Mobile Robotics, Localization: Particle Filters and Monte Carlo Localization

Material based on the book Probabilistic Robotics (Thrun, Burgard, Fox) [PR]; Chapter 4.3, 8.3

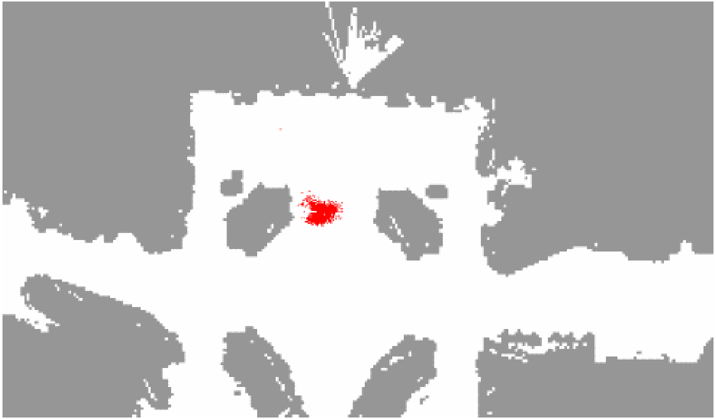
Part of the material is based on lectures from Cyrill Stachniss

Summary

- Introduction to Particle Filters
- Particle Filter [Chapter 4.3]
- Monte Carlo Localization [Chapter 8.3]

Intro to particle filters and Monte Carlo Localization

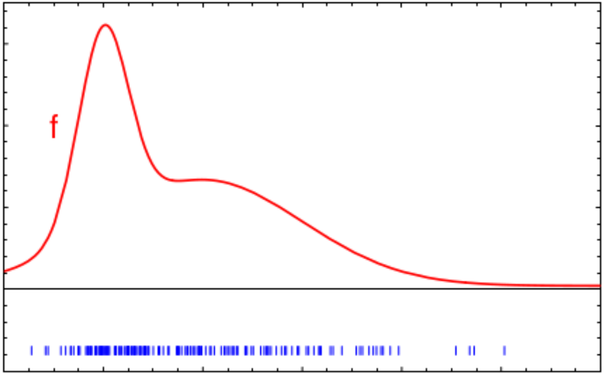
- Monte Carlo Localization: based on particle filters, non parametric



Visualization of Monte Carlo Localization, source [PR]

Key ideas for particle filters

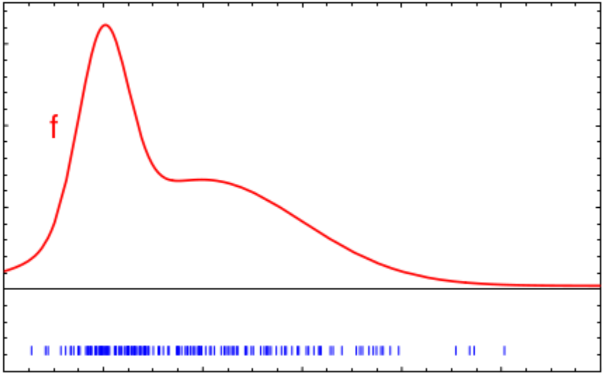
- **Goal**: represent arbitrary distributions
- Use samples to represent the distribution



Sample based representation of a generic function, source [PR]

Weighted samples

- Can reduce number of samples by using weights



Sample based representation of a generic function, source [PR]

Particle Set

- Set of weighted samples

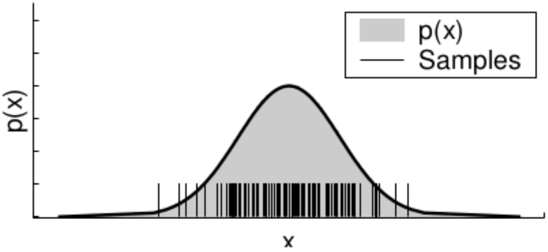
$$\mathcal{X} = \{\langle x^{[j]}, \omega^{[j]} \rangle\}_{j=1,\ldots,M}$$

- Samples represent the posterior

$$P(x) = \sum_{j=1}^{M} \omega^{[j]} \delta_{x^{[j]}}(x)$$

Generating samples

- Key point: we can efficiently sample from some distributions
  - Uniform $U(a, b)$: use pseudo-random generator $x \leftarrow rand(a, b)$
  - Gaussian $\mathcal{N}(0, \sigma)$: $x = \frac{1}{2} \sum_{i=1}^{12} rand(-\sigma, \sigma)$

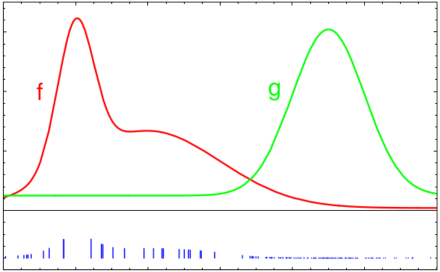


Samples drawn from a Gaussian distribution, source [PR]

- How to sample from other distributions ?

