

Localização II

Filtro de Kalman
Monte Carlo

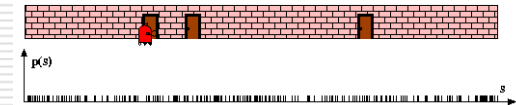
Monte Carlo

Filtro de partículas:

- Representa pdfs através de amostras
- Representa funções não-lineares, não gaussianas e multi-modais
- Eficiente computacionalmente
- Implmenetação simples.

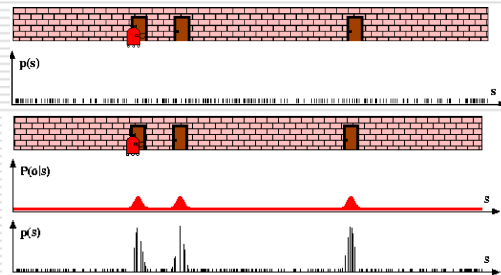
2

Filtro de Partículas

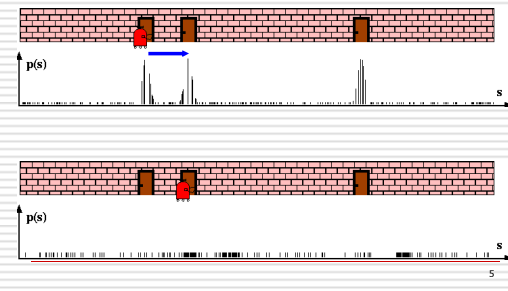


3

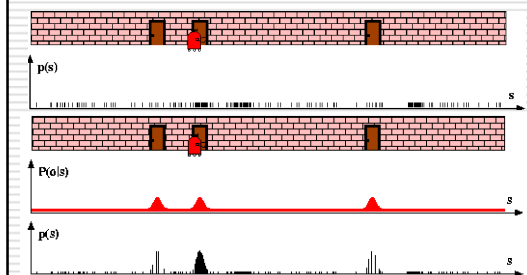
Filtro de Partículas: observação



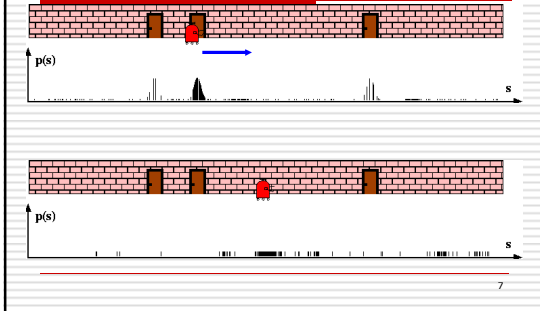
Filtro de Partículas: movimentação



Filtro de Partículas: observação



Filtro de Partículas: movimentação

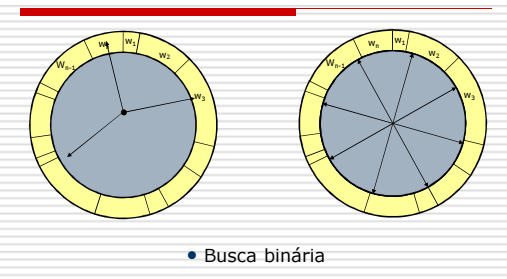


Algoritmo – filtro de partículas

```

1: Algorithm MCL( $\mathcal{X}_{t-1}, u_t, z_t, m$ ):
2:    $\bar{x}_t = \bar{x}_{t-1}$ 
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:      $\bar{x}_t = \bar{x}_t + (x_t^{[m]}, w_t^{[m]})$ 
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  $\mathcal{X}_t$ 
    
```

