

Robótica Móvel

Localização – Monte Carlo / Filtro de Partículas

Prof. Douglas G. Macharet
douglas.macharet@dcc.ufmg.br

Introdução

- Localização de Markov
 - Discretização do espaço
 - Alto custo de memória / processamento
 - Resolução fixa (não adaptável à crença)
- Como melhorar?
 - Técnicas baseadas em amostragem
 - Amostra → possível **configuração** no ambiente

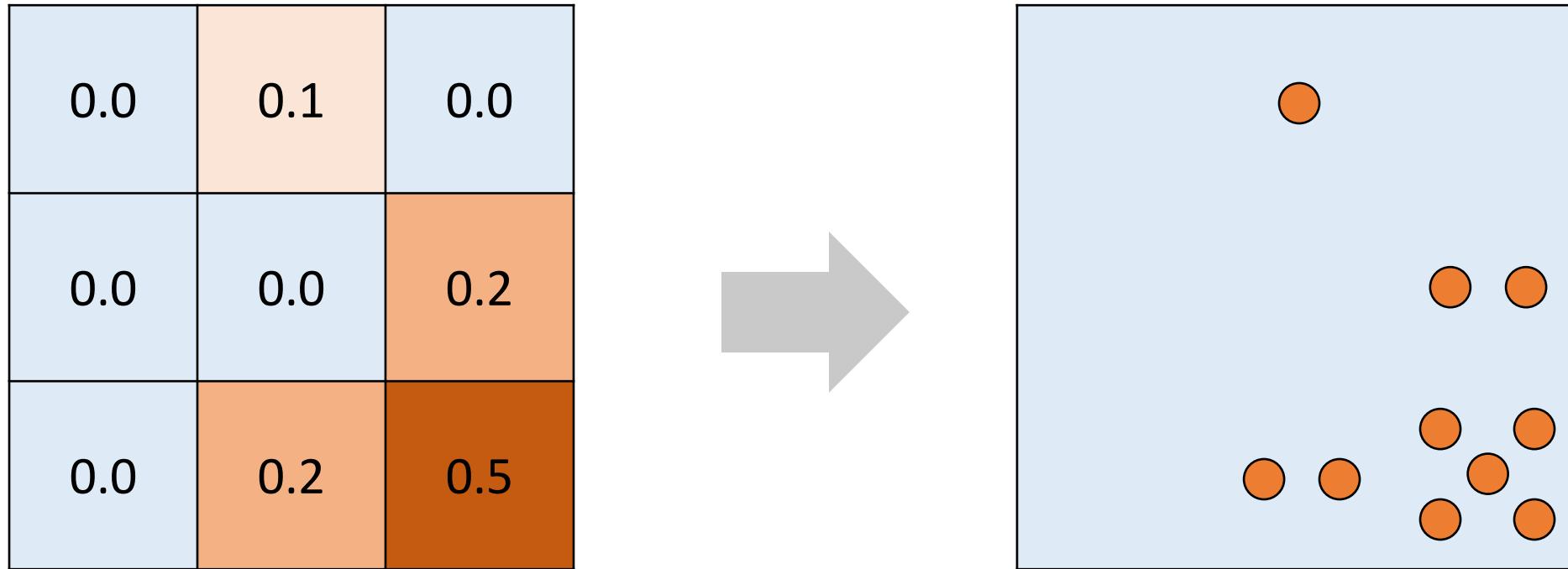
Introdução

Filtro de partículas

- Sistemas não-lineares
- Distribuições arbitrárias (não Gaussianas)
 - Representação aproximada de um modelo complexo em vez da representação exata do modelo simplificado (por exemplo, KF)
- Eficiente
 - Não atualiza todo o espaço de estados (por exemplo, grid), mas atualiza apenas o subespaço amostral definido pelas partículas

Introdução

Filtro de partículas



Localização de Monte Carlo

- A distribuição $bel(x_t)$ é representada por um conjunto \mathcal{X}_t de amostras com o estado (x_t) e um peso associado (ω_t)

$$\mathcal{X}_t = \left\{ \langle x_t^{[i]}, \omega_t^{[i]} \rangle \mid i = 1, \dots, M \right\}$$

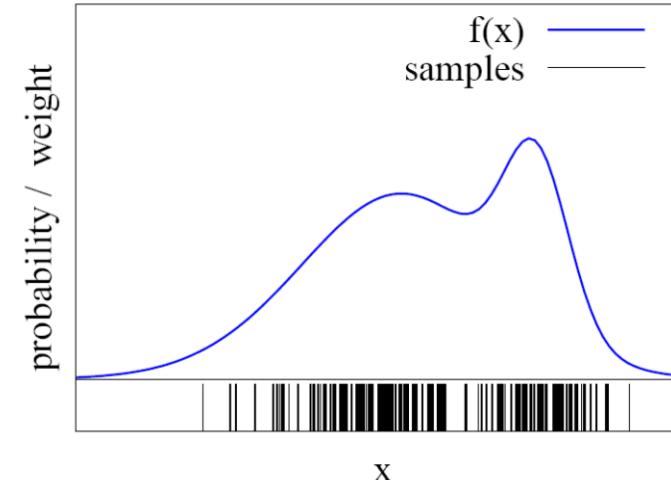
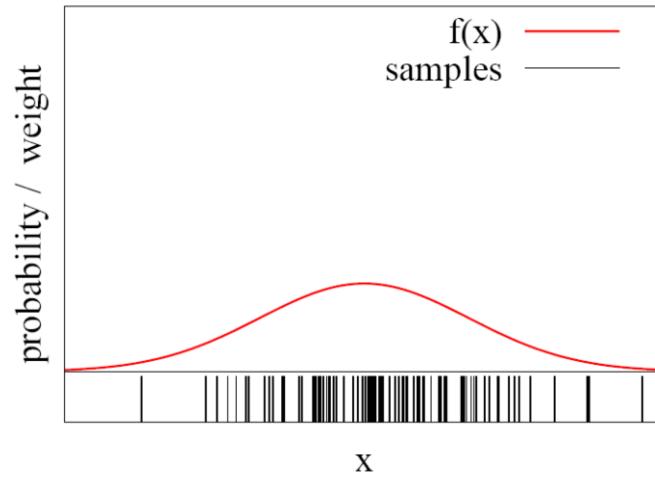
- onde

$$bel(x_t) \approx \sum_{i=1}^M \omega_t^{[i]} \delta(x_t - x_t^{[i]}) \quad \text{e} \quad \sum_{i=1}^M \omega_t^{[i]} = 1$$

https://en.wikipedia.org/wiki/Dirac_delta_function#Probability_theory

Localização de Monte Carlo

- Partículas usadas para aproximar funções arbitrárias



- Quanto maior o número de partículas em um intervalo, significa que maior a densidade de probabilidade na região

Fonte: *Probabilistic Robotics*



Localização de Monte Carlo

Rejection sampling

- PDF $f(x)$ não representável analiticamente, como amostrar?
- Considerar uma função $g(x)$, onde $f(x) < M \cdot g(x)$

I. Amostrar um par (x, u)

- a) $x \sim g(x)$
- b) $u \sim U(0,1)$

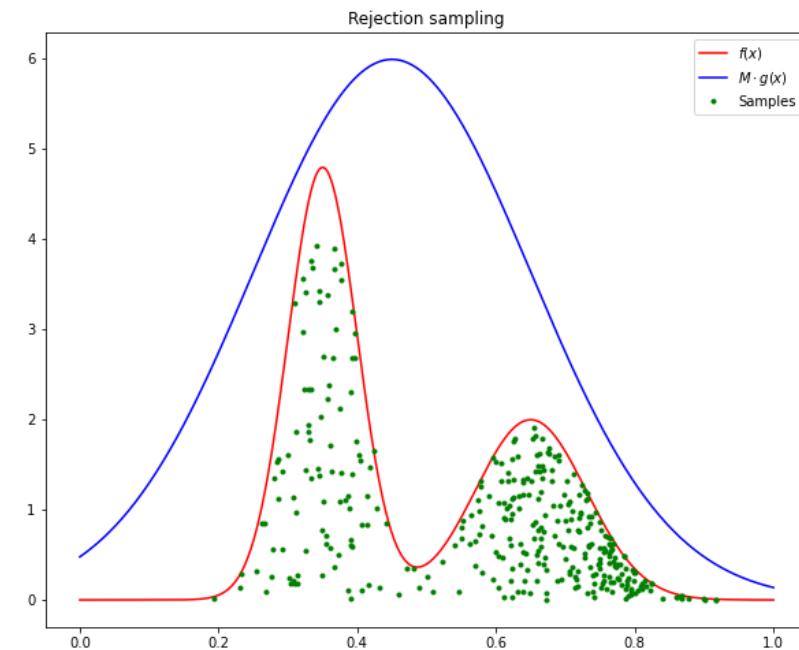
2. Aceitar x como amostra de $f(x)$ se

$$u < \frac{f(x)}{M \cdot g(x)}$$

https://en.wikipedia.org/wiki/Rejection_sampling

<https://www.jarad.me/teaching/2013/10/03/rejection-sampling>

<https://www.data-blogger.com/2016/01/24/the-mathematics-behind-rejection-sampling/>



