Robot Mapping

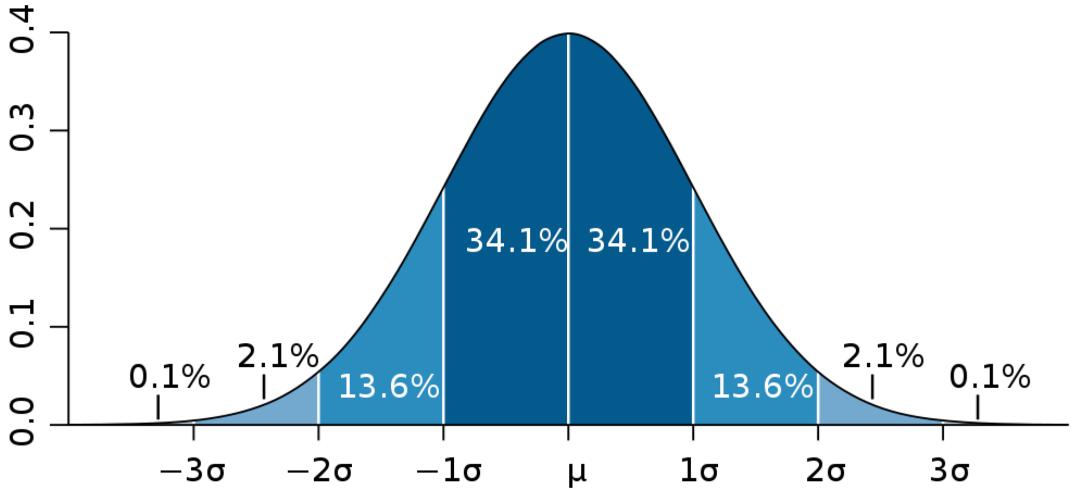
Short Introduction to Particle Filters and Monte Carlo Localization

Cyrill Stachniss

Gaussian Filters

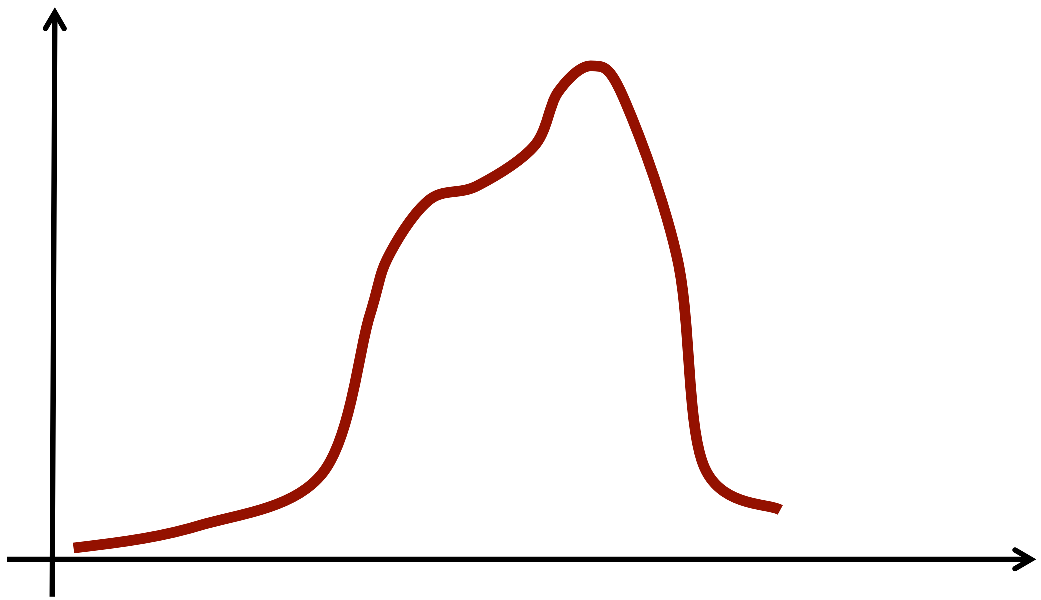
- The Kalman filter and its variants can only model **Gaussian distributions**

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



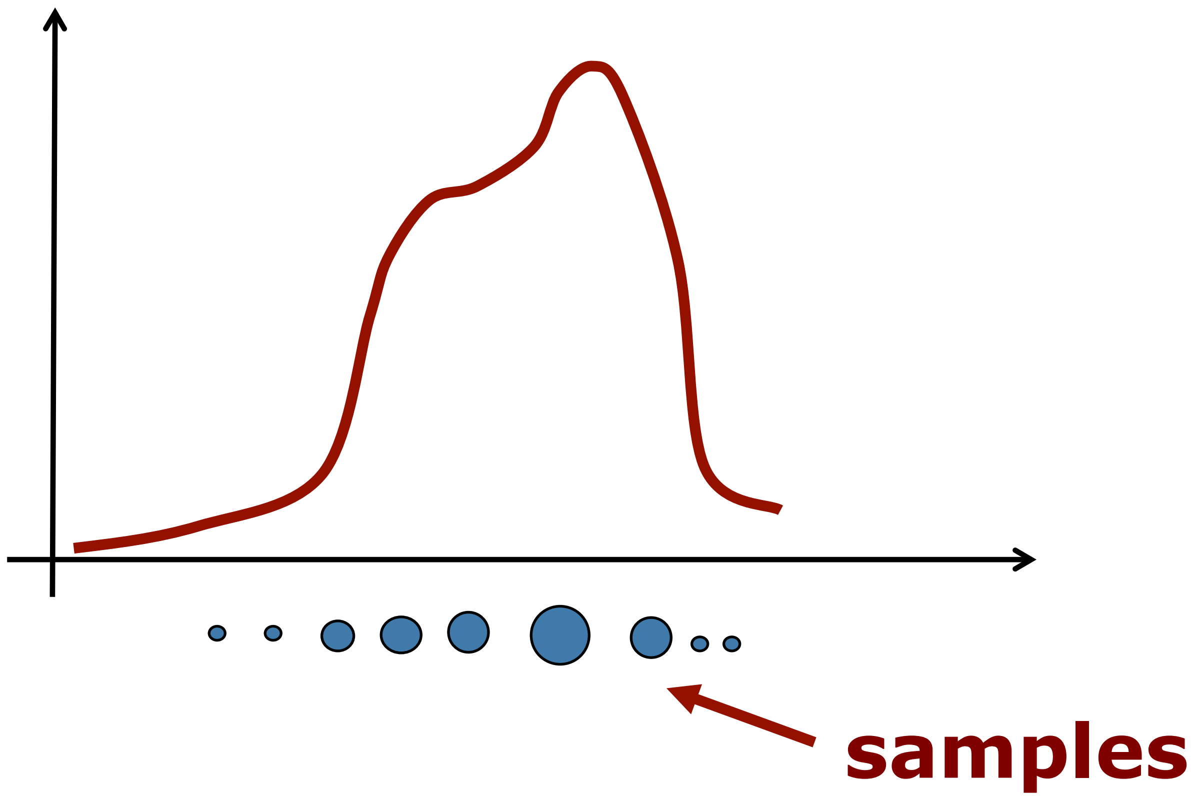
Motivation

- Goal: approach for dealing with **arbitrary distributions**



Key Idea: Samples

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



Particle Set

- Set of weighted samples

$$\mathcal{X} = \left\{ \left\langle x^{[j]}, w^{[j]} \right\rangle \right\}_{j=1,\dots,J}$$

state hypothesis

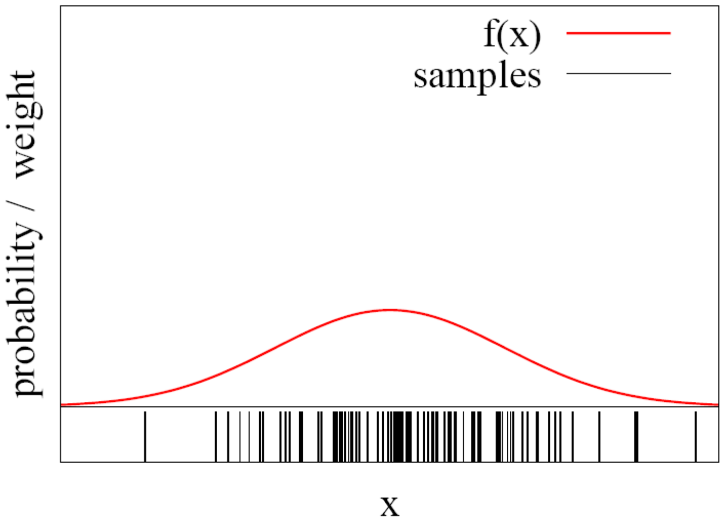
importance weight

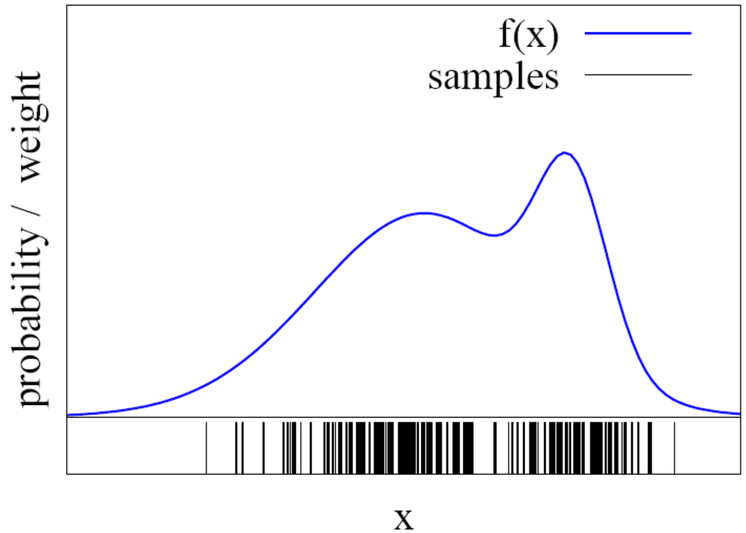
- The samples represent the posterior

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

Particles for Approximation

- Particles for function approximation



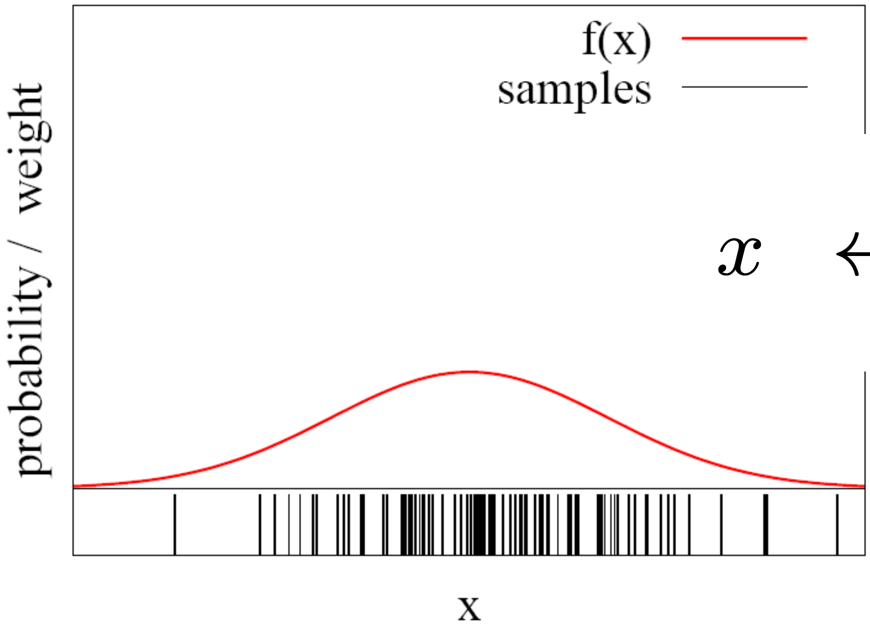


- The more particles fall into a region, the higher the probability of the region

How to obtain such samples?

