CS 3630!

Lecture 11:
Monte Carlo Inference

Topics

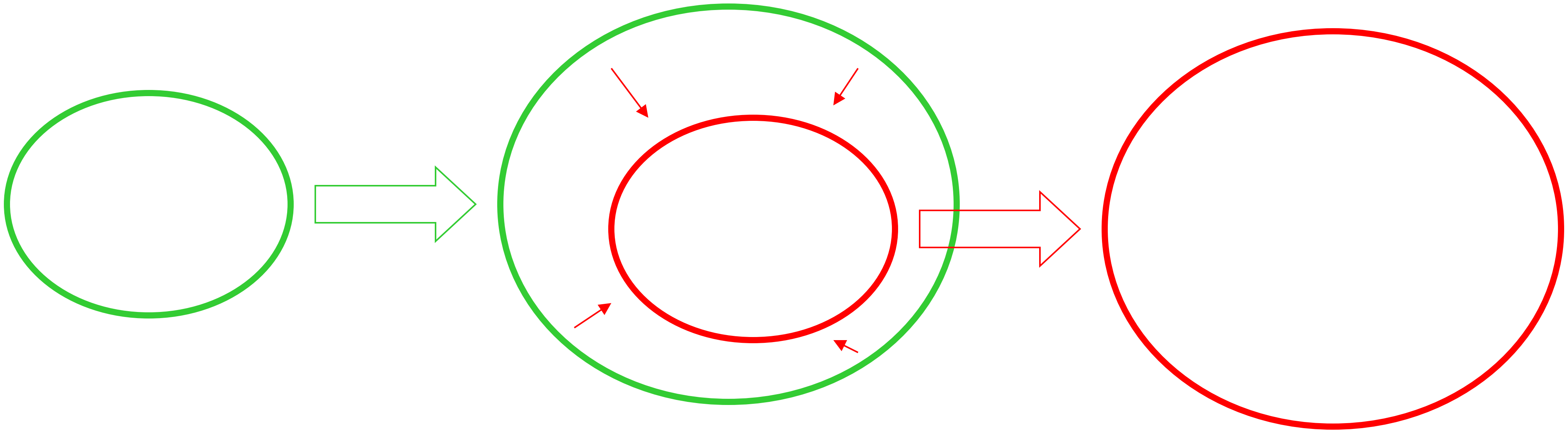
- Review: continuous models
- Sampling as Approximation
- Importance Sampling
- Particle Filters and Monte Carlo Localization

Motivation

- Robots live in a continuous world
- To localize the robot, we need probabilistic inference
- Many of the concepts we discussed before generalize
- In many cases exact inference is intractable -> sampling
- A popular class of algorithm: Particle filters & Monte Carlo Localization

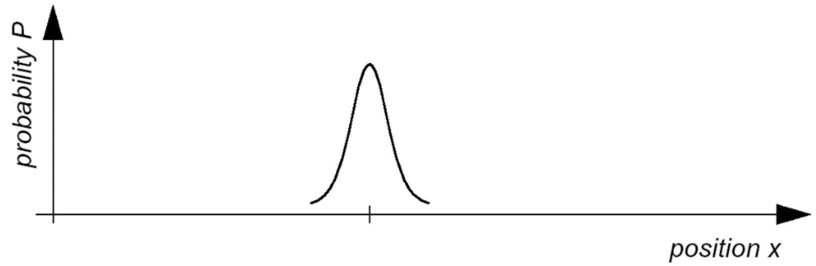
Remember: the Bayes Filter

- Two phases: a. Prediction Phase
  b. Measurement Phase

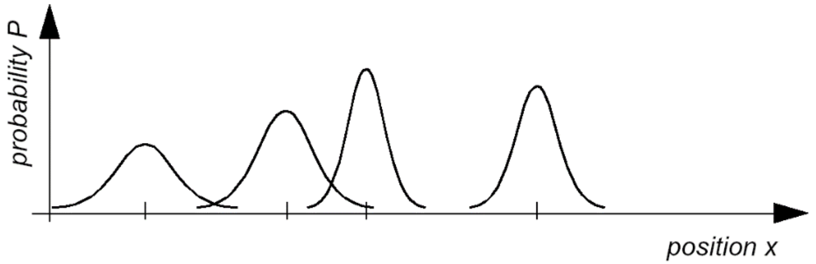


Belief representation: how do we represent our belief of where the robot is located?

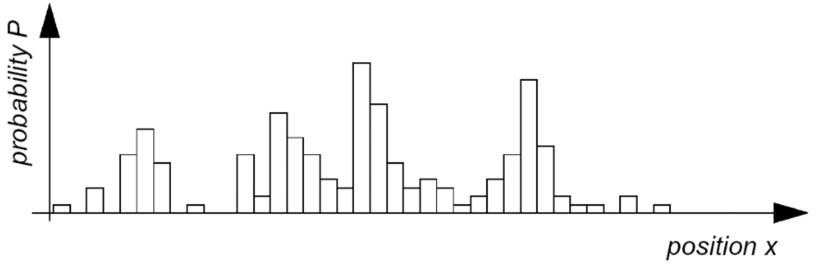
Continuous map with single hypothesis probability distribution



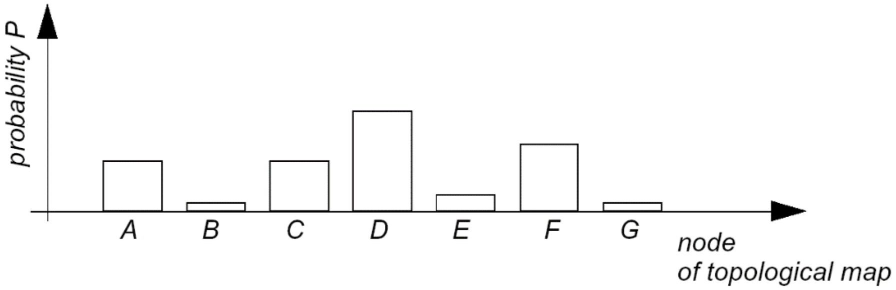
Continuous map with multiple hypotheses probability distribution



Discretized map with multiple hypotheses probability distribution

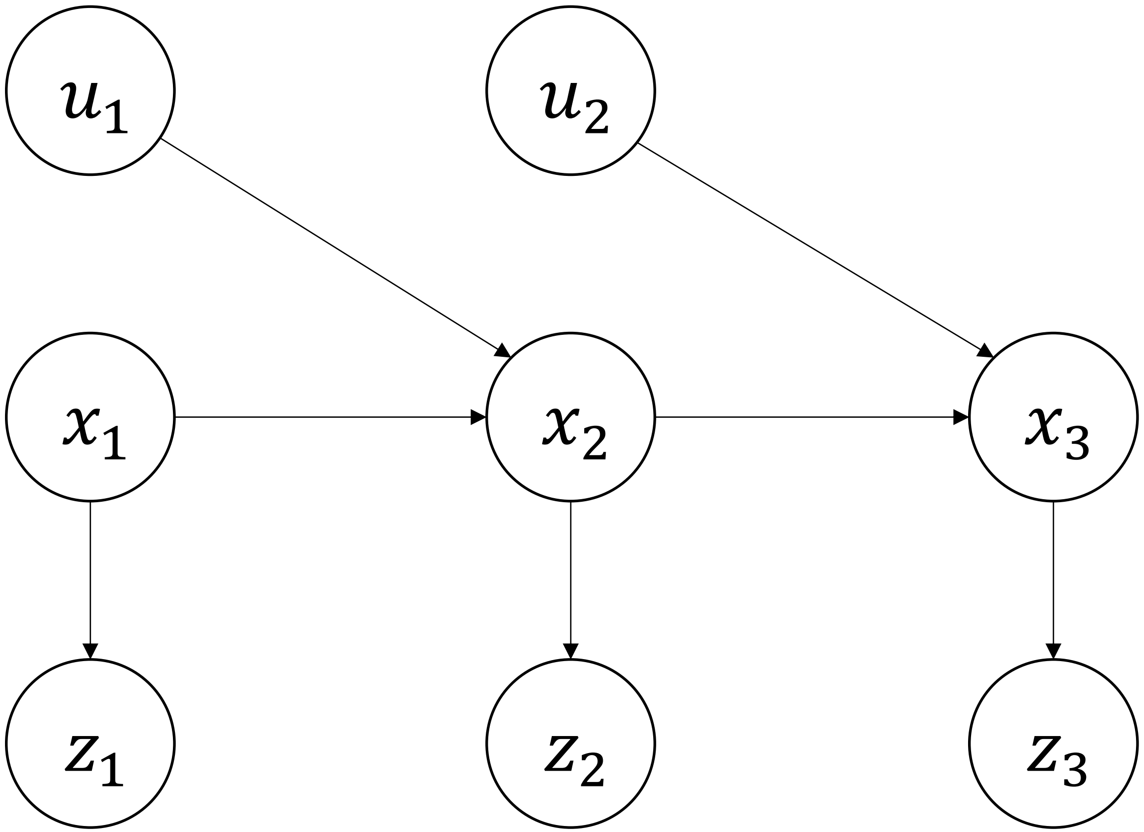


Discretized topological map with with multiple hypotheses probability distribution



Continuous Bayes Nets

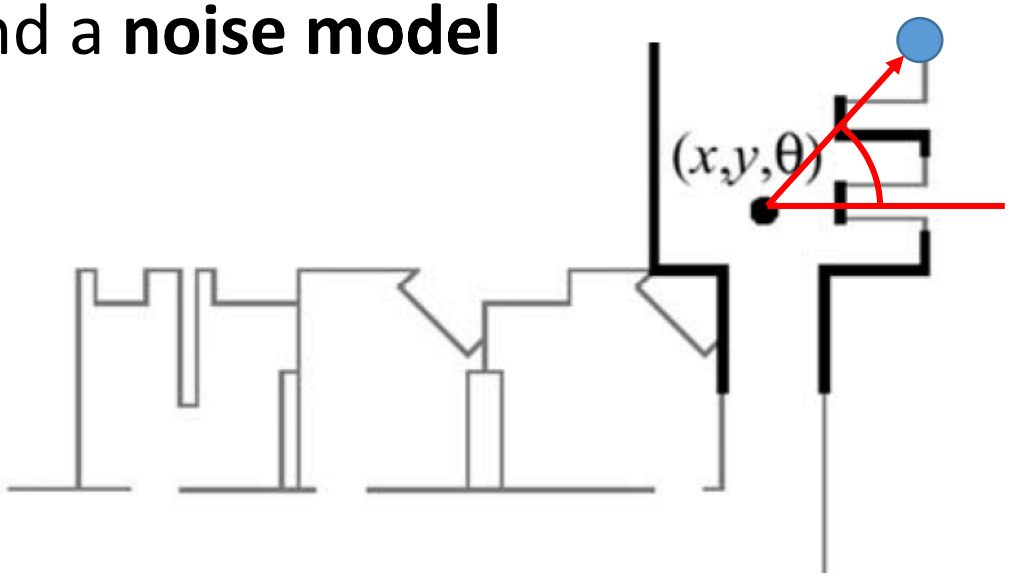
- As before, but now states S, observations O, and action A can all be continuous.
- Terminology: x, z, u
- Hence: measurement models and state transition models are continuous.



Continuous Measurement Models

- We need a **measurement function** and a **noise model**
- Example: bearing to a landmark l:

$$h(x, l) = atan2(l_y - x_y, l_x - x_x)$$



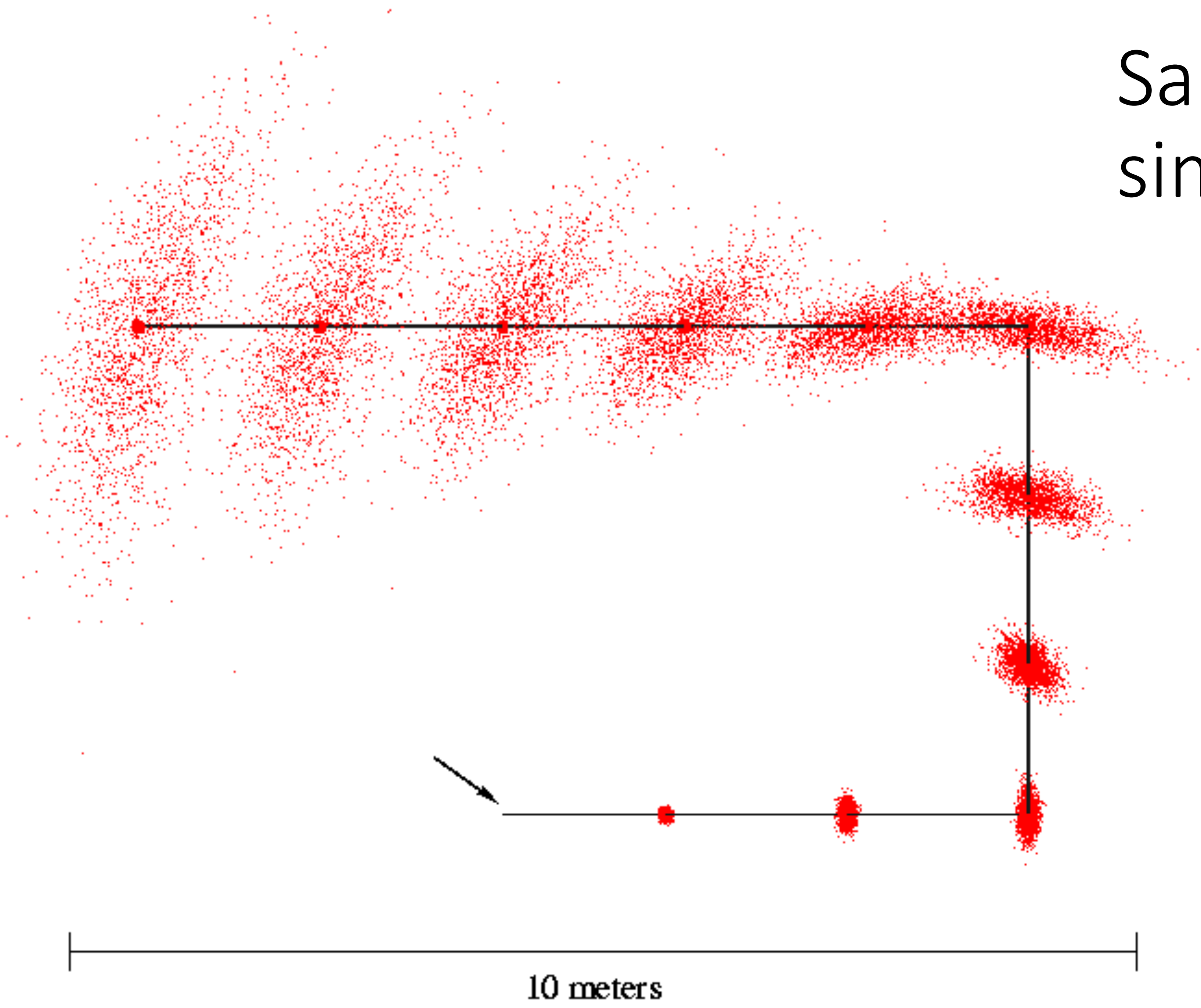
$$p(z|x, l) = \frac{1}{\sqrt{|2\pi R|}} \exp\left\{-\frac{1}{2}\|atan2(l_y - x_y, l_x - x_x) - z\|_R^2\right\}$$