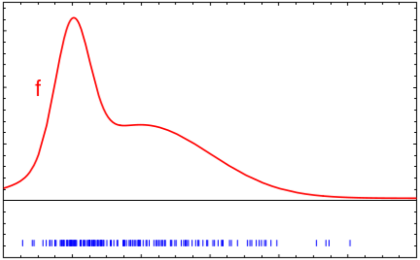
Importance Sampling

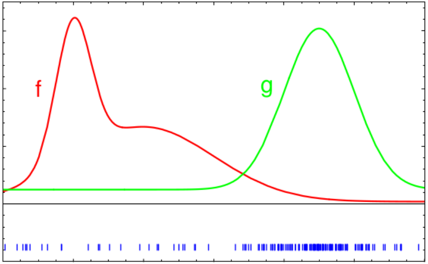
- **Goal**: generate samples from a **target** distribution f
- we can use a different distribution g called **proposal**
- account for difference between the two distributions by using weights
  - $\omega = \frac{f(x)}{g(x)}$
- **pre-condition**: $f(x) > 0 \rightarrow g(x) > 0$

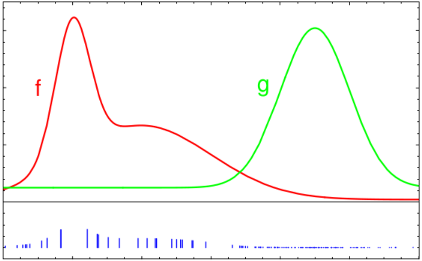


Visualization of importance sampling, source [PR]

Importance Sampling: Visual example







Visual example of importance sampling, source [PR]

Particle Filter for Dynamic State Estimation

- Recursive Bayes Filter
- Non-parametric approach
  - Models the distribution by **samples**
- **Key Ideas**
  - **Prediction**: drawing samples from the proposal
  - **Correction**: weighting by the ration between target and proposal
- The more samples the better the estimate

Particle Filter Algorithm, informal

- 1. Sample particles using proposal distribution

$$x_t^{[j]} \sim proposal(x_t | \ldots)$$

- 2. Compute the importance weights

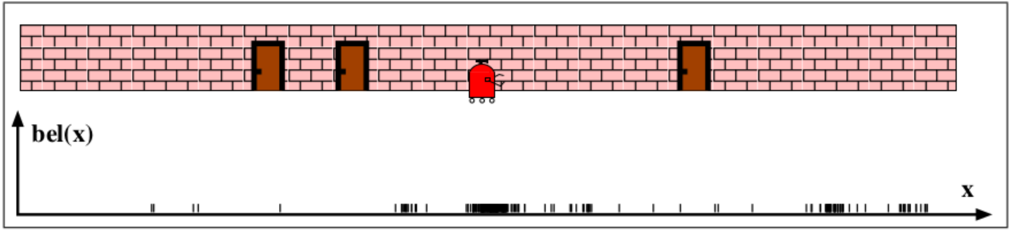
$$\omega_t^{[j]} = \frac{target(x_t^{[j]})}{proposal(x_t^{[j]})}$$

- 3. Resampling: draw sample i with probability $\omega_t^{[j]}$ and repeat M times

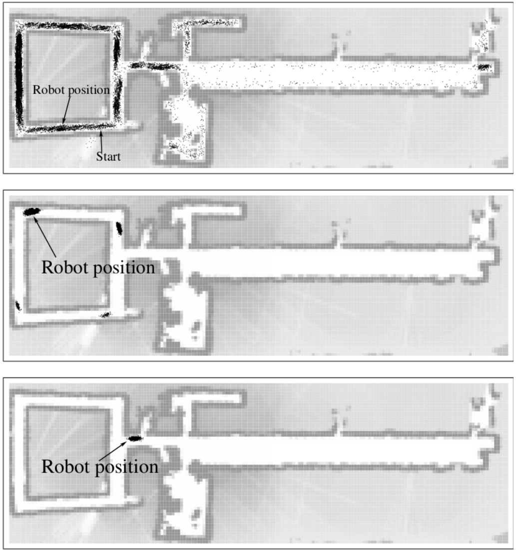
Particle Filter Algorithm, pseudo-code

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

Monte Carlo Localization, overview



Monte Carlo Localization in office environment



Evolution of particles while robot moves in the environment, source [PR]

Monte Carlo Localization

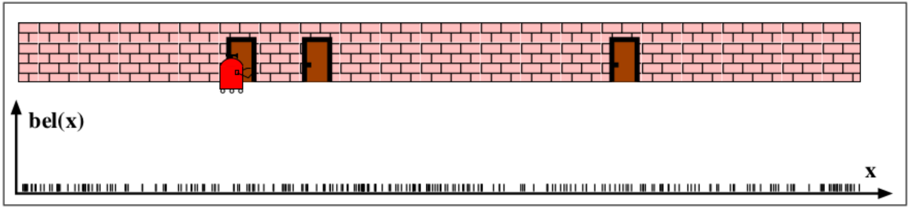
- Each particle is a pose hypothesis
- Proposal is motion model