Localização de Monte Carlo

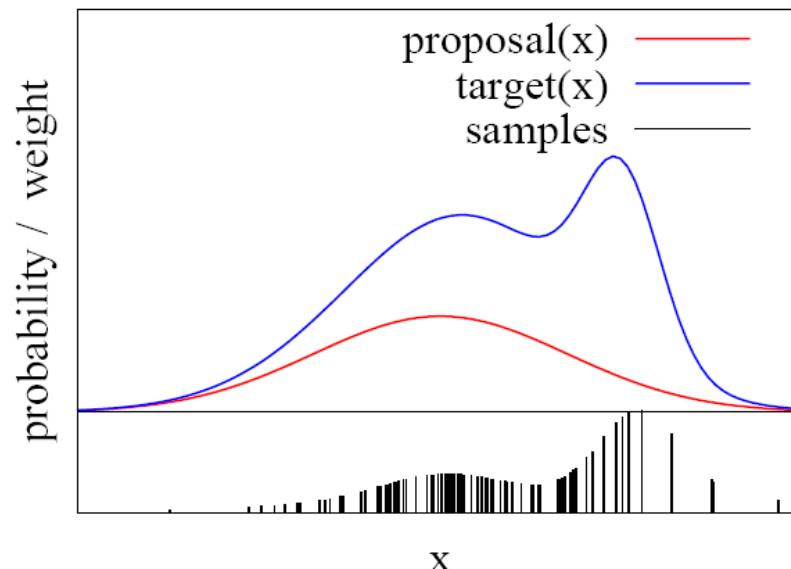
Importance sampling

- Utilizar amostras de uma distribuição $g(x)$ (proposal) para estimar propriedades de outra distribuição $f(x)$ (target)
- Introdução de um fator de importância (peso): $\omega = f(x)/g(x)$

Pré-condição:

$$f(x) > 0 \rightarrow g(x) > 0$$

Garantir que vamos gerar amostras que cobrem toda a extensão de $f(x)$



Fonte: *Probabilistic Robotics*

https://en.wikipedia.org/wiki/Importance_sampling

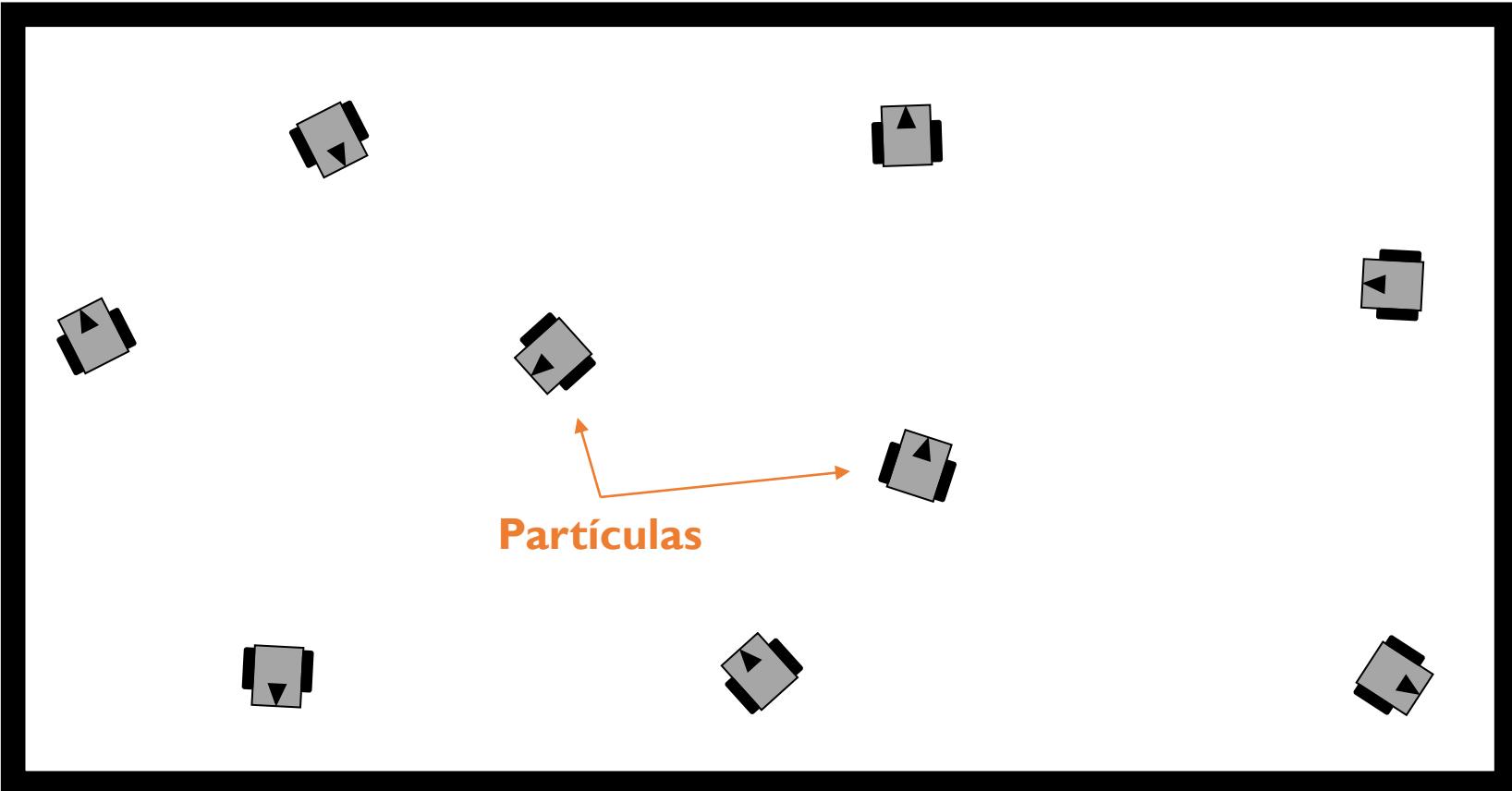


Localização de Monte Carlo

- Predição
 - Atualizar a amostra (estado) aplicando o modelo de transição
- Atualização
 - Atribuir pesos às amostras utilizando o modelo de observação
- Reamostragem
 - Redistribuir as amostras considerando os pesos calculados
 - Substituir amostras “improváveis” por outras mais prováveis