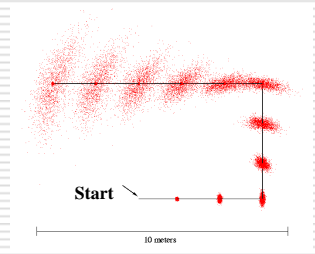
Resorteio



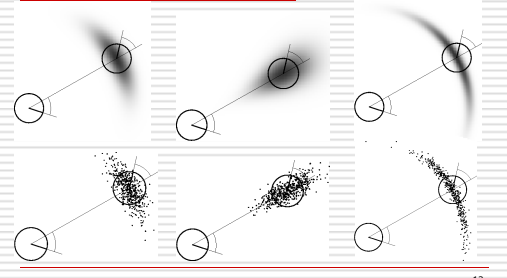
Sample Odometry Motion Model

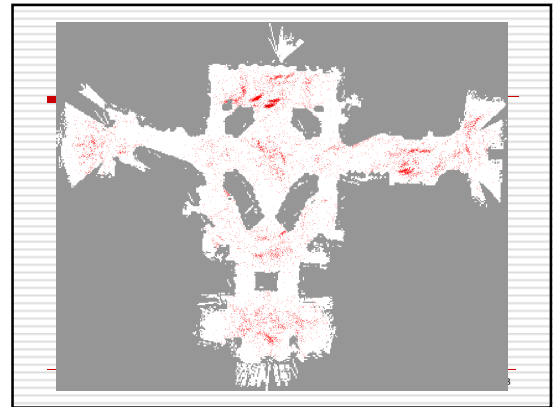
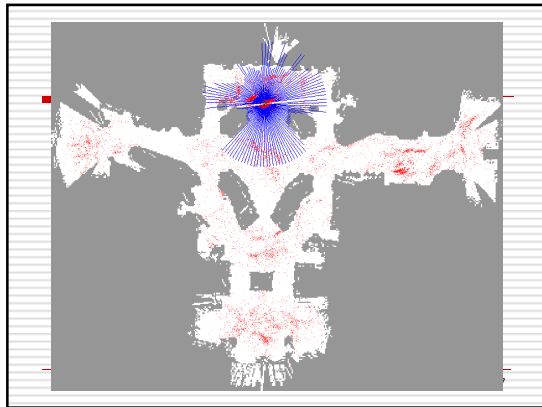
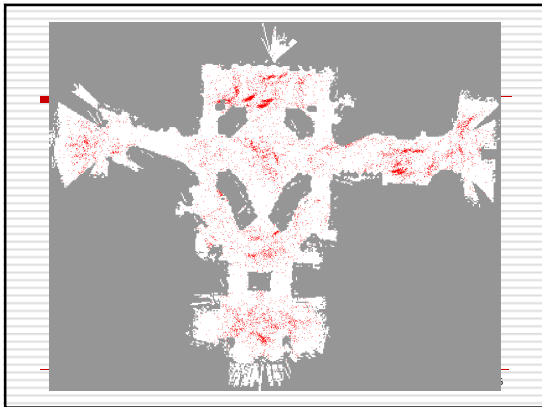
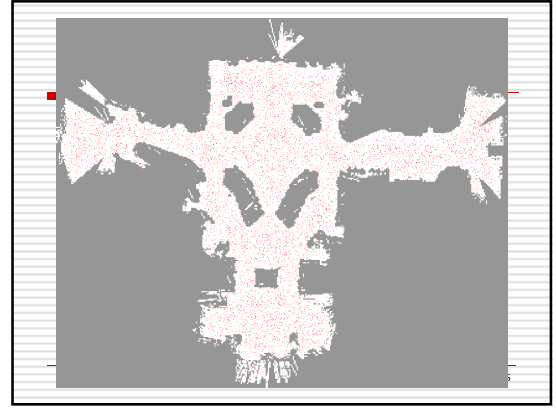
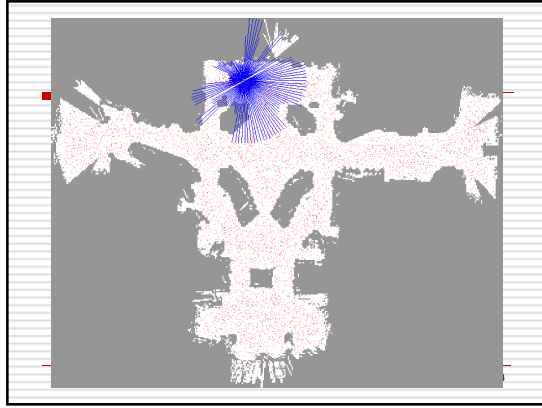
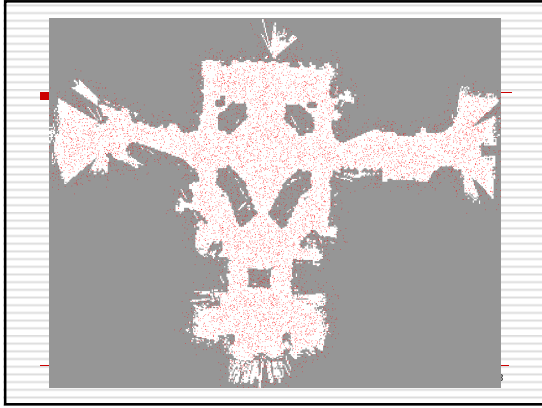
1. Algorithm **sample_motion_model**(u, x):
 - $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle, x = \langle x, y, \theta \rangle$
 - 1. $\hat{\delta}_{rot1} = \delta_{rot1} + \text{sample}(\alpha_1 | \delta_{rot1} | + \alpha_2 \delta_{trans})$
 - 2. $\hat{\delta}_{trans} = \delta_{trans} + \text{sample}(\alpha_1 \delta_{trans} + \alpha_2 (|\delta_{rot1}| + |\delta_{rot2}|))$
 - 3. $\hat{\delta}_{rot2} = \delta_{rot2} + \text{sample}(\alpha_1 |\delta_{rot2}| + \alpha_2 \delta_{trans})$
 - 4. $x' = x + \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1})$
 - 5. $y' = y + \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1})$ **sample_normal_distribution**
 - 6. $\theta' = \theta + \hat{\delta}_{rot1} + \hat{\delta}_{rot2}$
7. Return $\langle x', y', \theta' \rangle$