Continuous Motion Models

- Similar for state transition, but we now have a motion model
- Motion model $g(x,u)$ takes state x and control u
- Multivariate noise model with covariance Q:

$$p(x_{t+1}|x_t, u_t) = \frac{1}{\sqrt{|2\pi Q|}} \exp\left\{-\frac{1}{2}\|g(x_t, u_t) - x_{t+1}\|_Q^2\right\}$$

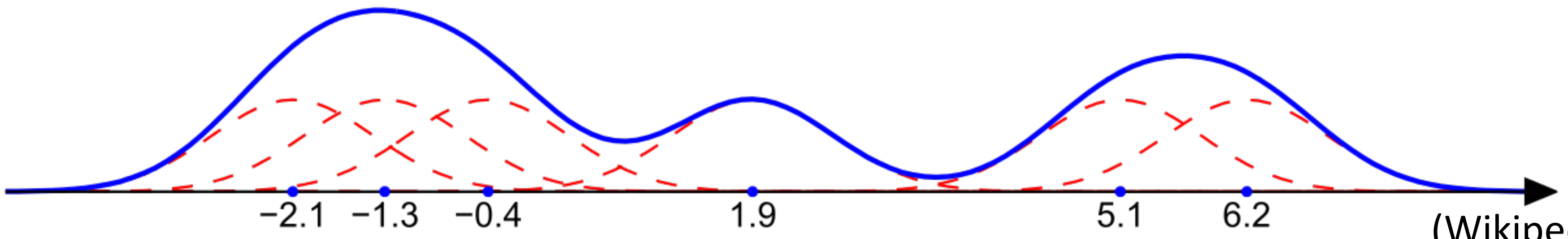
Sampling for simulation



- The infamous "banana density"
- Happens because we also sample heading \theta
- Clearly non-Gaussian!

Sampling to Approximate Densities

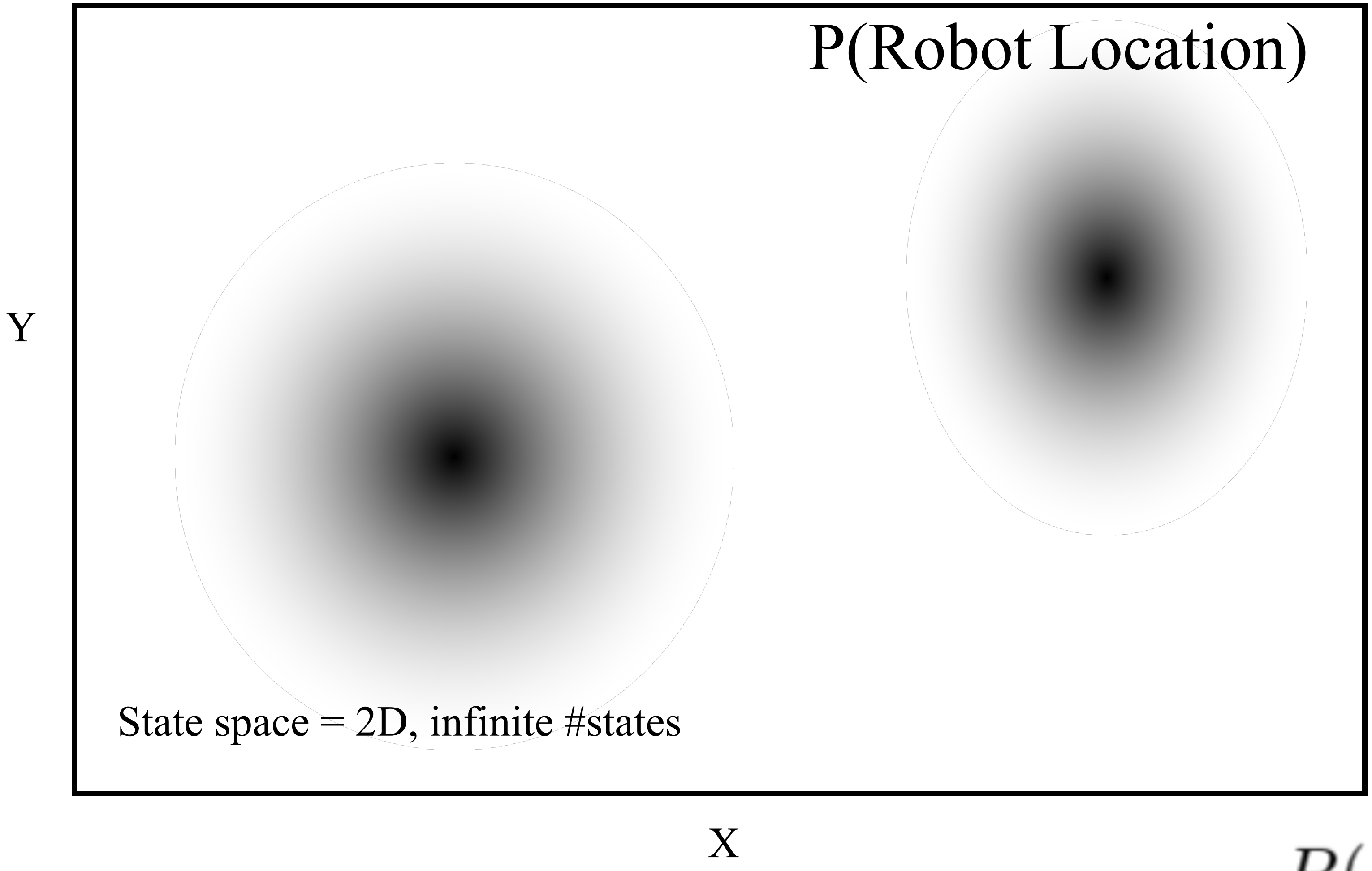
- As banana distribution illustrates, densities can become arbitrarily complex, even when noise models are Gaussian
- Issue is nonlinear measurement and noise models
- One way out: **Parzen window density estimation** (mixtures!)



(Wikipedia)

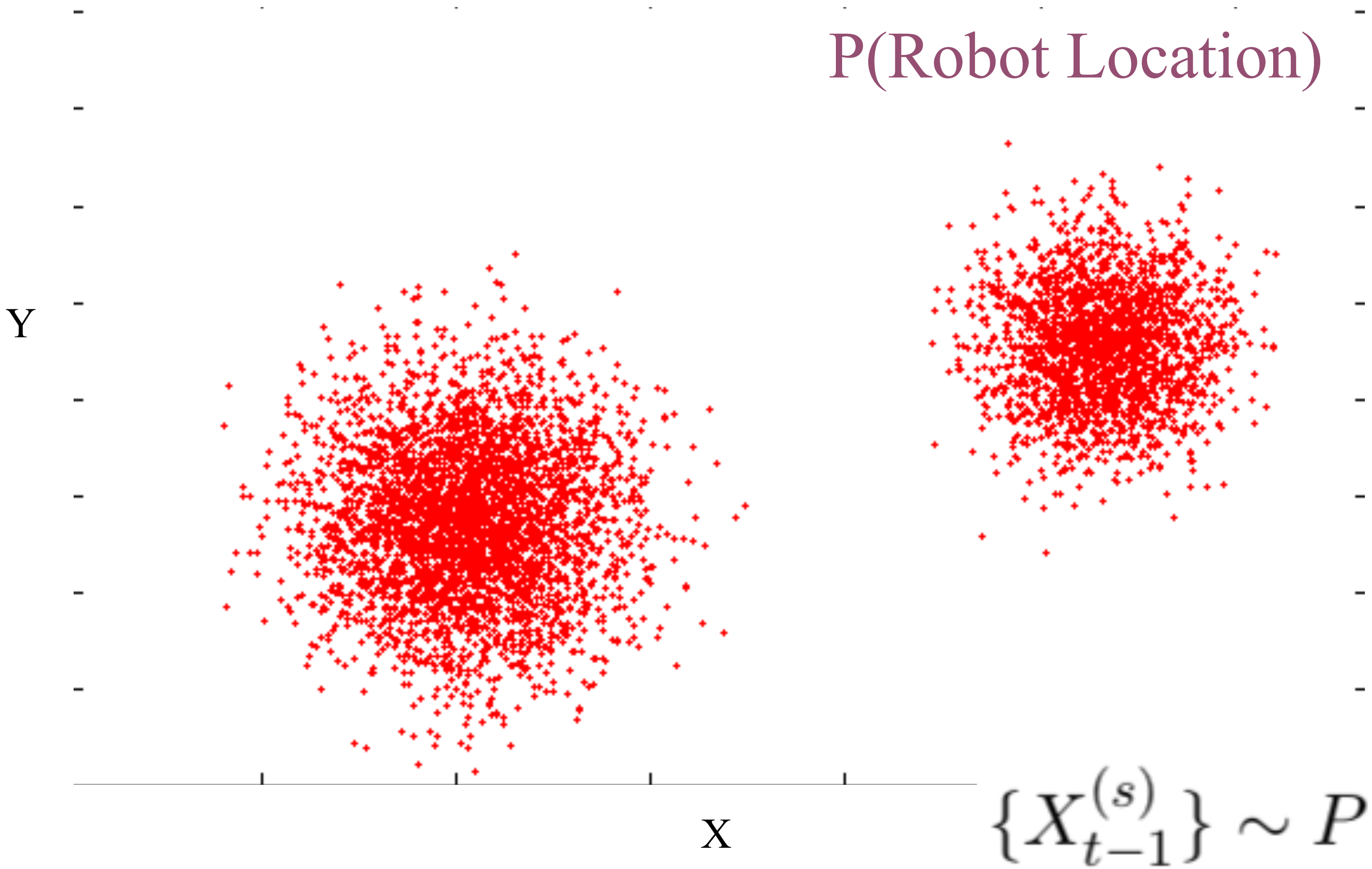
- Other way out: ***sampling!***

Probability of Robot Location



$$P(X_{t-1}|Z^{t-1})$$

Sampling as Representation



$$\{X_{t-1}^{(s)}\} \sim P(X_{t-1}|Z^{t-1})$$

Sampling Advantages

- Arbitrary densities
- Memory = O(#samples)
- Only in “Typical Set”
- Great visualization tool !

- minus: Approximate

Importance Sampling

- Additionally, use weights to represent a density

$$\{X_{t-1}^{(r)}, \pi_{t-1}^{(r)}\} \sim P(X_{t-1}|Z^{t-1})$$

- Generic importance sampling idea:
  - We want to sample from $p(x)$, but we don't know how
  - sample $x^{(r)}$ from $q(x)$, which some way we can sample from
  - give each sample $x^{(r)}$ an importance weight equal to $p(x)/q(x)$
- Specific example: Bayes law:
  - Sample $x^{(r)}$ from prior $p(x)$
  - weight each sample $x^{(r)}$ with likelihood $l(x;z)$

