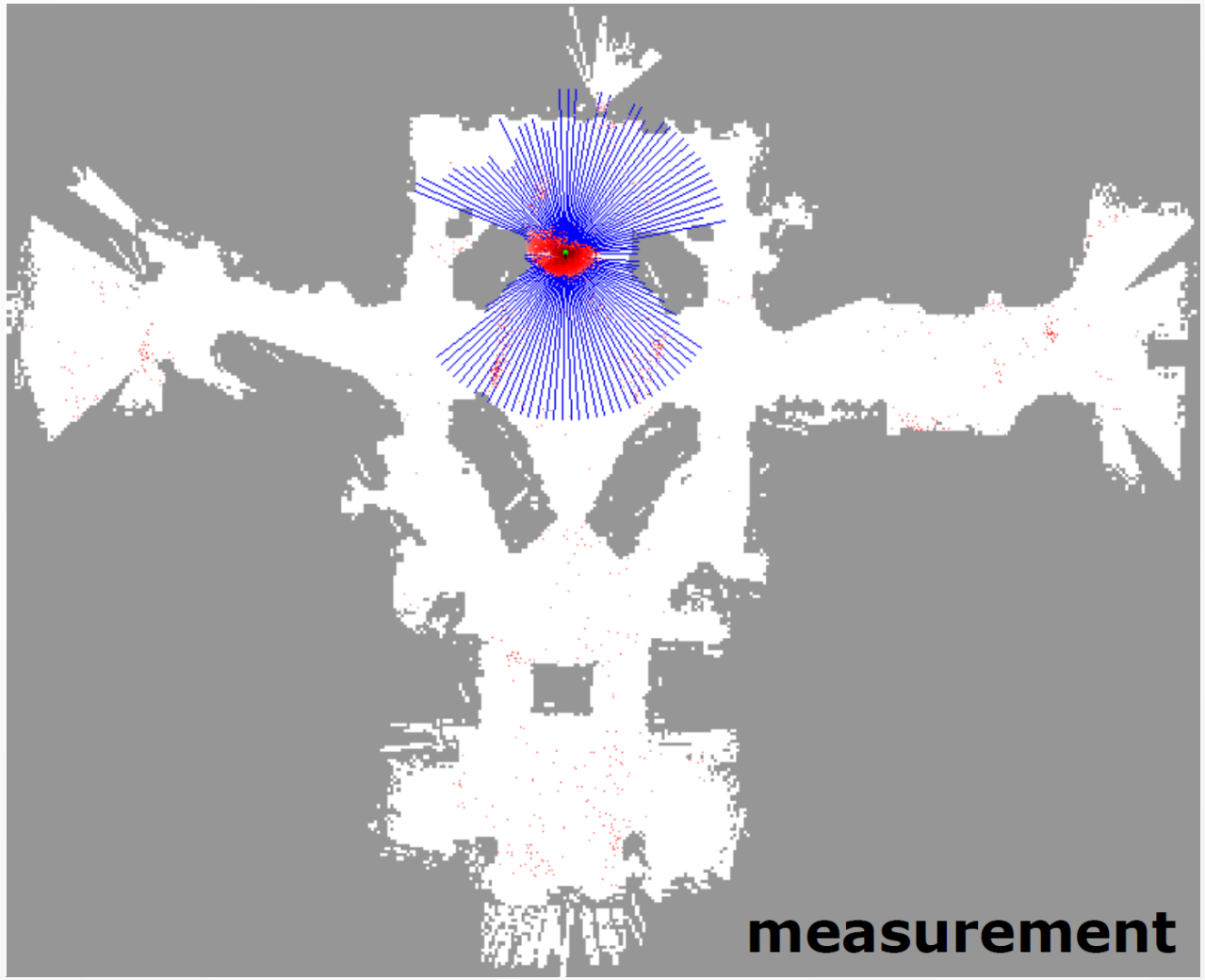
Example

- Assumption of existence of a known map



Example

- Assumption of existence of a known map

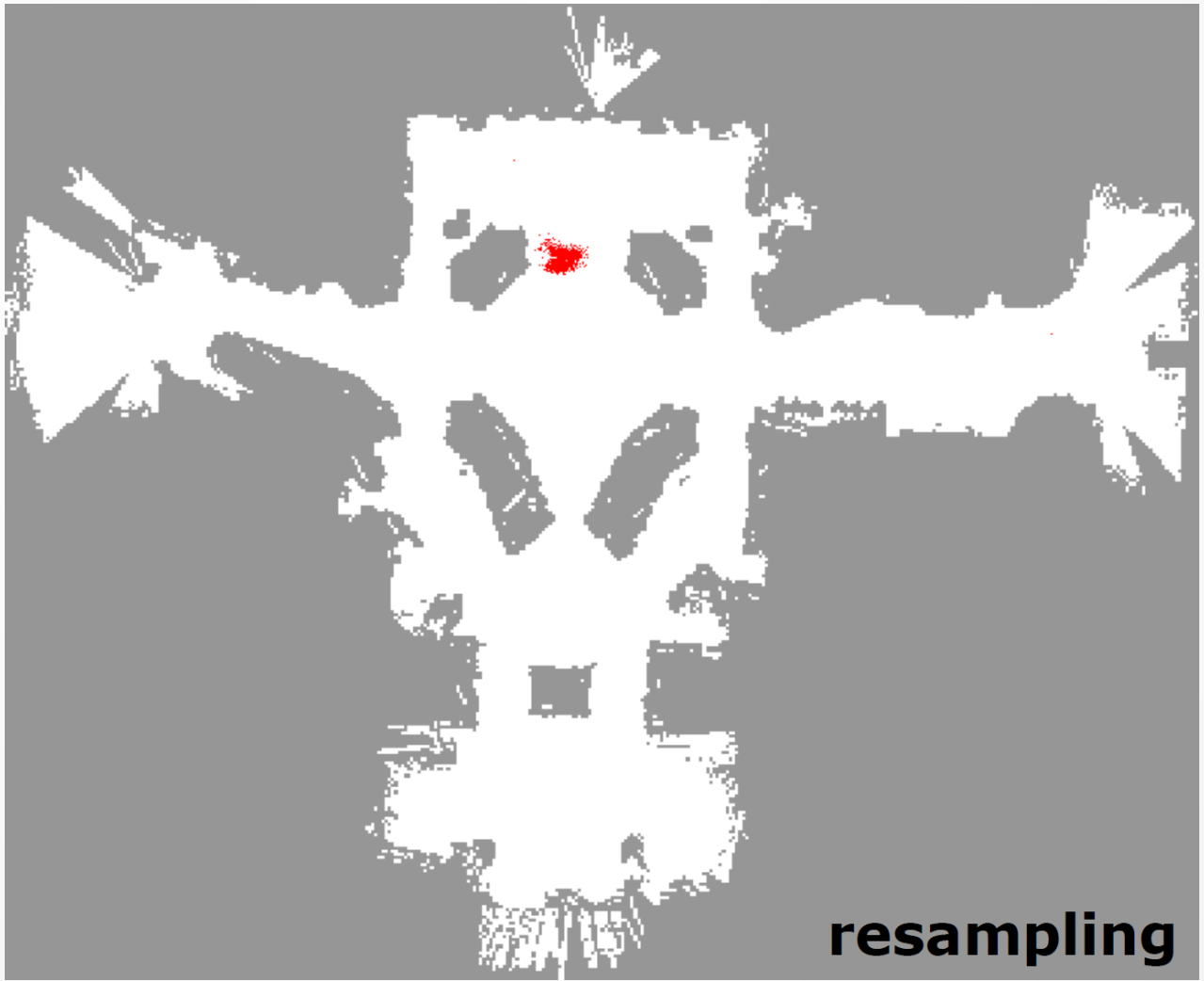


Example

- Assumption of existence of a known map

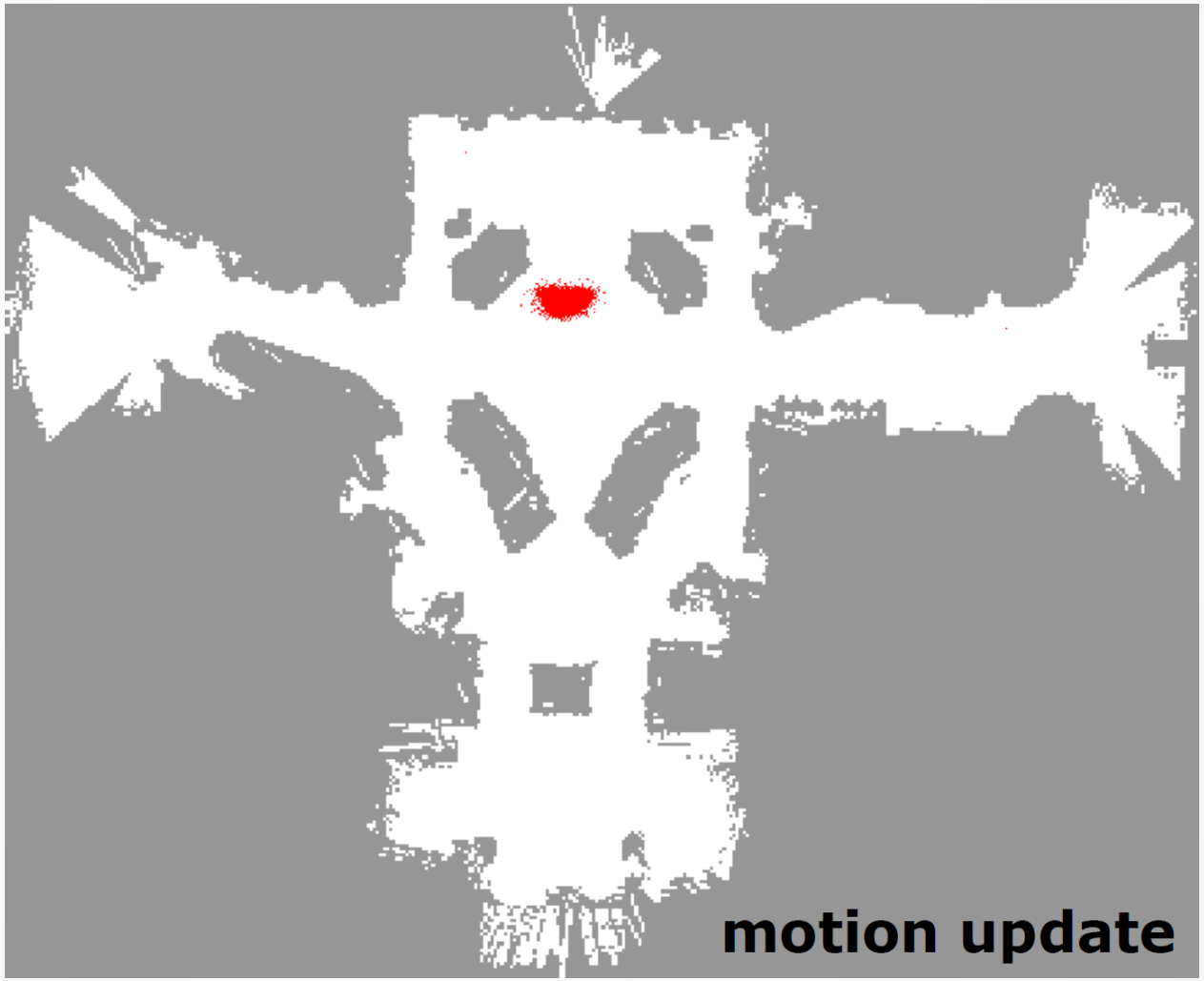