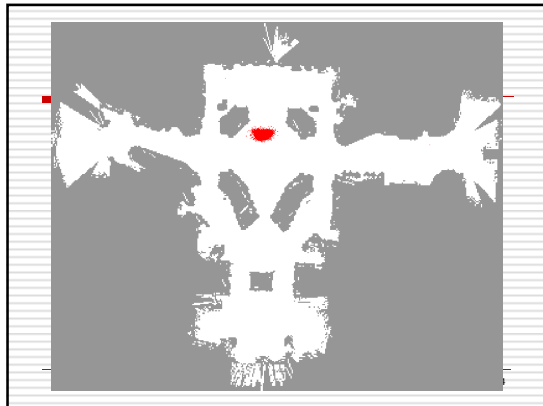
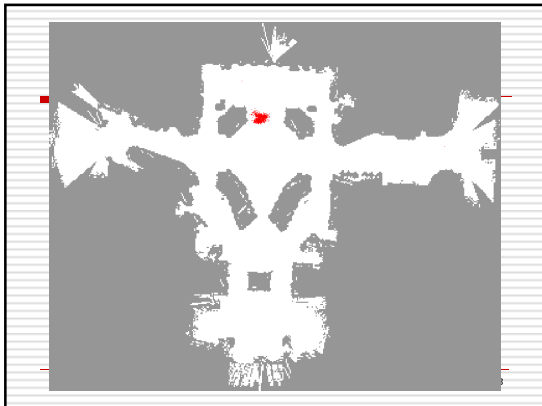
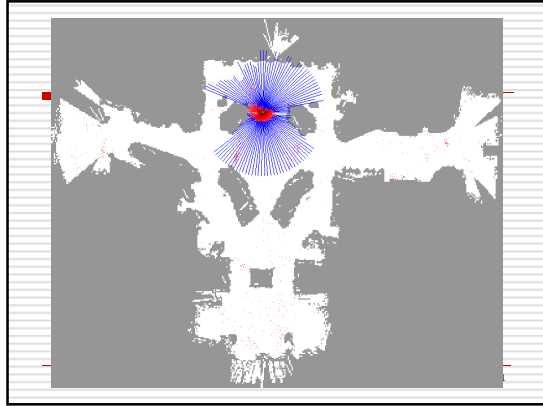
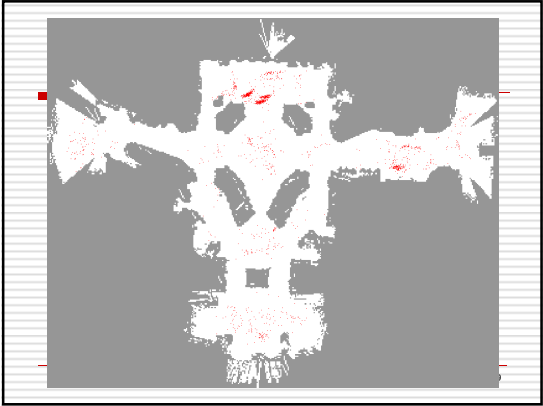
Propagação das partículas