Importance Sampling

- Sample $x^{(r)}$ from $q(x)$
- $\pi_r = p(x^{(r)})/q(x^{(r)})$

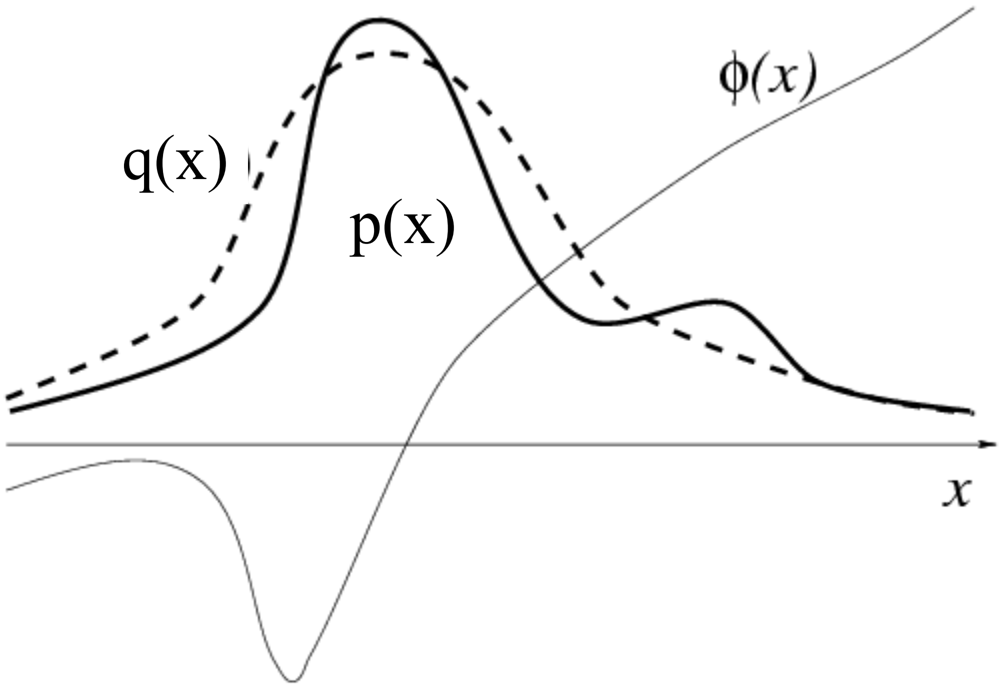
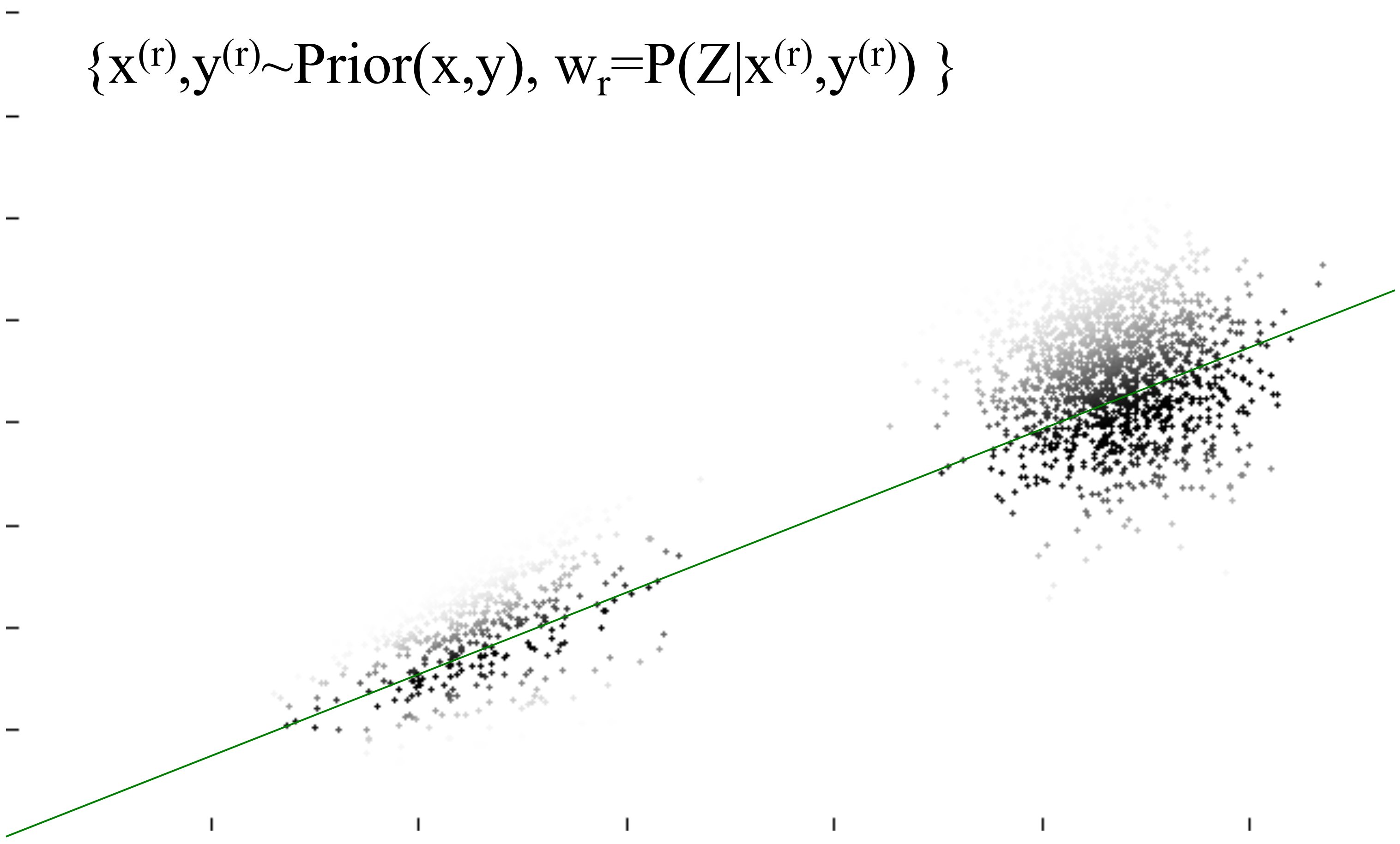


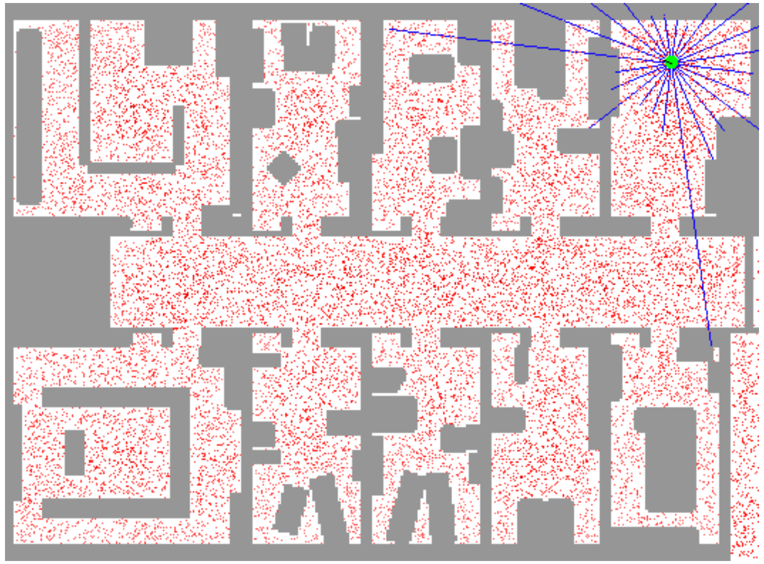
Image by MacKay

Example: Bayes law via importance sampling

$\{x^{(r)}, y^{(r)} \sim \text{Prior}(x,y),\ w_r = P(Z|x^{(r)}, y^{(r)})\ \}$



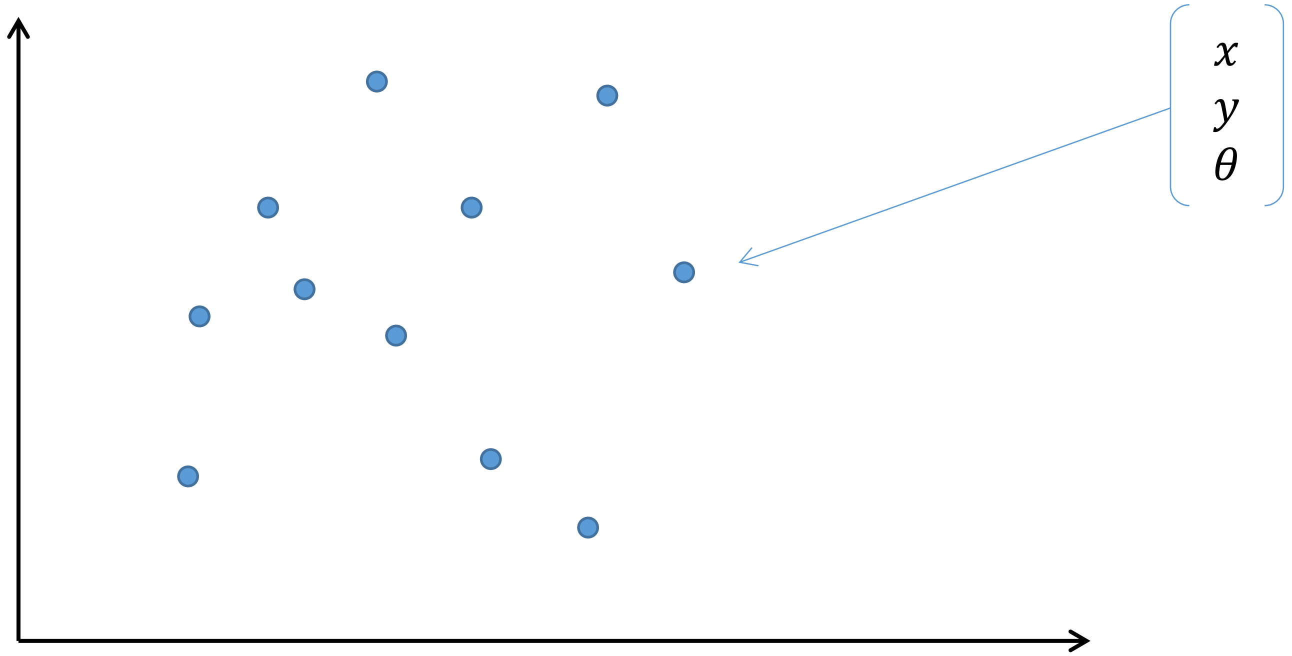
Particle Filters & Monte Carlo Localization



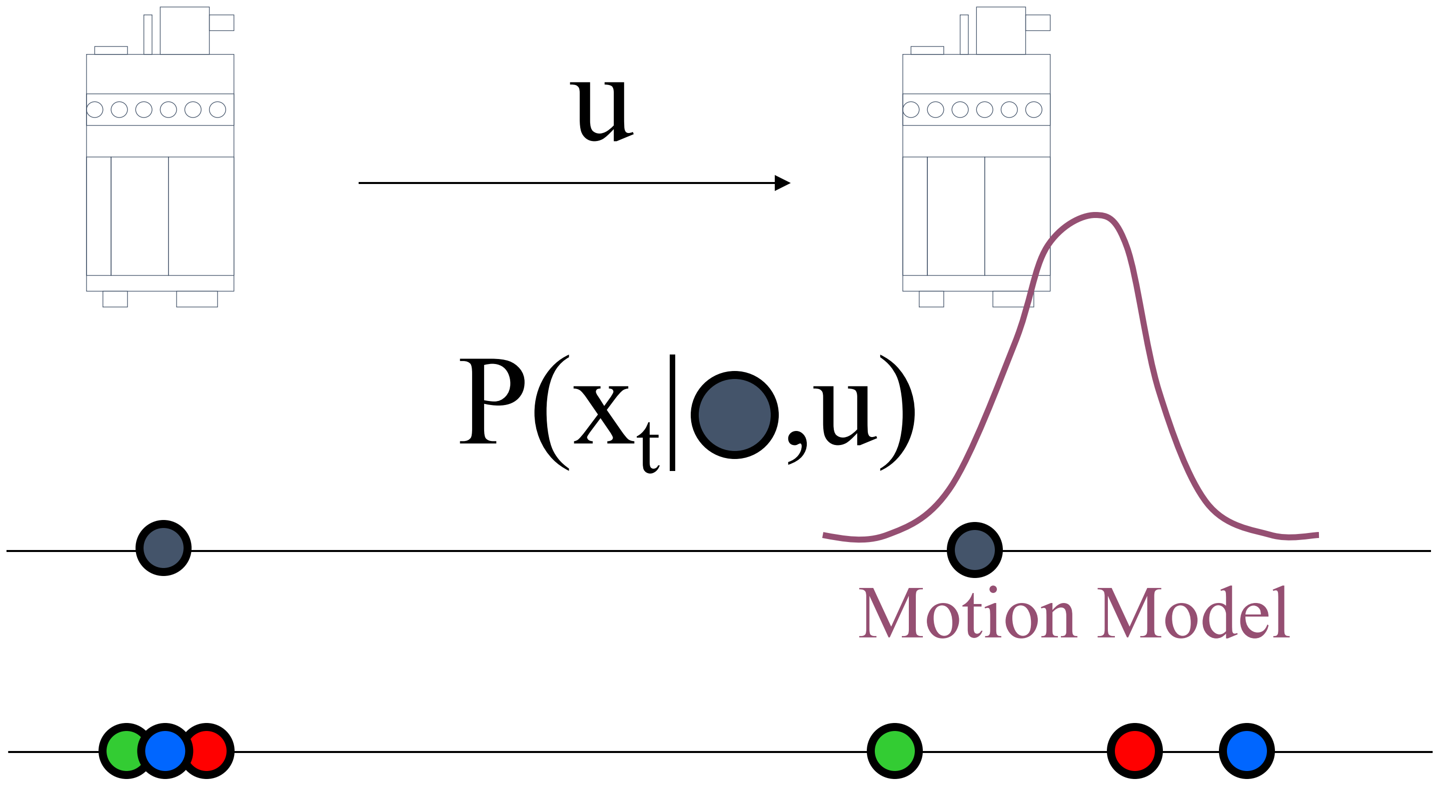
- Bayes filter using importance sampling for Bayes law
- First appeared in 70's, re-discovered by Kitagawa,
- Isard & Blake rediscovered in computer vision, as CONDENSATION
- Monte Carlo Localization in robotics

Particles

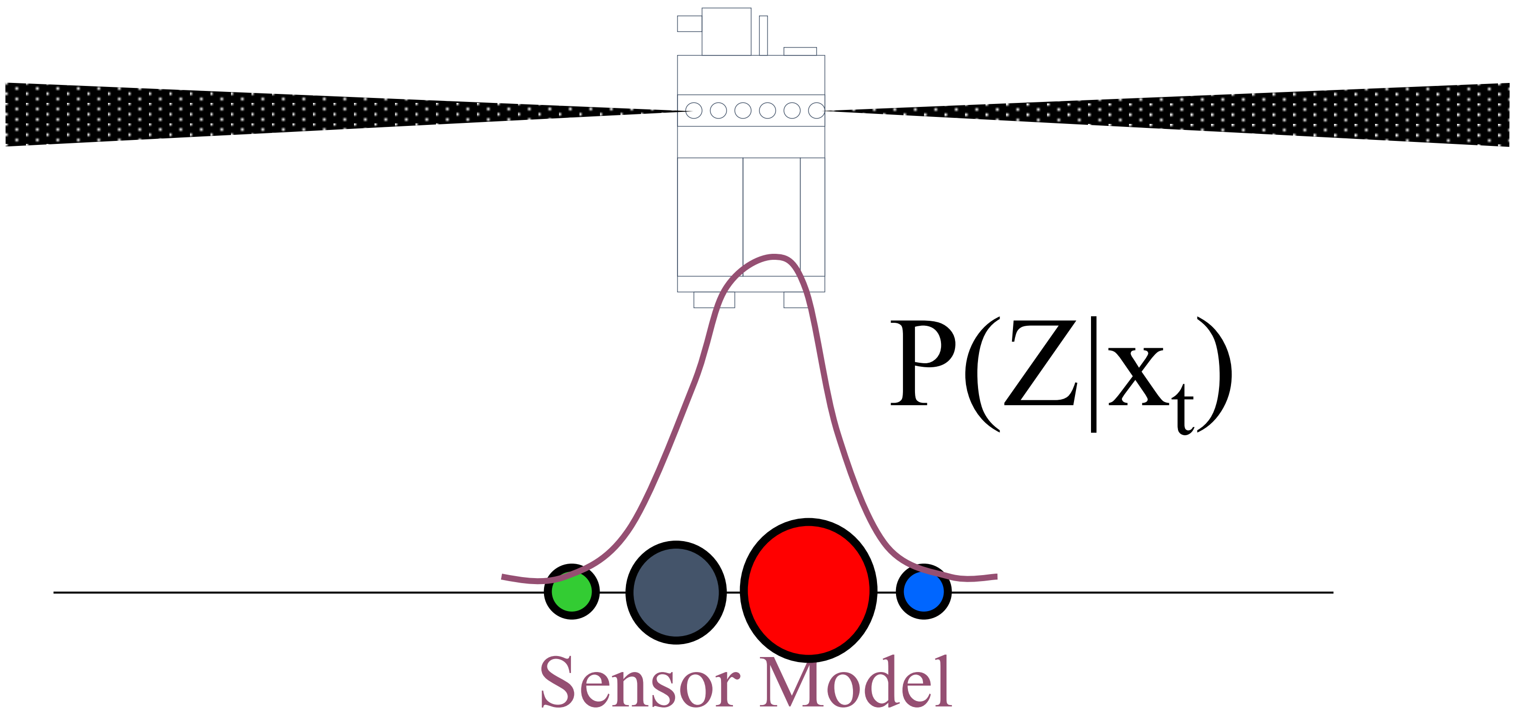
- Each particle is a guess about where the robot might be



