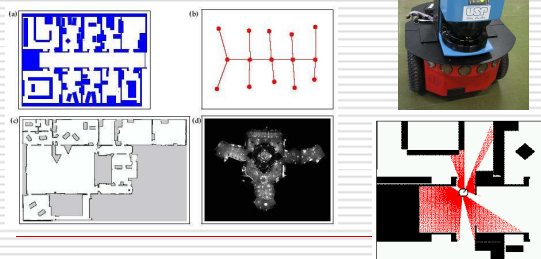


SCE5880
Algoritmos de Estimação para
Robótica Móvel

Localização I

Localização

Estimar a posição de um robô a partir de um mapa e de informações obtidas por sensores.



Localização: classificação

Tipo de problema

- Tracking
 - Posição inicial é conhecida
 - Busca local
 - Representação unimodal é apropriada
-

Localização: classificação

- Localização global
 - Posição inicial não é conhecida
 - Busca global
 - Representação multimodal é apropriada
 - Kidnaped robot (teletransporte)
 - O robô pode ser movido para qualquer posição do ambiente
 - Exige solução robusta, capaz de se recuperar de falhas
-

Tipo de ambiente

- Estático
 - A única variável a ser estimada é a posição do robô
 - Formulação matemática concisa
 - Dinâmico
 - Outras entidades, além do robô, se movem no ambiente
 - Problema complexo
-

Ambiente dinâmico - soluções

- Solução 1: estimar a posição dos objetos móveis
 - Aumento considerável da complexidade do problema
 - Solução 2: identificar e filtrar os dados originados por entidades móveis
 - É necessário descartar informações, aumentando o tempo de convergência e diminuindo a precisão
 - Mantém a formulação de ambiente estático
-

Tipo de localização

- Passiva
 - Módulo de localização apenas observa a operação do robô

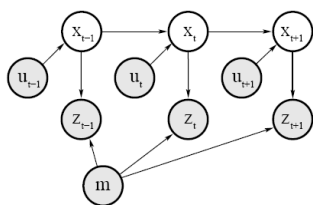
- Ativa
 - Módulo de localização atua no controle do robô, de forma a facilitar a convergência na localização

Número de robôs

- Single robot
 - Todos os dados são obtidos pelo mesmo robô.
 - Não é necessário comunicação

- Multi-robot
 - Robôs podem trocar informações
 - Custo da comunicação
 - Complexidade na integração nos dados
 - Convergência mais rápida e robusta