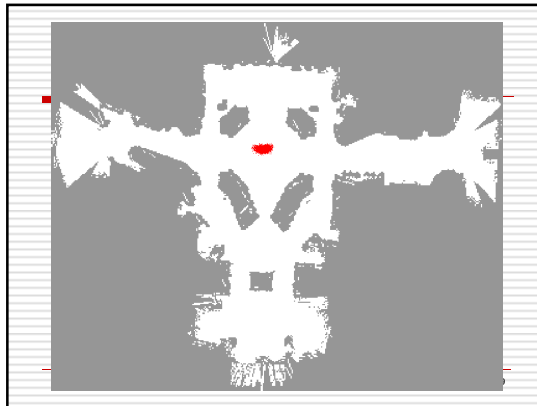
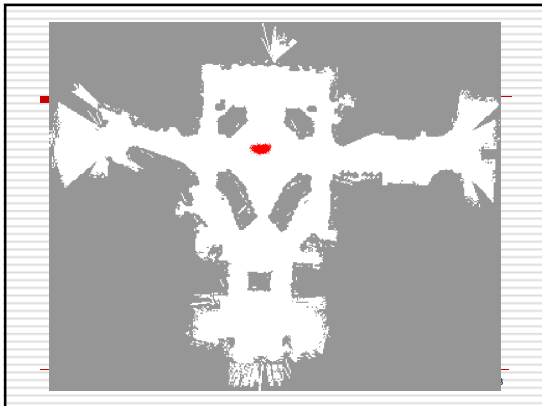
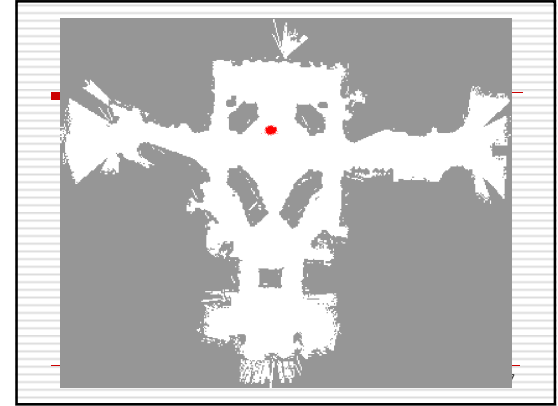
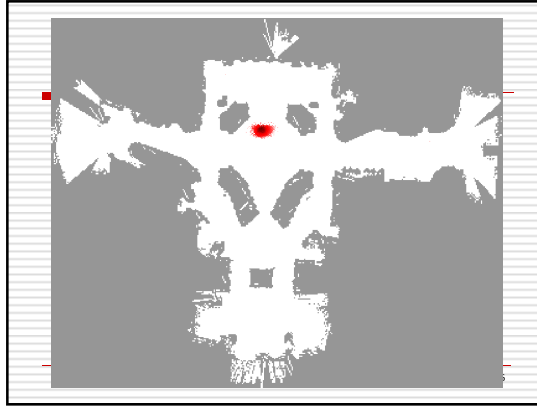
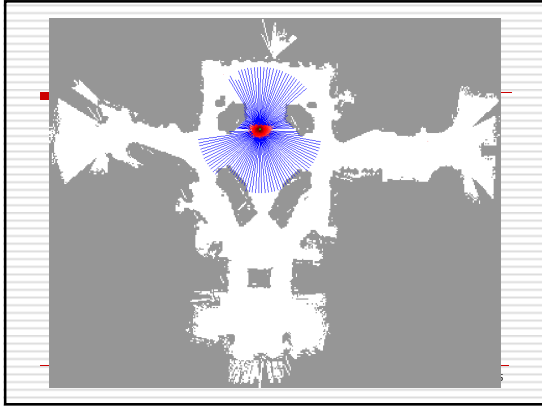


Exemplos





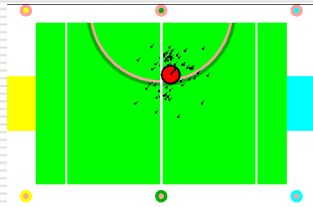




Exemplo de Localização

30

Exemplo de Localização



31

Limitações

- O método de Monte Carlo é capaz de:
 - tracking
 - Busca global
- E quanto ao problema do robô teletransportado (kidnapped robot)?