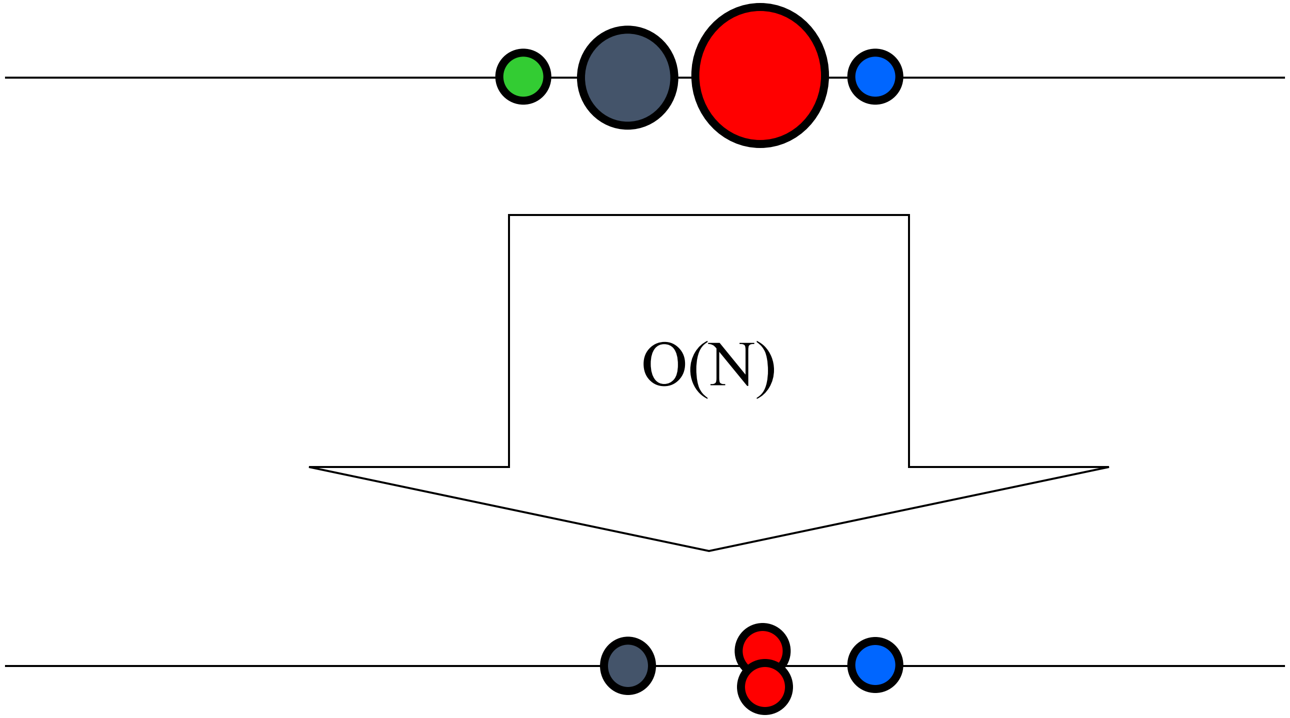
1. Prediction Phase



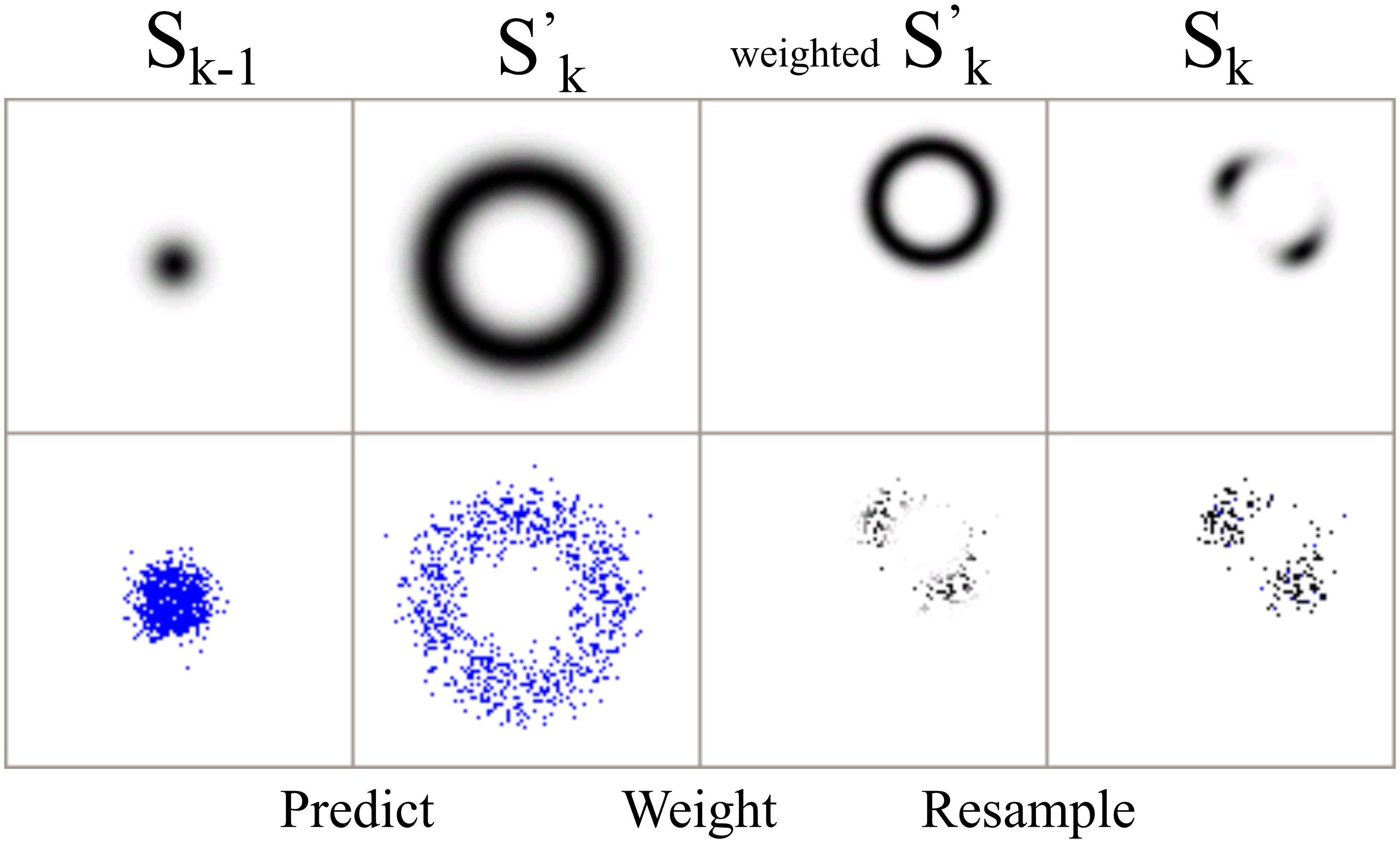
2. Measurement Phase



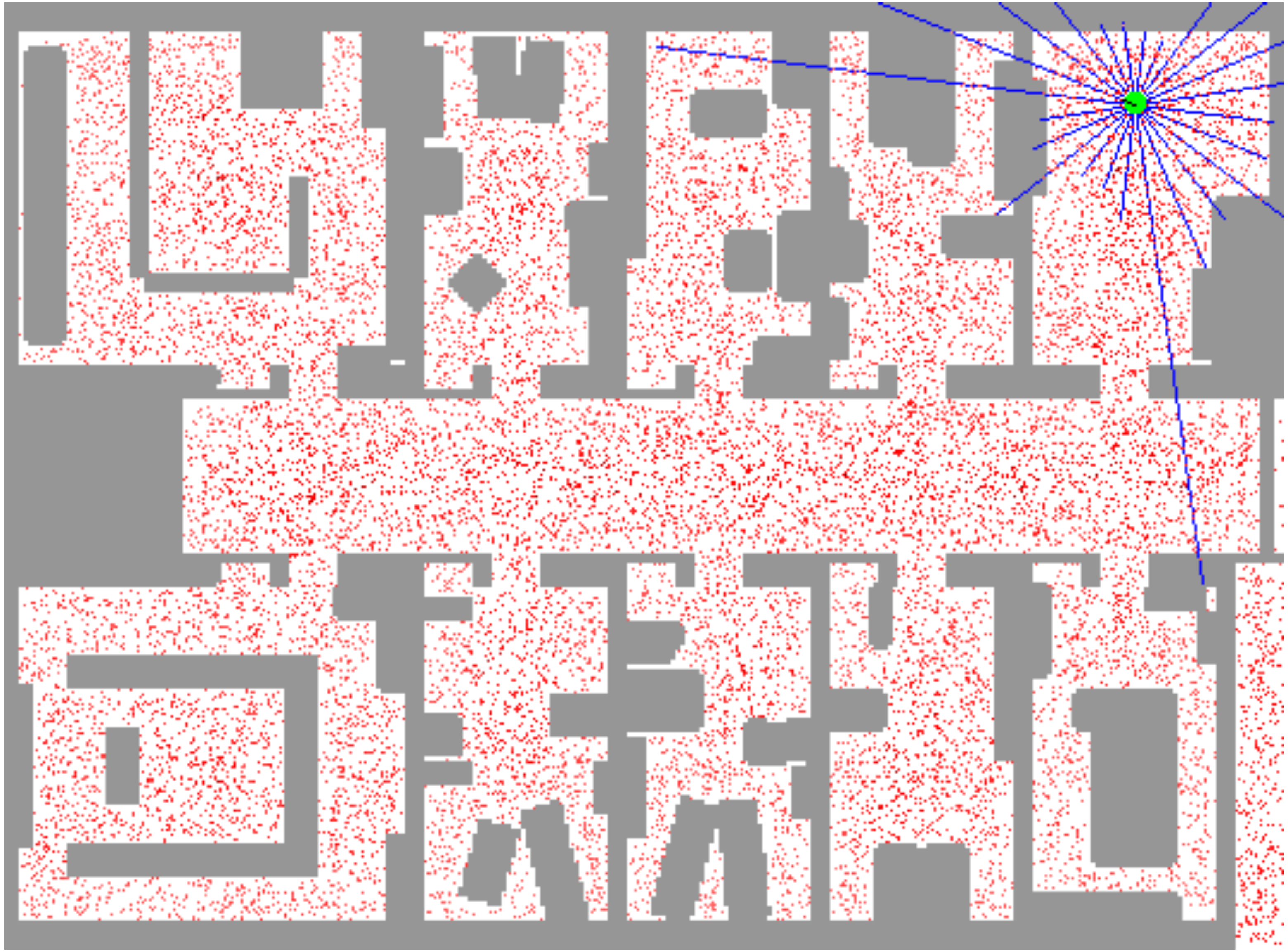
3. Resampling Step



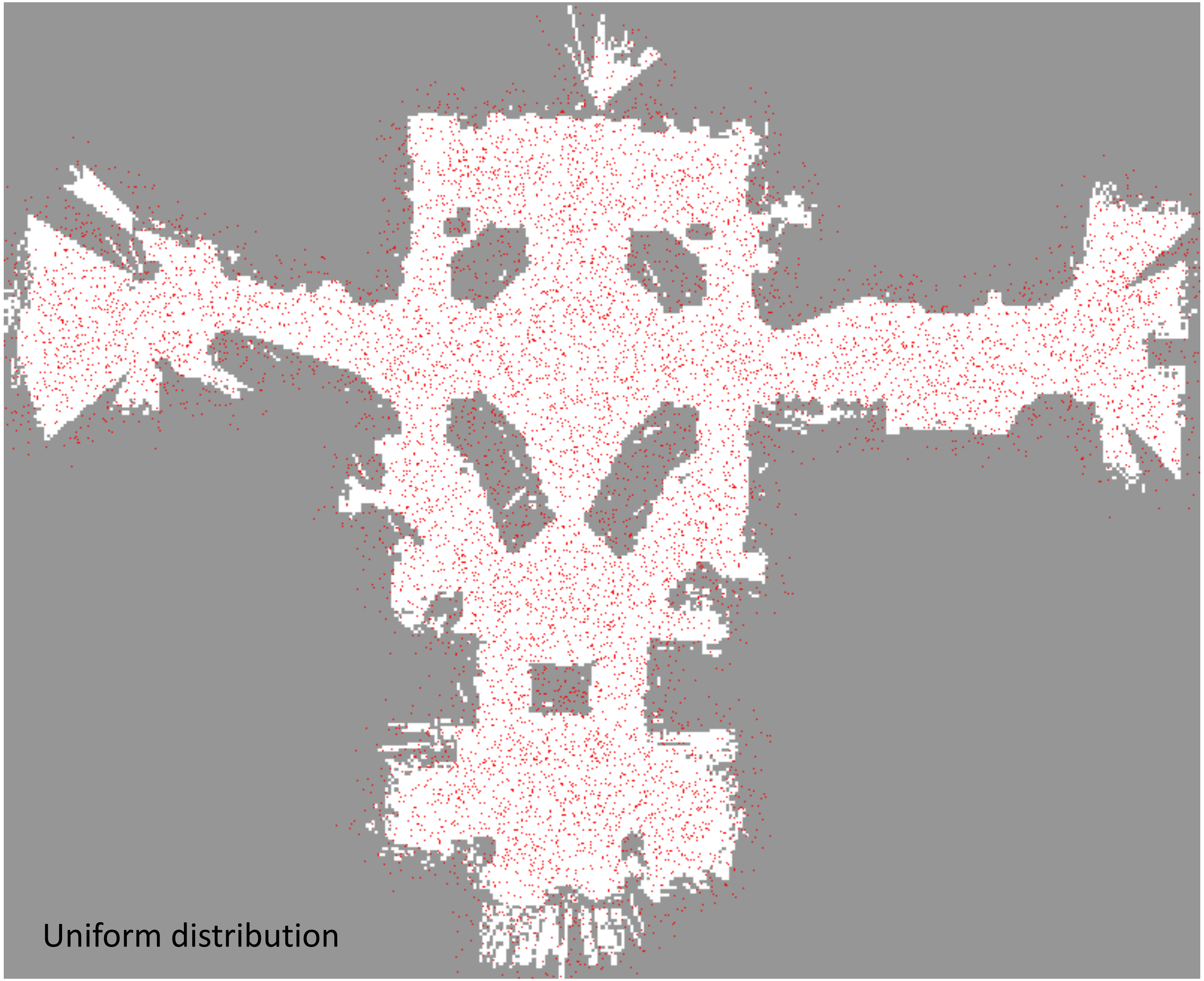
Monte Carlo Localization (ICRA 1999)