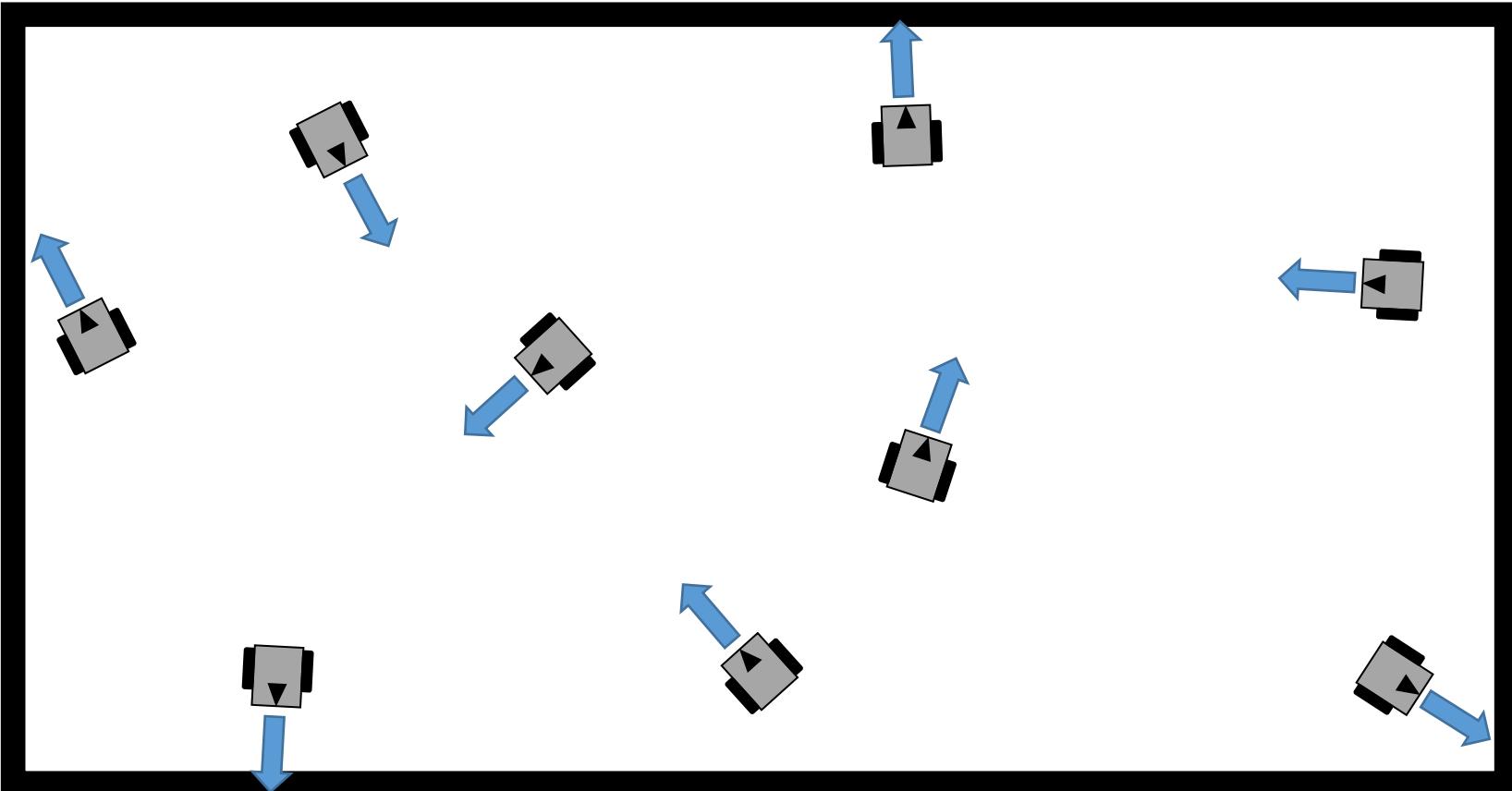
Localização de Monte Carlo

Inicialização



Localização de Monte Carlo

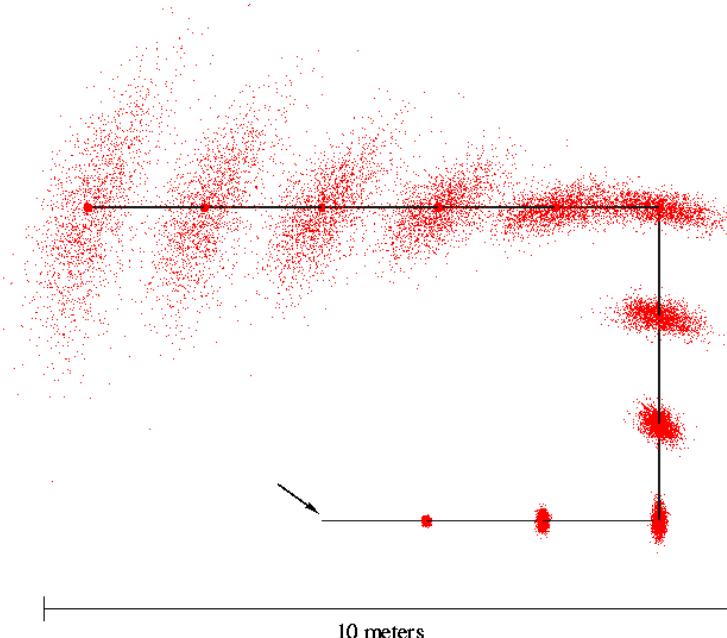
Predição



Localização de Monte Carlo

Predição

- Aplicar o modelo de transição (ação)
- Amostrar usando a distribuição proposta (g)



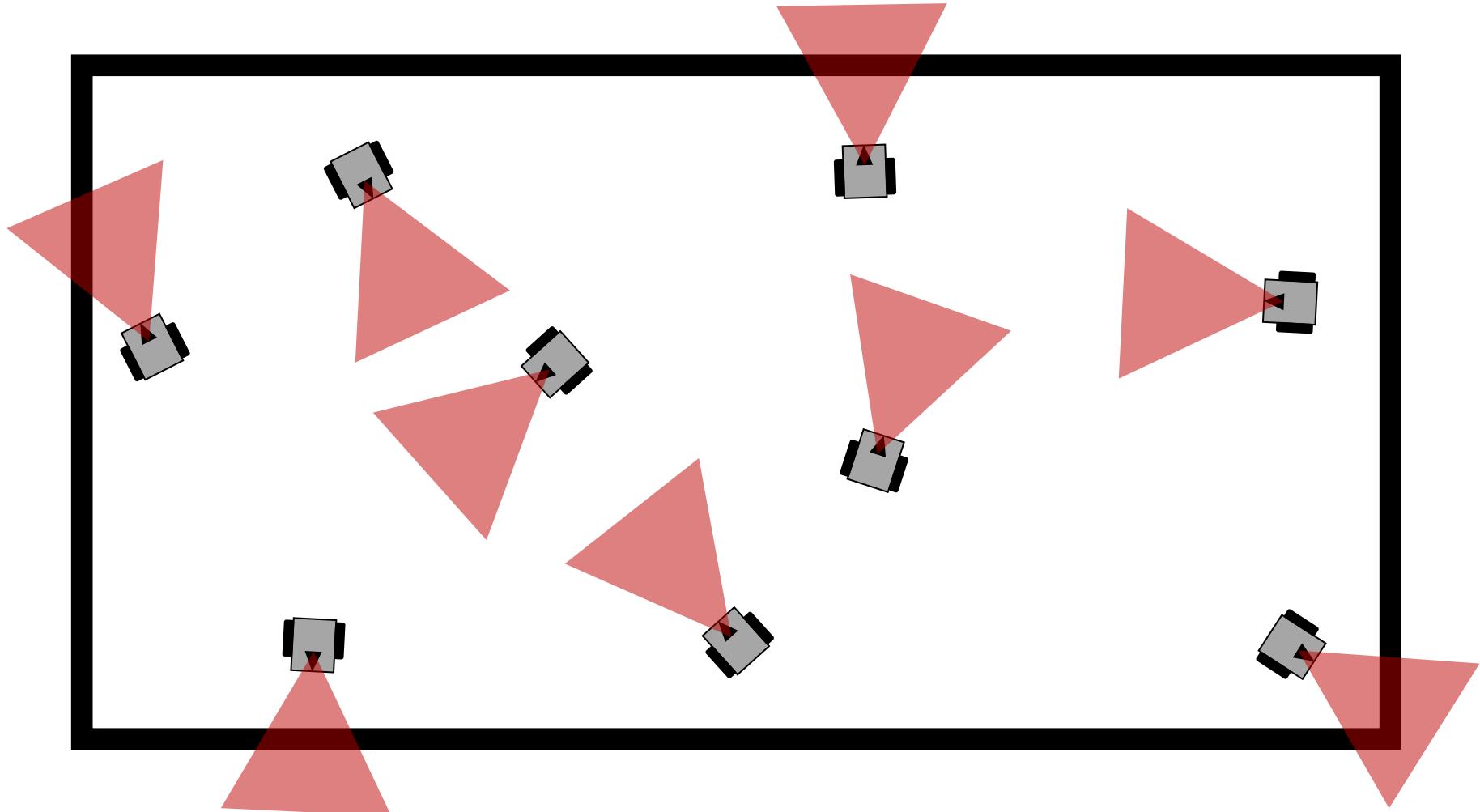
$$x_t^{[i]} \sim p(x_t^{[i]} | x_{t-1}^{[i]}, u_t)$$