Example

- Assumption of existence of a known map



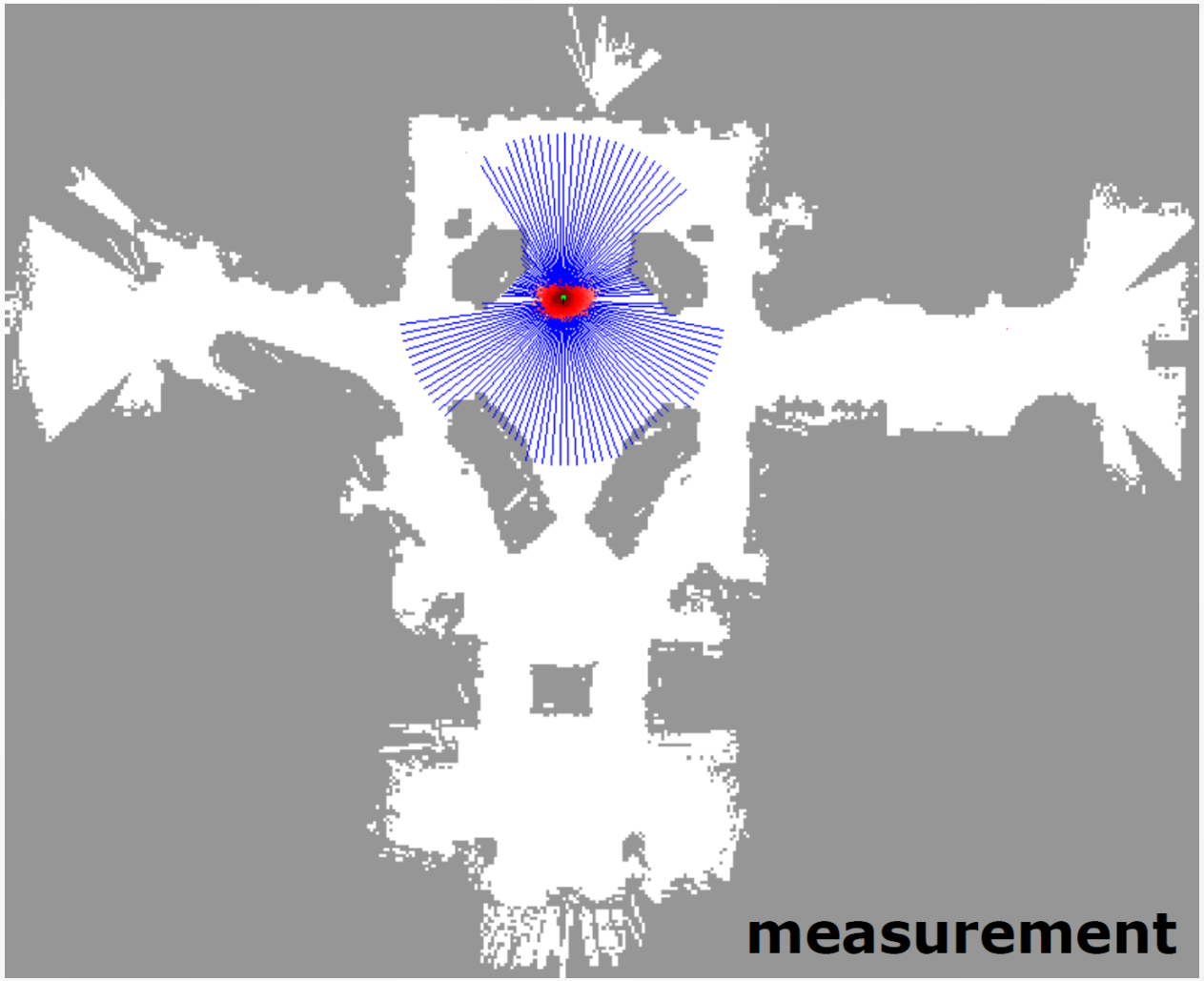
Example

- Assumption of existence of a known map



Example

- Assumption of existence of a known map

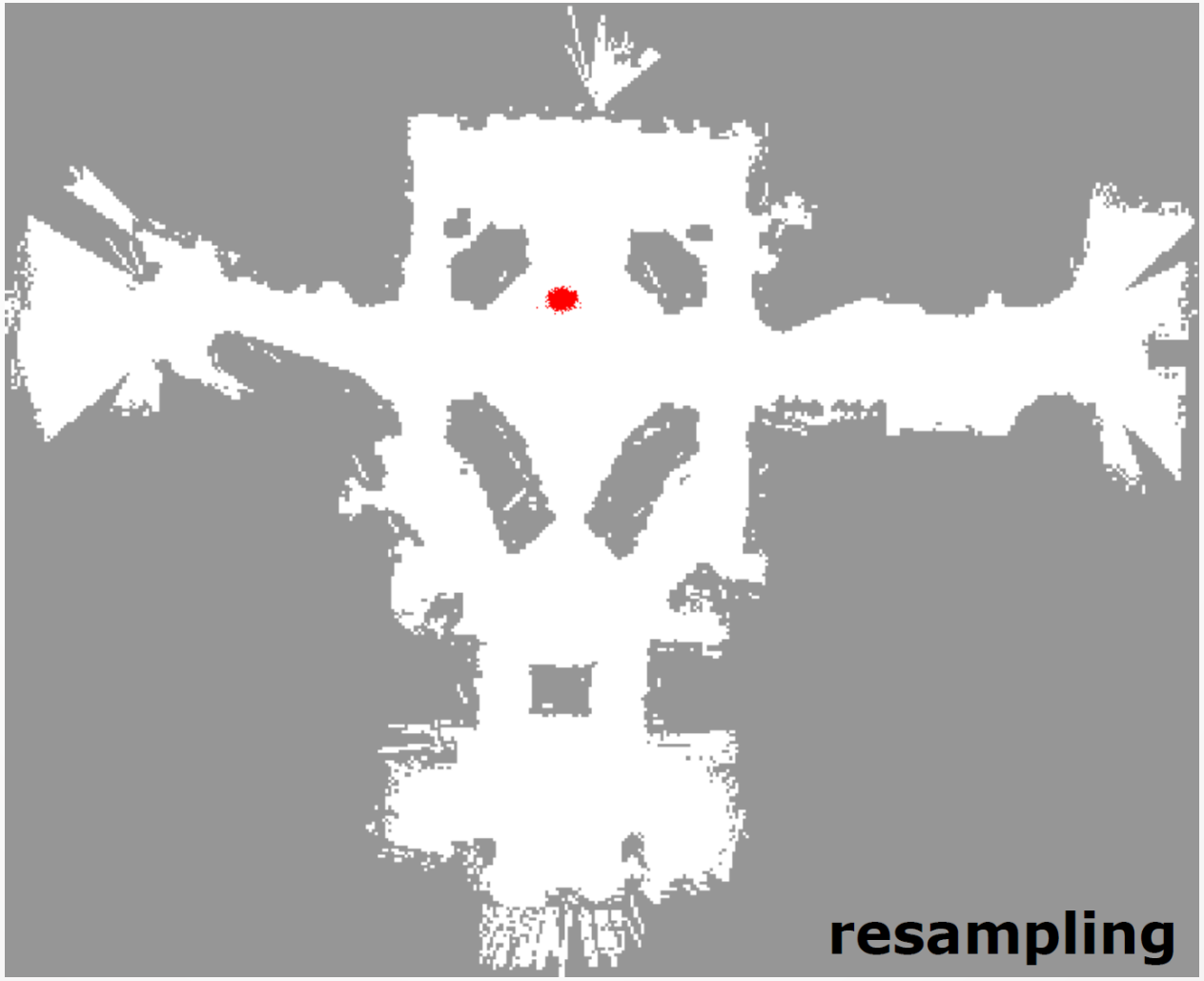


Example