$$x_t^{[m]} \sim P(x_t | u_t, x_{t-1}^{[m]})$$

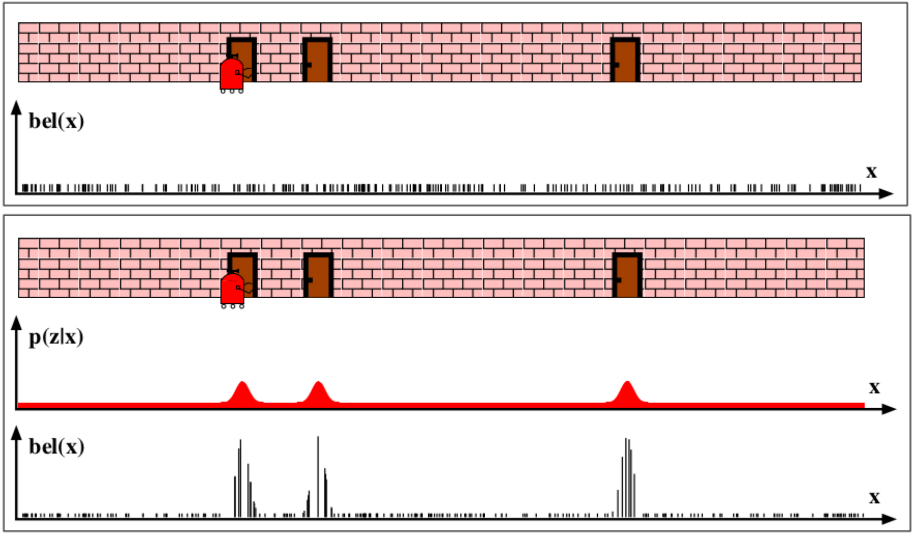
- Correction performed through observation model

$$\omega_t^{[m]} = \frac{target}{proposal} \propto P(z_t | x_t^{[m]}, m)$$

MCL: correction

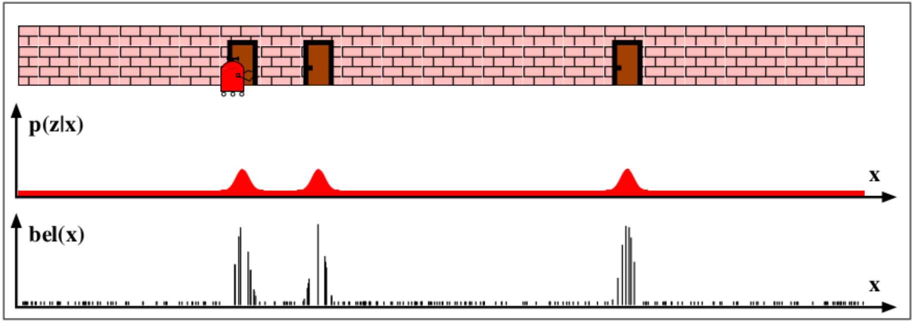


MCL: correction

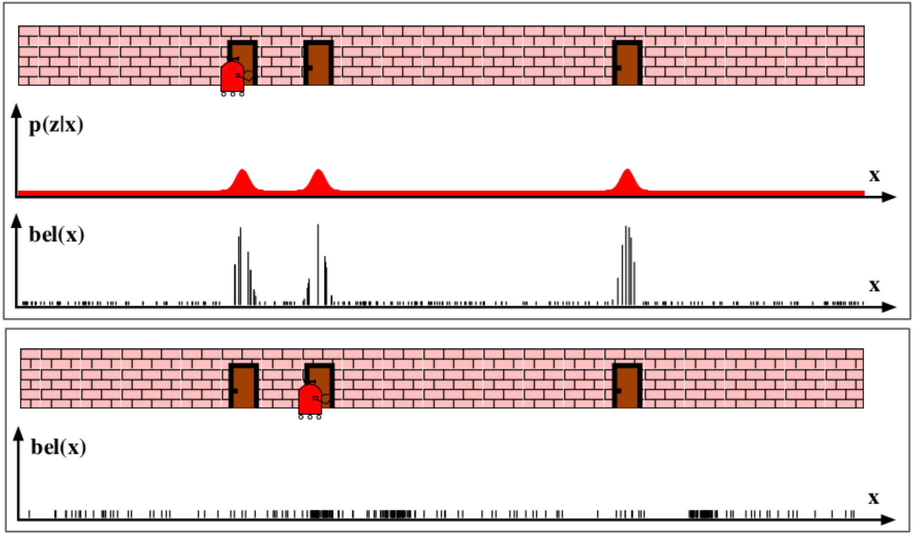


Source [PR]