Localização de Markov

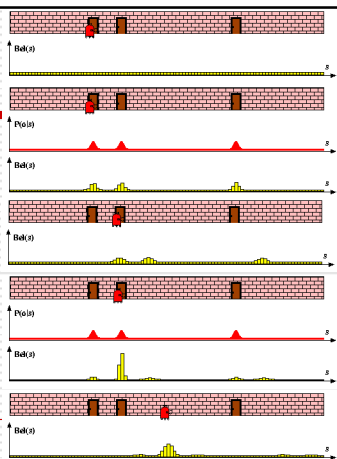


Localização de Markov

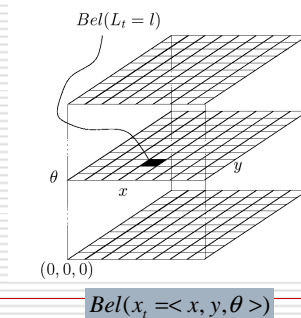
```

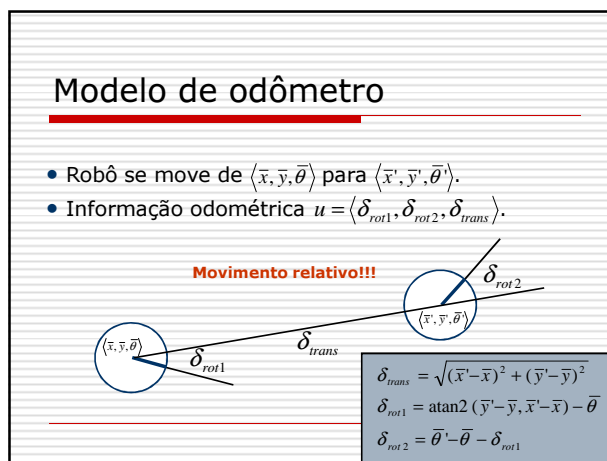
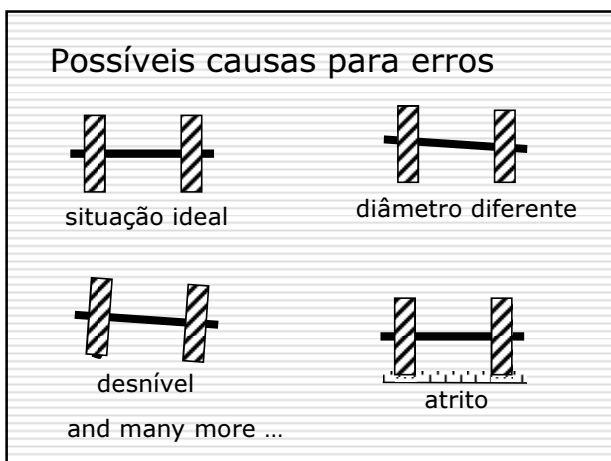
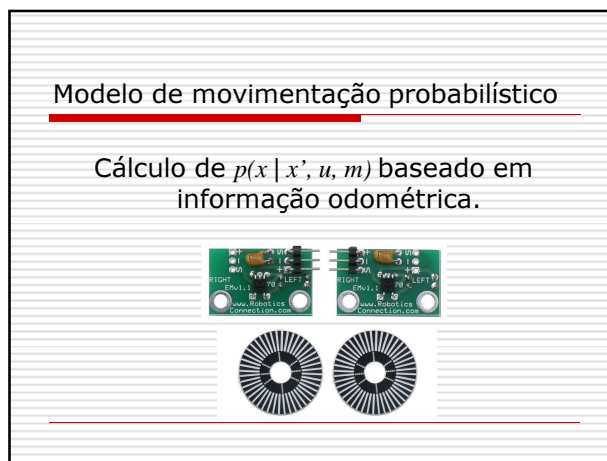
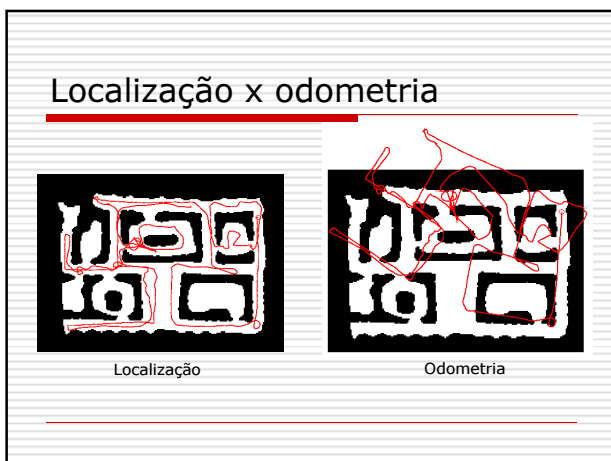
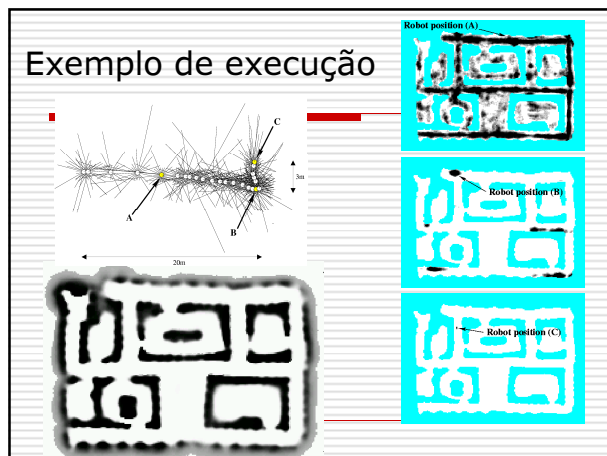
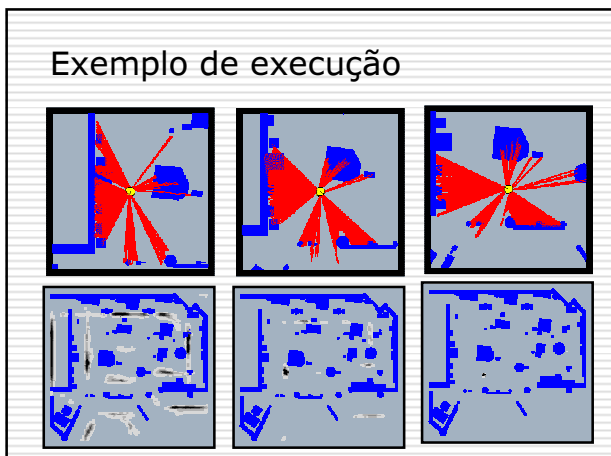
1:  Algorithm Markov_Localization(bel(x_{t-1}), u_t, z_t, m):
2:    for all x_t do
3:      bel(x_t) = ∫ p(x_t | u_t, x_{t-1}, m) bel(x_{t-1}) dx
4:      bel(x_t) = η p(z_t | x_t, m) bel(x_t)
5:    endfor
6:    return bel(x_t)
    
```