Closed Form Sampling is Only Possible for a Few Distributions

- Example: Gaussian

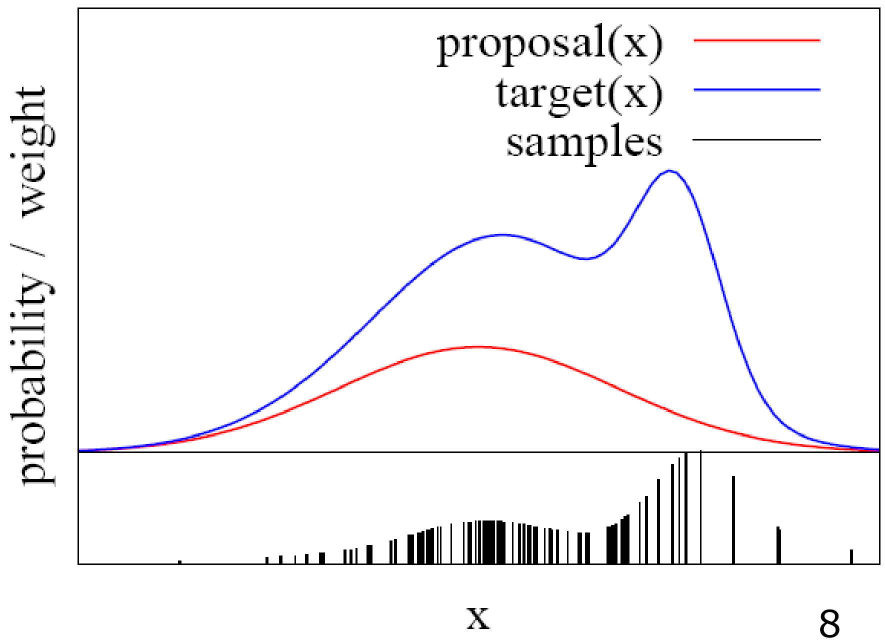


$$x \leftarrow \frac{1}{2} \sum_{i=1}^{12} \mathrm{rand}(-\sigma, \sigma)$$

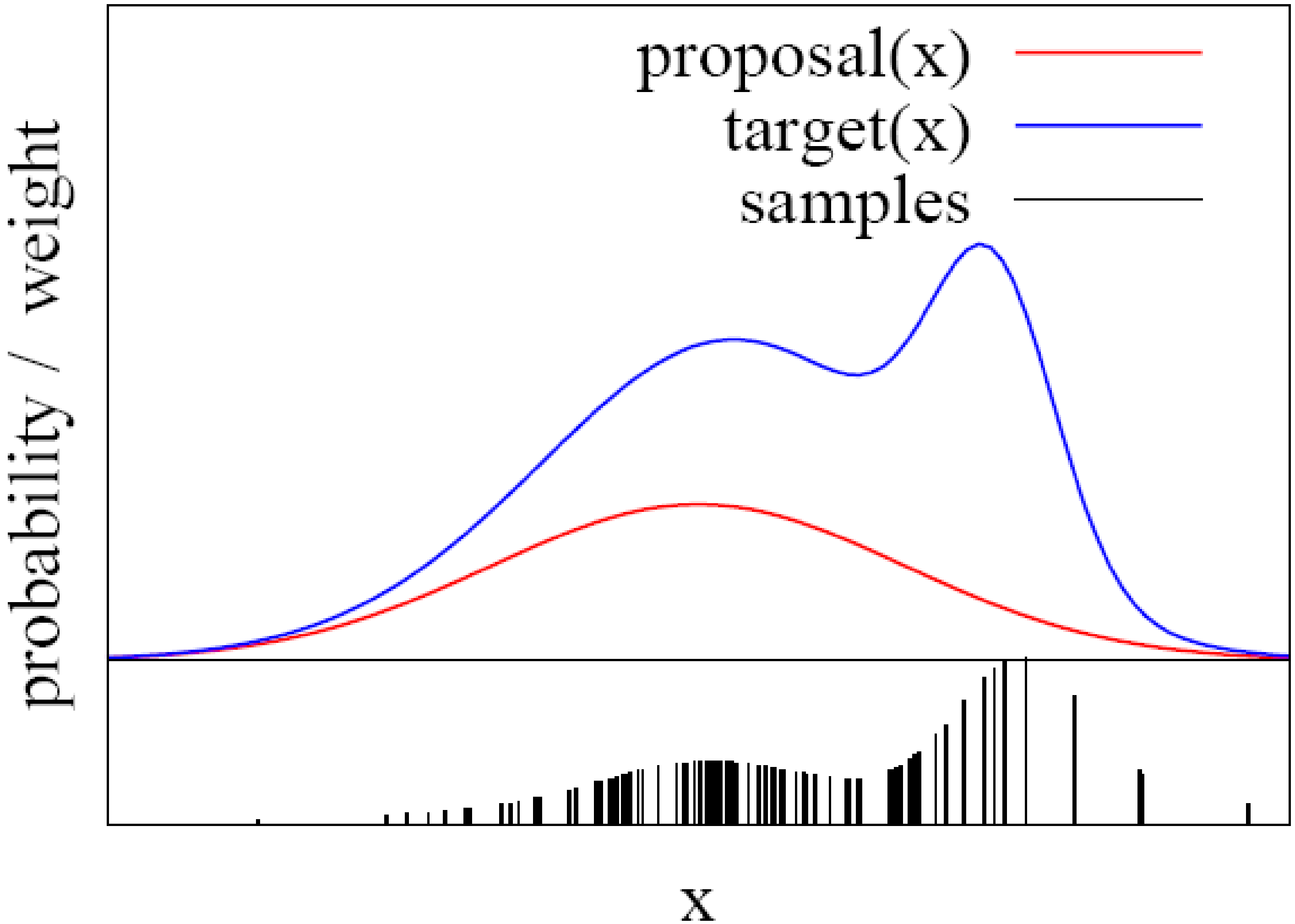
How to sample from **other** distributions?

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
- Correction: weighting by the ratio of target and proposal

The more samples we use, the better is the estimate!

Particle Filter Algorithm

1. Sample the particles using the proposal distribution

$$x_t^{[j]} \sim \pi(x_t \mid \ldots)$$

2. Compute the importance weights

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

- Resampling: Draw sample i with probability $w_t^{[i]}$ and repeat J times

Particle Filter Algorithm

Particle_filter($\mathcal{X}_{t-1}, u_t, z_t$)**:**

$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
for $j = 1$ *to* J *do*
  sample $x_t^{[j]} \sim \pi(x_t)$
  $w_t^{[j]} = \frac{p(x_t^{[j]})}{\pi(x_t^{[j]})}$
  $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[j]}, w_t^{[j]} \rangle$
endfor
for $j = 1$ *to* J *do*
  draw $i \in 1, \ldots, J$ *with probability* $\propto w_t^{[i]}$
  add $x_t^{[i]}$ *to* $\mathcal{X}_t$
endfor
return $\mathcal{X}_t$

Monte Carlo Localization

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

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

- Correction via the observation model

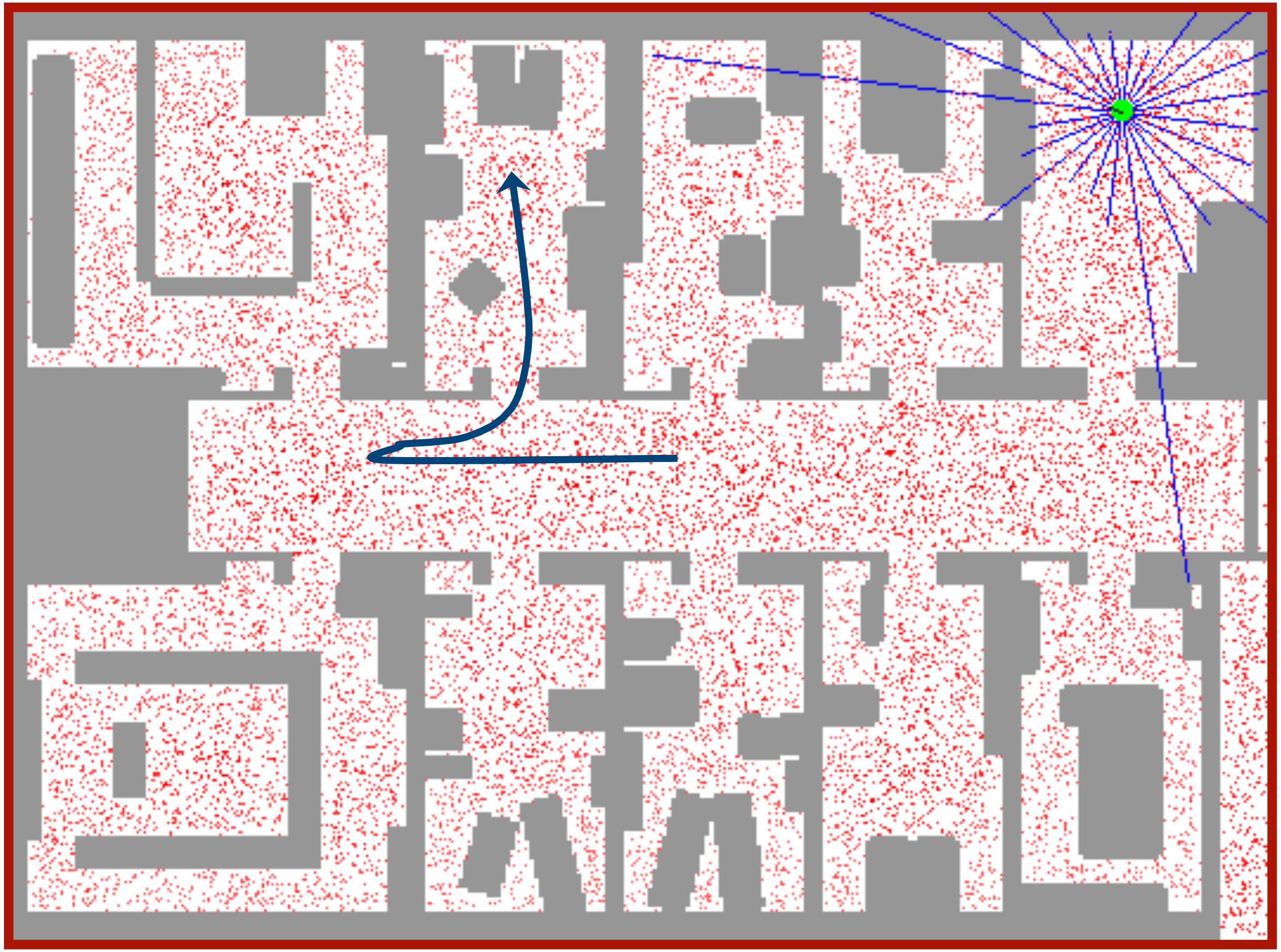
$$w_t^{[j]} = \frac{target}{proposal} \propto p(z_t \mid x_t, m)$$