MCL: Prediction and Resampling

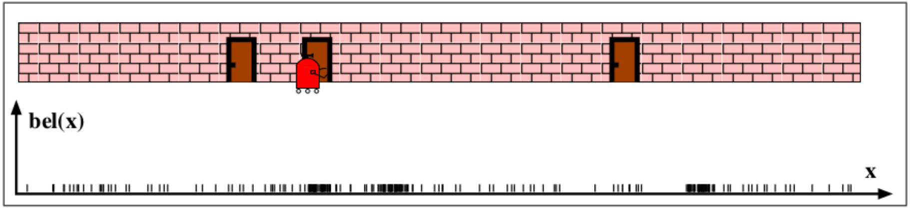


MCL: Prediction and Resampling

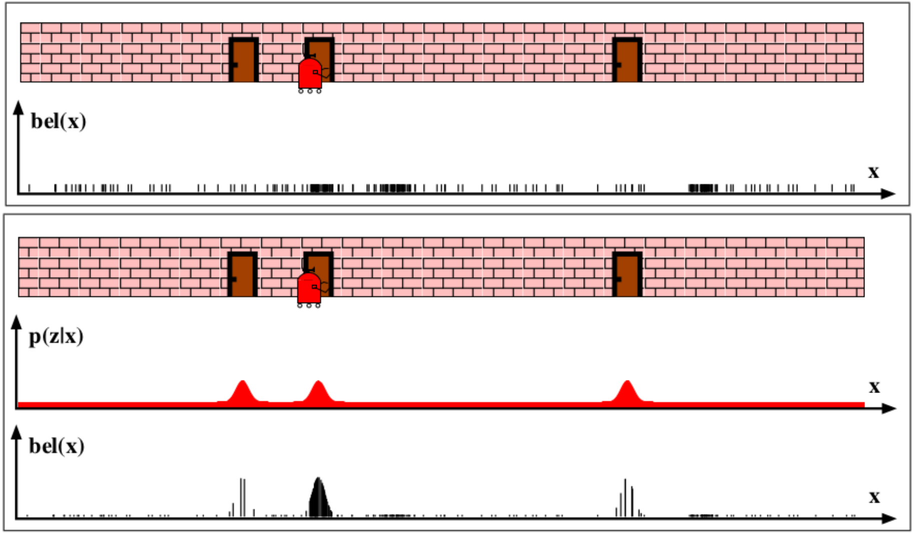


Source [PR]

MCL: Second Correction

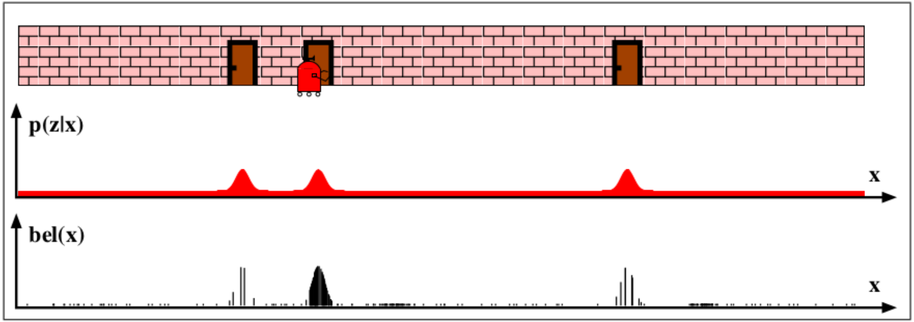


MCL: Second Correction

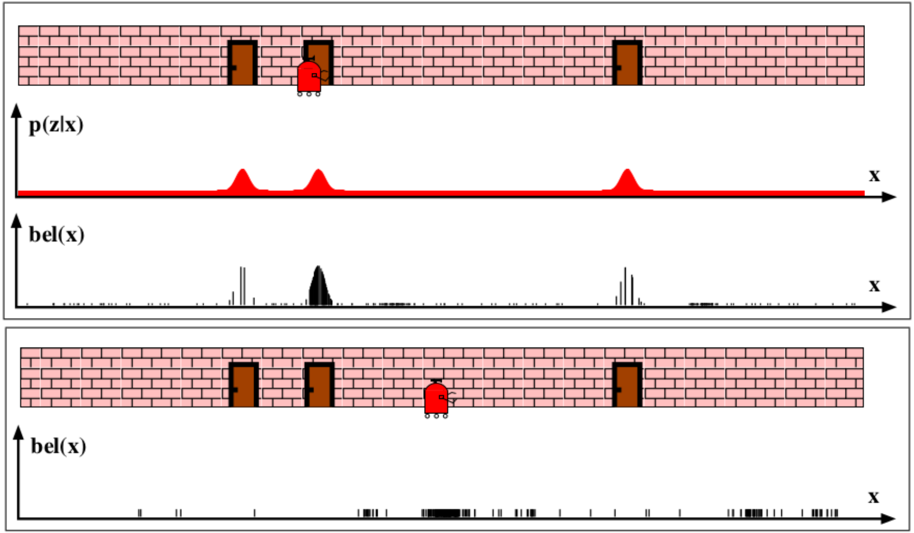


Source [PR]

MCL: Second Prediction and Resampling

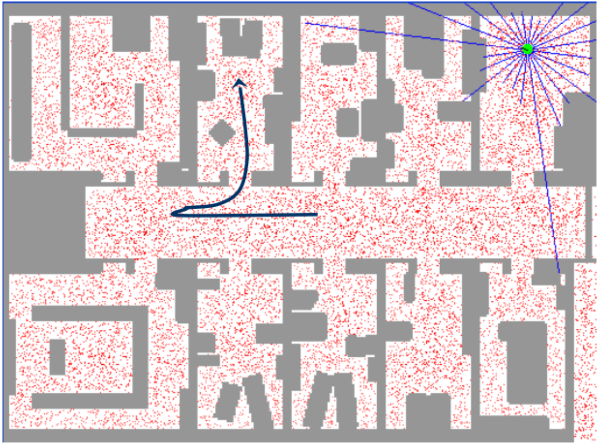


MCL: Second Prediction and Resampling



Source [PR]

