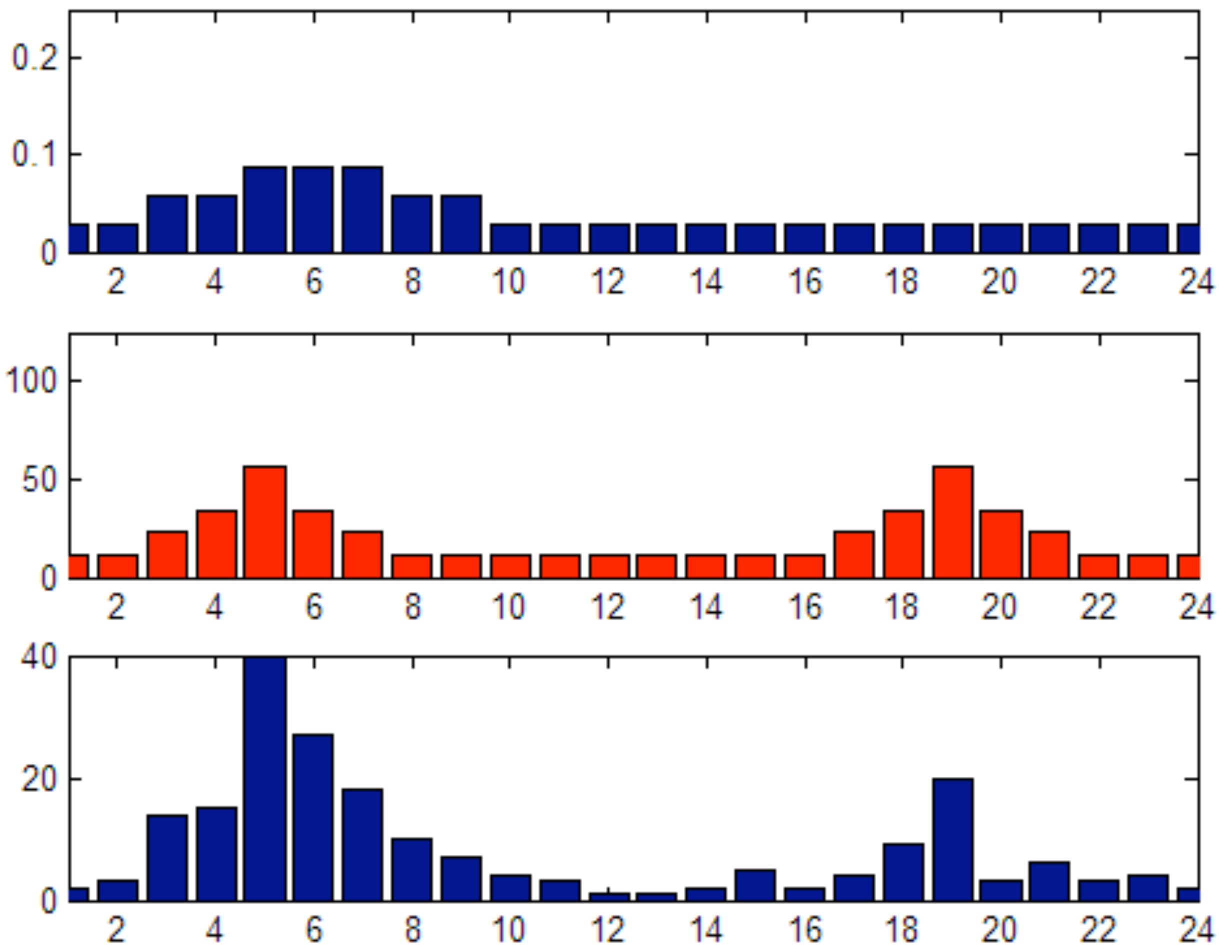
Particle Filters &
Monte Carlo Localization

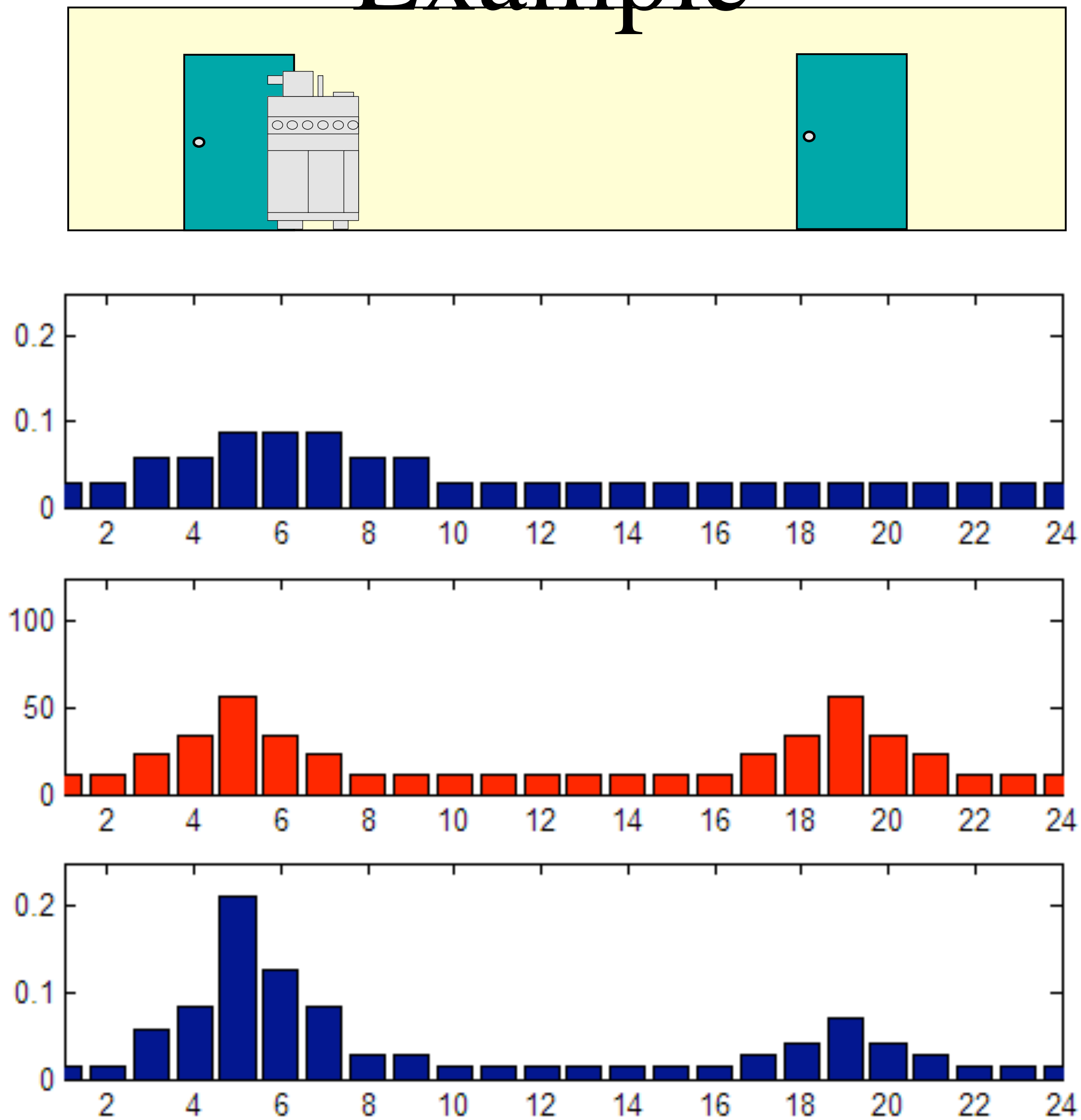
CS 3630 Intro to Perception and Robotics