Particle Filter for Localization

Particle_filter($\mathcal{X}_{t-1}, u_t, z_t$)**:**

$\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
    for $j = 1$ *to* J *do*
        sample $x_t^{[j]} \sim p(x_t \mid u_t, x_{t-1}^{[j]})$
        $w_t^{[j]} = p(z_t \mid x_t^{[j]})$
        $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[j]}, w_t^{[j]} \rangle$
    endfor
    for $j = 1$ *to* J *do*
        draw $i \in 1, \ldots, J$ *with probability* $\propto w_t^{[i]}$
        add $x_t^{[i]}$ *to* $\mathcal{X}_t$
    endfor
    return $\mathcal{X}_t$

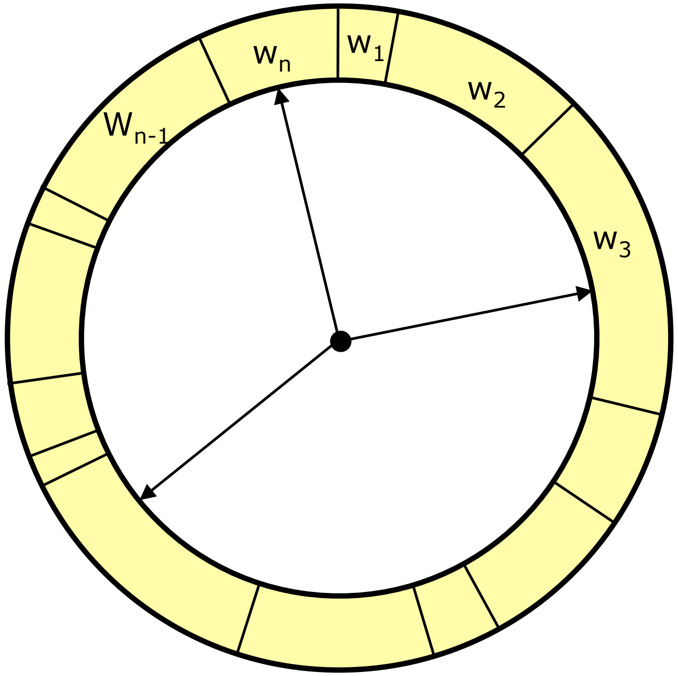
Application: Particle Filter for Localization (Known Map)



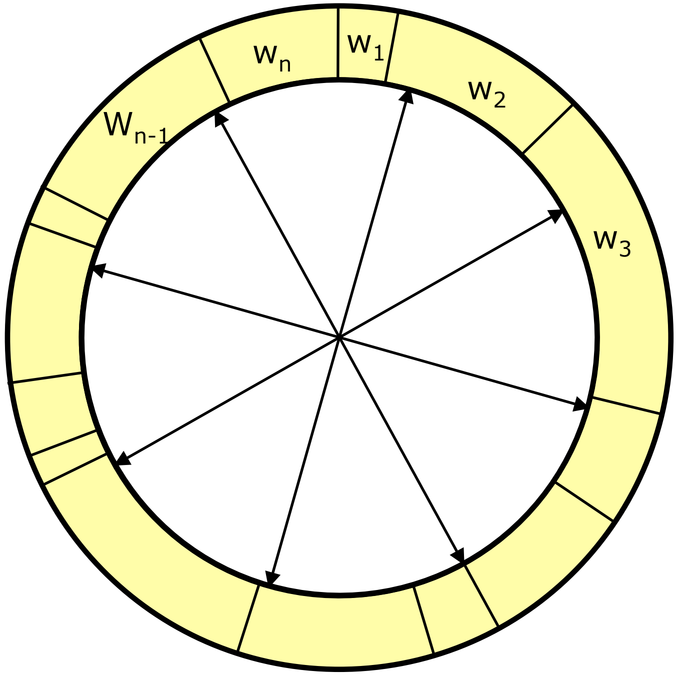
Resampling

- Draw sample i with probability $w_t^{[i]}$. Repeat J times.
- Informally: “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(J log J)