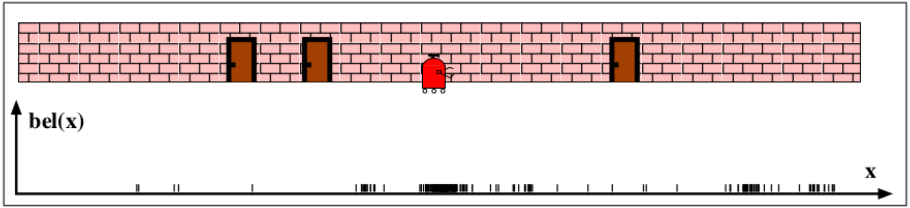
MCL in Action



Courtesy: Dieter Fox

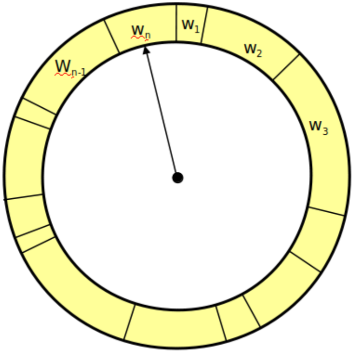
Resampling

- **Goal**: Maintain informative particles, avoid wasting memory
- Informally: replace unlikely samples by more likely ones
- Survival of the fittest
- "Trick" to avoid that many samples cover unlikely states (waste of memory)
- Draw sample i with probability $\omega_t^{[i]}$

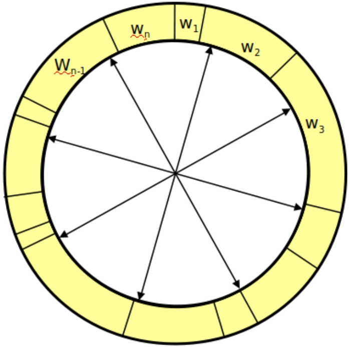


Need to resample to focus on more likely area, source [PR]

Resampling



Roulette wheel, binary search ($O(n \log n)$), source [PR] slides

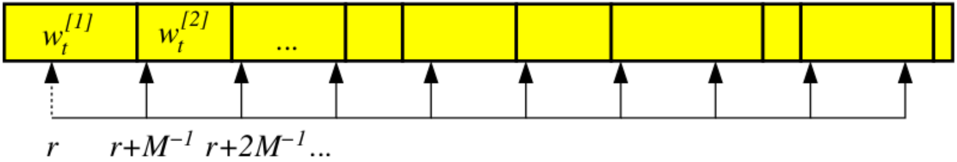


Stochastic universal sampling, systematic resampling, linear time ($O(n)$), low variance, source [PR] slides

Issues with roulette sampling

- Roulette wheel is easy to understand and implement but is sub-optimal
- Can lead to bad estimate (high variance) in specific situations
- What happens to roulette sampling if all samples have same weight ?

Low variance resampling



◇ **1.** Draw a random number between 0 and $1/M$ ◇ **2.** Pick $M-1$ particles at distance $1/M$

Low variance resampling, pseudocode

Efficient implementation of the low variance sampling procedure, source [PR]

Algorithm Low_variance_sampler($\mathcal{X}_t, \mathcal{W}_t$):
  $\bar{\mathcal{X}}_t = \emptyset$
  $r = \text{rand}(0; M^{-1})$
  $c = w_t^{[1]}$
  $i = 1$
  for $m = 1$ to M do
    $u = r + (m-1) \cdot M^{-1}$
    while $u > c$
      $i = i + 1$
      $c = c + w_t^{[i]}$
    endwhile
    add $x_t^{[i]}$ to $\bar{\mathcal{X}}_t$
  endfor
  return $\bar{\mathcal{X}}_t$

Low variance resampling, features

- $\diamondsuit$ Faster than roulette wheel: $O(M)$ vs. $O(M \log M)$
- $\diamondsuit$ **Most important**: performs resampling that keeps the samples in case of same weights

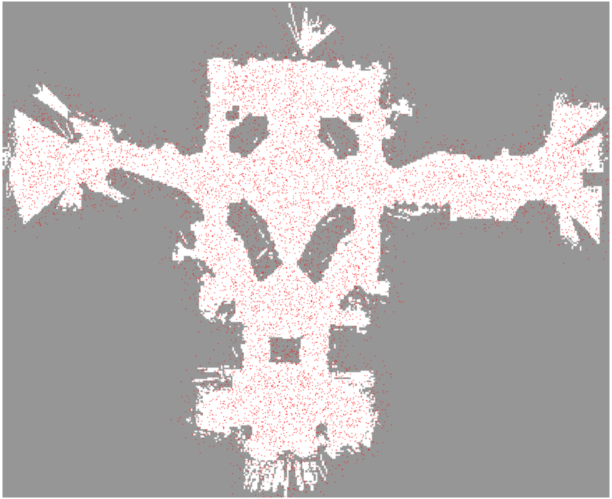
Always use low variance resampling!