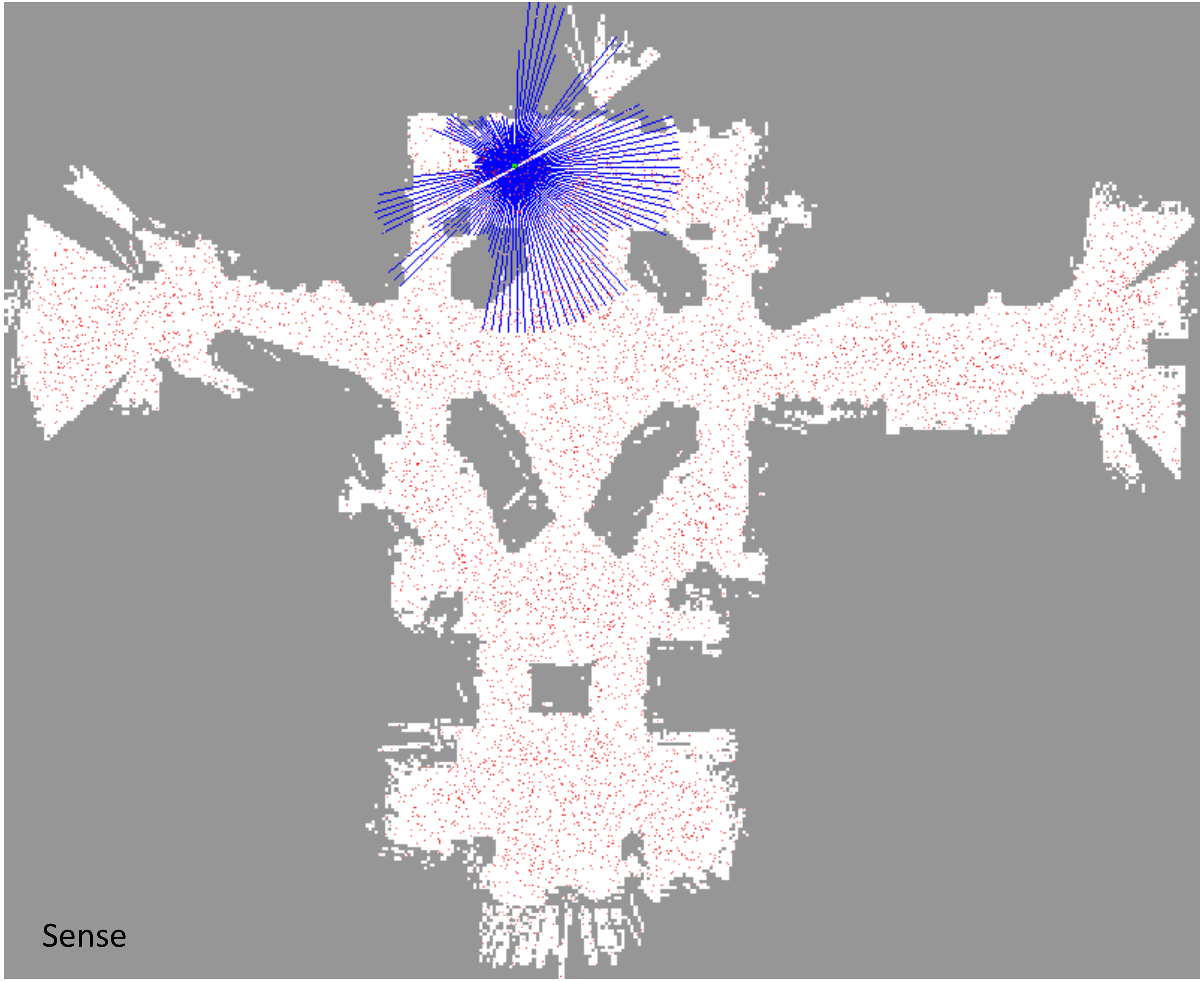


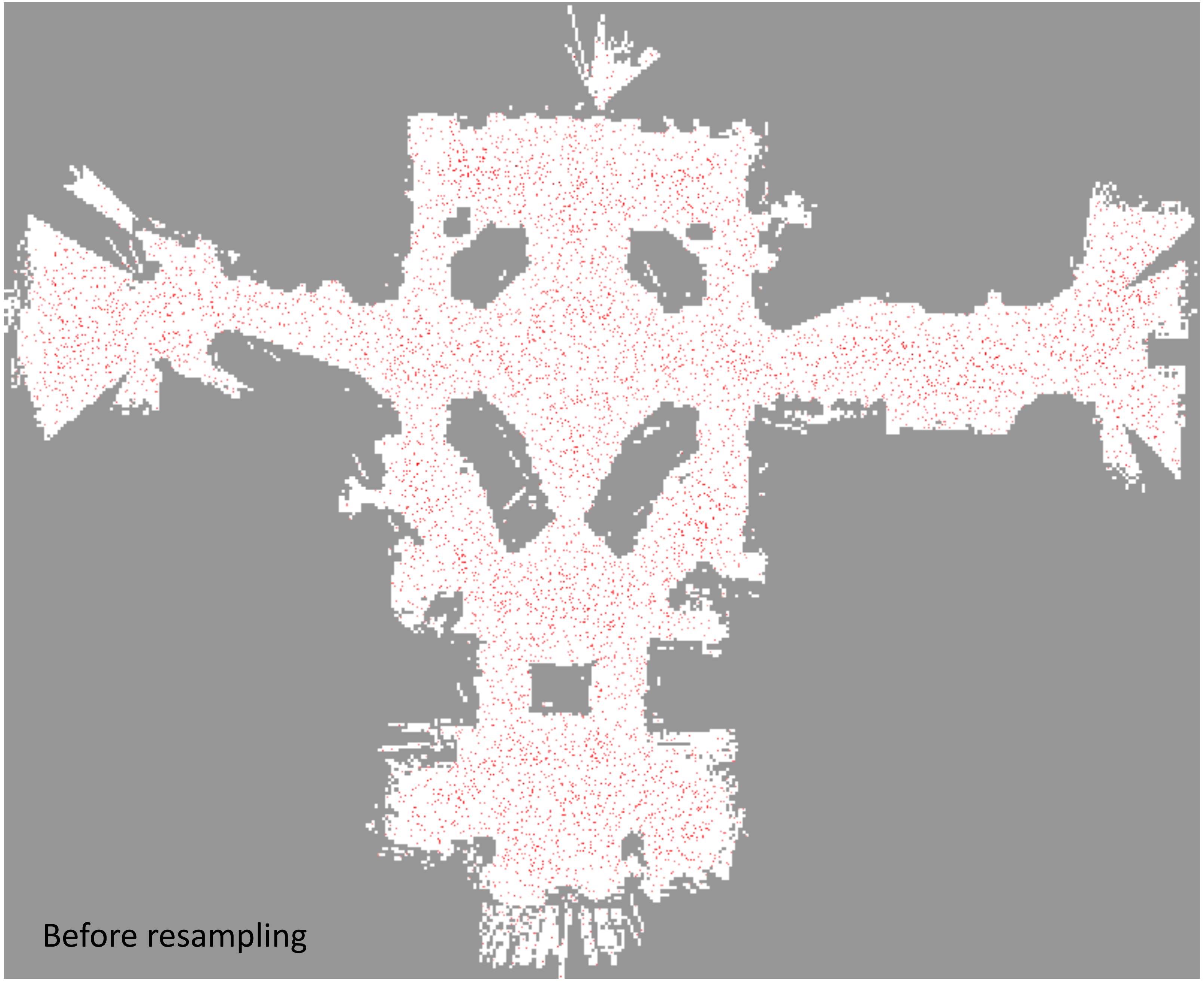
Monte Carlo Localization

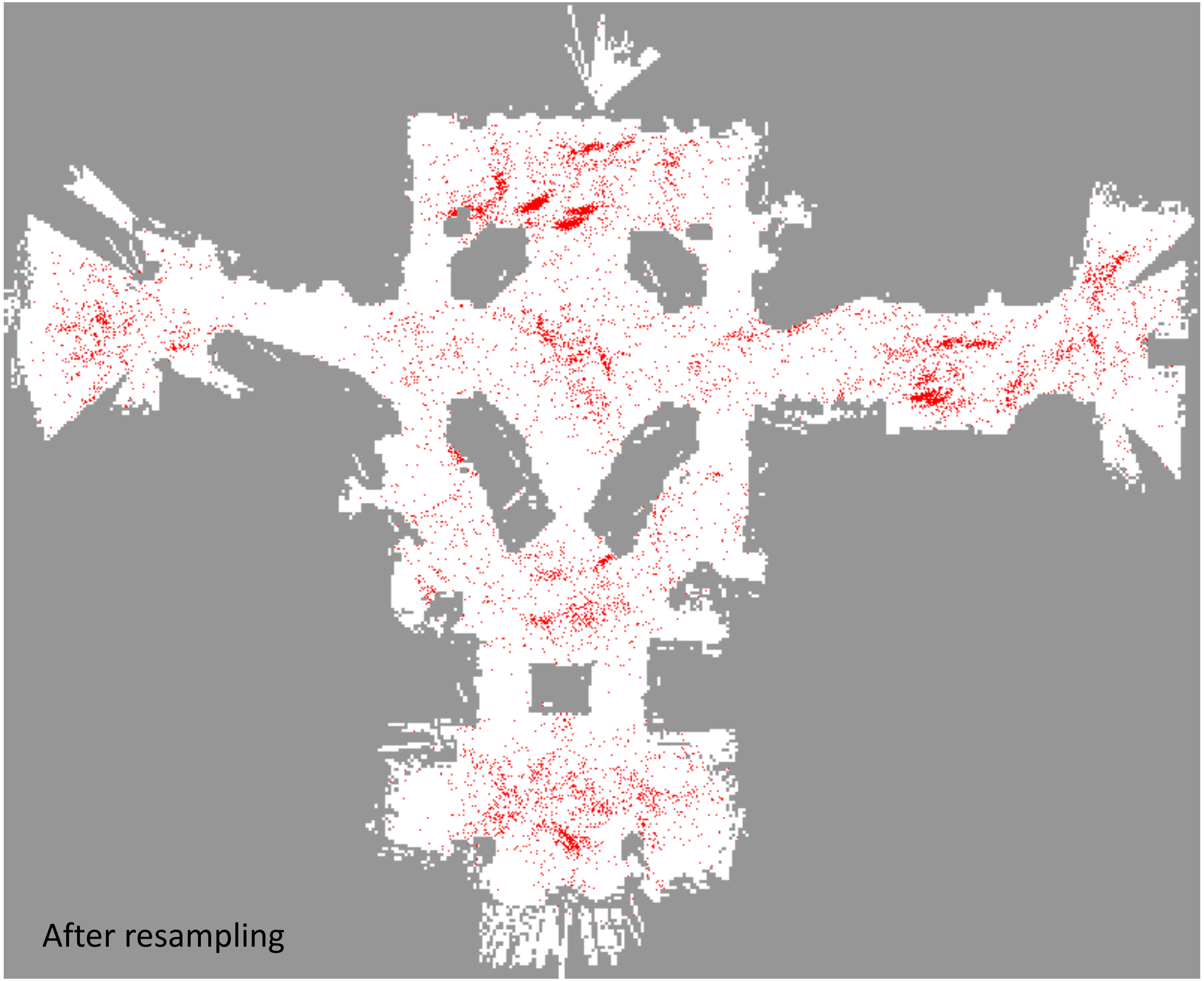


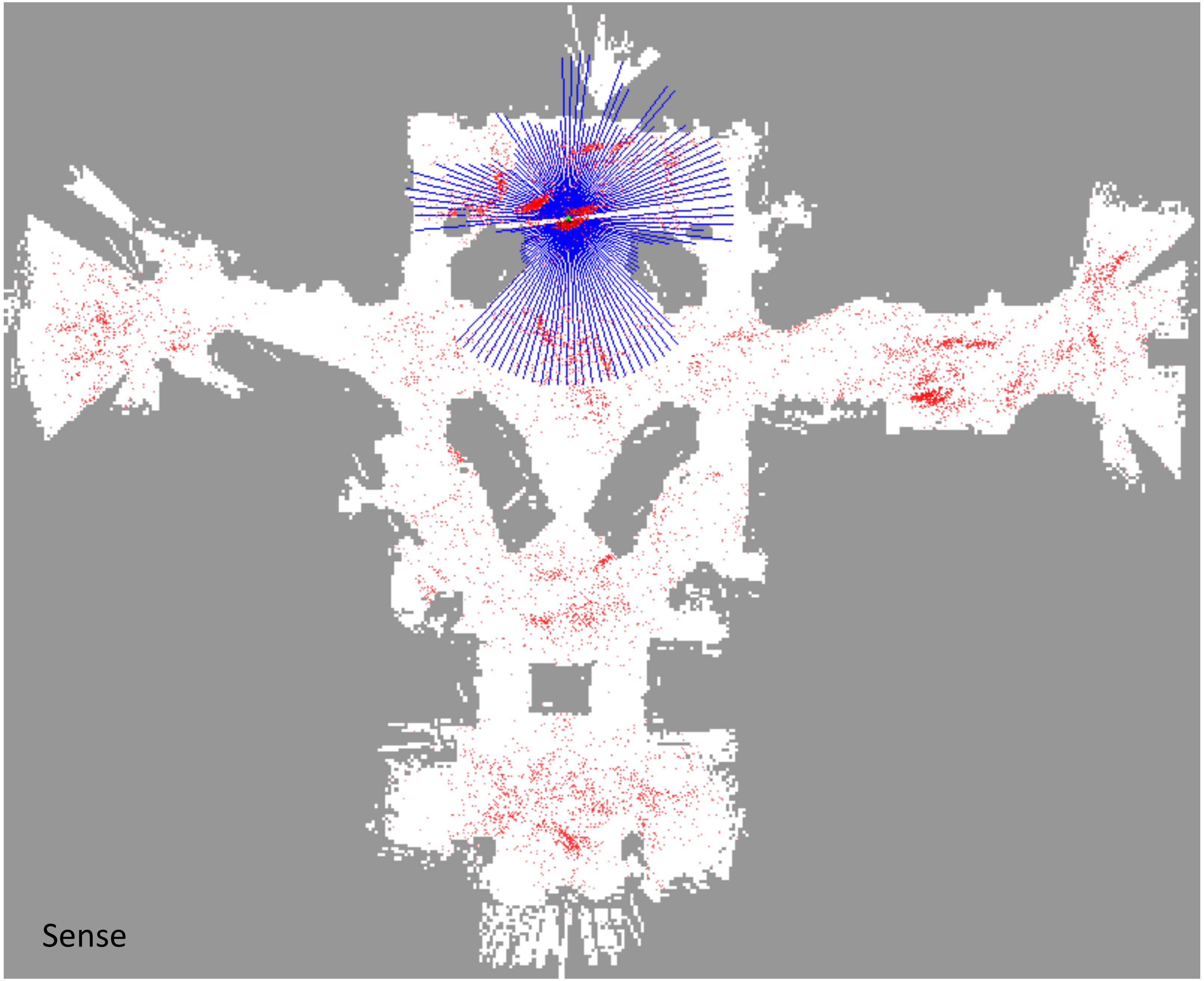
Monte Carlo Localization in NMAH (published CVPR'99)