Frank Dellaert

1D Importance Sampling



Monte Carlo Localization: a 1D Example

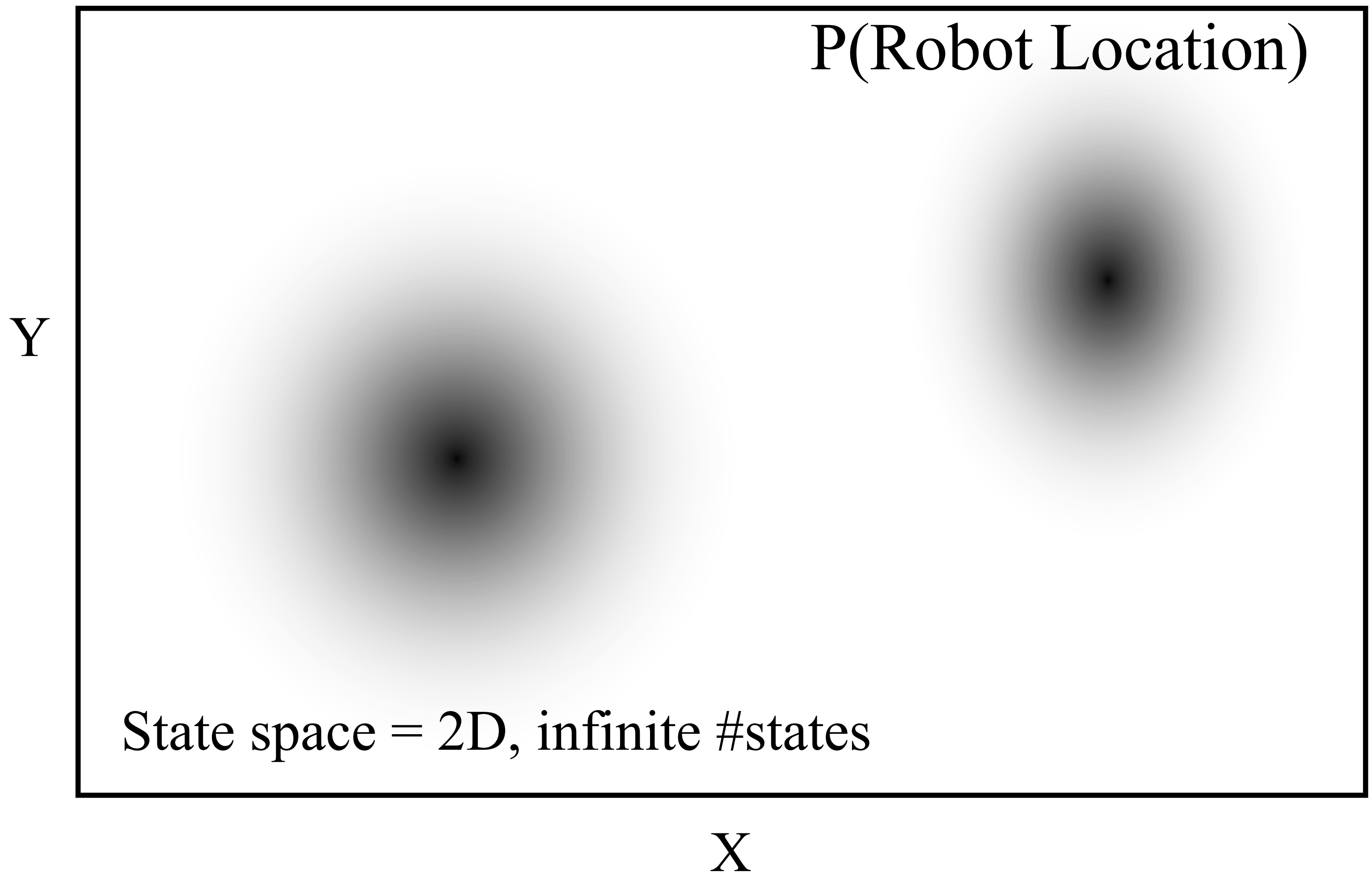


Prior P(X)

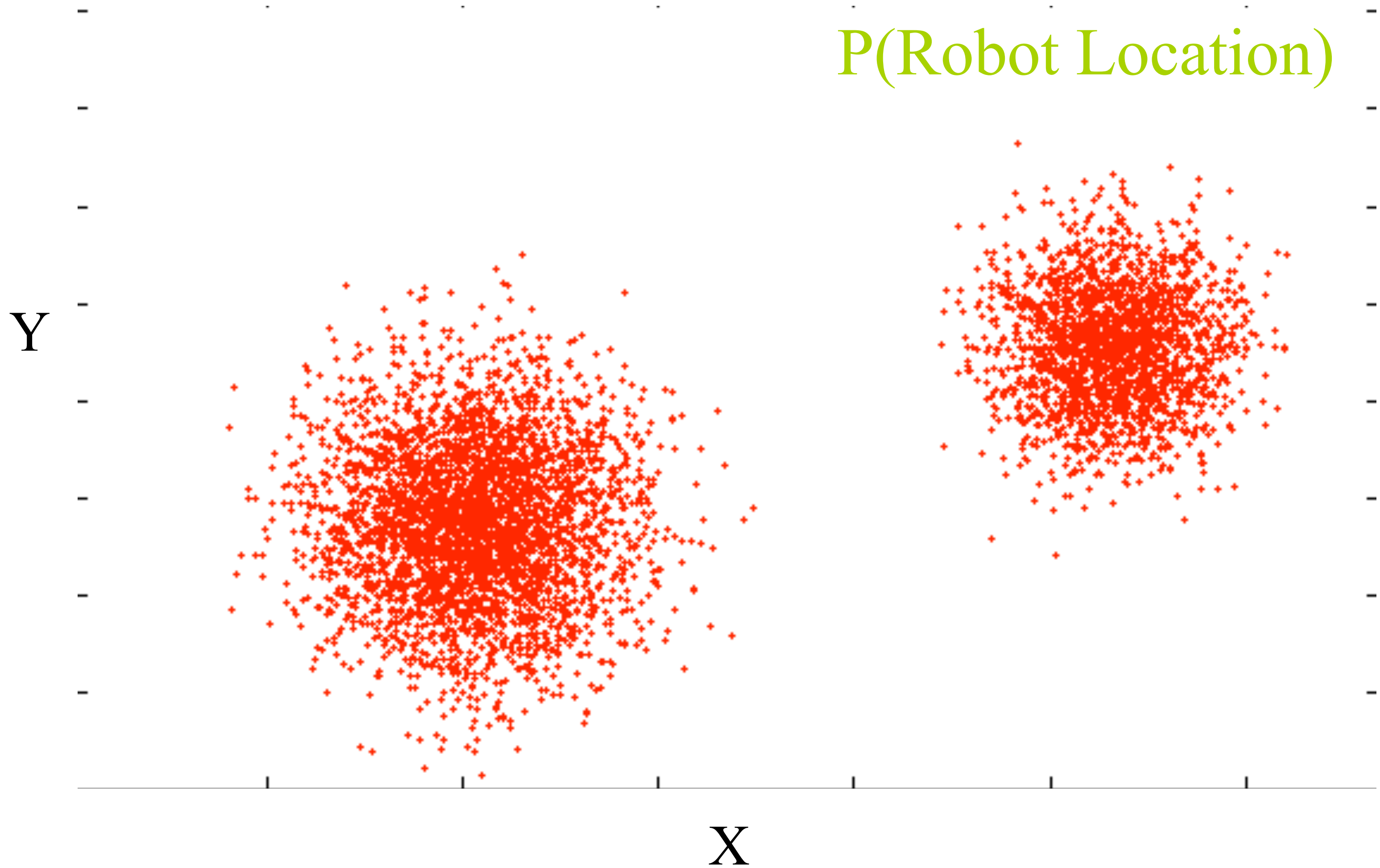
Likelihood L(X;Z)