Use of MCL for mobile robot localization

◇ Rhino, Minerva, Albert ($\approx$ 1998)

Courtesy of Burgard, Fox, Thrun

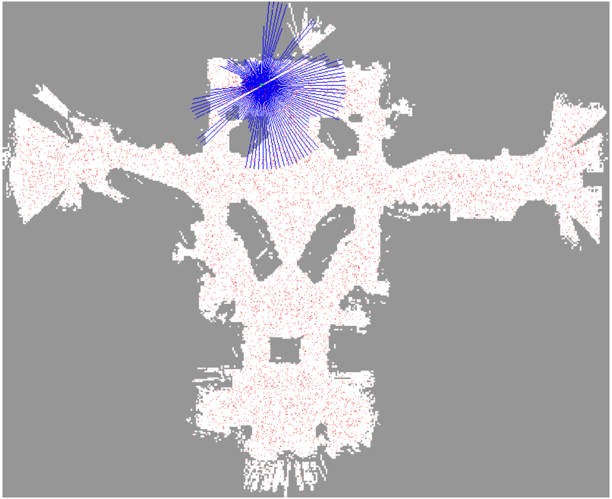
MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

Initialization

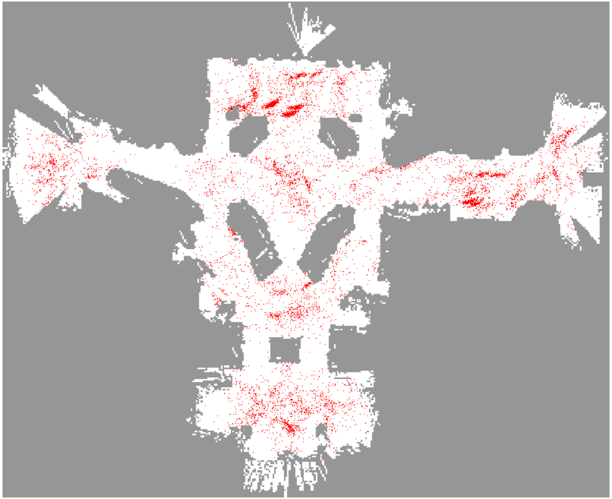
MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

Observation

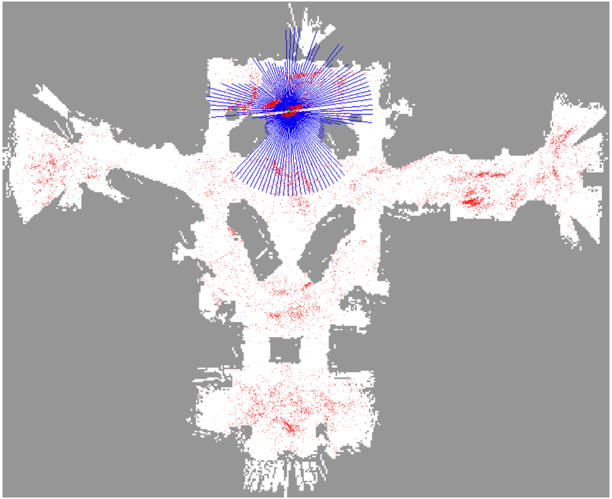
MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

Resampling and motion update

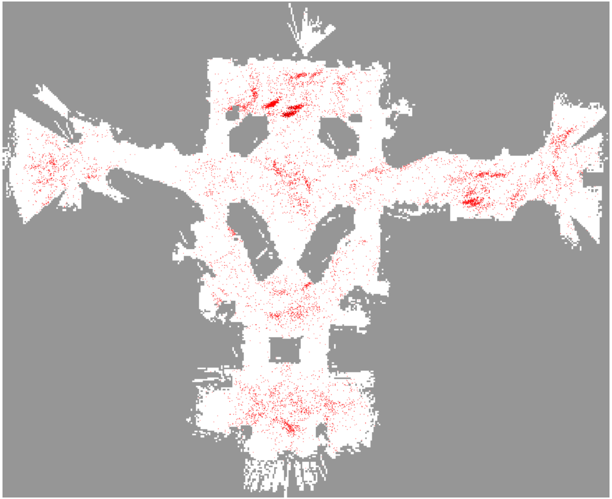
MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