Malha de células



Células de ocupação





Calculando x , baseado em x' e u

1. Algorithm **motion_model_odometry**(x, x', u)
2. $\hat{\delta}_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$
3. $\hat{\delta}_{rot1} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$ → odometry values (u)
4. $\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$
5. $\hat{\delta}_{trans} = \sqrt{(x' - x)^2 + (y' - y)^2}$ → values of interest (x, x')
6. $\hat{\delta}_{rot1} = \text{atan2}(y' - y, x' - x) - \bar{\theta}$
7. $\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$
8. $p_1 = \text{prob}(\hat{\delta}_{rot1} - \hat{\delta}_{rot1}, \alpha_1 | \hat{\delta}_{rot1} | + \alpha_2 \hat{\delta}_{trans})$
9. $p_2 = \text{prob}(\hat{\delta}_{trans} - \hat{\delta}_{trans}, \alpha_3 \hat{\delta}_{trans} + \alpha_4 (|\hat{\delta}_{rot1}| + |\hat{\delta}_{rot2}|))$
10. $p_3 = \text{prob}(\hat{\delta}_{rot2} - \hat{\delta}_{rot2}, \alpha_1 | \hat{\delta}_{rot2} | + \alpha_2 \hat{\delta}_{trans})$
11. return $p_1 \cdot p_2 \cdot p_3$

Aproximação de distribuições probabilísticas

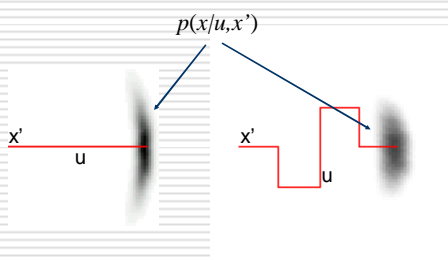
```

1: Algorithm prob_normal_distribution(a, b):
2:   return  $\frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2} \frac{a^2}{b}}$ 
3:
4: Algorithm prob_triangular_distribution(a, b):
5:   if |a| >  $\sqrt{6b}$ 
6:     return 0
7:   else
8:     return  $\frac{\sqrt{6b} - |a|}{6b}$ 
    
```

Probabilidade de a , dados média 0 e desvio padrão b

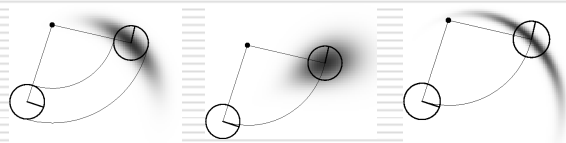
Aplicação

Aplicar o modelo repetidamente para movimentos curtos



Modelo de odômetro

Exemplos:



Utilização do mapa

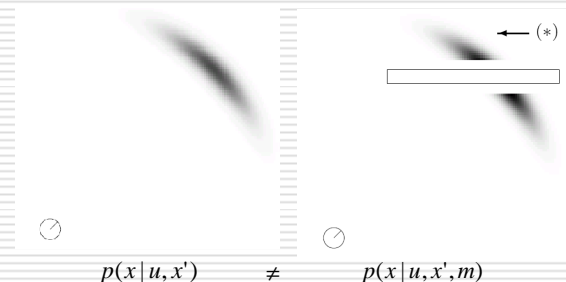
$$p(x | x', u) \neq p(x | x', u, m)$$

$$p(x | x', u, m) = \eta \cdot p(x | x', u) \cdot p(x | m)$$

$$p(x | m) = 1 \text{ se } m \text{ estiver livre}$$

$$p(x | m) = 0 \text{ se } m \text{ estiver ocupado}$$

Utilização do mapa

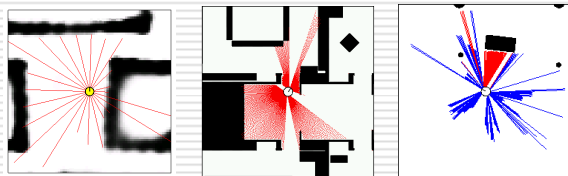


Modelo de percepção

Problema: calcular $p(z|x)$

Calcular $p(z|x, m)$

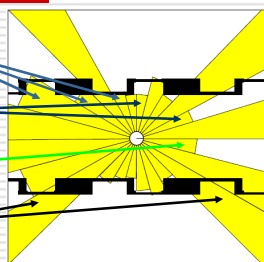
Sensores de distância



$$P(z|x, m) = \prod_{k=1}^K P(z_k|x, m)$$

Leituras de sensores de distância

1. Raios refletidos em obstáculos
2. Raios refletidos por pessoas/obst. móveis
3. Erros aleatórios
4. Distância máxima



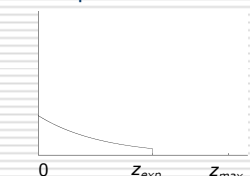
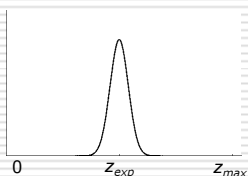
Modelo de raio-x

- Calcular a distância que deveria ser medida baseada no mapa e na posição do robô.
- Comparar com a distância lida do sensor

Modelo de raio-x

Measurement noise

Unexpected obstacles



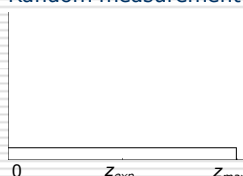
$$P_{hit}(z|x, m) = \eta \frac{1}{\sqrt{2\pi}b} e^{-\frac{1}{2} \left(\frac{z-z_{exp}}{b}\right)^2}$$

$$P_{unexp}(z|x, m) = \begin{cases} \eta \lambda e^{-\lambda z} & z < z_{exp} \\ 0 & otherwise \end{cases}$$

Modelo de raio-x

Random measurement

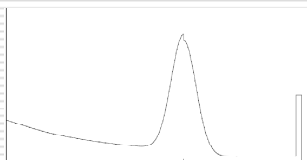
Max range



$$P_{rand}(z|x, m) = \eta \frac{1}{z_{max}}$$

$$P_{max}(z|x, m) = \eta \frac{1}{z_{small}}$$

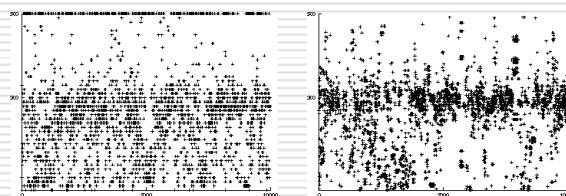
pdf resultante



$$P(z|x,m) = \begin{pmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{uncsp}} \\ \alpha_{\text{max}} \\ \alpha_{\text{rand}} \end{pmatrix}^T \begin{pmatrix} P_{\text{hit}}(z|x,m) \\ P_{\text{uncsp}}(z|x,m) \\ P_{\text{max}}(z|x,m) \\ P_{\text{rand}}(z|x,m) \end{pmatrix}$$