- Assumption of existence of a known map

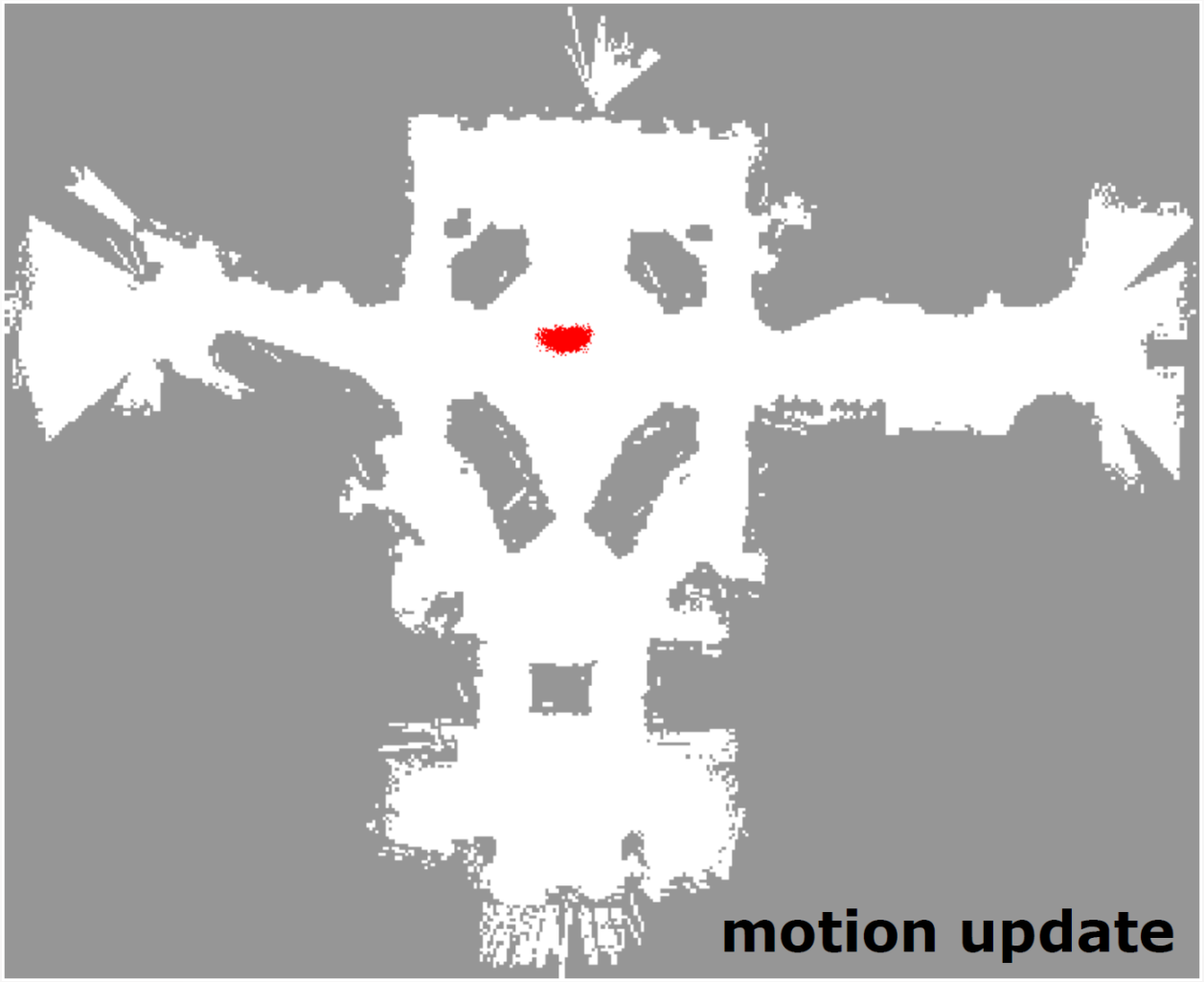
Example

- Assumption of existence of a known map



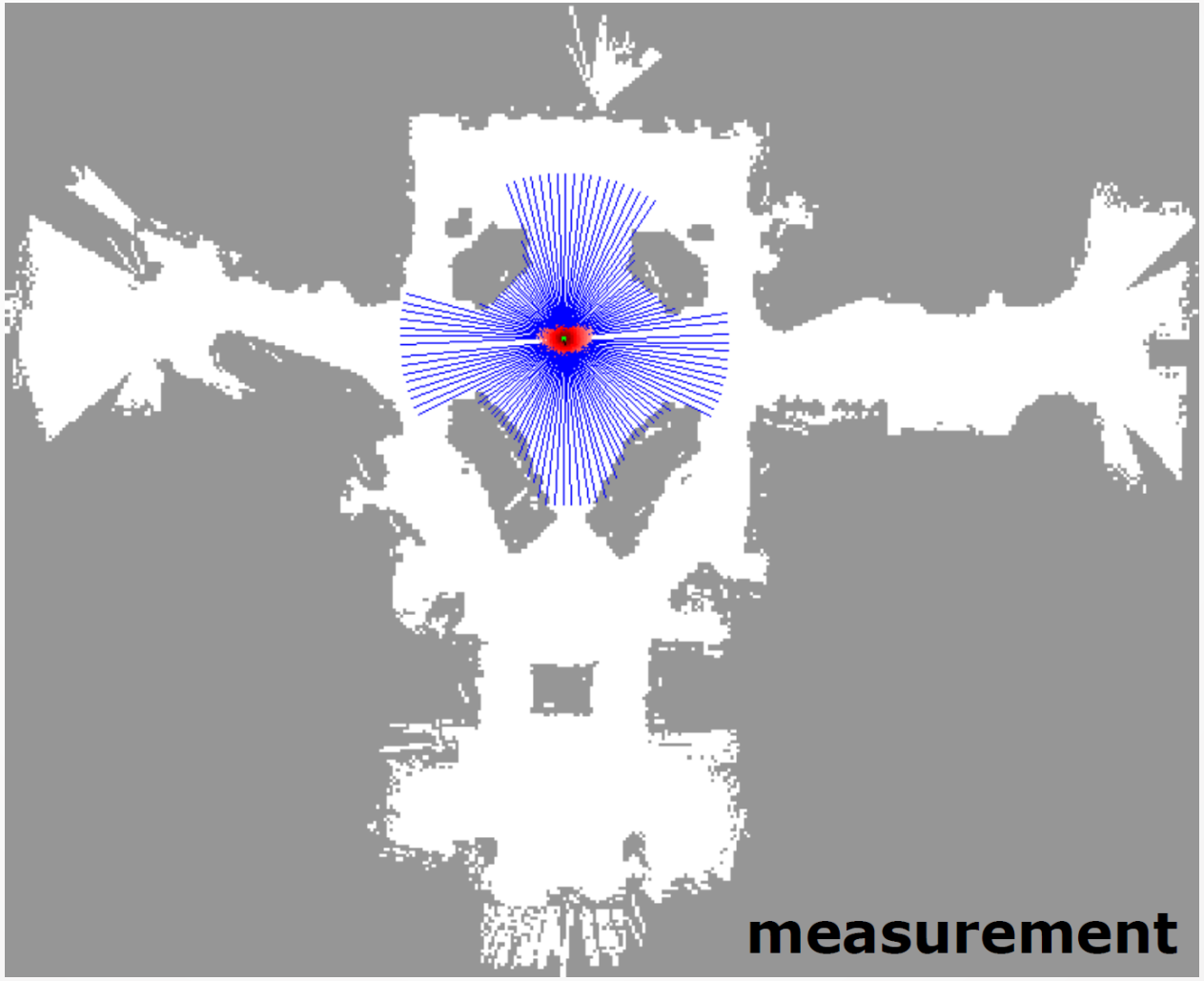
Example

- Assumption of existence of a known map



Example

- Assumption of existence of a known map



Summary – Particle Filters

- Particle filters are non-parametric
- Posterior is represented by a set of weighted samples
- Not limited to Gaussians
- Proposal to draw new samples
- Weight to account for the differences between the proposal and the target
- Works well in low-dimensional spaces

How does this apply to my project?

- State estimation is the way to use robot sensors to infer the robot state. You may use it for estimating your robot pose or its map, to track and object and be able to follow it etc.

Find out more

- Dieter Fox, Wolfram Burgard, Frank Dellaert, Sebastian Thrun, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots", Proc. 16th National Conference on Artificial Intelligence, AAAI'99, July 1999
- Dieter Fox, Wolfram Burgard, Sebastian Thrun, "Markov Localization for Mobile Robots in Dynamic Environments", J. of Artificial Intelligence Research 11 (1999) 391-427
- Sebastian Thrun, "Probabilistic Algorithms in Robotics", Technical Report CMU-CS-00-126, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA, 2000

Thank you!

Please ask your question!