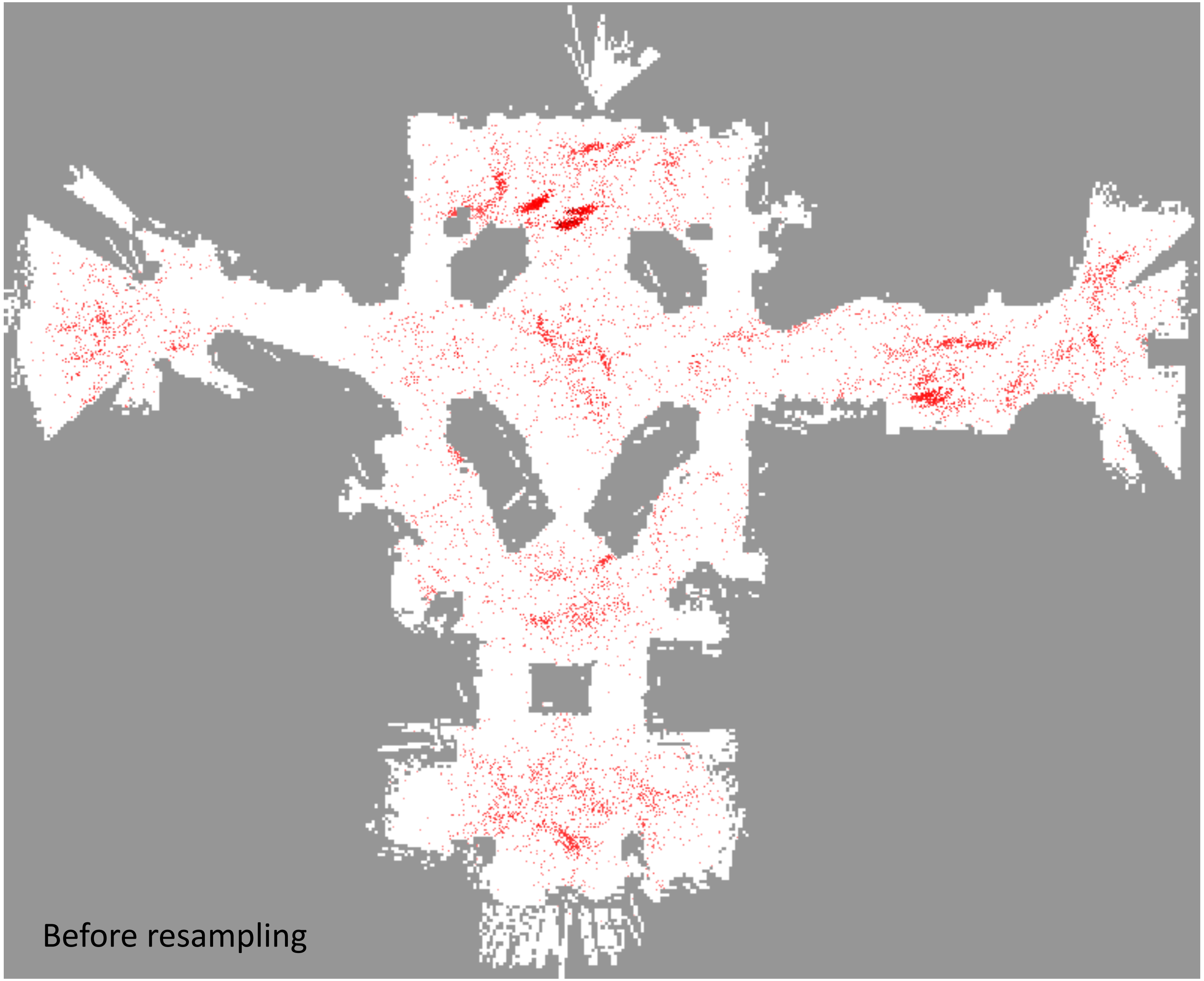

Uniform distribution

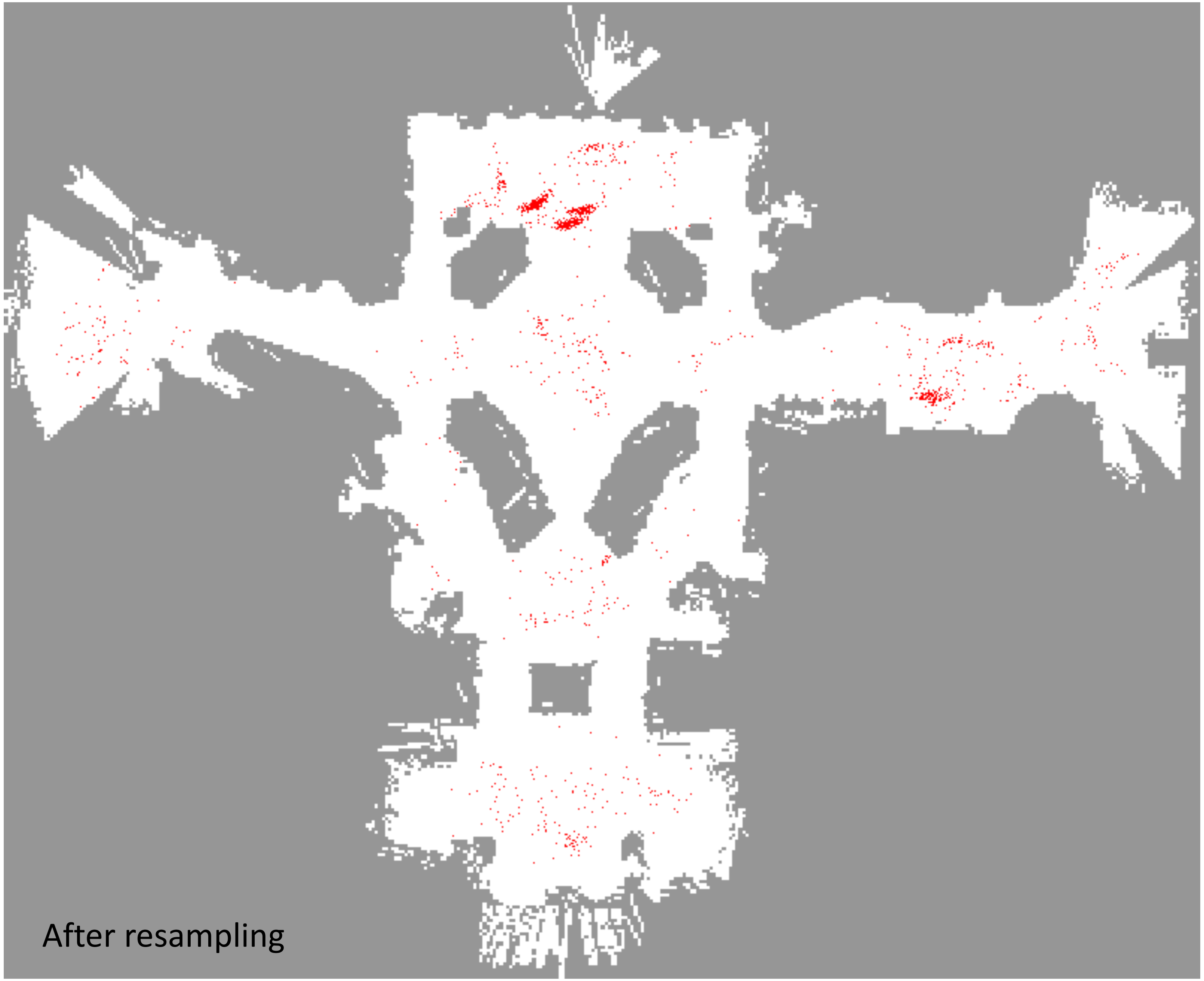

Sense


Before resampling

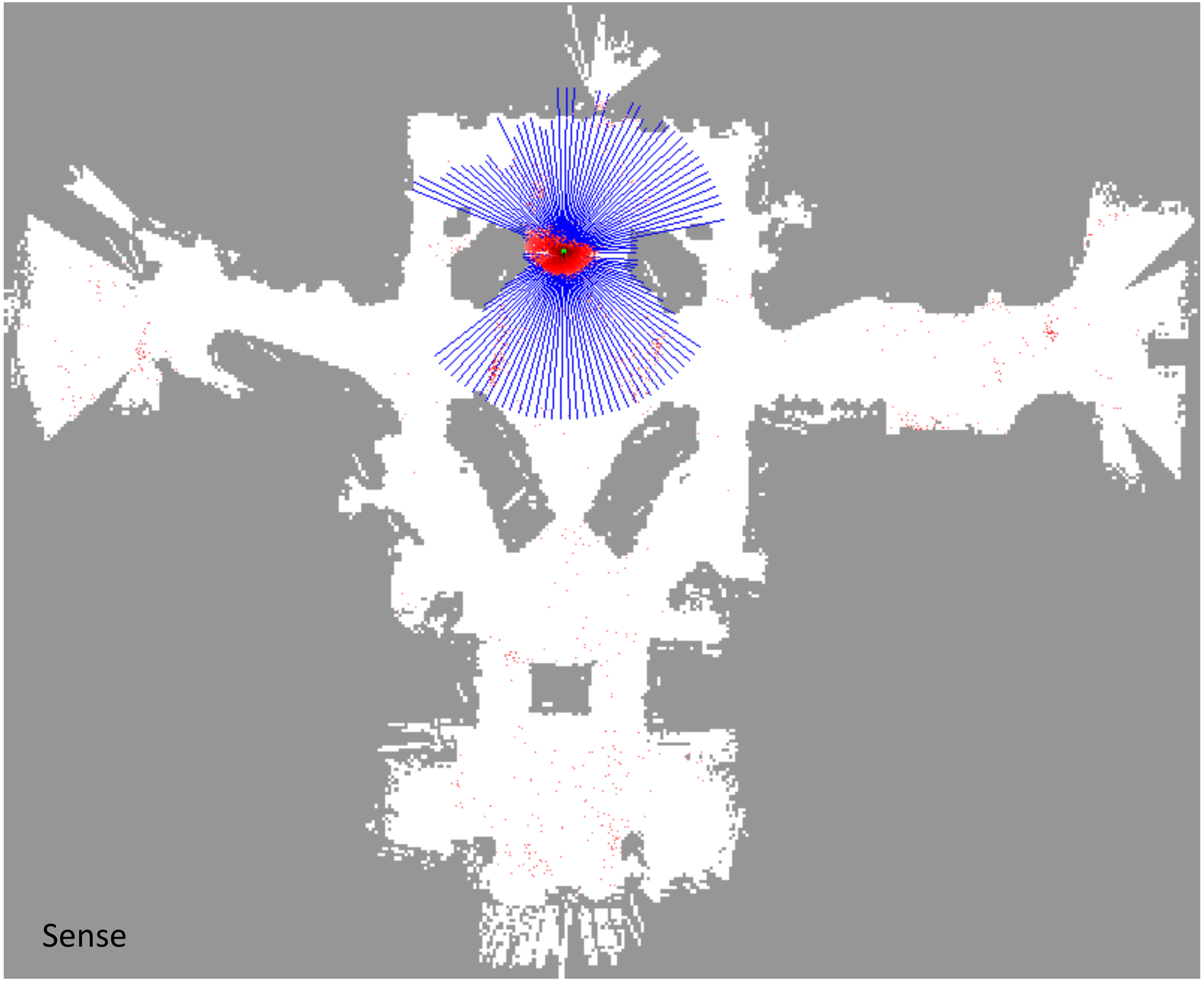

After resampling


Sense

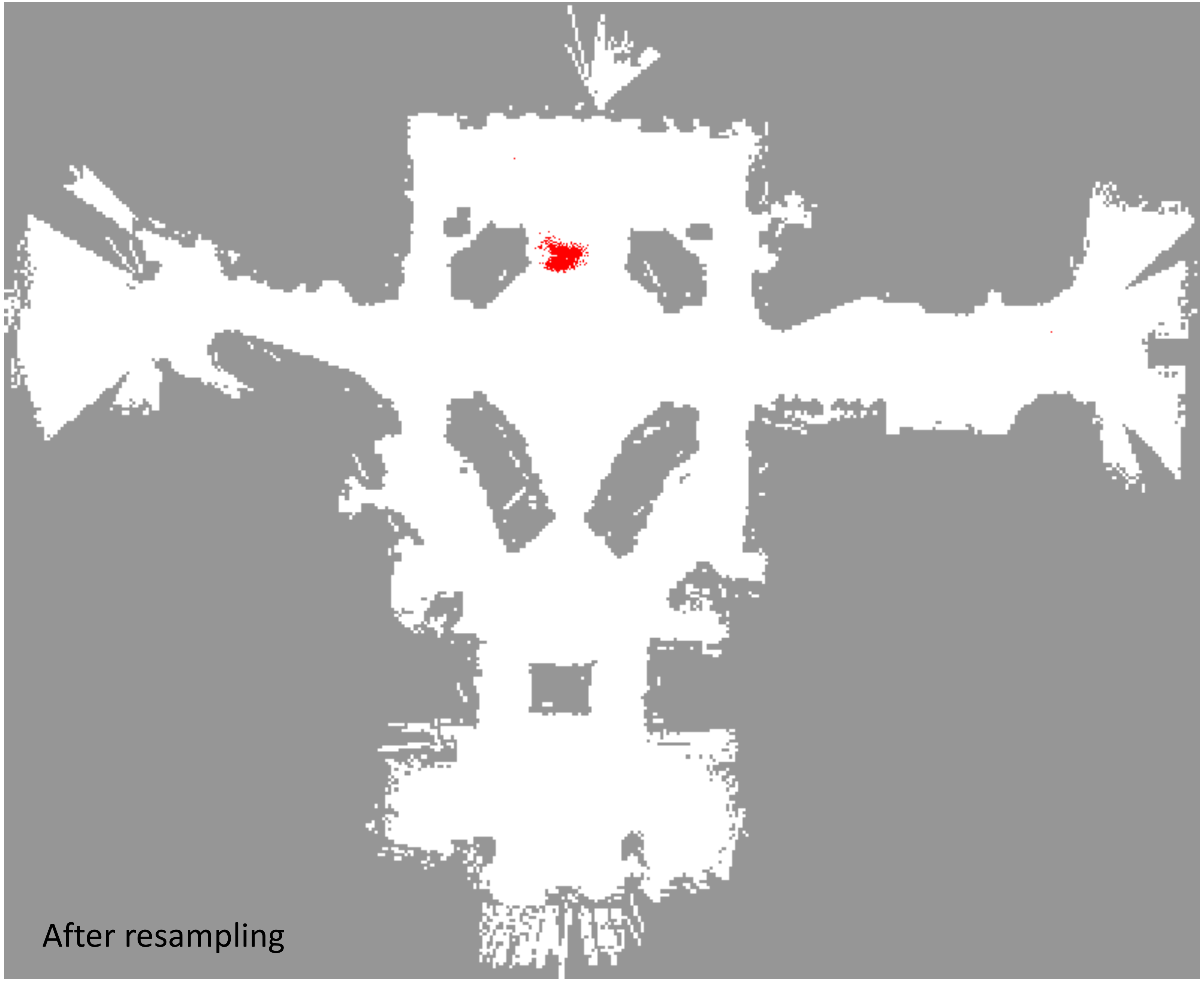

Before resampling

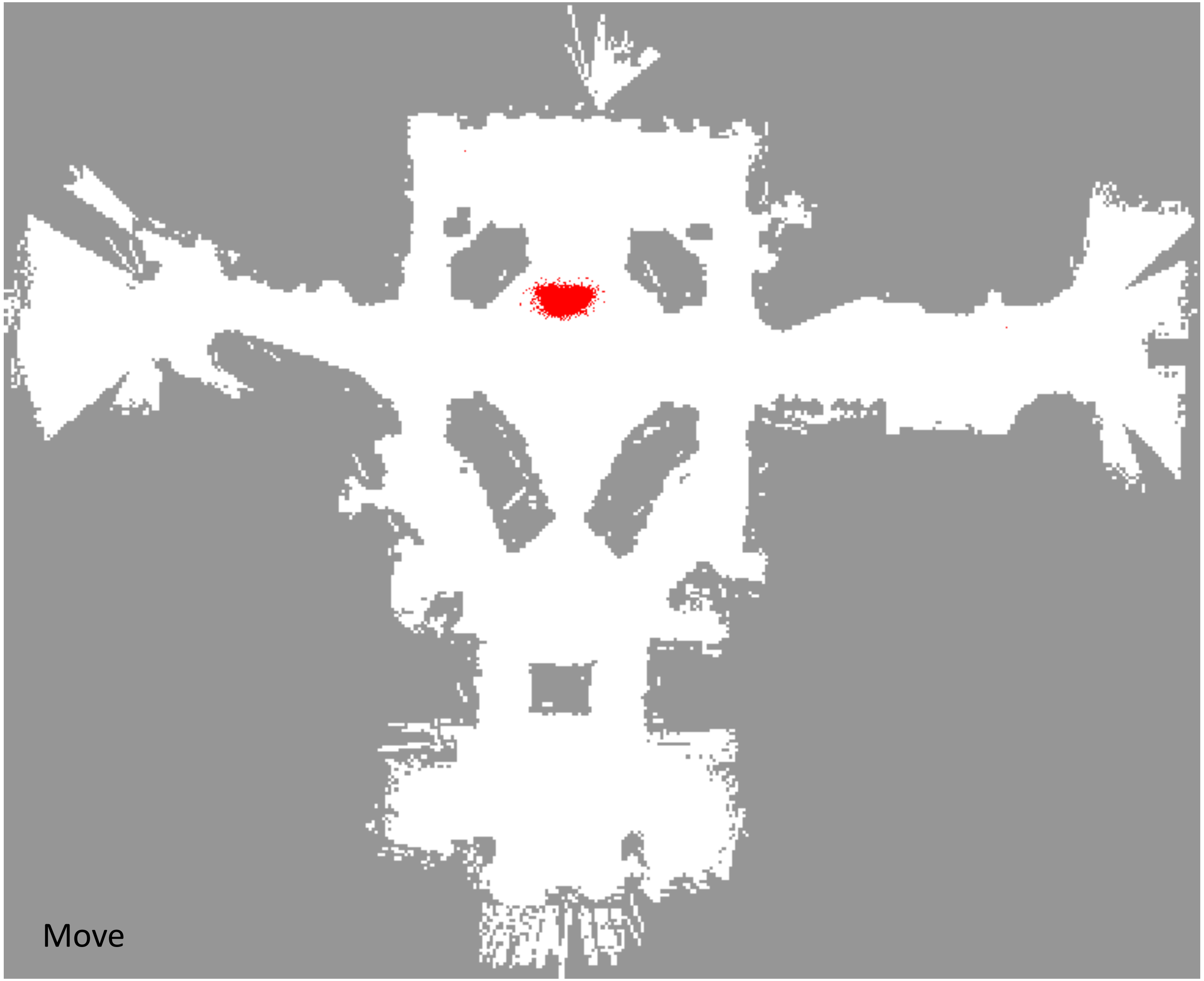

After resampling

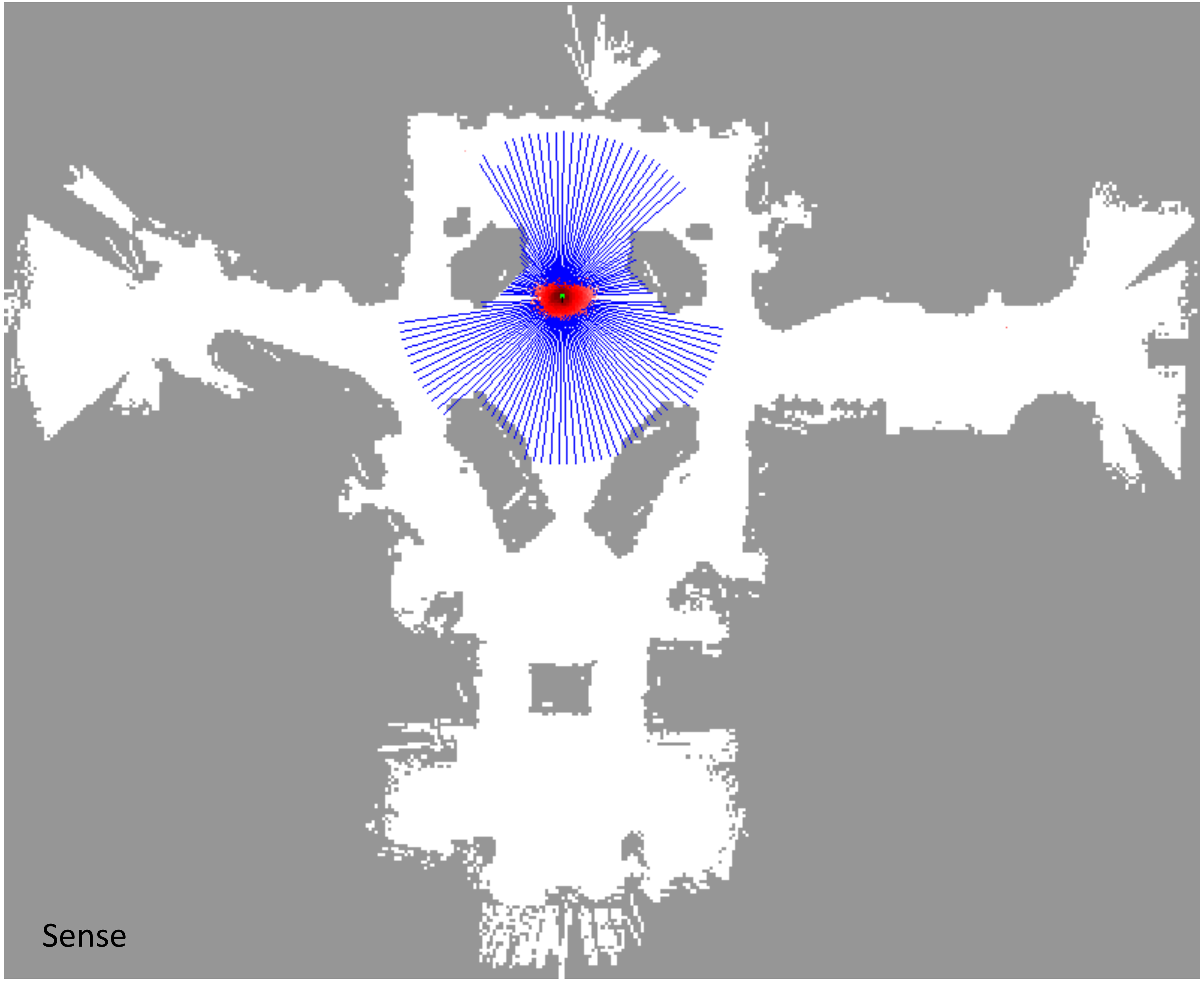

Move

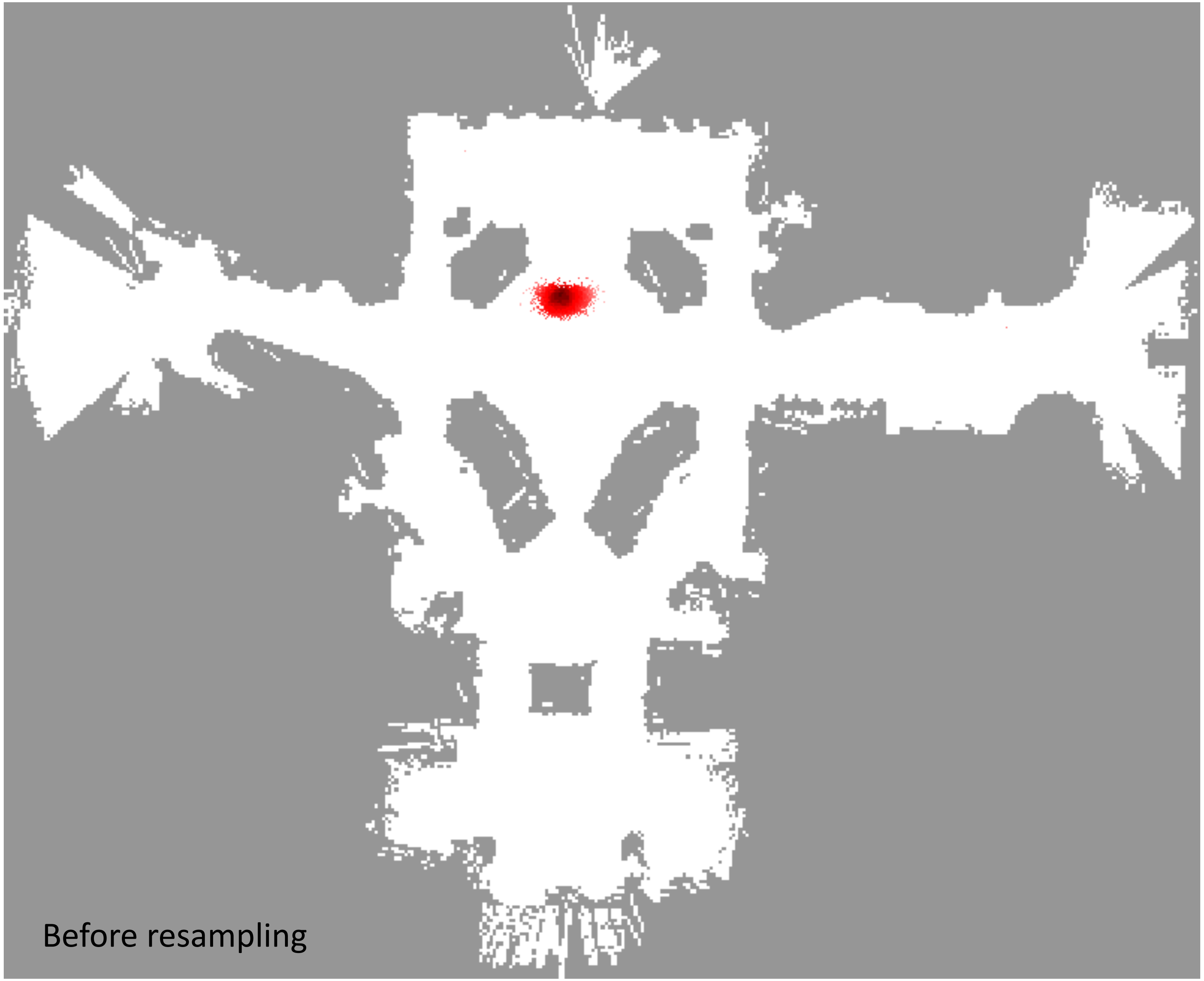

Sense

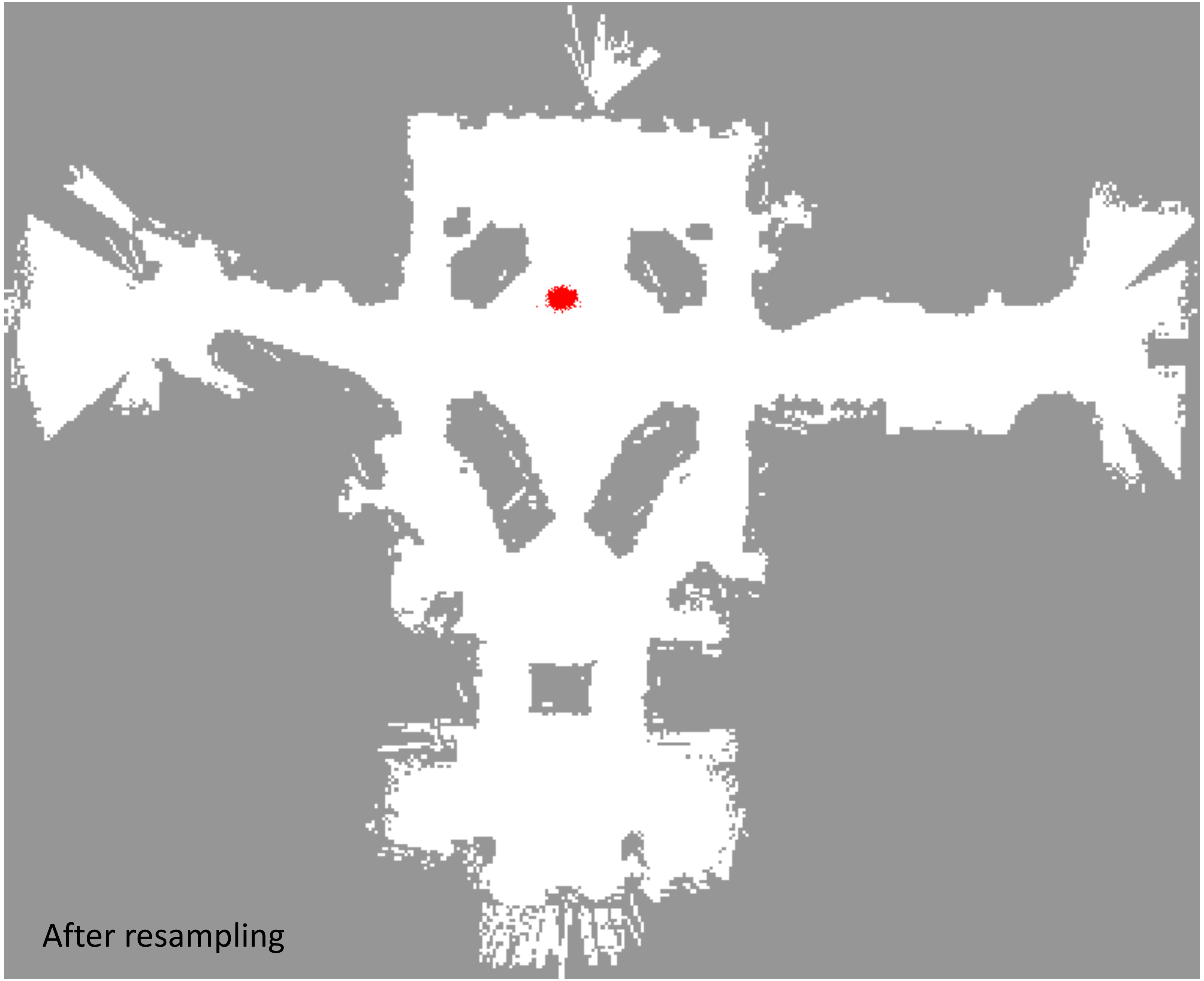

Before resampling

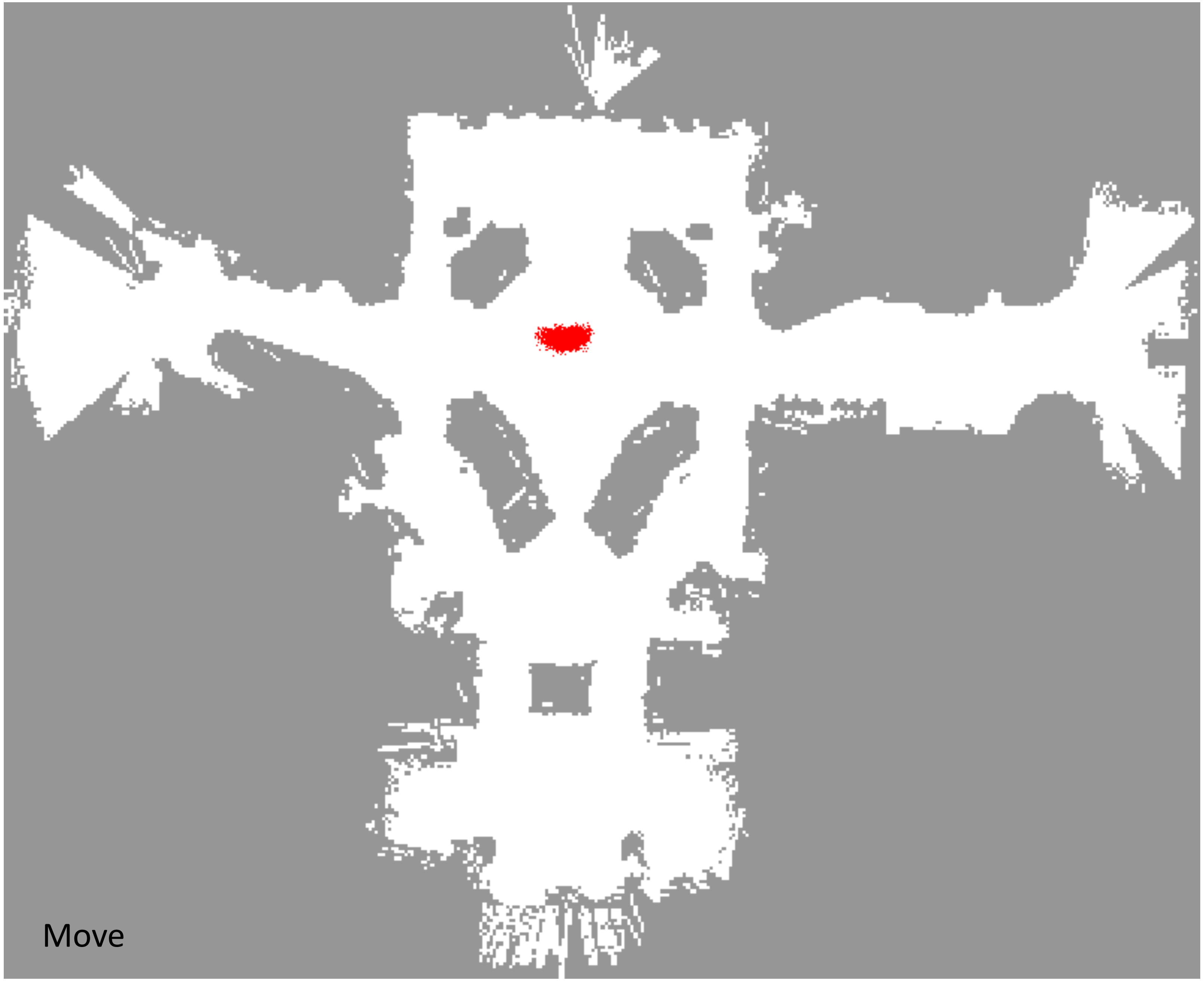

After resampling


Move


Sense