Posterior P(X|Z)

Probability of Robot Location

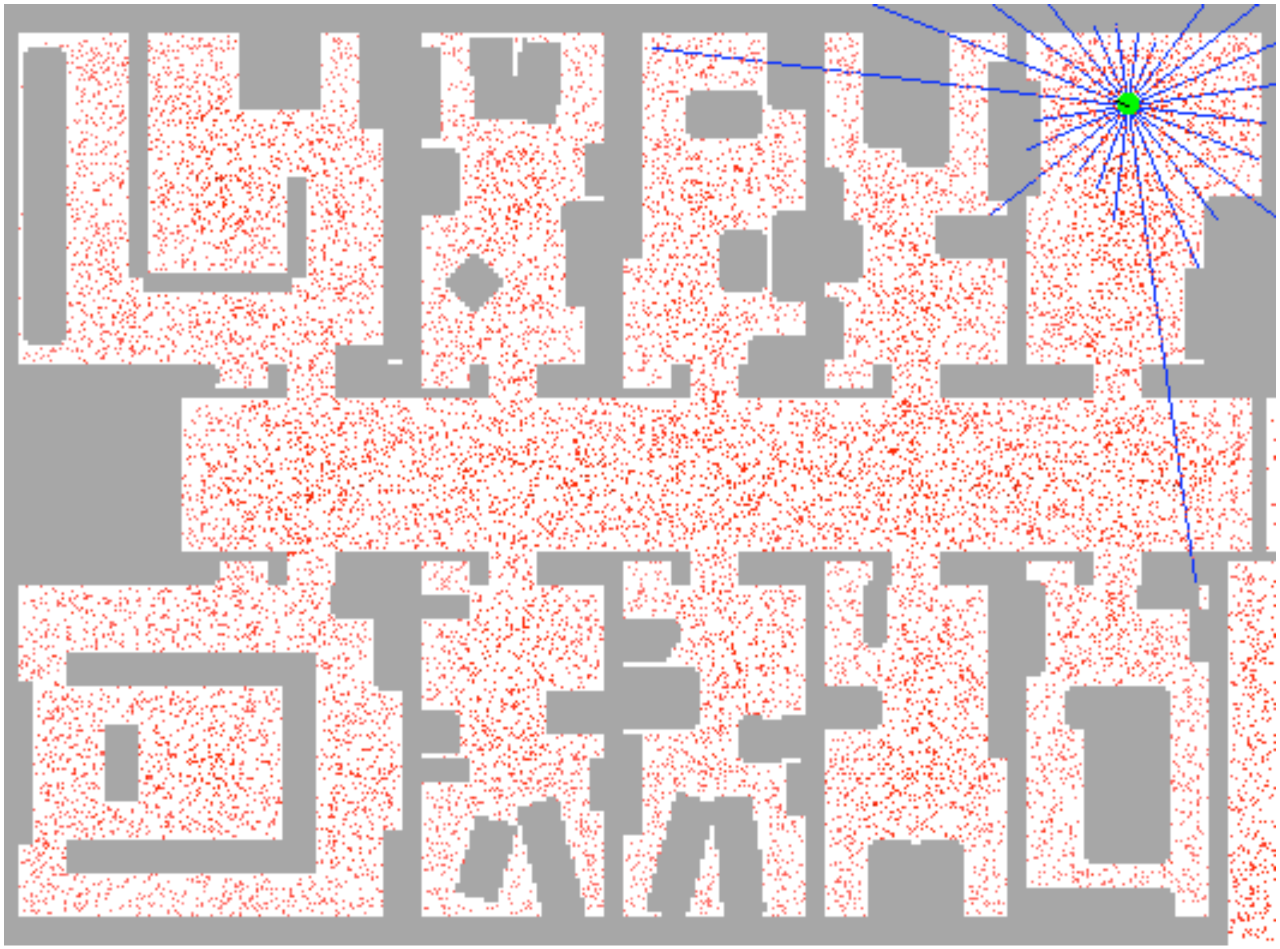


Sampling as Representation



3D Particle filter for robot pose: Monte Carlo Localization

Dellaert, Fox & Thrun ICRA 99



Sampling Advantages

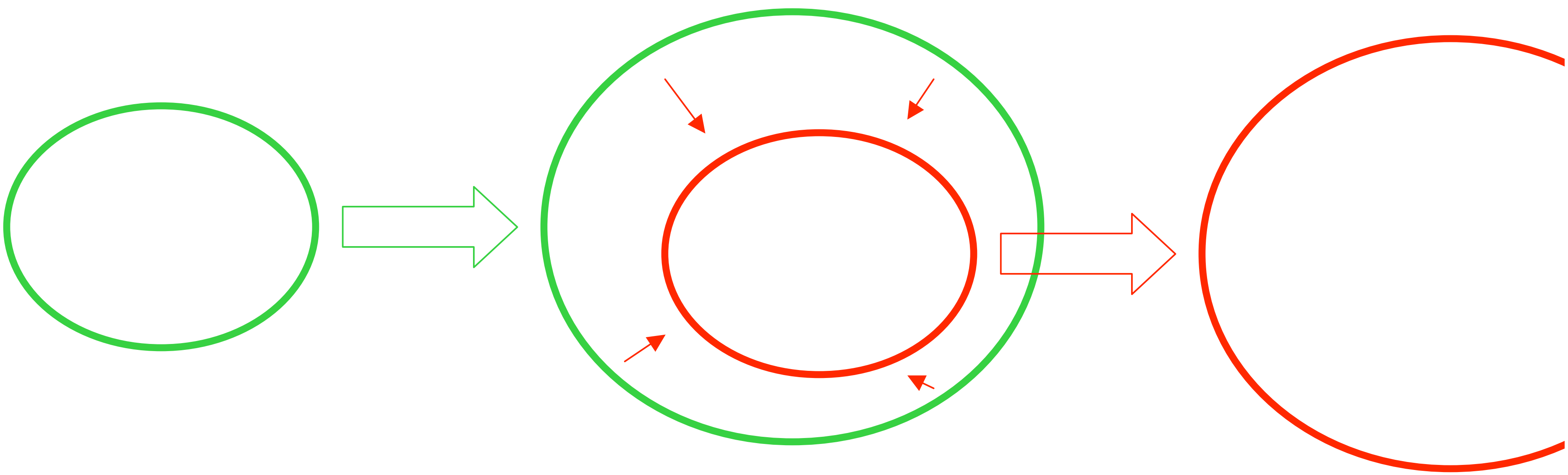
- Arbitrary densities
- Memory = O(#samples)
- Only in “Typical Set”
- Great visualization tool !