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

Low Variance Resampling

Low_variance_resampling$(\mathcal{X}_t, \mathcal{W}_t)$**:**

$\bar{\mathcal{X}}_t = \emptyset$

$r = \mathrm{rand}(0; J^{-1})$

$c = w_t^{[1]}$

$i = 1$

for $j = 1$ *to* J *do*

  $U = r + (j-1)J^{-1}$

  while $U > c$

    $i = i + 1$

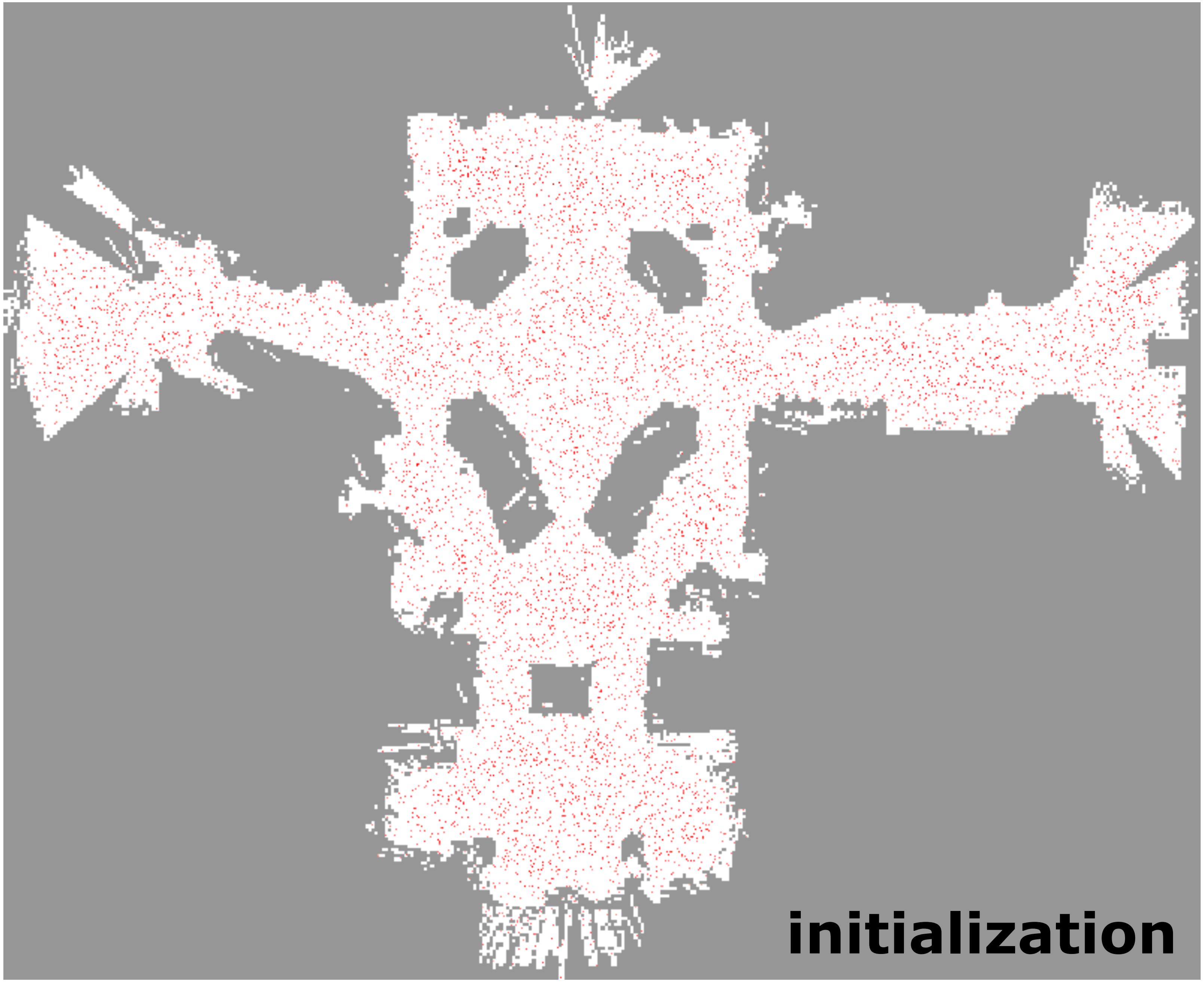
    $c = c + w_t^{[i]}$

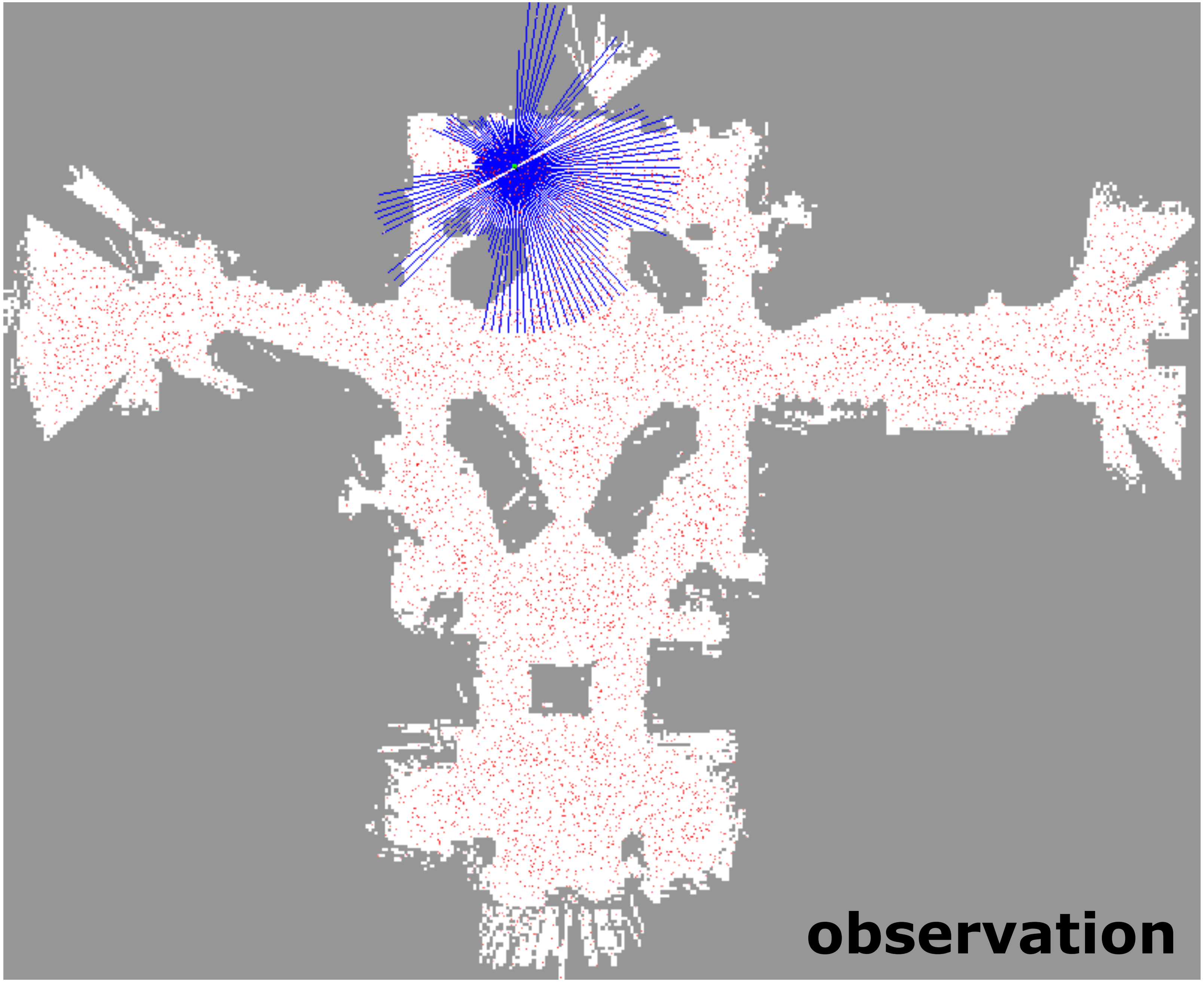
  endwhile

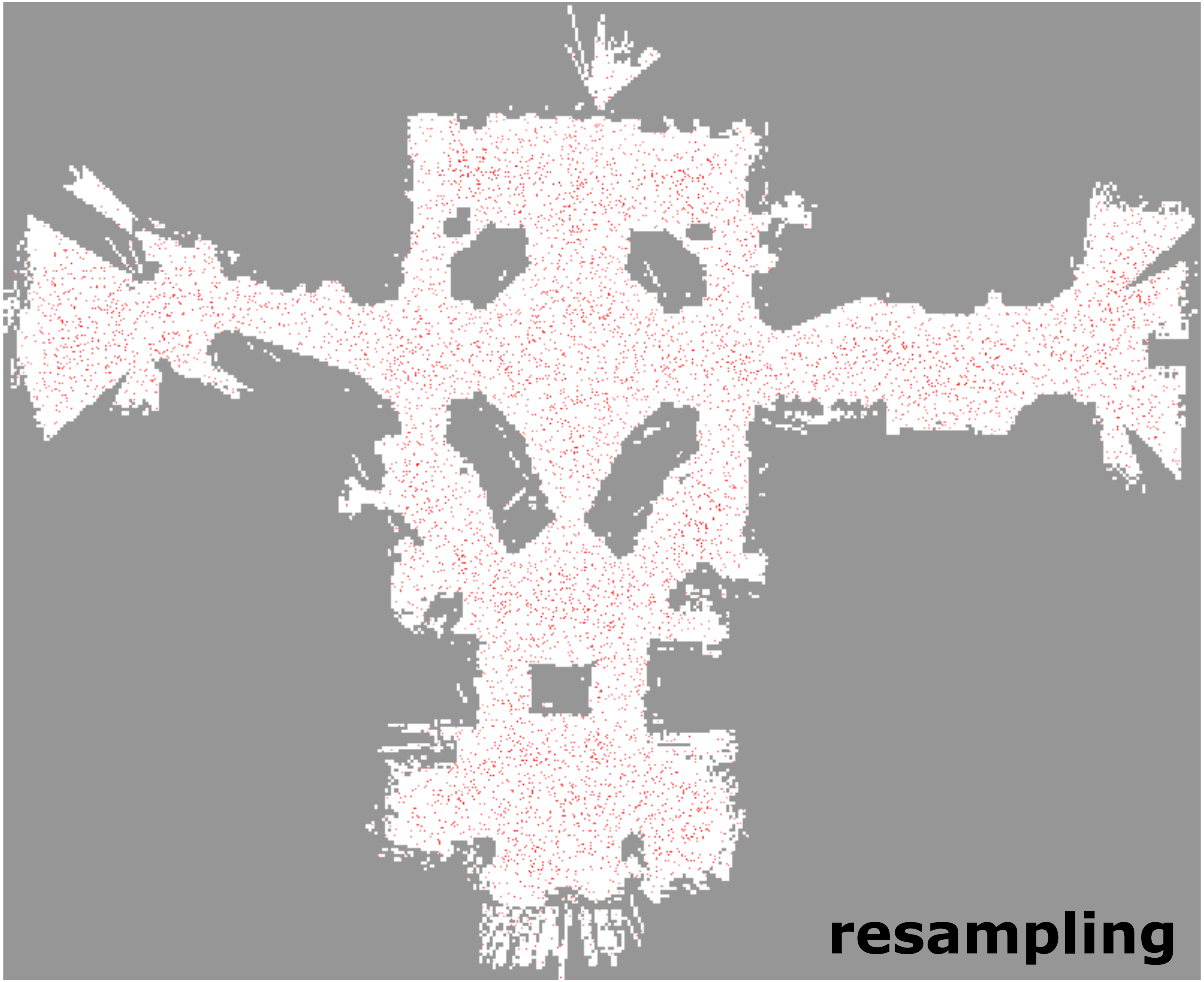
  add $x_t^{[i]}$ *to* $\bar{\mathcal{X}}_t$