Before resampling


After resampling


Move

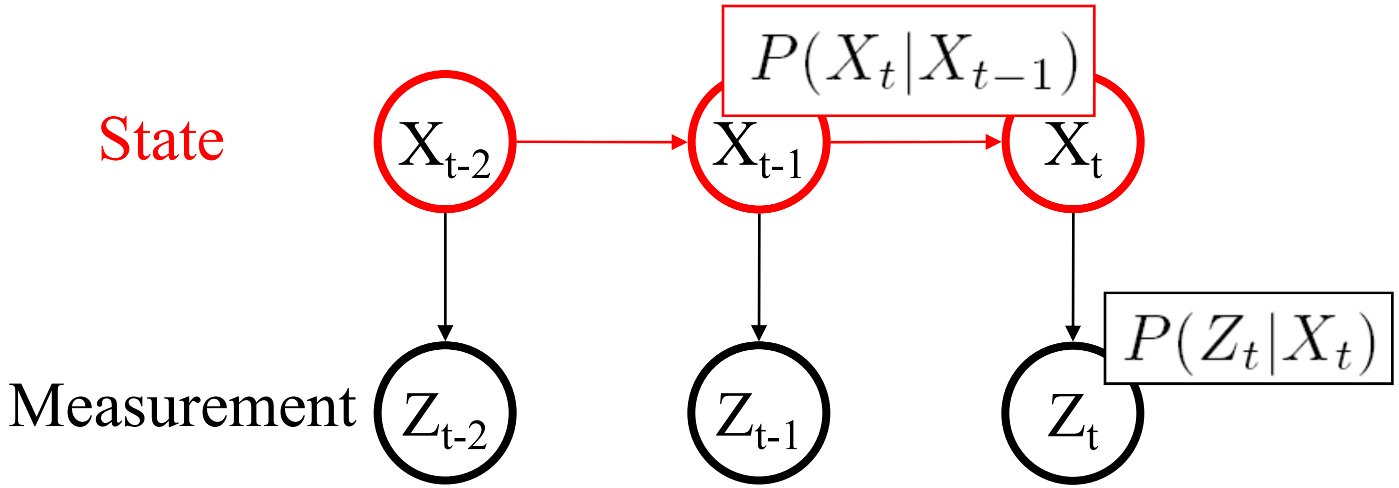
Summary

- Continuous models are for real robots
- Sampling can approximate densities
- Importance Sampling implements Bayes law
- Monte Carlo Localization uses simulation and resampling to implement a Bayes filter

- Bonus slides: you can interpret a particle filter as using a mixture model as predictive density

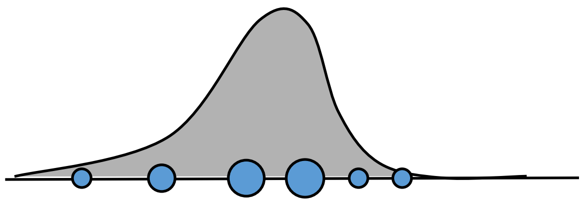
Bonus slides:
Particle filter prediction as a mixture density

Particle Filter Tracking