- minus: Approximate

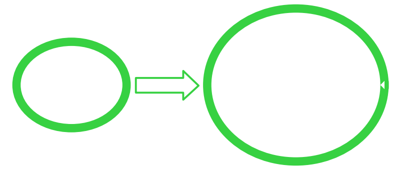
First appeared in 70’s, re-discovered by Kitagawa,
Isard & Blake in computer vision,
Monte Carlo Localization in robotics

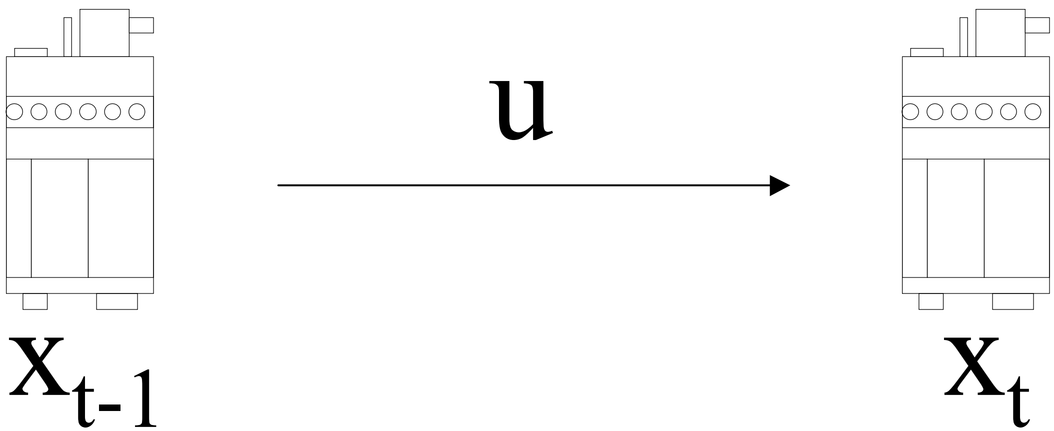
Bayesian Filtering

- Two phases: 1. Prediction Phase
  2. Measurement Phase



1. Prediction Phase

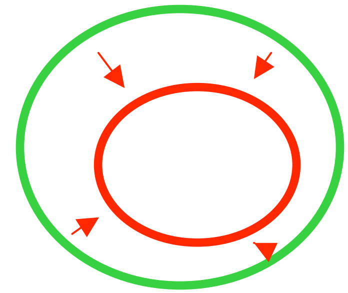