Fonte: *Probabilistic Robotics*



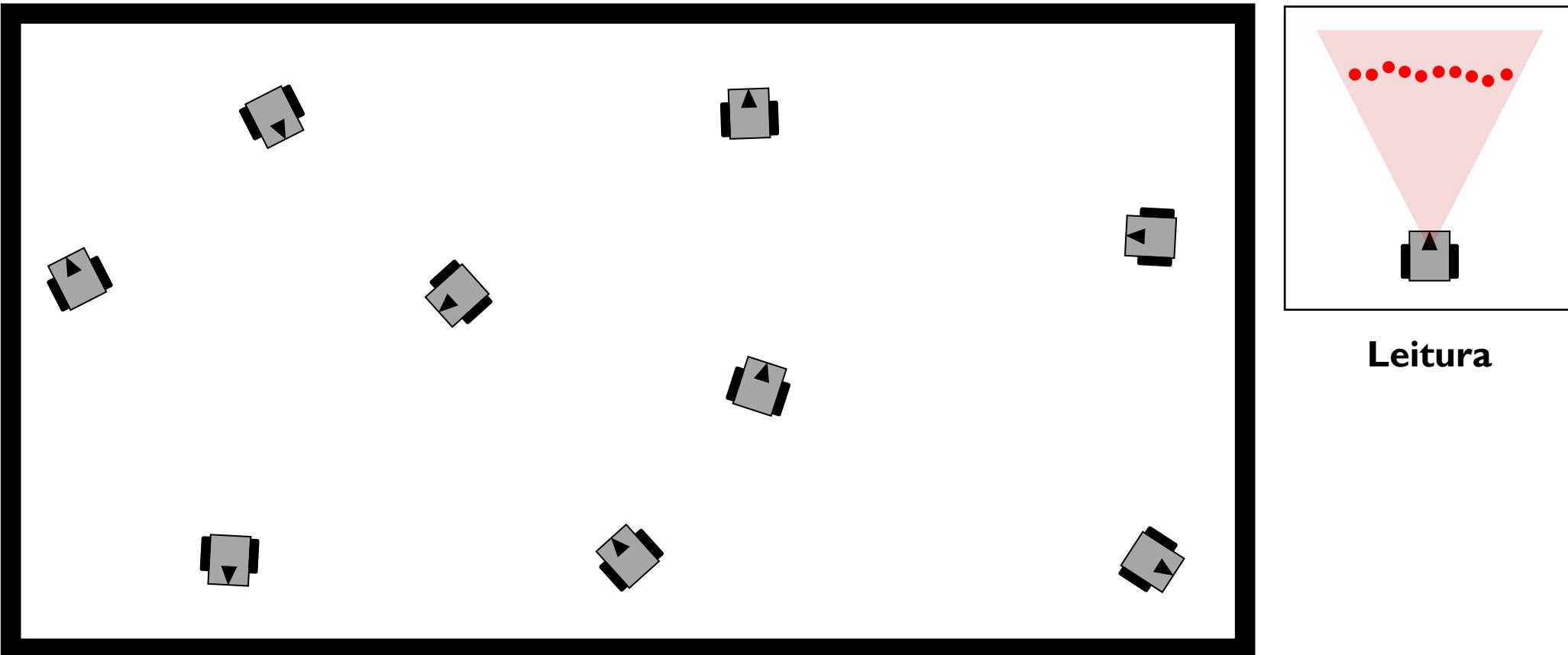
Localização de Monte Carlo

Atualização



Localização de Monte Carlo

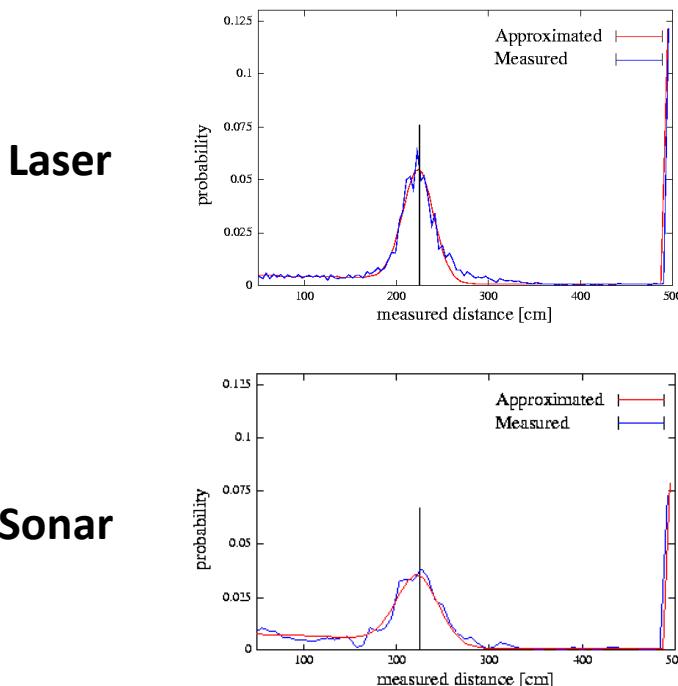
Atualização



Localização de Monte Carlo

Atualização

- Aplicar o modelo de observação (correção)
- Determinar os pesos das amostras



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

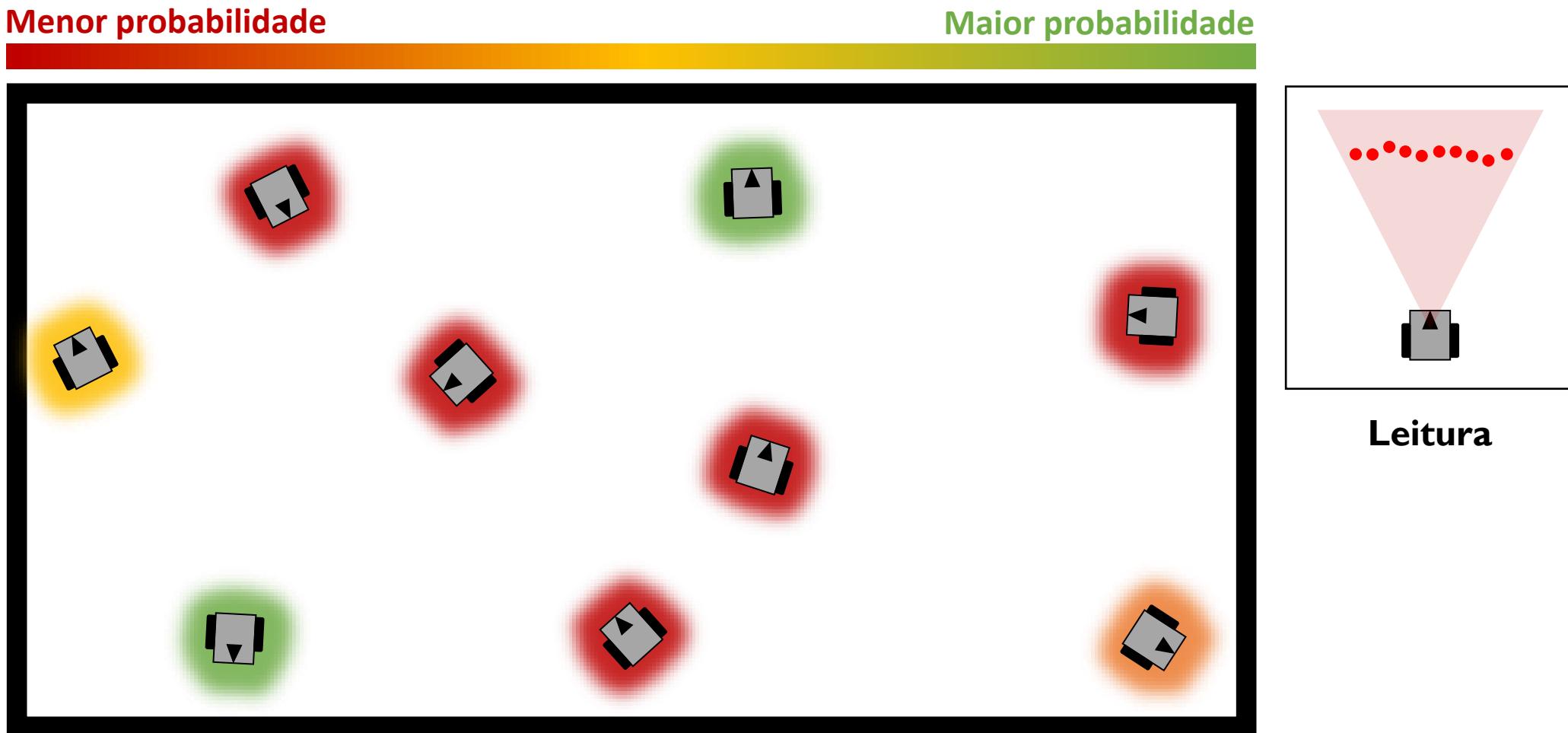
$$\omega_t^{[i]} = \frac{\omega_i^{[i]}}{\sum_M \omega_i^{[i]}}$$