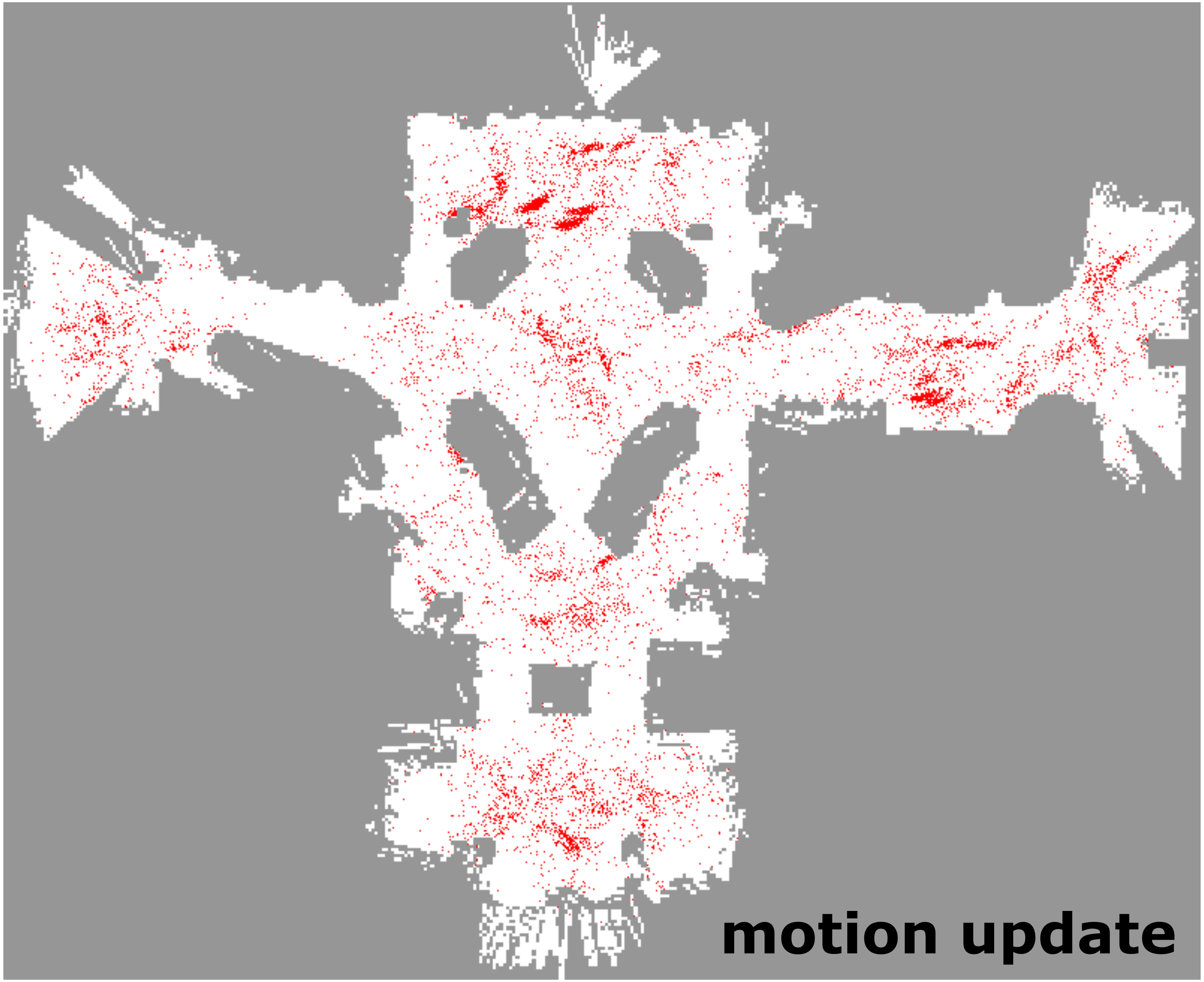
endfor

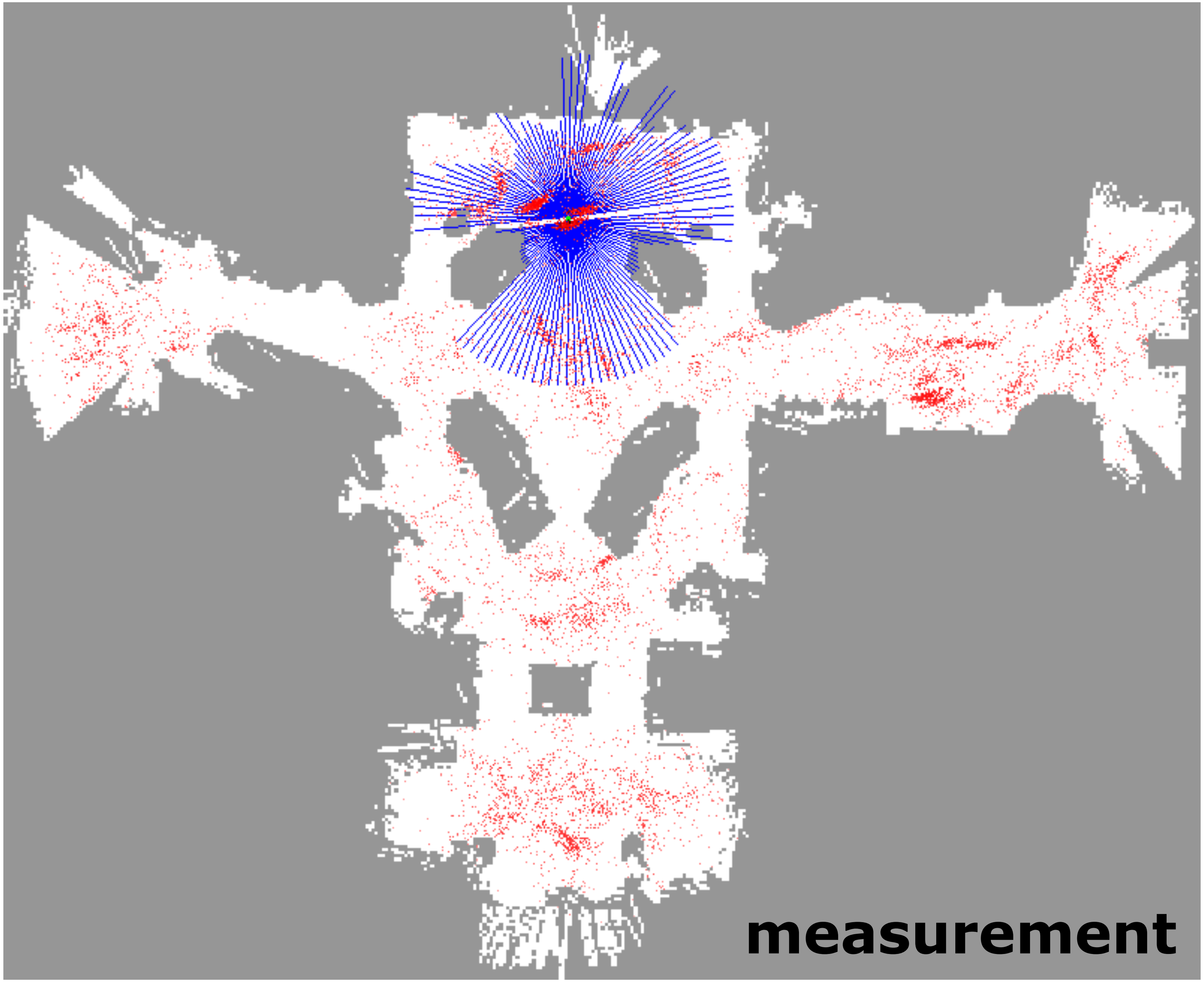
return $\bar{\mathcal{X}}_t$


initialization

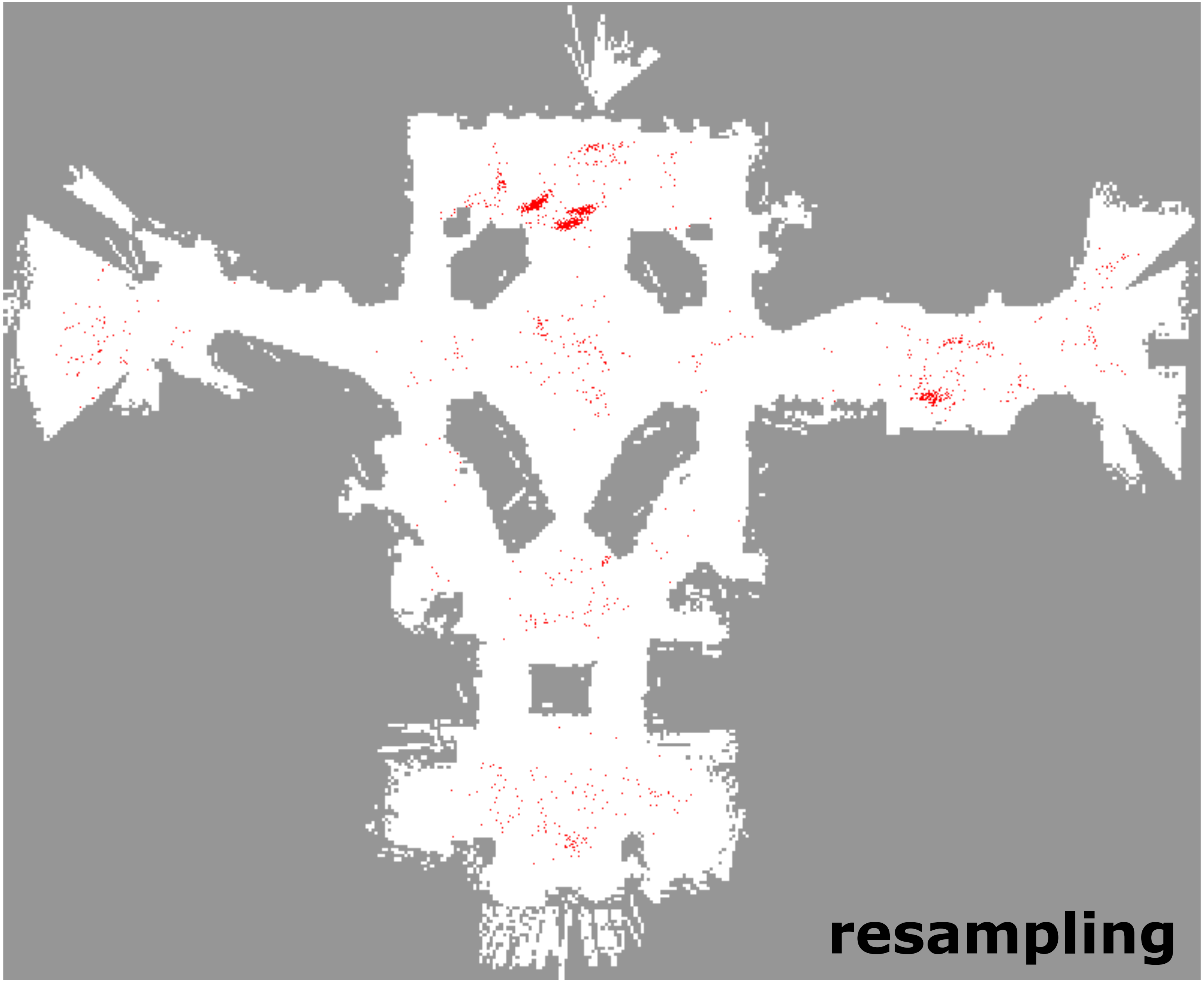

observation

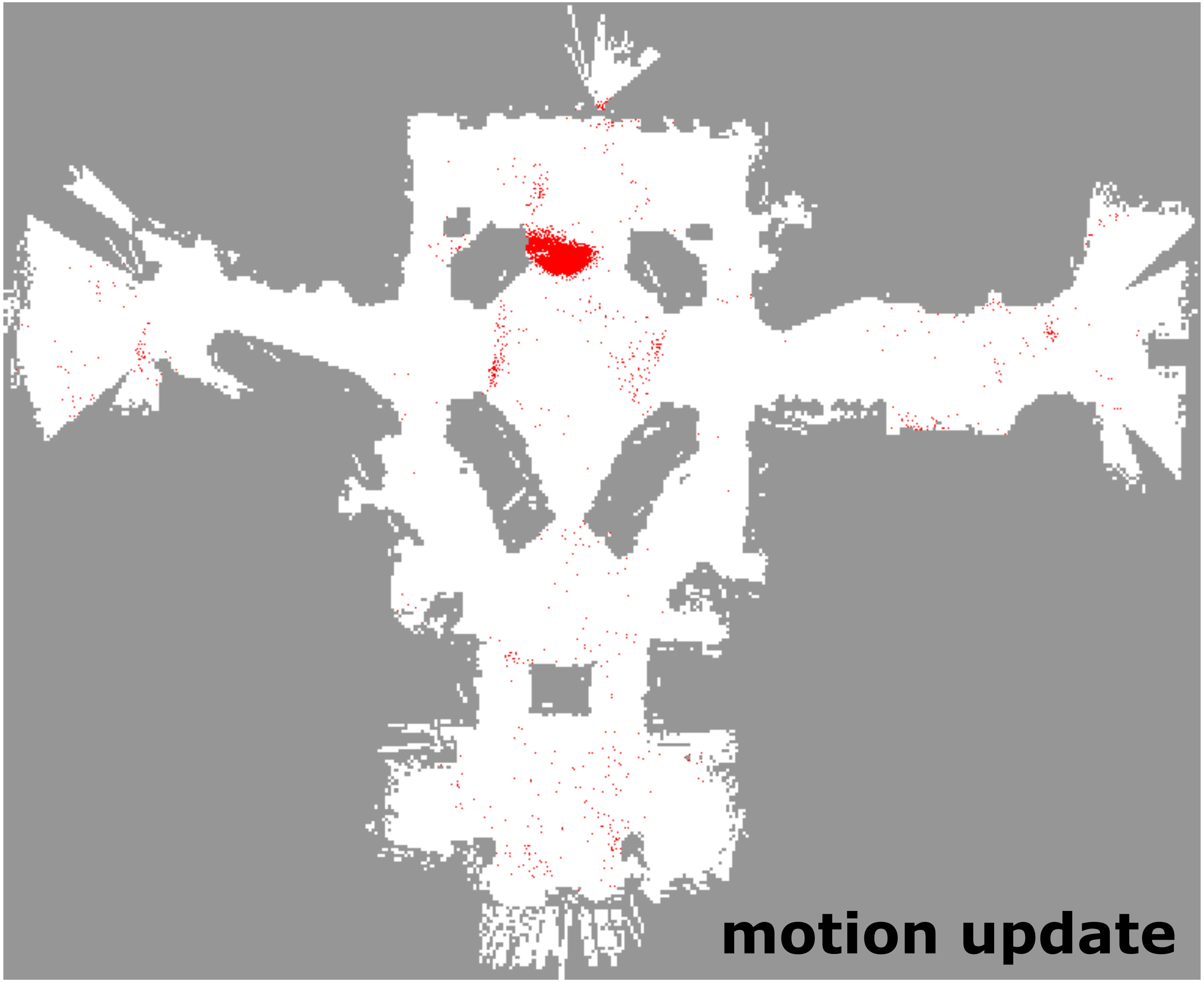

resampling

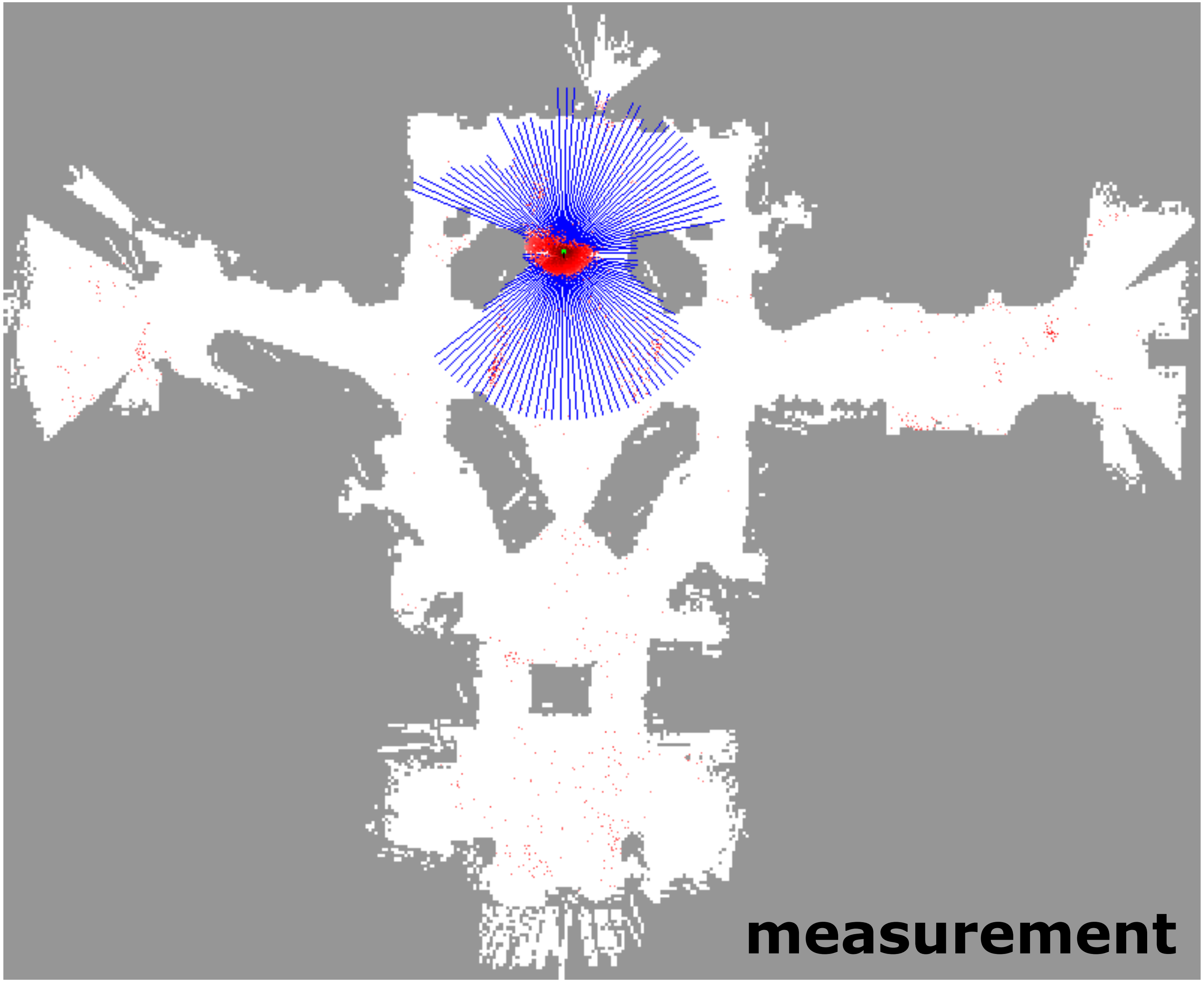

motion update


measurement

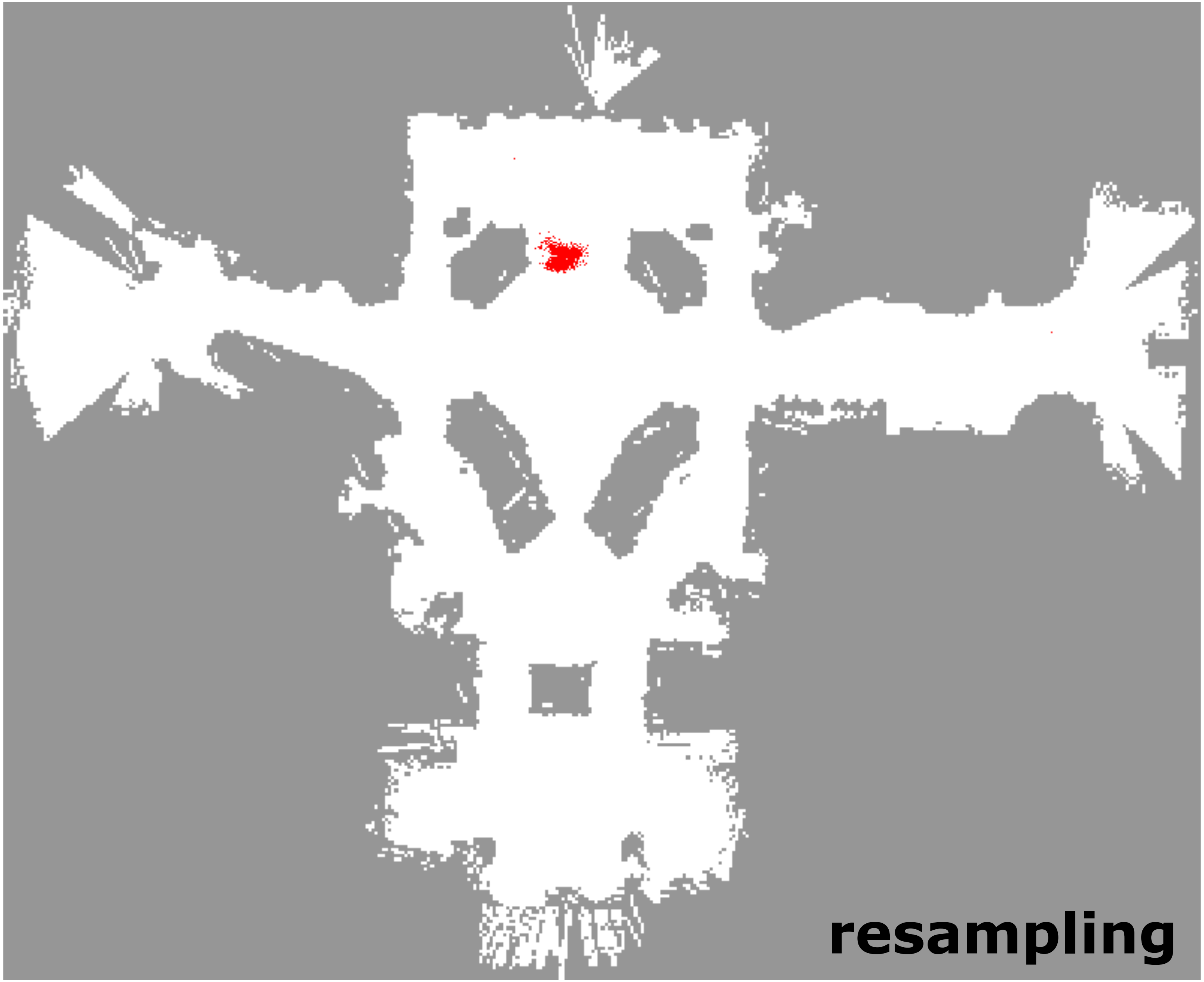

weight update

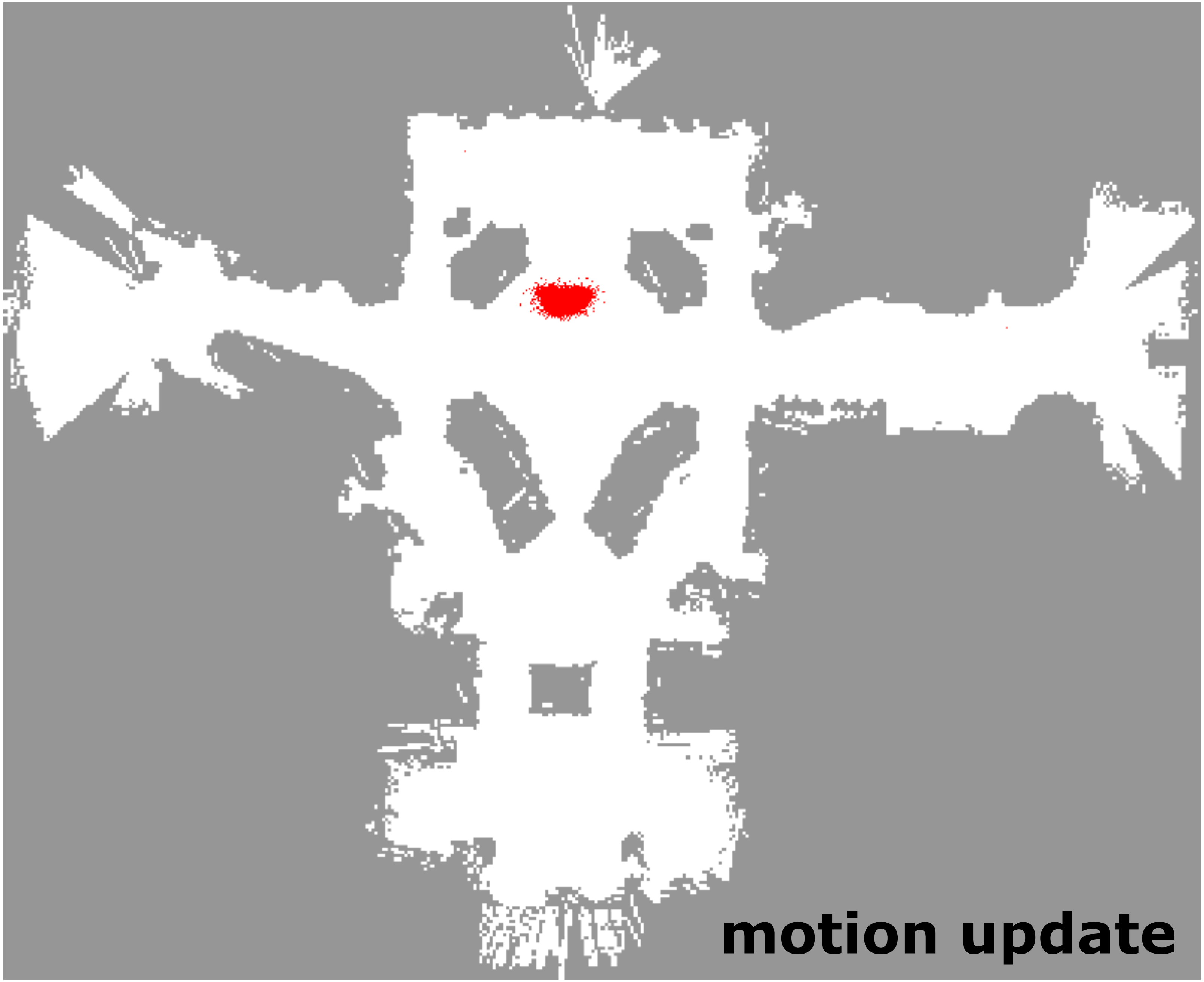