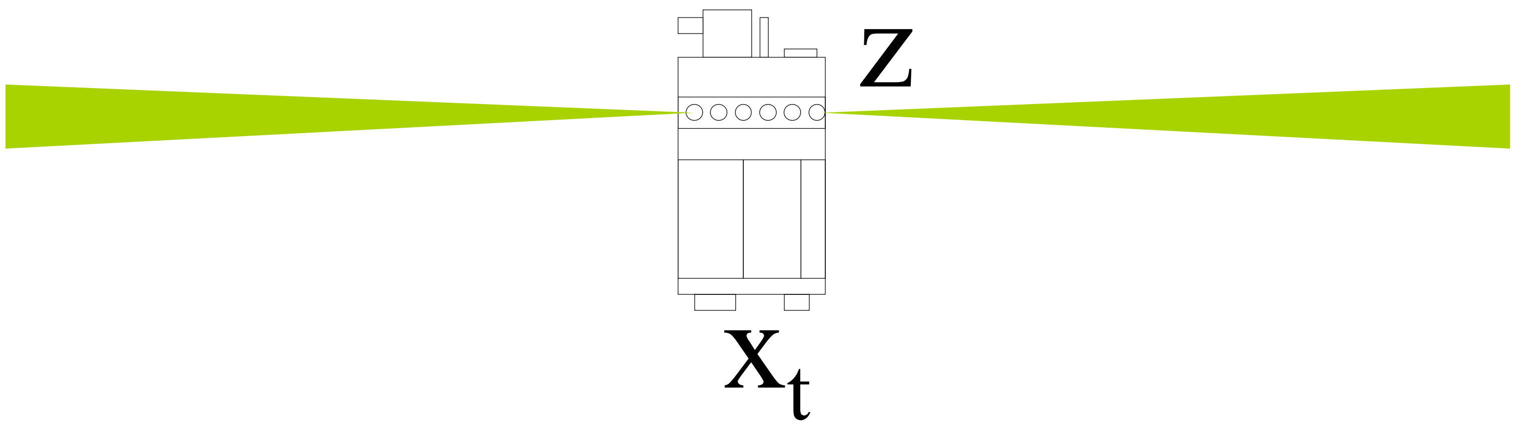


$$P(x_t) = \sum P(x_t|x_{t-1},u)\ P(x_{t-1})$$

Motion Model

2. Measurement Phase

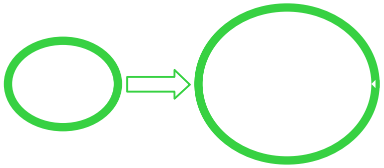


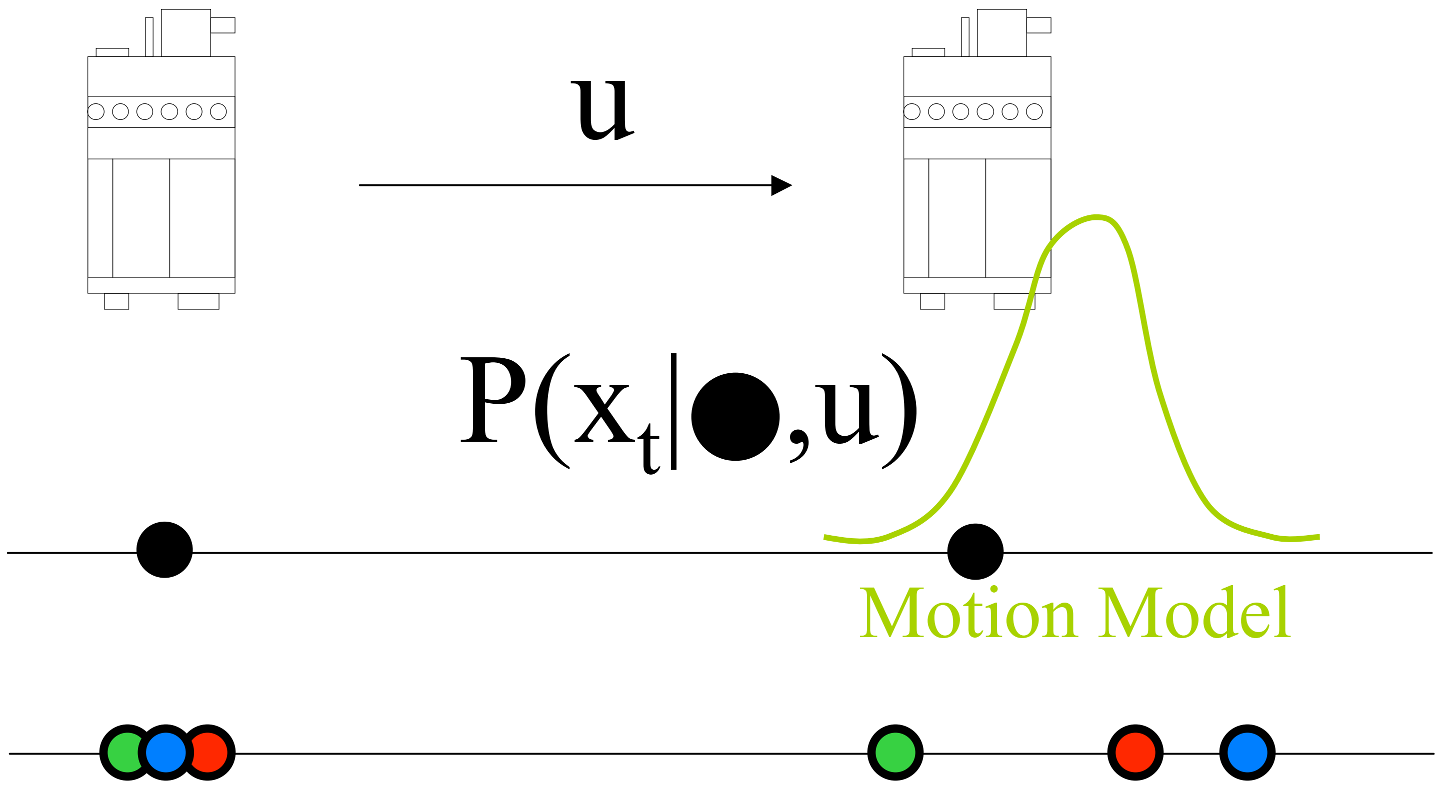


$$P(x_t|z) = k\, P(z|x_t)\, P(x_t)$$

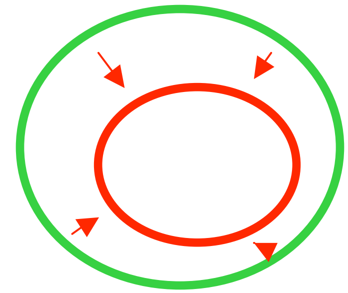
Sensor Model

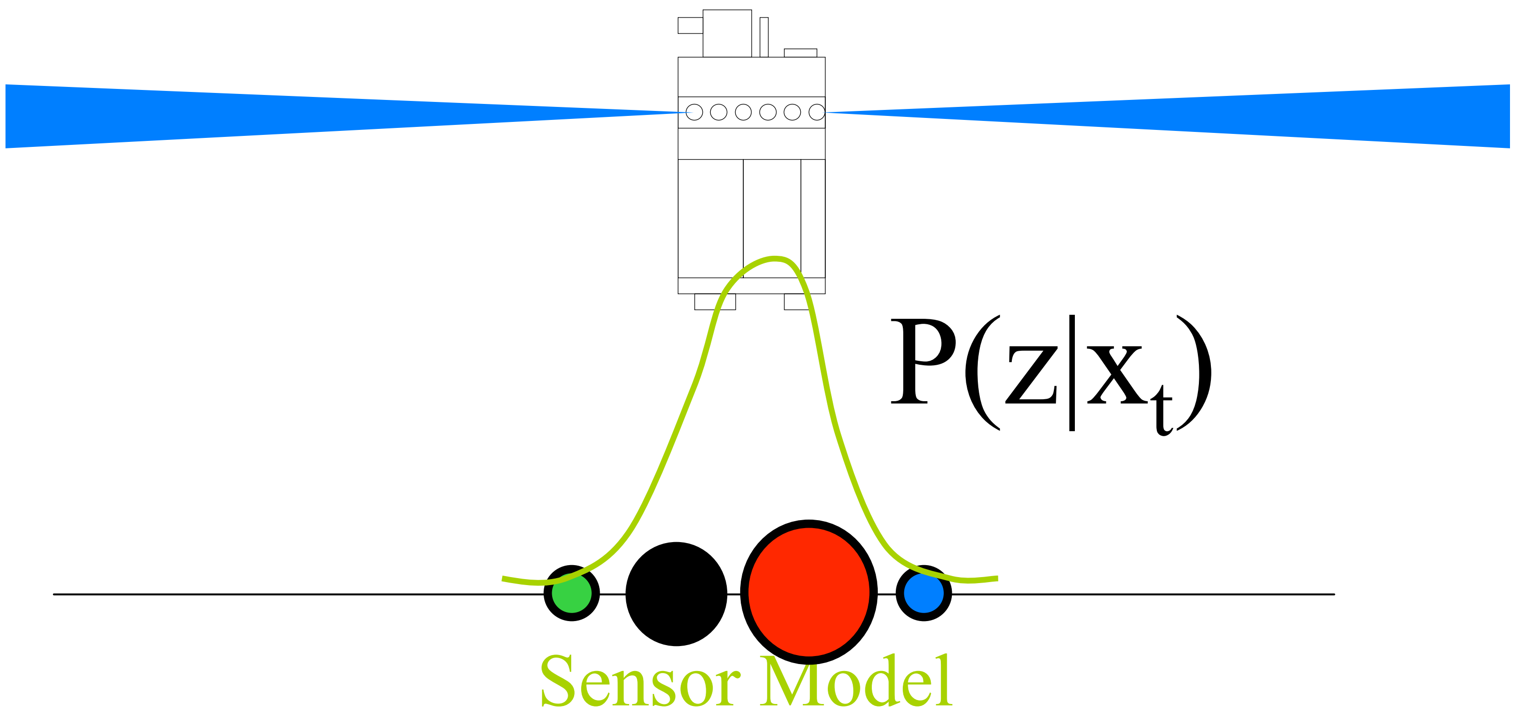
1. Prediction Phase