Fonte: *Probabilistic Robotics*



Localização de Monte Carlo

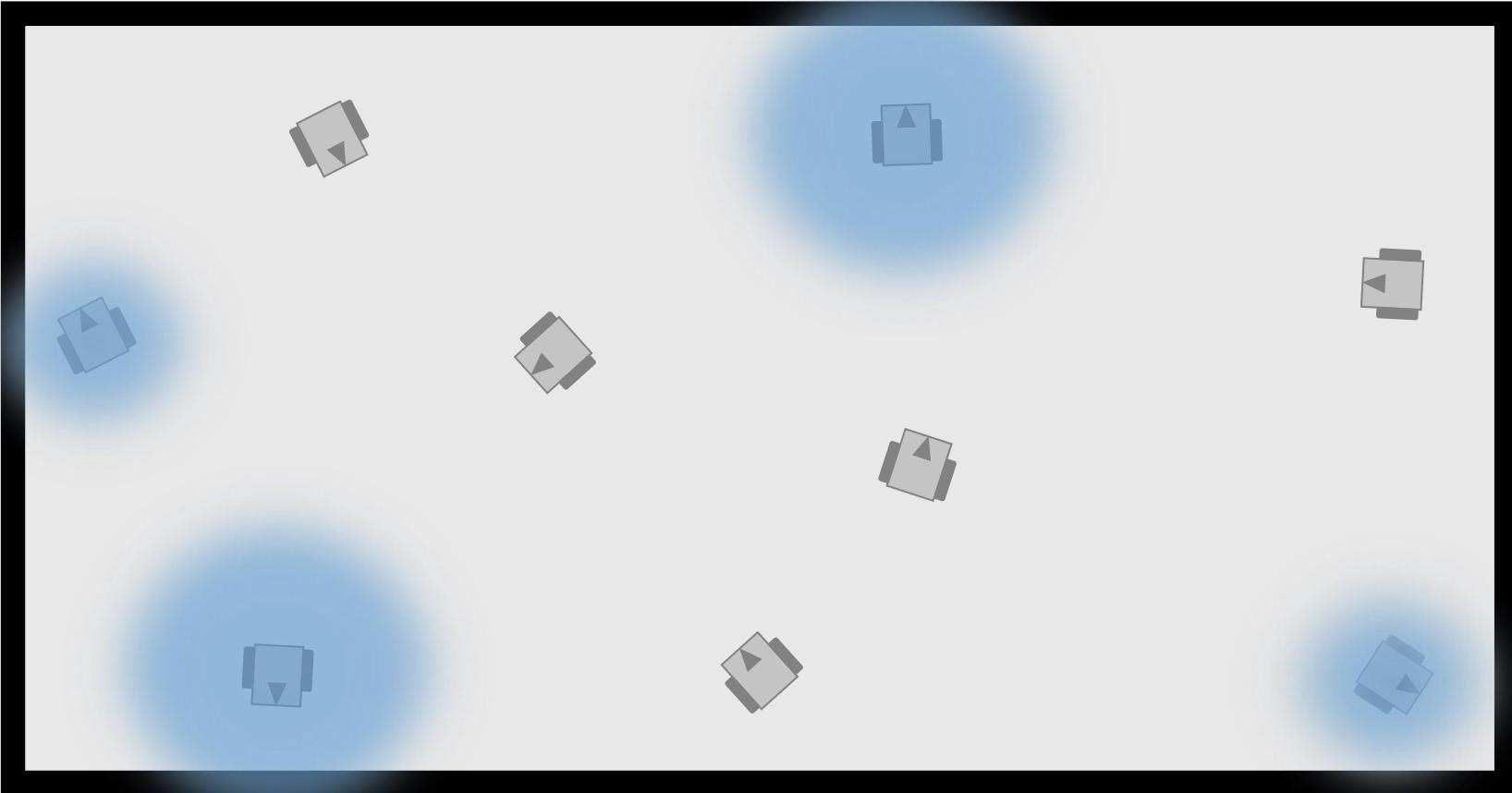
Atualização



Localização de Monte Carlo

Atualização

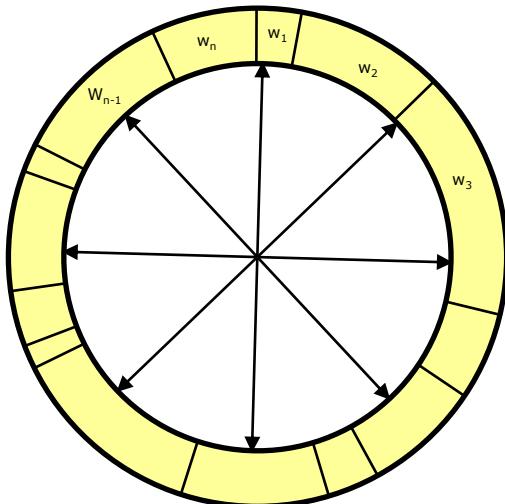
Distribuição de probabilidade



Localização de Monte Carlo

Reamostragem

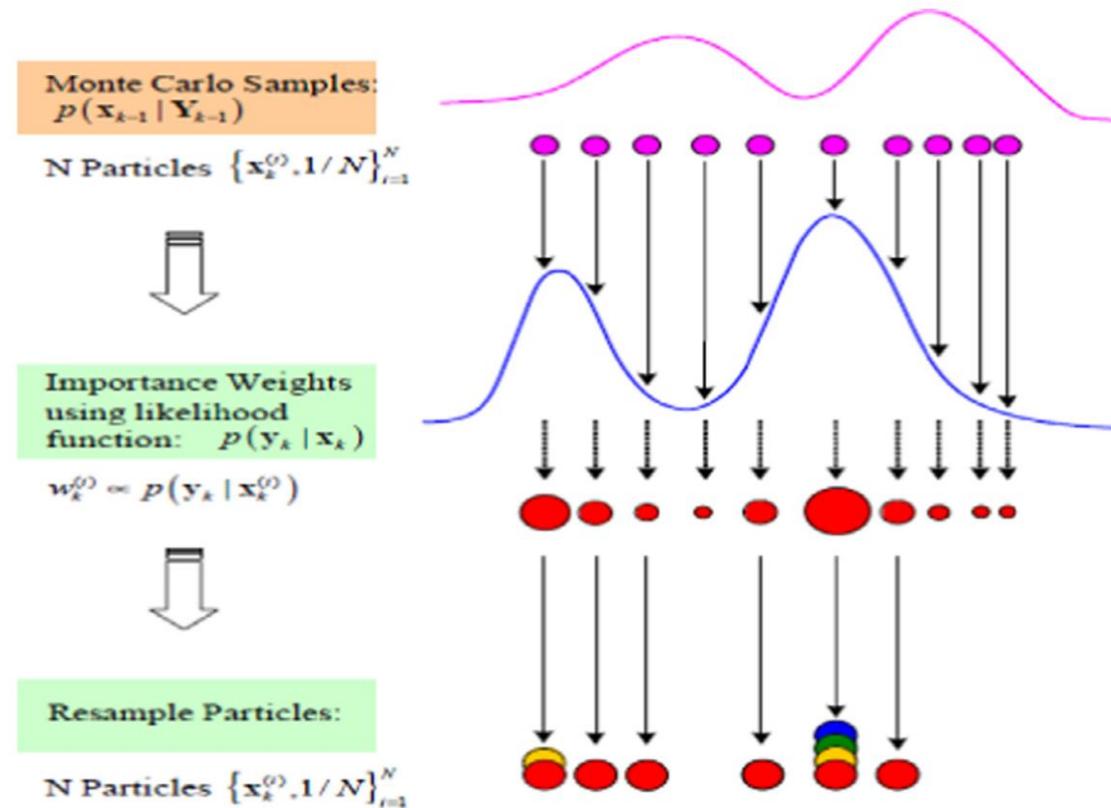
- Dado um conjunto ponderado de amostras \mathcal{X}
- Gerar um novo conjunto de amostras, onde a probabilidade de se manter uma amostra x_i é proporcional ao peso ω_i



```
1: Algorithm Low_variance_sampler( $\mathcal{X}_t, \mathcal{W}_t$ ):  
2:    $\bar{\mathcal{X}}_t = \emptyset$   
3:    $r = \text{rand}(0; M^{-1})$   
4:    $c = w_t^{[1]}$   
5:    $i = 1$   
6:   for  $m = 1$  to  $M$  do  
7:      $U = r + (m - 1) \cdot M^{-1}$   
8:     while  $U > c$   
9:        $i = i + 1$   
10:       $c = c + w_t^{[i]}$   
11:    endwhile  
12:    add  $x_t^{[i]}$  to  $\bar{\mathcal{X}}_t$   
13:  endfor  
14:  return  $\bar{\mathcal{X}}_t$ 
```

Localização de Monte Carlo

Reamostragem

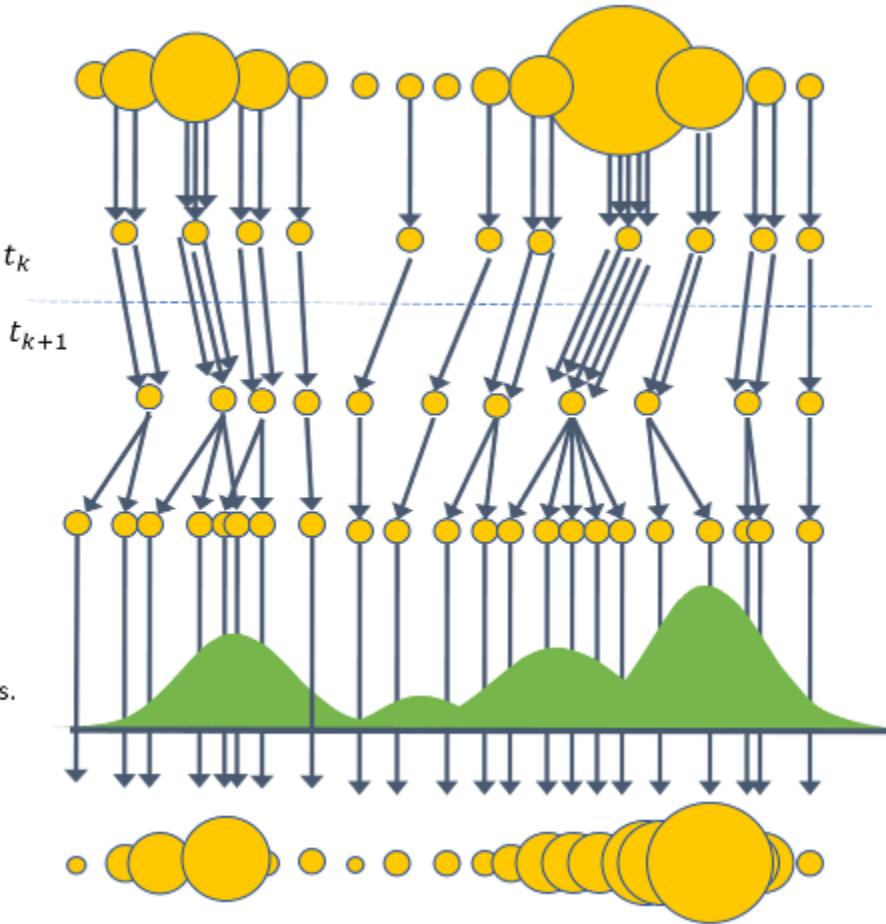


Sharifian, M. S., Rahimi, A., & Pariz, N. (2016). Classifying the weights of particle filters in nonlinear systems. In: Communications in Nonlinear Science and Numerical Simulation.

Localização de Monte Carlo

Algoritmo

1. Weighted particles from initialization or last update. The weight is represented by the size of each particle.
2. Extract particles according to their weights.
3. Propagate the particle state one step forward using dynamic model.
4. Create multiple sub-particles according to the weight of the original particle following a certain distribution.
5. Resampling: Compare the predicted measurements given by these particles and the true measurements to update the particle weights.
6. New particles are obtained with weights.