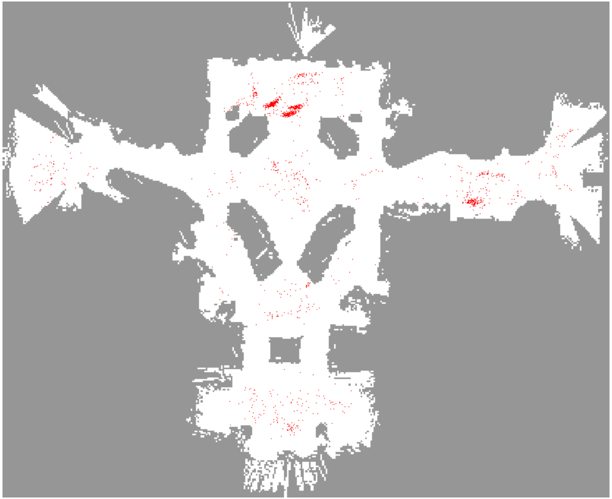
Observation

MCL Example of application, Minerva in the Smithsonian Museum


Courtesy of Burgard, Fox, Thrun

Weight update

MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

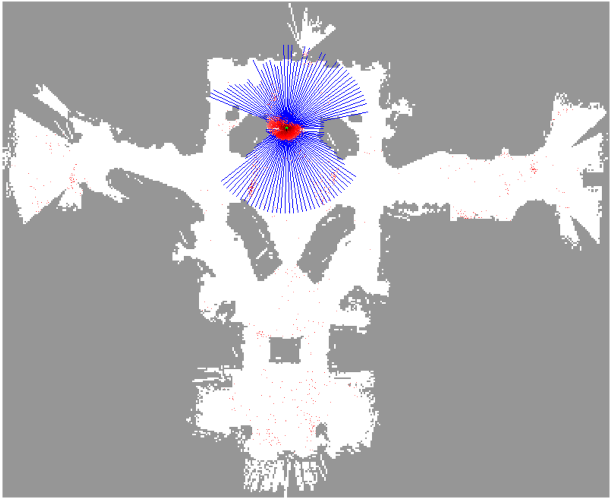
Resampling

MCL Example of application, Minerva in the Smithsonian Museum

Courtesy of Burgard, Fox, Thrun

Motion update

MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

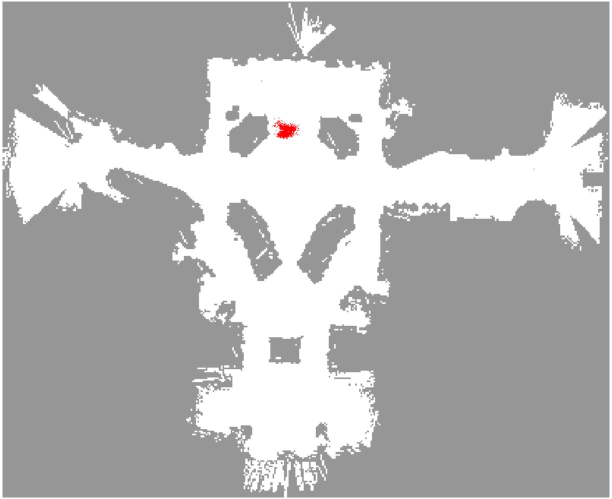
Observation

MCL Example of application, Minerva in the Smithsonian Museum

Courtesy of Burgard, Fox, Thrun

Weight update

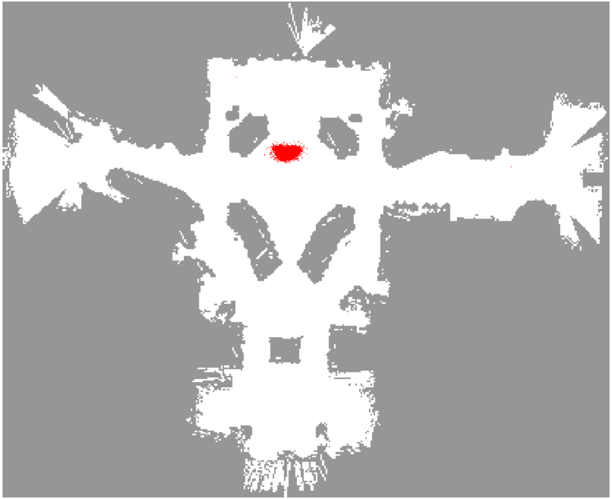
MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

Resampling

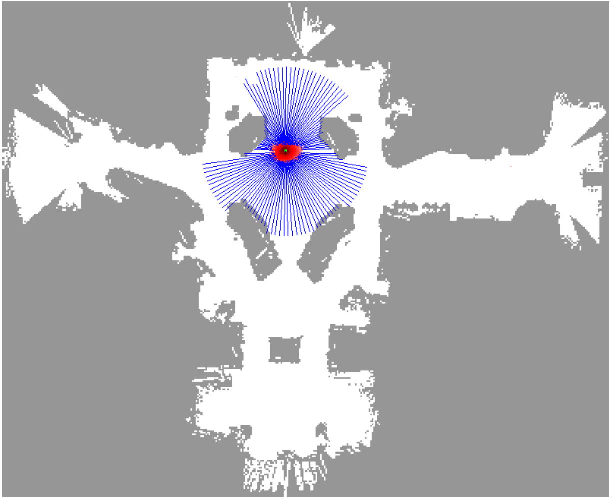
MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