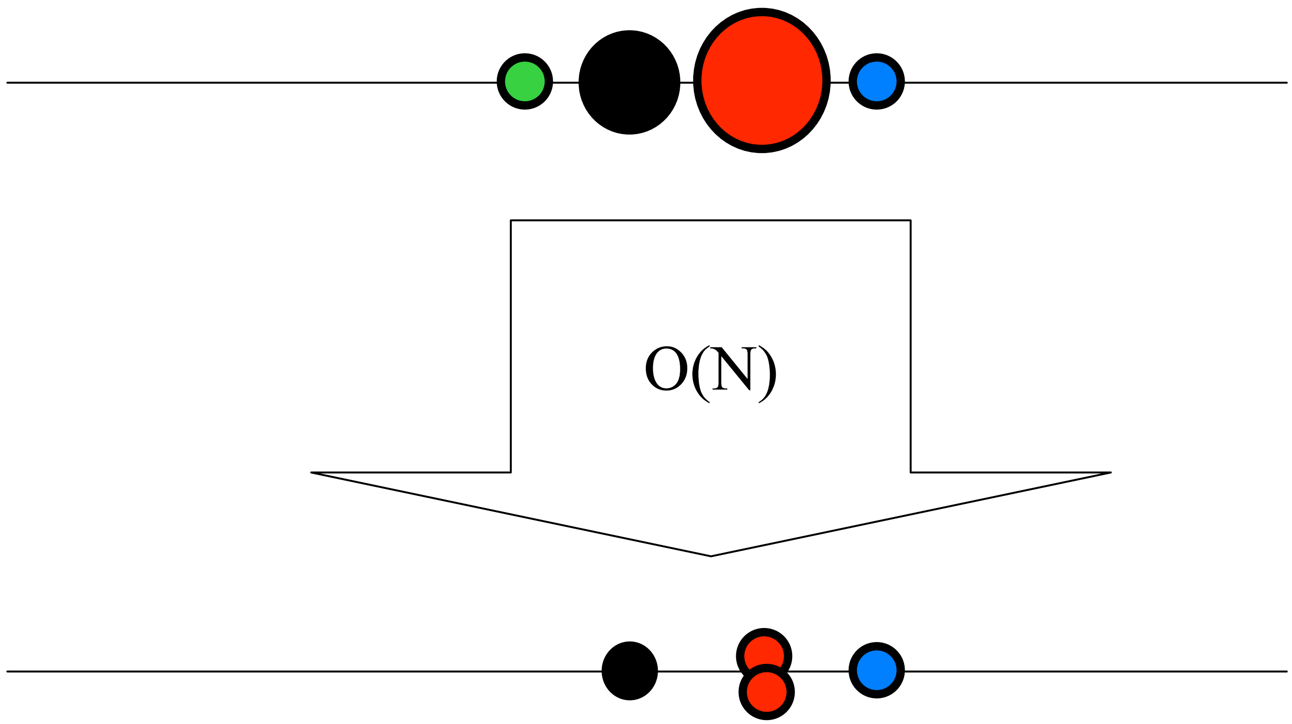


2. Measurement Phase

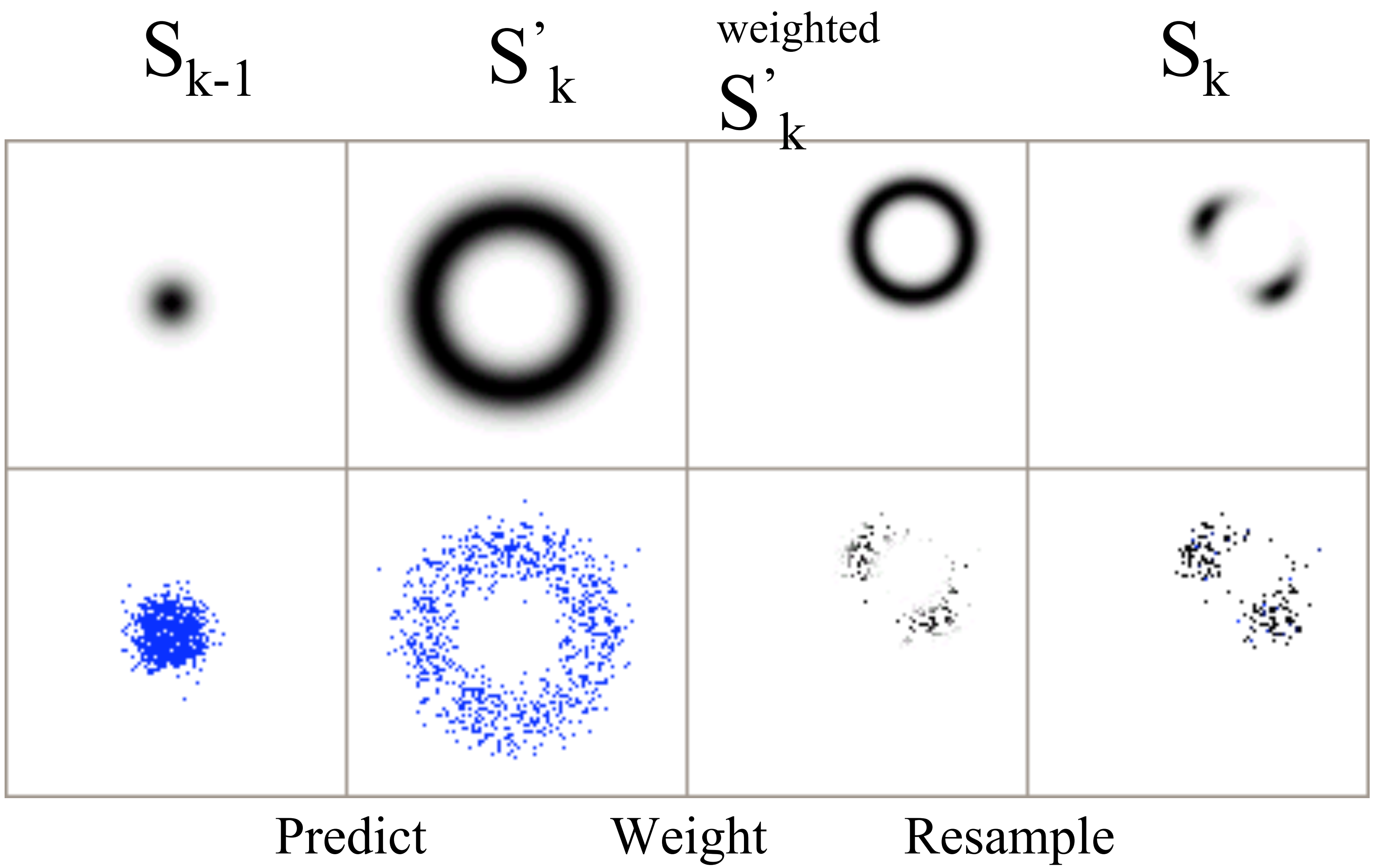




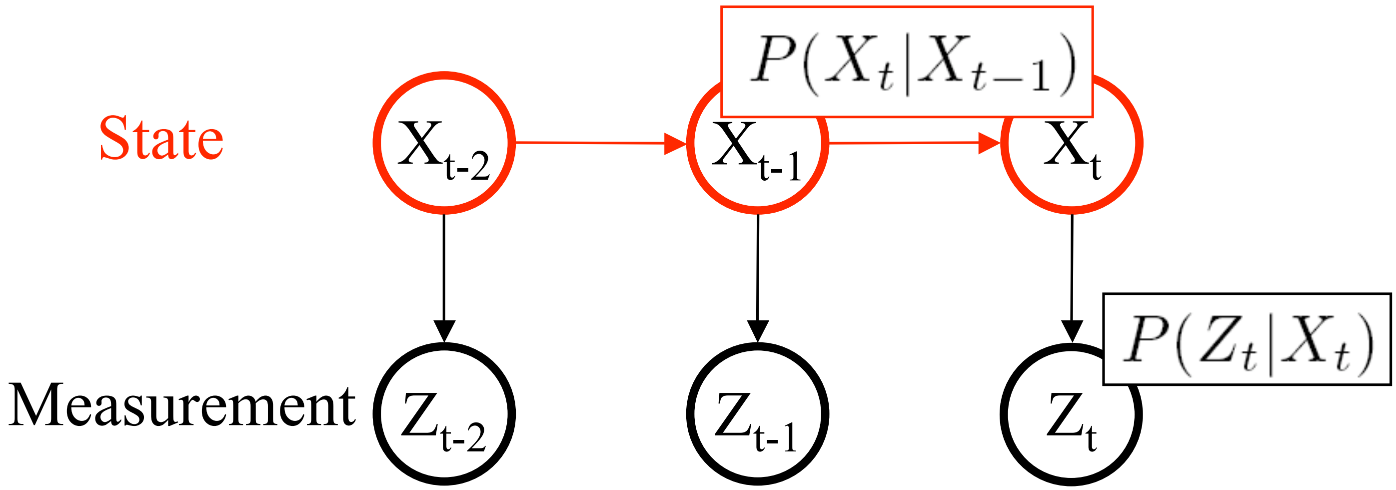
3. Resampling Step



Monte Carlo Localization

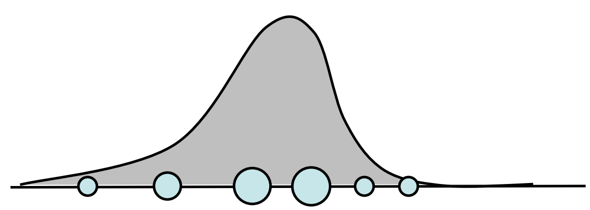


Particle Filter Tracking



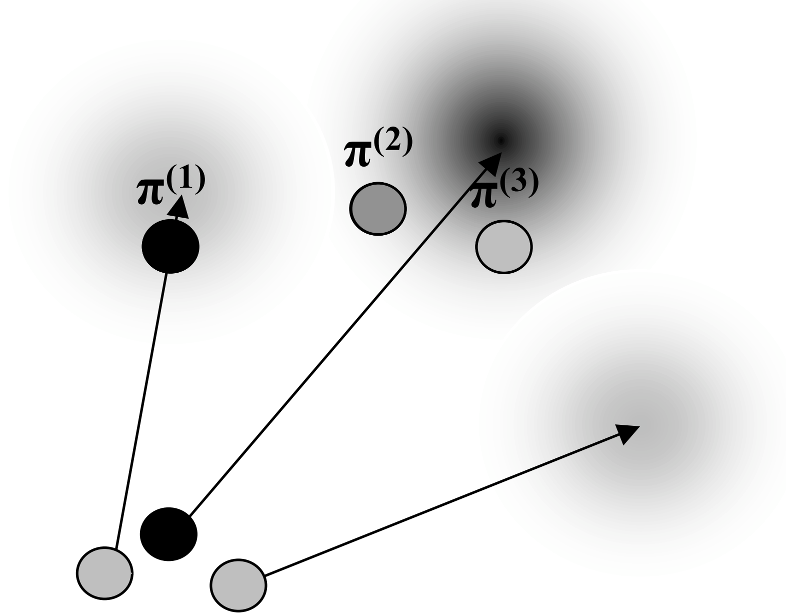
Monte Carlo Approximation of Posterior:

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



A Two-step View of the Particle Filter

Empirical predictive density = Mixture Model



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

Bayes Filter and Particle Filter

Motion Model

Recursive Bayes Filter Equation:

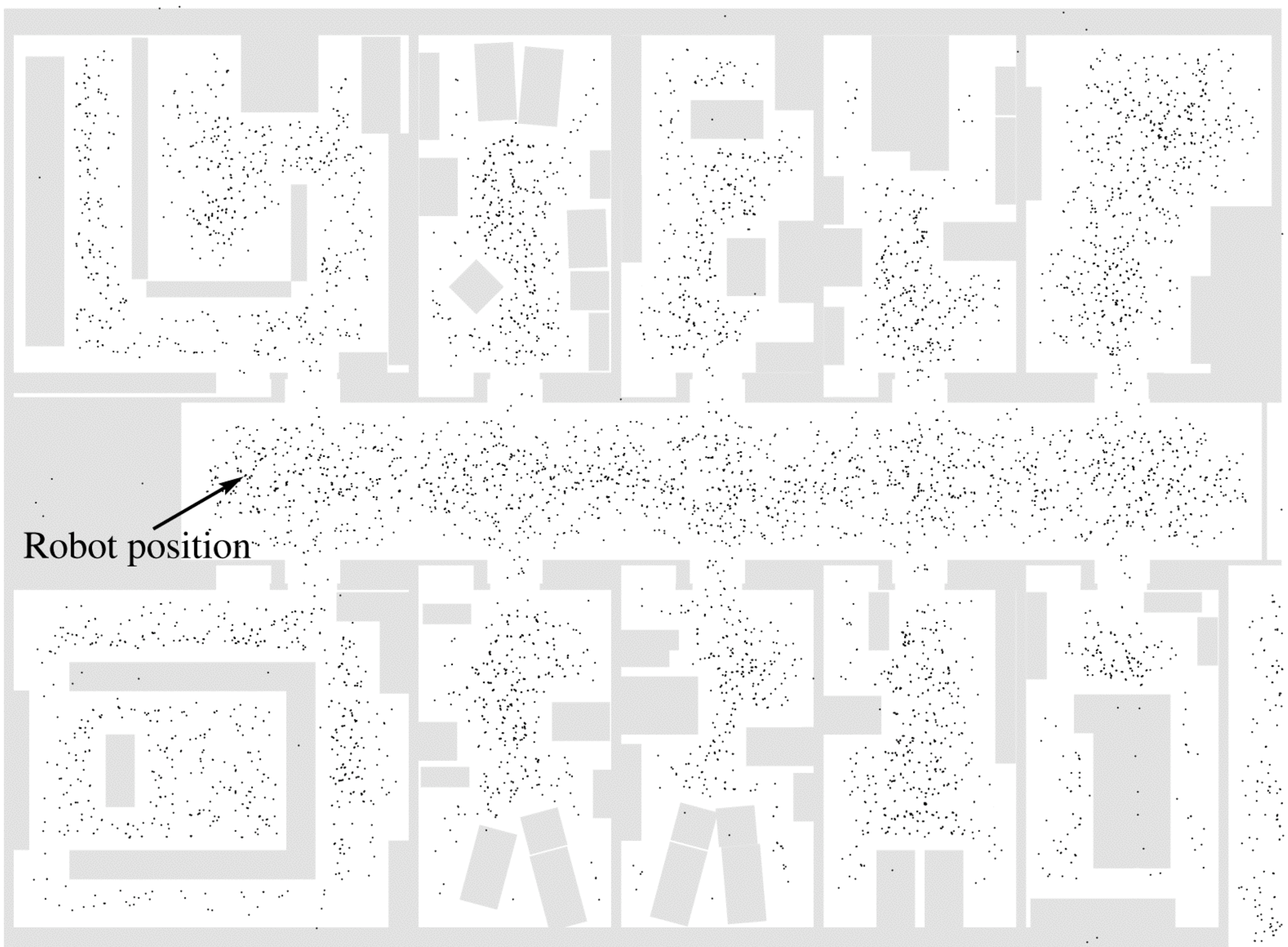
$$P(X_t|Z^t) = kP(Z_t|X_t)\int_{X_{t-1}} P(X_t|X_{t-1})P(X_{t-1}|Z^{t-1})$$