Sample Motion Observation

Fonte: <https://ww2.mathworks.cn/help/fusion/ug/introduction-to-estimation-filters.html>

Localização de Monte Carlo

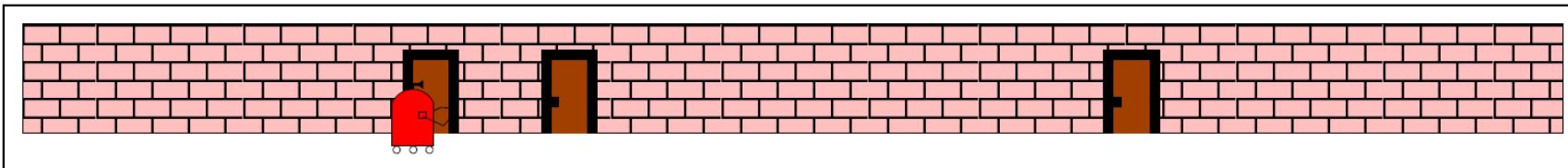
Algoritmo

```
1: Algorithm MCL( $\mathcal{X}_{t-1}, u_t, z_t, m$ ):
2:    $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$ 
3:   for  $m = 1$  to  $M$  do
4:      $x_t^{[m]} = \text{sample\_motion\_model}(u_t, x_{t-1}^{[m]})$ 
5:      $w_t^{[m]} = \text{measurement\_model}(z_t, x_t^{[m]}, m)$ 
6:      $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$ 
7:   endfor
8:   for  $m = 1$  to  $M$  do
9:     draw  $i$  with probability  $\propto w_t^{[i]}$ 
10:    add  $x_t^{[i]}$  to  $\mathcal{X}_t$ 
11:   endfor
12:   return  $\mathcal{X}_t$ 
```

Localização de Monte Carlo

Exemplo 1

- Localização em um corredor (1D)
- Contém três portas idênticas
 - Robô consegue identificar a presença de portas
 - Possuímos um mapa do ambiente (posição das portas)
- Robô conhece a sua direção de movimento



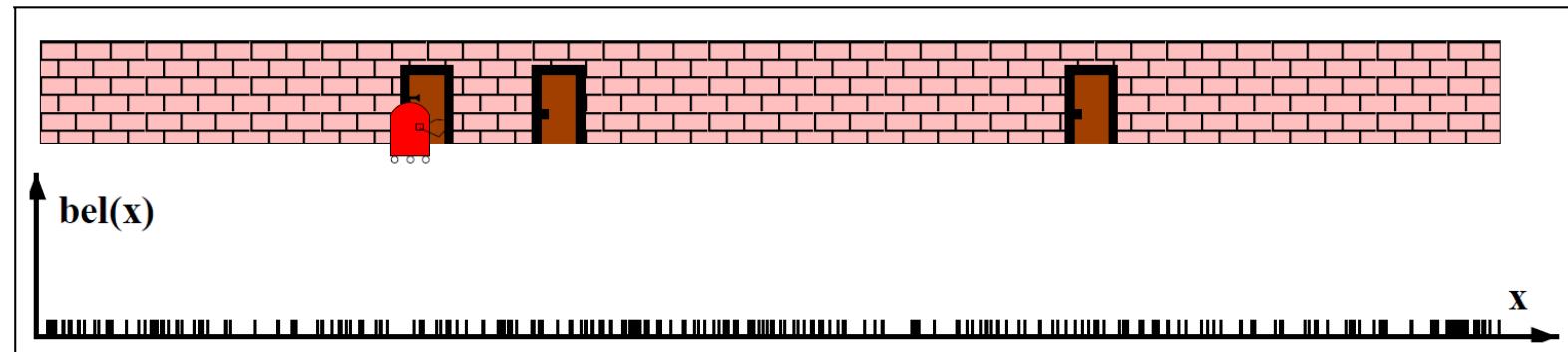
Fonte: *Probabilistic Robotics*



Localização de Monte Carlo

Exemplo 1

- Posição inicial não é inicialmente conhecida
- Estimativa $bel(x_0)$ gerada de maneira aleatória e uniforme



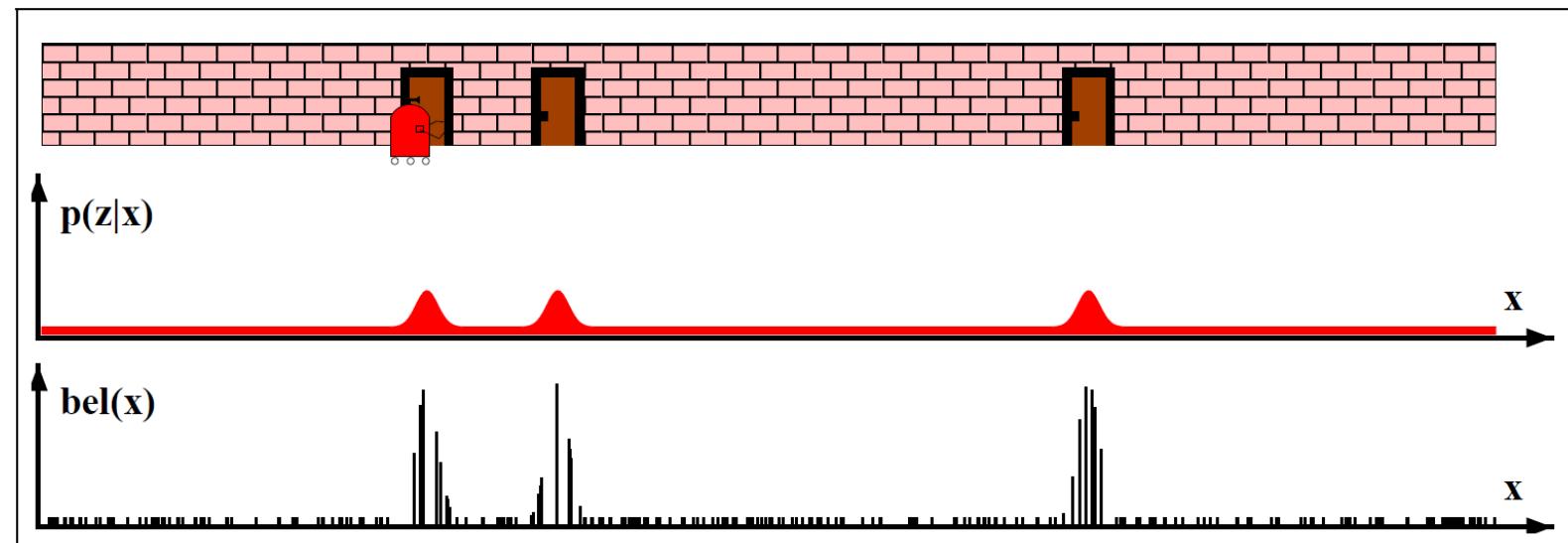
Fonte: *Probabilistic Robotics*



Localização de Monte Carlo

Exemplo 1

- É feita uma leitura inicial e uma porta é observada
- Modelo de observação é usado para ponderar as amostras



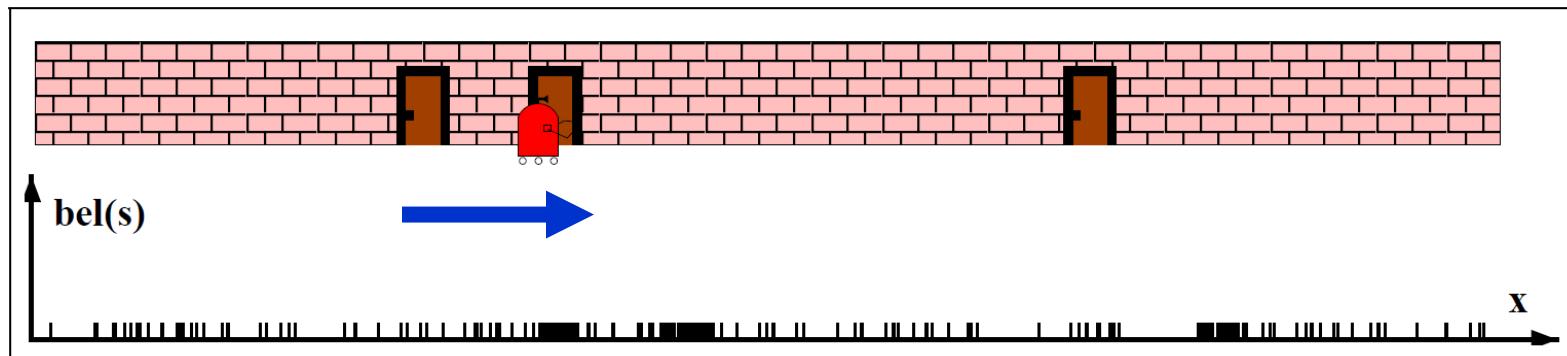
Fonte: *Probabilistic Robotics*



Localização de Monte Carlo

Exemplo 1

- Reamostragem → Nova distribuição das amostras
- Predição → Modelo de transição atualiza os estados