32

Soluções

- Inserir amostras randômicas (supondo que o robô pode estar em qualquer lugar).
- Inserir amostras proporcionais a verossemelhança de cada partícula (supondo que o robô tem maiores chances de estar nos locais onde as partículas tem maior verossemelhança).

33

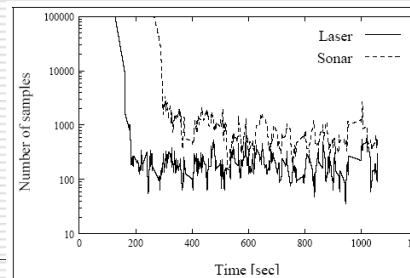
Algoritmo MCL (c/ recover)

```

1: Algorithm Augmented_MCL( $\lambda_{t-1}, u_t, z_t, m$ ):
2:   static  $w_{slow}, w_{fast}$ 
3:    $\hat{\lambda}_t = \hat{\lambda}_{t-1} = \emptyset$ 
4:   for  $m = 1$  to  $M$  do
5:      $x_t^{[m]} = \text{sample\_motion\_model}(u_t, x_{t-1}^{[m]})$ 
6:      $w_t^{[m]} = \text{measurement\_model}(z_t, x_t^{[m]}, m)$ 
7:      $\hat{\lambda}_t = \hat{\lambda}_t + (x_t^{[m]}, w_t^{[m]})$ 
8:      $w_{avg} = w_{avg} + \frac{1}{M} w_t^{[m]}$ 
9:   endfor
10:   $w_{slow} = w_{slow} + \alpha_{slow}(w_{avg} - w_{slow})$ 
11:   $w_{fast} = w_{fast} + \alpha_{fast}(w_{avg} - w_{fast})$ 
12:  for  $m = 1$  to  $M$  do
13:    with probability  $\max(0.0, 1.0 - w_{fast}/w_{slow})$  do
14:      add random pose to  $\hat{\lambda}_t$ 
15:    else
16:      draw  $i \in \{1, \dots, N\}$  with probability  $\propto w_i^{[i]}$ 
17:      add  $x_t^{[i]}$  to  $\hat{\lambda}_t$ 
18:    endwith
19:  endfor
20:  return  $\hat{\lambda}_t$ 
    
```

34

Gráficos – KLD Sampling



35

Localização baseada em mapa

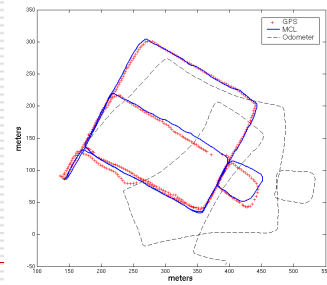
Cada partícula representa a possibilidade do robô estar naquela posição específica. É atribuído um peso a cada partícula proporcional a sua chance de representar a posição do robô. Partículas com peso baixo são excluídas. Dado número suficiente de partículas, é provado que o método converge. Depois que o robô é localizado, a trajetória completa é estimada propagando-se as partículas no sentido contrário.



Nova estimativa peso propagação Estimativa anterior

$$Bel(x_t) = \eta p(a_t | z_t) \int p(x_t | x_{t-1}, a_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

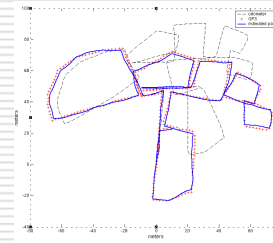
Resultados da localização



37

Localização baseada em GPS

- Cada partícula representa uma possível trajetória completa do robô
- É atribuído um peso a cada partícula de acordo com sua proximidade do GPS.
- Partículas que divergem do GPS recebem peso baixo e são eliminadas.



38

Resumo

- Filtros de partículas são implementações de filtros de Bayes
- As pdfs são representadas por amostras.
- No contexto da localização, as partículas são propagadas de acordo com o odômetro
- No contexto da localização, as partículas recebem seus pesos baseados no modelo de observação .

39