Predictive Density

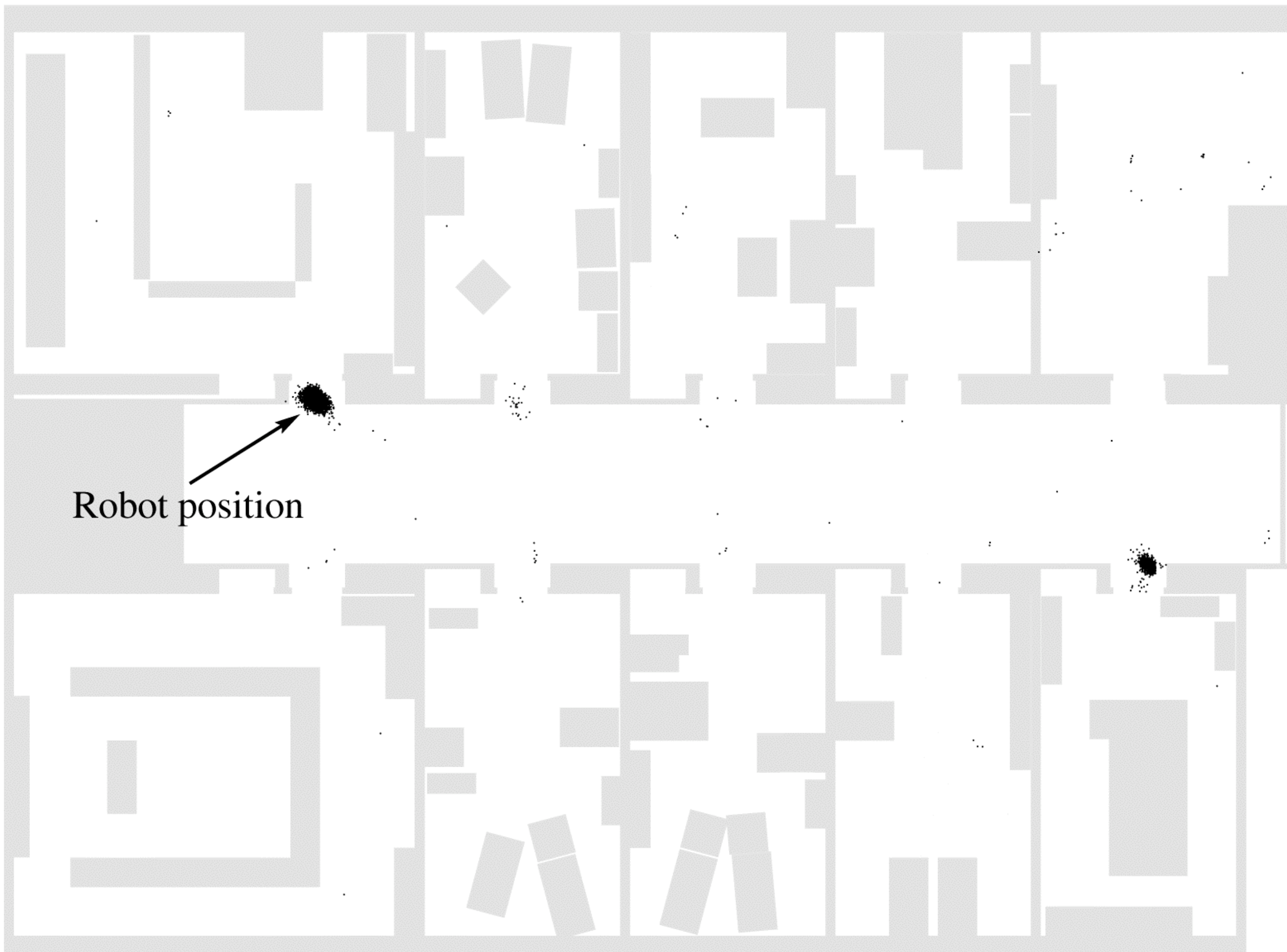
Monte Carlo Approximation:

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

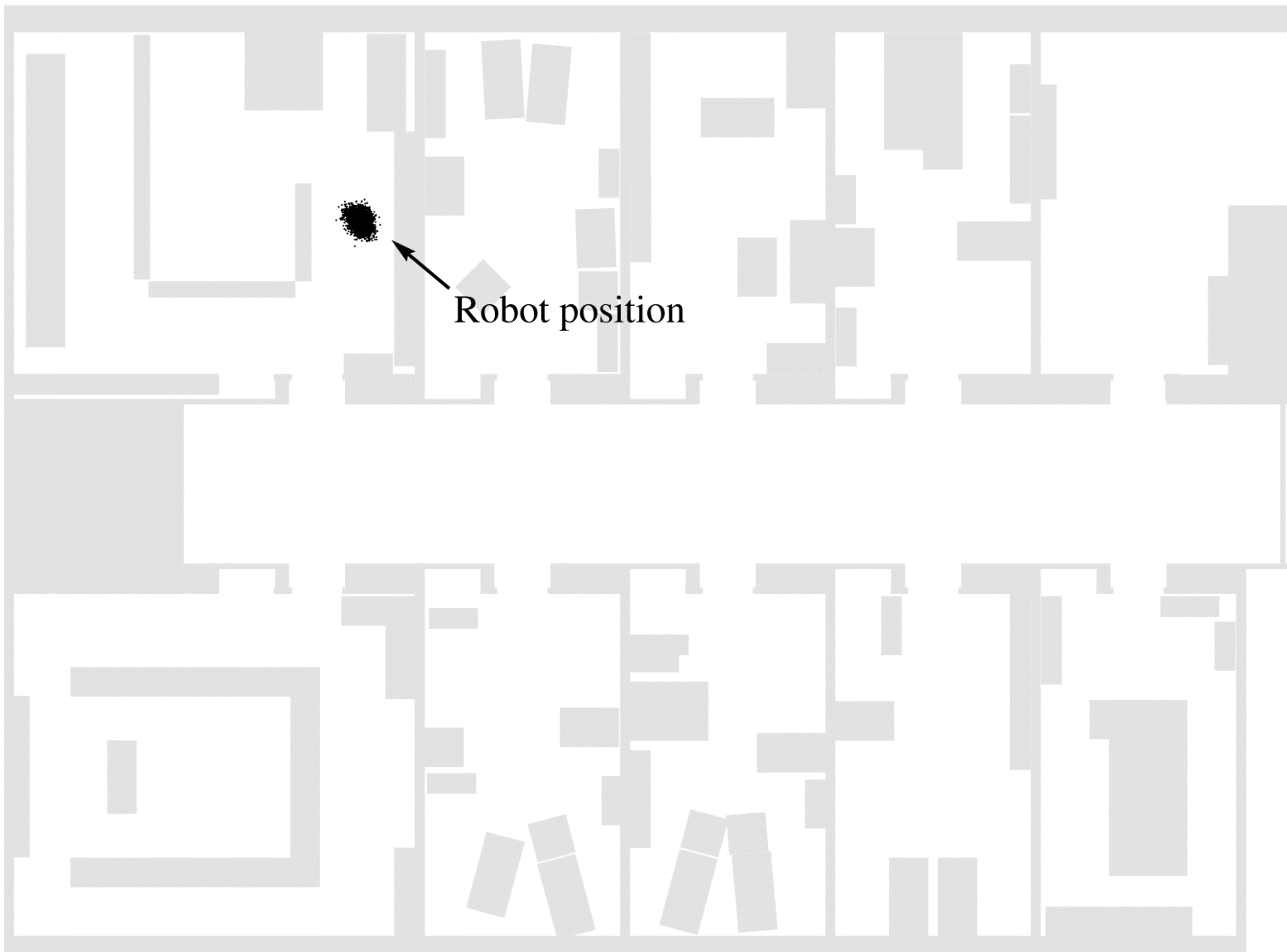
Global Localization



Global Localization (2)



Global Localization (3)



Conclusions

- Monte Carlo Localization:
  Powerful yet efficient
  Significantly less memory and CPU
  Very simple to implement

Take Home Message

Representing uncertainty using samples is powerful, fast, and simple !

Questions