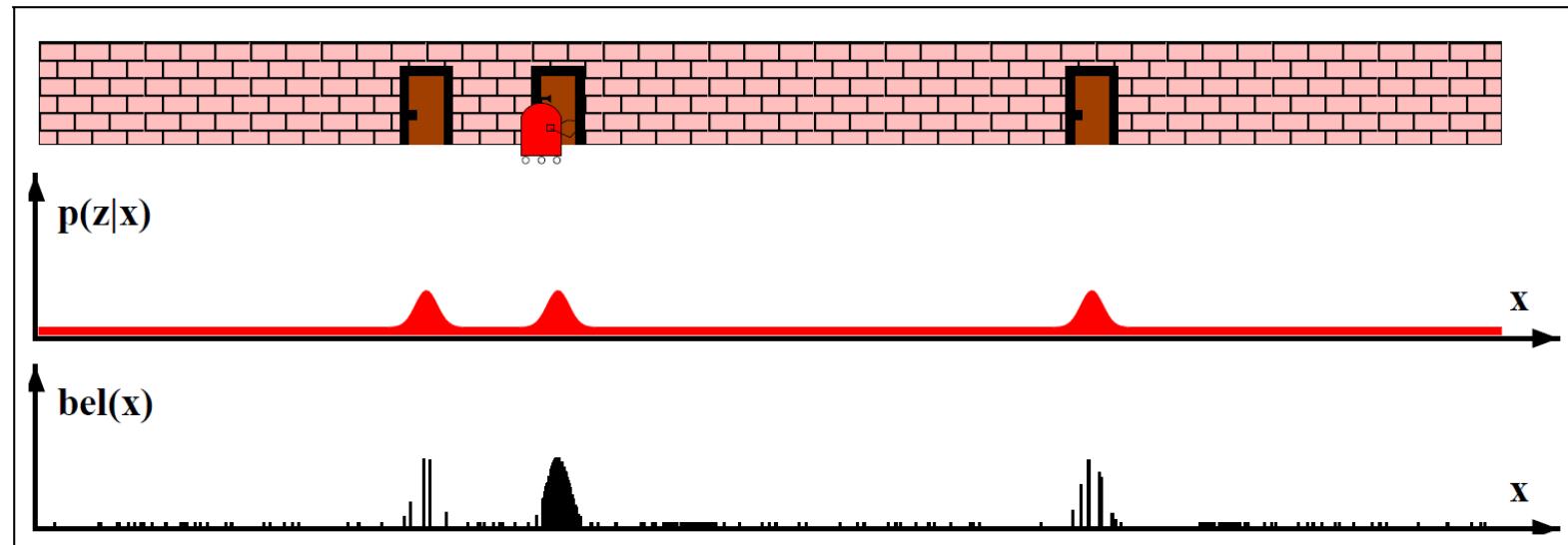
Fonte: *Probabilistic Robotics*



Localização de Monte Carlo

Exemplo 1

- Atualização → Modelo de observação



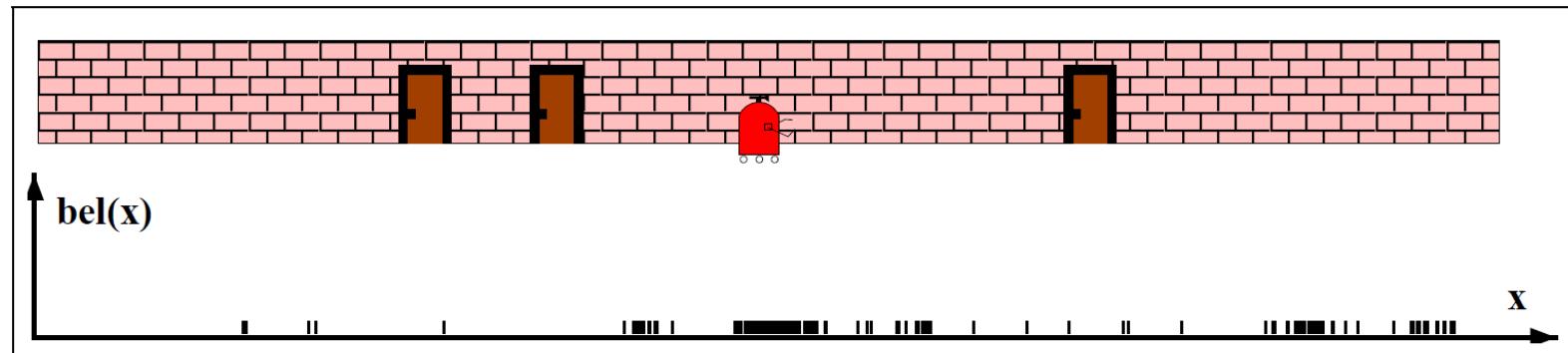
Fonte: *Probabilistic Robotics*



Localização de Monte Carlo

Exemplo 1

- Reamostragem / Predição

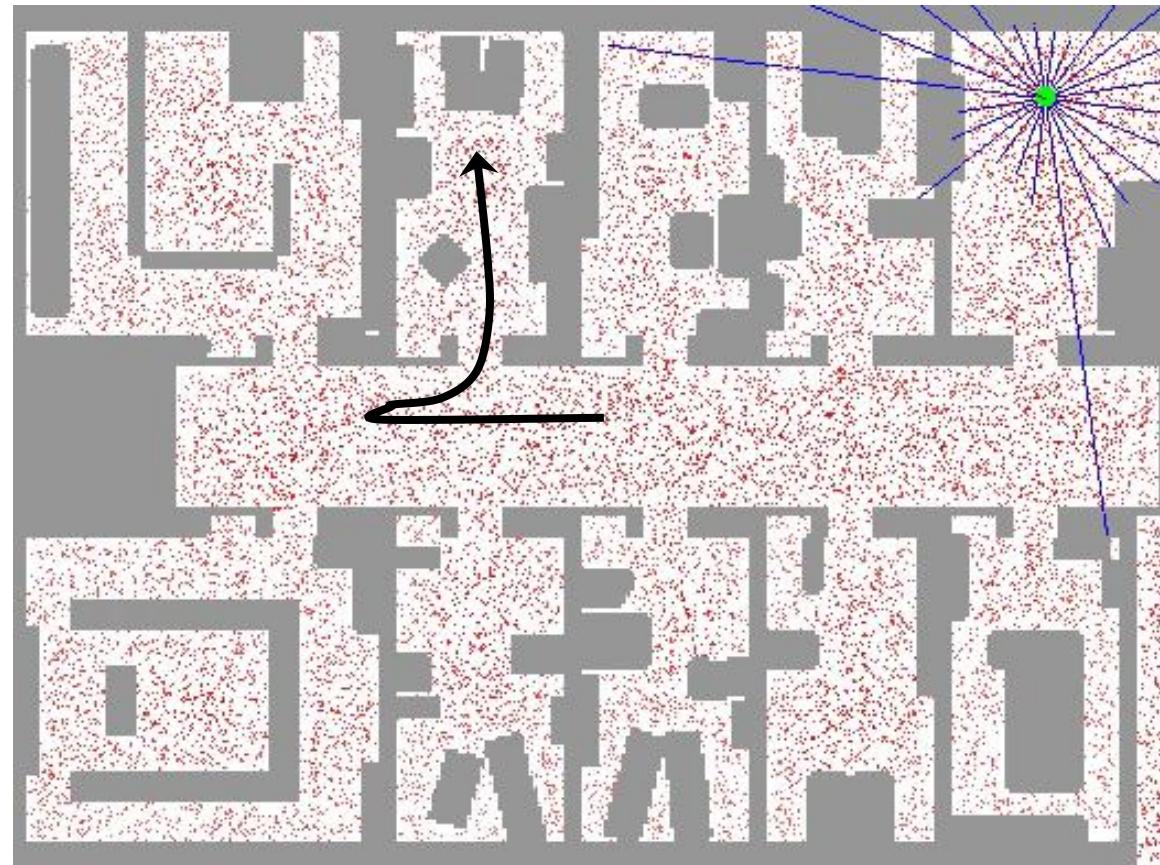


Fonte: *Probabilistic Robotics*



Localização de Monte Carlo

Exemplo 2

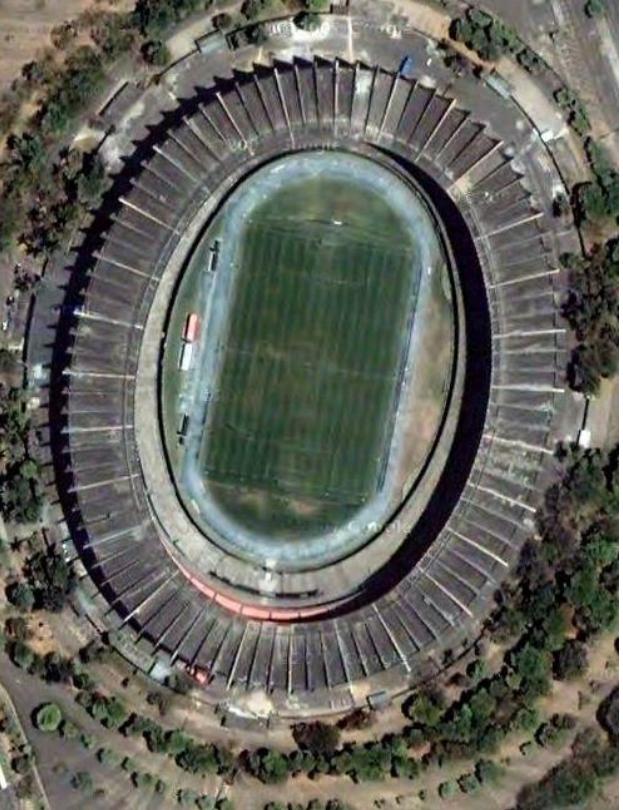


Fonte: <https://robots.stanford.edu/movies/sca80a0.avi>

Localização de Monte Carlo

Exemplo 3

Observação



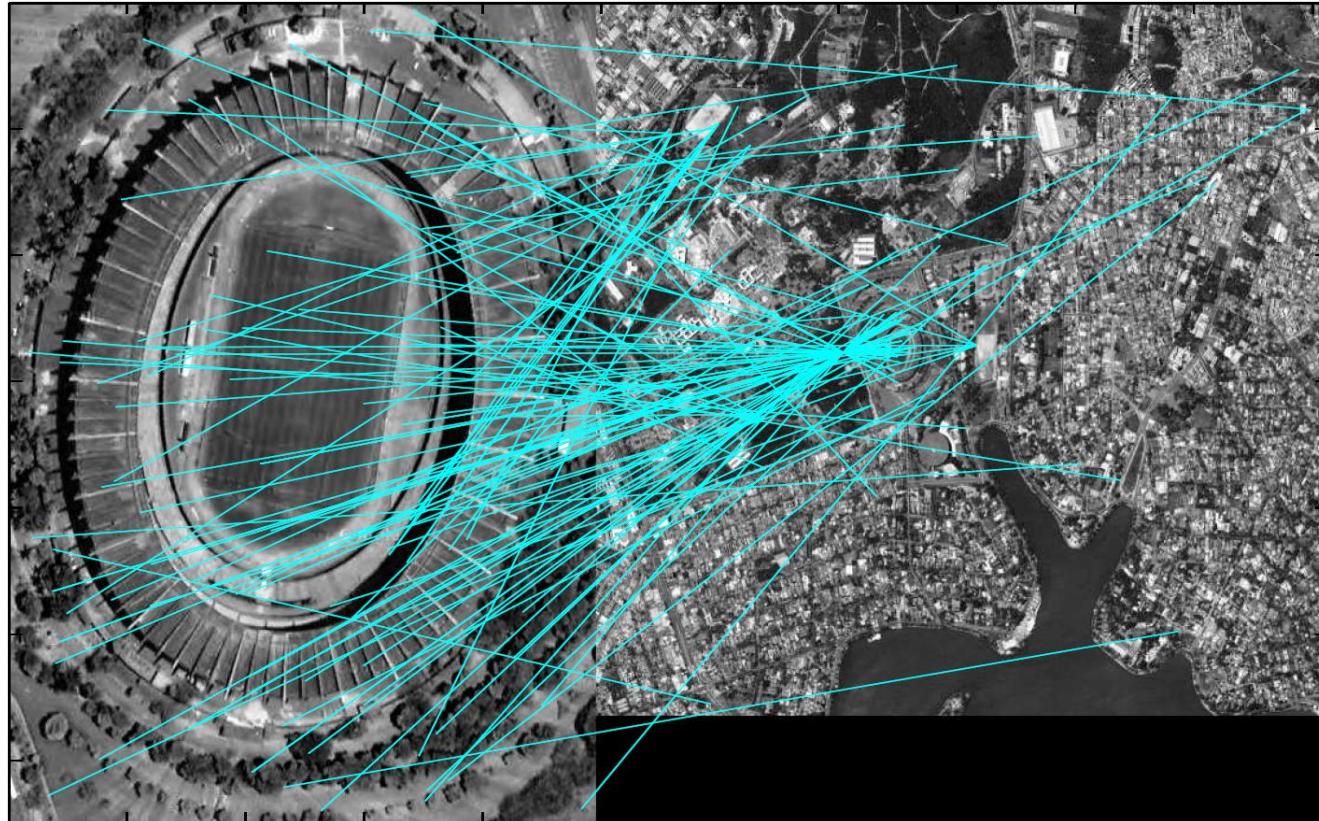
Mapa



D. G. Macharet, A. Alves Neto, V. C. S. Campos, M. F. M. Campos. Computer Vision Based Monte Carlo Localization For Autonomous Aerial Vehicles. In: Anais of SBAI, 2009.

Localização de Monte Carlo

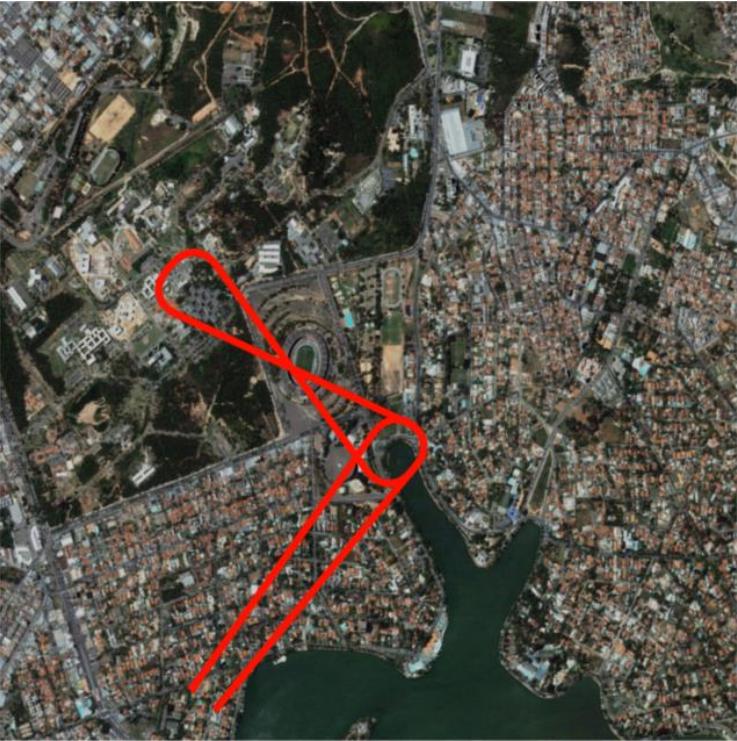
Exemplo 3