Monte Carlo Approximation of Filtering Density:

$$P(X_{t-1}|Z^{t-1}) \Longleftrightarrow \{X_{t-1}^{(i)}, \pi_{t-1}^{(i)}\}_{i=1}^{N}$$



Bayes Filter and Particle Filter

Motion Model

Recursive Bayes Filter Equation:

$$P(X_t|Z^t) = kP(Z_t|X_t)\int_{X_{t-1}} P(X_t|X_{t-1})P(X_{t-1}|Z^{t-1})$$

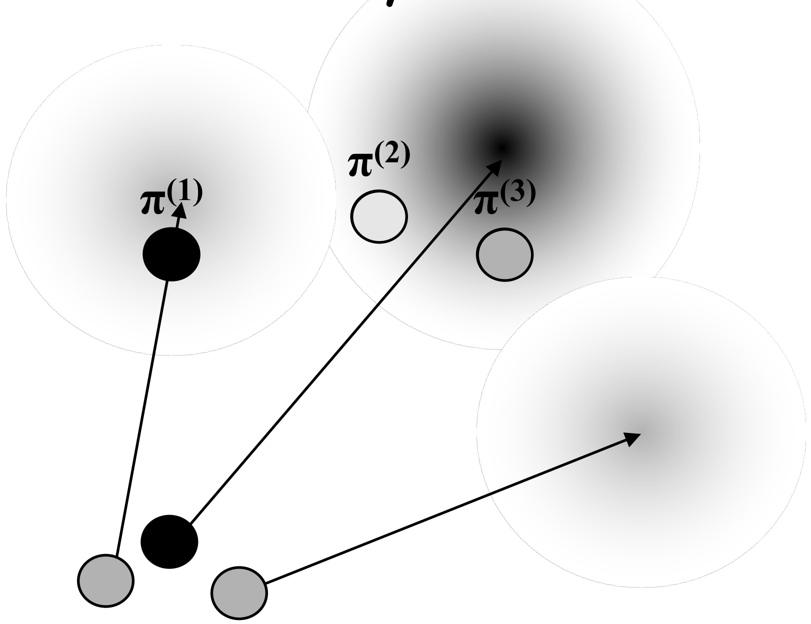
Predictive Density

Monte Carlo Approximation:

$$P(X_t|Z^t) \approx kP(Z_t|X_t)\sum_r \pi_{t-1}^{(r)} P(X_t|X_{t-1}^{(r)})$$

Particle Filter

Empirical predictive density = Mixture Model



$$\pi_t^{(s)} = P(Z_t | X_t^{(s)})$$