resampling

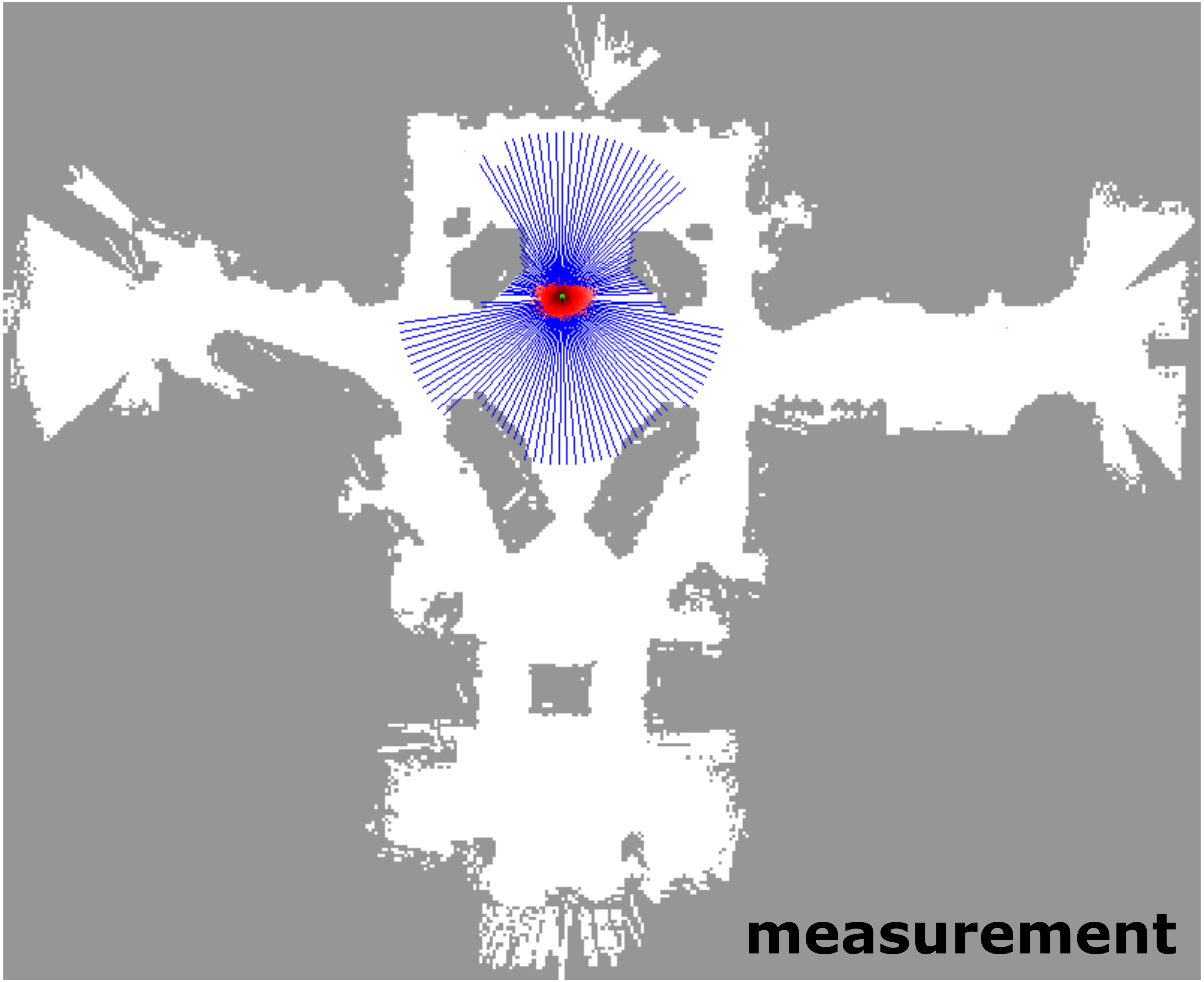

motion update

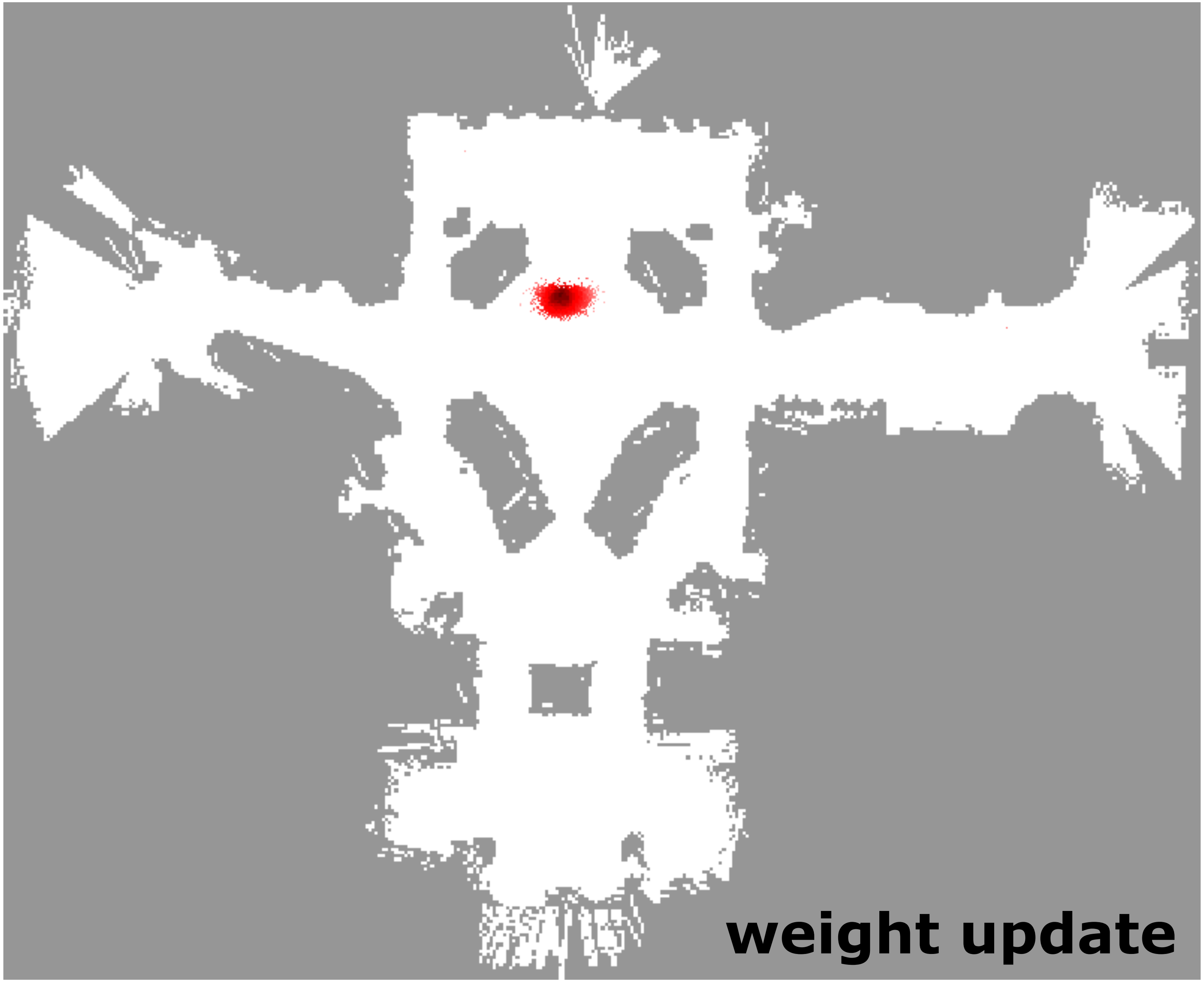

measurement

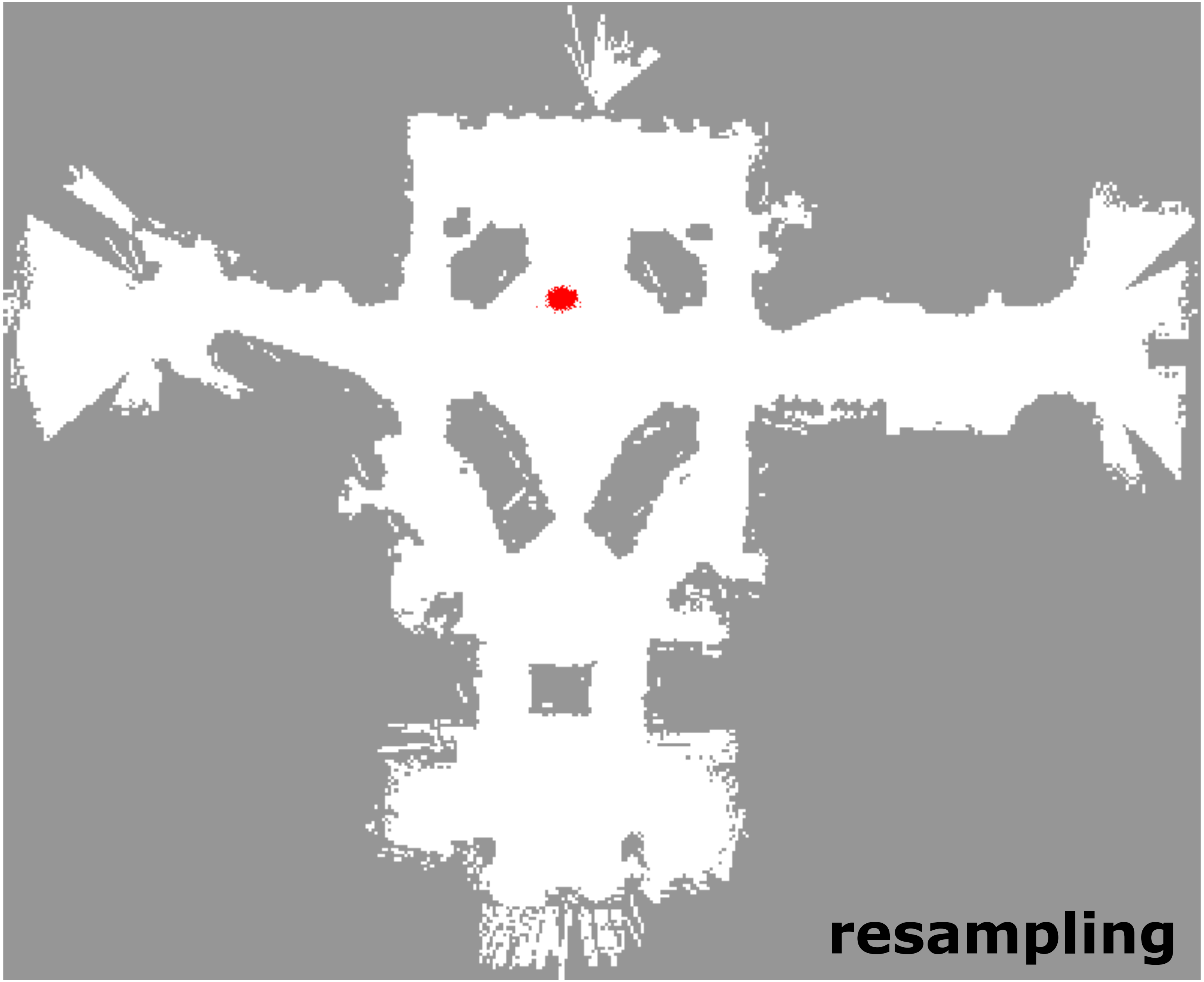

weight update

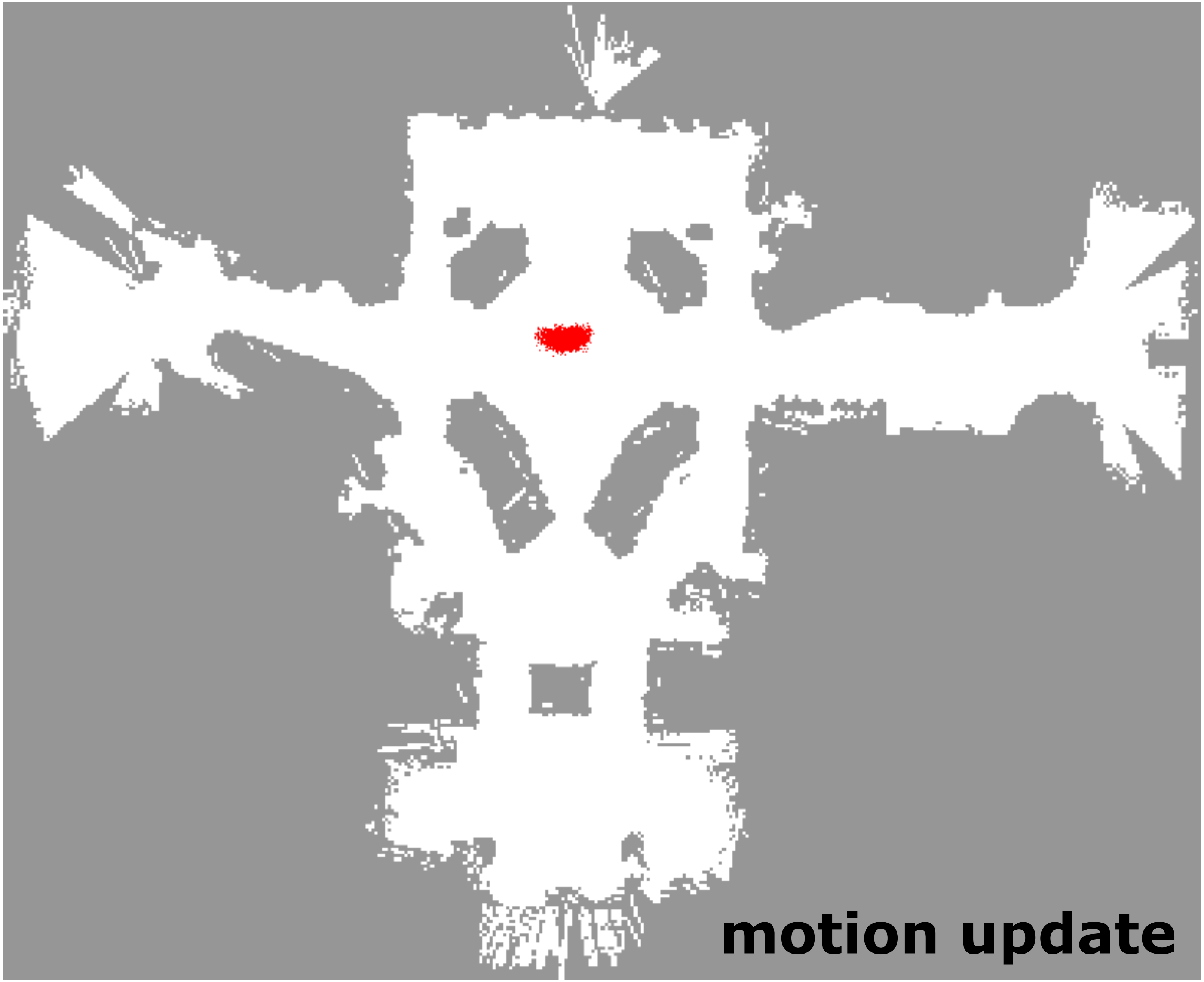

resampling

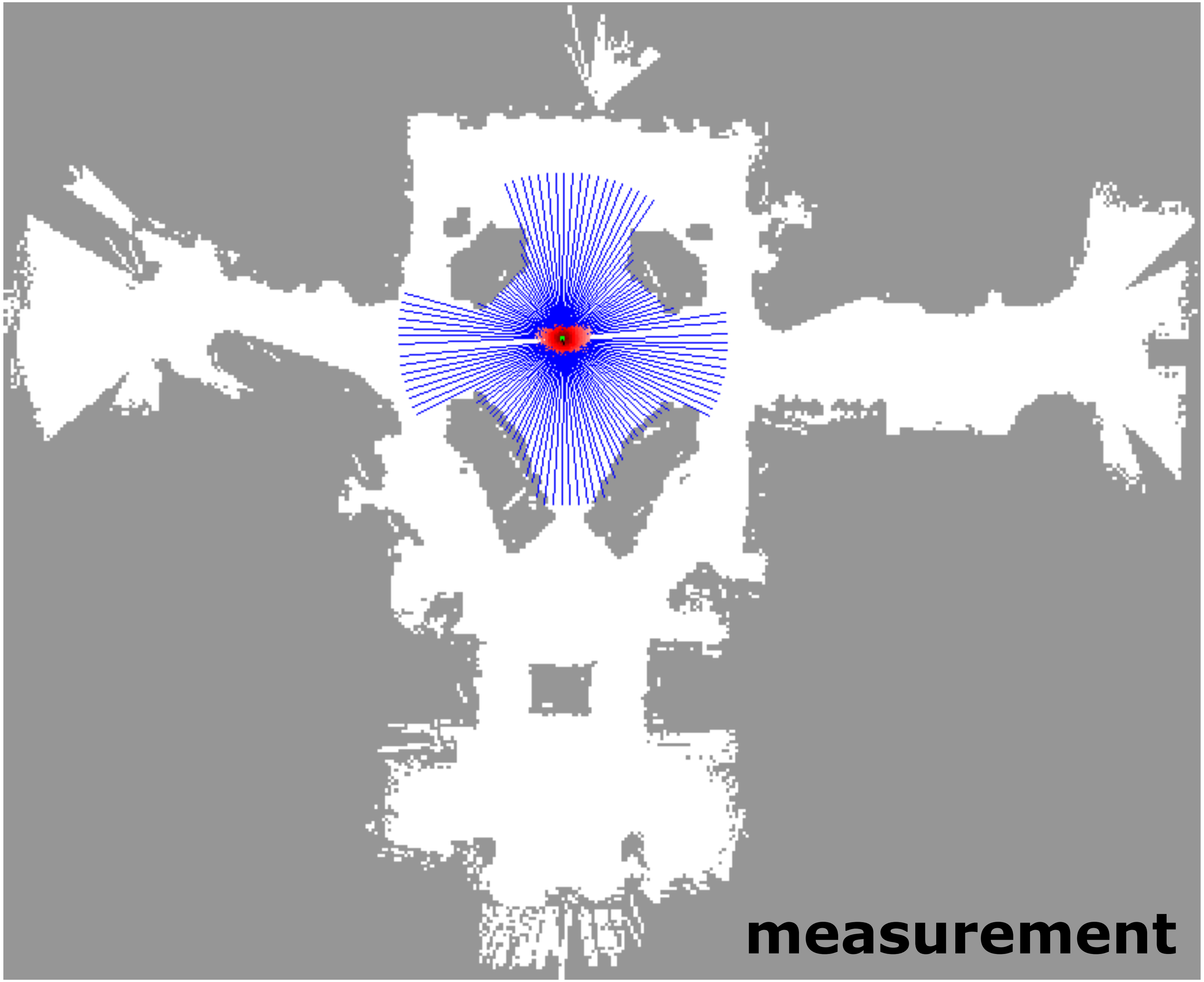

motion update


measurement


weight update


resampling


motion update


measurement

Summary – Particle Filters

- Particle filters are non-parametric, recursive Bayes filters
- Posterior is represented by a set of weighted samples
- Proposal to draw the samples for t+1
- Weight to account for the differences between the proposal and the target
- Work well in low-dimensional spaces

Summary – PF Localization

- Particles are propagated according to the motion model
- They are weighted according to the likelihood of the observation
- Called: Monte-Carlo localization (MCL)
- MCL is the gold standard for mobile robot localization today

Literature

On Monte Carlo Localization

- Thrun et al. “Probabilistic Robotics”, Chapter 8.3

On the particle filter

- Thrun et al. “Probabilistic Robotics”, Chapter 3

On motion and observation models

- Thrun et al. “Probabilistic Robotics”, Chapters 5 & 6