Dados obtidos de sensores

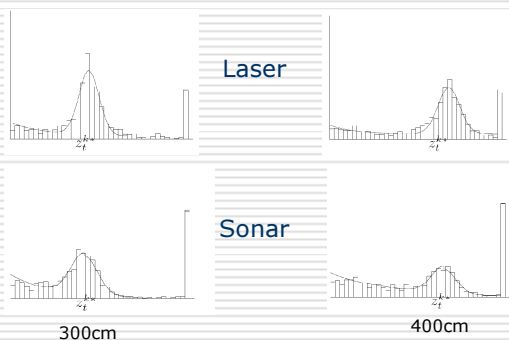
Dados obtidos em uma distância de 3m.



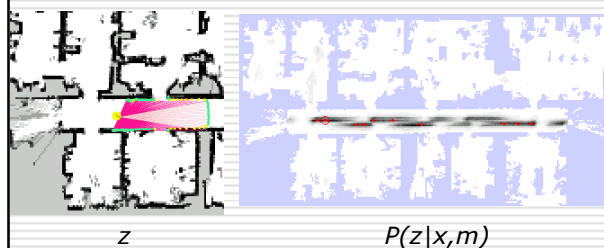
Sonar

Laser

Aproximação



Exemplo



z

$P(z|x,m)$

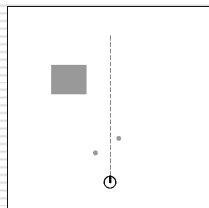
Modelo de raio-x

- Alto custo computacional
- Não tolera pequenos erros de localização

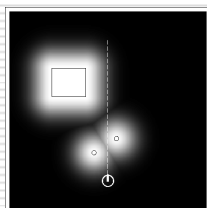
Modelo de mapa de probabilidade

- Pré-calcula a probabilidade de cada posição do mapa
- Compara o ponto obtido com os dados do sensor com o mapa de probabilidade e obtém $p(z|x,m)$ diretamente

Exemplos



Mapa

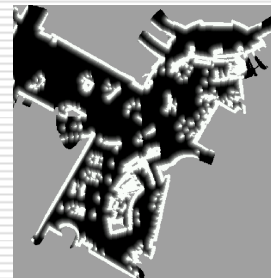


Mapa probabilidade

Exemplo



Occupancy grid map



Likelihood field

Mapa de likelihood

- Eficiente computacionalmente
- Comporta pequenos erros de localização
- Ignora a física do mapa e do sensor de distância

Landmarks

- Sensores ativos (radio, GPS, laser)
- Sensores passivos (cameras)
- Triangulação
 - Distância
 - Ângulo

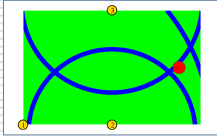
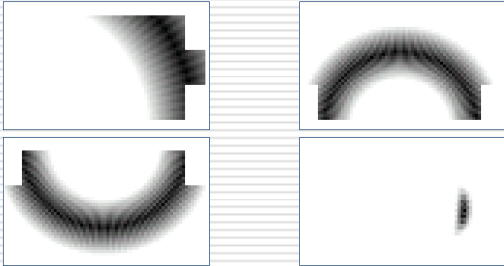
Distance and Bearing



Modelo probabilístico

1. Algorithm `landmark_detection_model(z,x,m)`:
 $z = \langle i, d, \alpha \rangle, x = \langle x, y, \theta \rangle$
2. $\hat{d} = \sqrt{(m_x(i) - x)^2 + (m_y(i) - y)^2}$
3. $\hat{\alpha} = \text{atan2}(m_y(i) - y, m_x(i) - x) - \theta$
4. $p_{\text{det}} = \text{prob}(\hat{d} - d, \epsilon_d) \cdot \text{prob}(\hat{\alpha} - \alpha, \epsilon_\alpha)$
5. Return p_{det}

Distributions



Próxima aula

Localização II

- Monte Carlo
- Filtro de Kalman