D. G. Macharet, A. Alves Neto, V. C. S. Campos, M. F. M. Campos. Computer Vision Based Monte Carlo Localization For Autonomous Aerial Vehicles. In: Anais of SBAI, 2009.

Localização de Monte Carlo

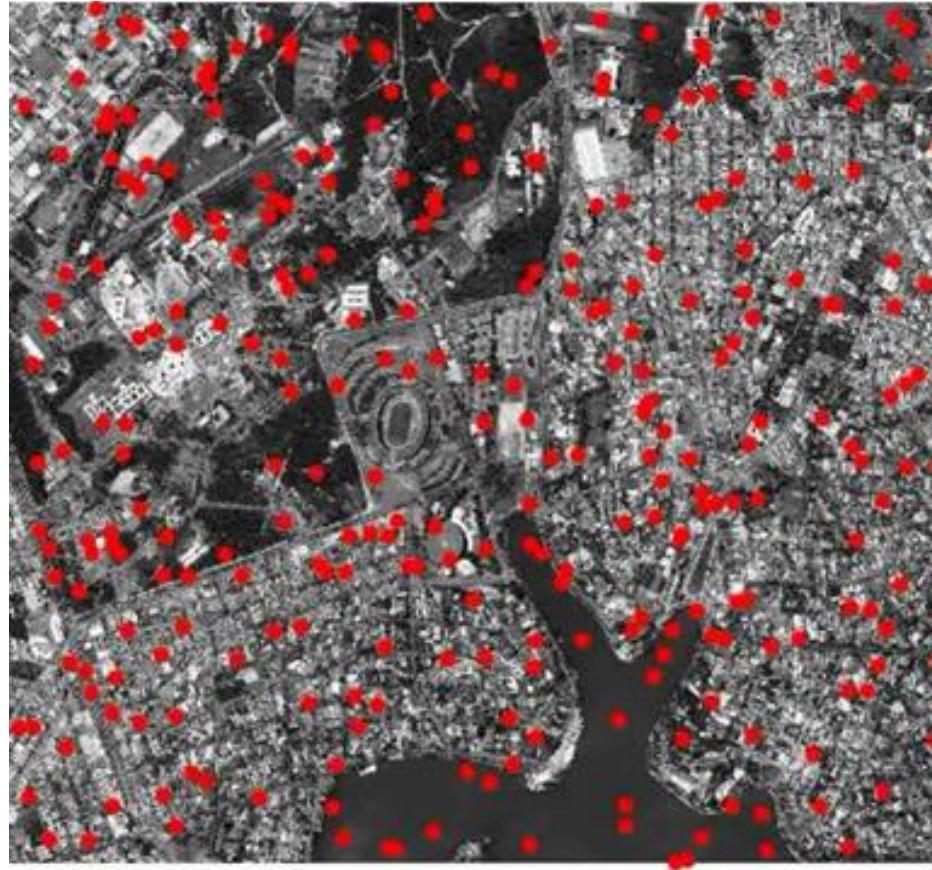
Exemplo 3



D. G. Macharet, A. Alves Neto, V. C. S. Campos, M. F. M. Campos. Computer Vision Based Monte Carlo Localization For Autonomous Aerial Vehicles. In: Anais of SBAI, 2009.

Localização de Monte Carlo

Exemplo 3



https://youtu.be/_DRzAgEPEjY

D. G. Macharet, A. Alves Neto, V. C. S. Campos, M. F. M. Campos. Computer Vision Based Monte Carlo Localization For Autonomous Aerial Vehicles. In: Anais of SBAI, 2009.

Considerações finais

- Solução simples e eficiente
 - Position tracking
 - Localização global
- Distribuição de probabilidades
 - Conjunto de partículas
 - Propagação → Modelo de transição
 - Ponderação → Modelo de observação