Motion update

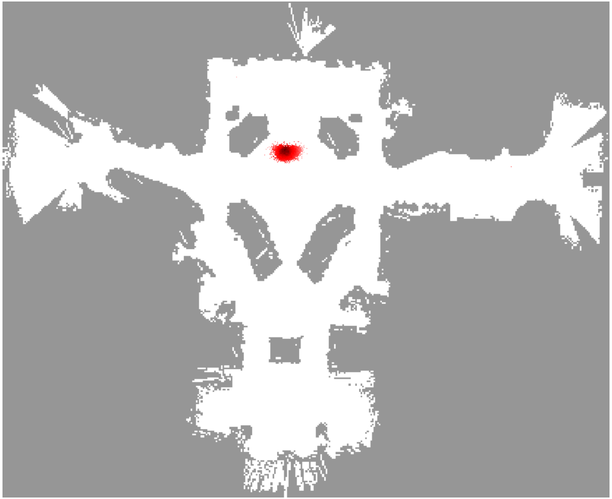
MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

Observation

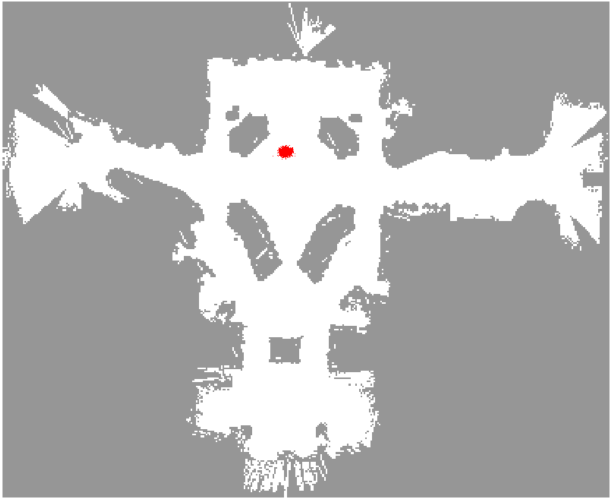
MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

Weight update

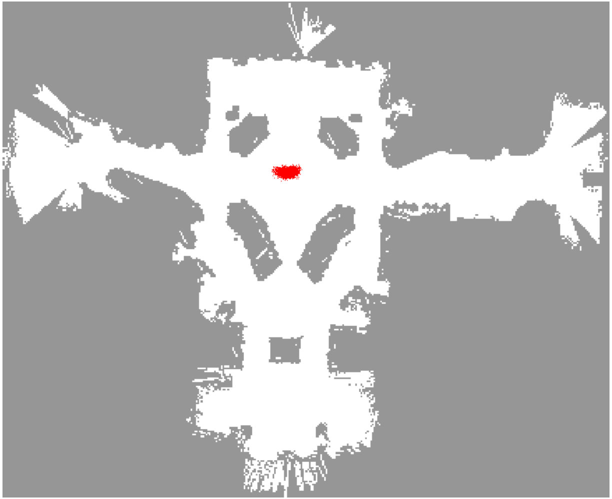
MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

Resampling

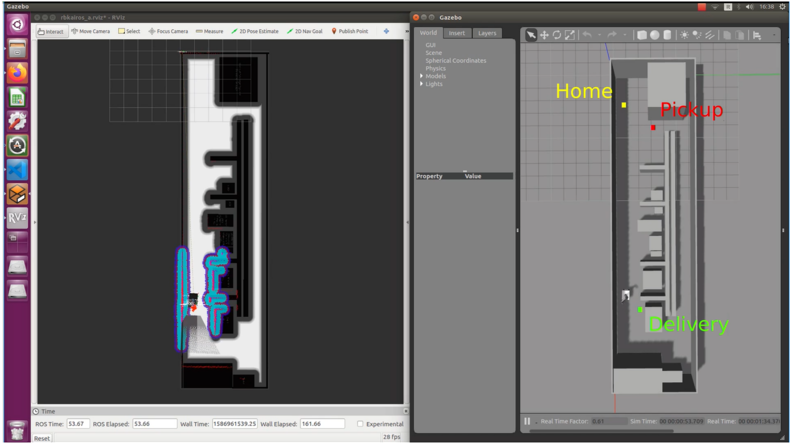
MCL Example of application, Minerva in the Smithsonian Museum



Courtesy of Burgard, Fox, Thrun

Motion update

MCL in ICE Lab



PF for localization

- **Cons**
  - Problematic in high dimensional spaces
  - problematic in situation with high uncertainty
  - Particle depletion problem

- **Pros**
  - Handle directly non Gaussian distributions
  - Handle well data association ambiguities
  - Can easily incorporate different sensing modalities
  - Robust
  - Easy to implement

Variants

- Real-time particle filters
  - Deal with data acquired at different frame rates
- Delayed state particle filters
  - Deal with delays in sensor data streams
- Rao-Blackwellized Particle Filters
  - Deal with high dimensional state spaces

Summary

- Particle Filters: non-parametric recursive Bayes filters
- Belief is represented by a set of weighted samples
- Use proposal (motion model) to draw samples
- Use weight to correct (observation model)
- Particle filter for localization: Monte Carlo Localization
- **Key point**: design appropriate motion and observation models
- MCL is the gold standard for indoor mobile robot localization