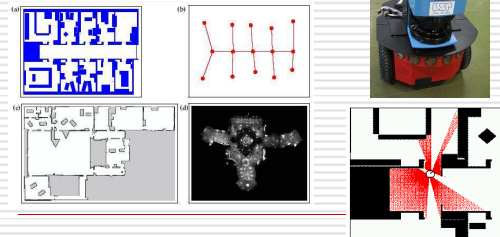


SCE5880  
Algoritmos de Estimação para  
Robótica Móvel

## 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

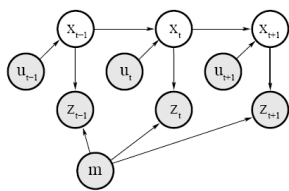
### Tipo de localização

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

### Localização de Markov

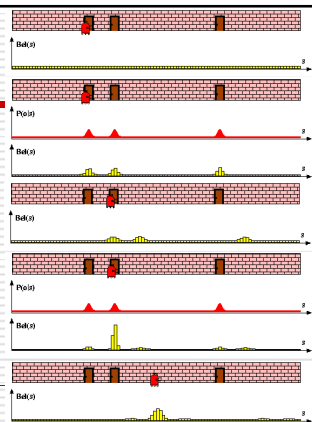


### Localização de Markov

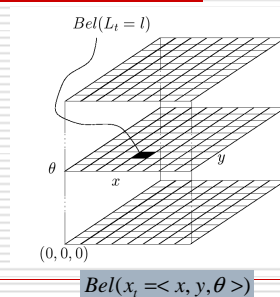
```

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

### Malha de células



### Células de ocupação



### Exemplo de execução

### Exemplo de execução

### Localização x odometria

### Modelo de movimentação probabilístico

Cálculo de  $p(x | x', u, m)$  baseado em informação odométrica.

### Possíveis causas para erros

and many more ...

### Modelo de odômetro

- Robô se move de  $\langle \bar{x}, \bar{y}, \bar{\theta} \rangle$  para  $\langle \bar{x}', \bar{y}', \bar{\theta}' \rangle$ .
- Informação odométrica  $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$ .

**Movimento relativo!!!**

$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$

$$\delta_{rot1} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$

### Calculando $x$ , baseado em $x'$ e $u$

1. Algorithm **motion\_model\_odometry**( $x, x', u$ )
2.  $\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$
3.  $\delta_{rot1} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$   $\rightarrow$  odometry values ( $u$ )
4.  $\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$
5.  $\hat{\delta}_{trans} = \sqrt{(x' - x)^2 + (y' - y)^2}$
6.  $\hat{\delta}_{rot1} = \text{atan2}(y' - y, x' - x) - \bar{\theta}$   $\rightarrow$  values of interest ( $x, x'$ )
7.  $\hat{\delta}_{rot2} = \bar{\theta}' - \bar{\theta} - \hat{\delta}_{rot1}$
8.  $p_1 = \text{prob}(\hat{\delta}_{rot1} - \hat{\delta}_{rot1} \cdot \alpha_1 | \hat{\delta}_{rot1} | + \alpha_2 \hat{\delta}_{trans})$
9.  $p_2 = \text{prob}(\hat{\delta}_{trans} - \hat{\delta}_{trans} \cdot \alpha_3 \hat{\delta}_{trans} + \alpha_4 (|\hat{\delta}_{rot1}| + |\hat{\delta}_{rot2}|))$
10.  $p_3 = \text{prob}(\hat{\delta}_{rot2} - \hat{\delta}_{rot2} \cdot \alpha_1 | \hat{\delta}_{rot2} | + \alpha_5 \hat{\delta}_{trans})$
11. return  $p_1 \cdot p_2 \cdot p_3$

### Aproximação de distribuições probabilísticas

```

1: Algorithm prob_normal_distribution(a, b):
2:   return  $\frac{1}{\sqrt{2\pi}b}$   $e^{-\frac{1}{2} \frac{a^2}{b^2}}$ 

3: Algorithm prob_triangular_distribution(a, b):
4:   if |a| >  $\sqrt{6b}$ 
5:     return 0
6:   else
7:     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

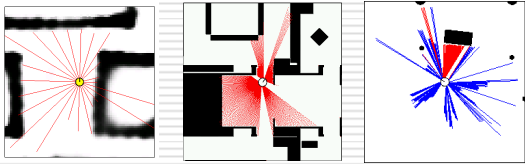
$p(x | u, x') \neq p(x | u, x', m)$

### Modelo de percepção

Problema: calcular  $p(z|x)$

**Calcular  $p(z | x, m)$**

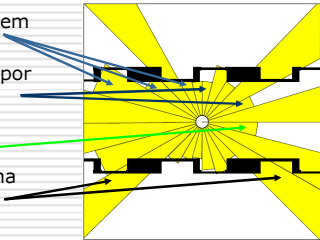
### Sensores de distância



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

### Leituras de sensores de distância

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

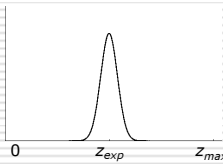


### Modelo de raio-x

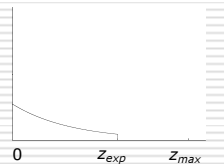
- Calcular a distância que deveria ser medida baseada no mapa e na posição do robô.
- Comparar com a distância lida do sensor

### Modelo de raio-x

Measurement noise



Unexpected obstacles




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


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

### Modelo de raio-x

Random measurement

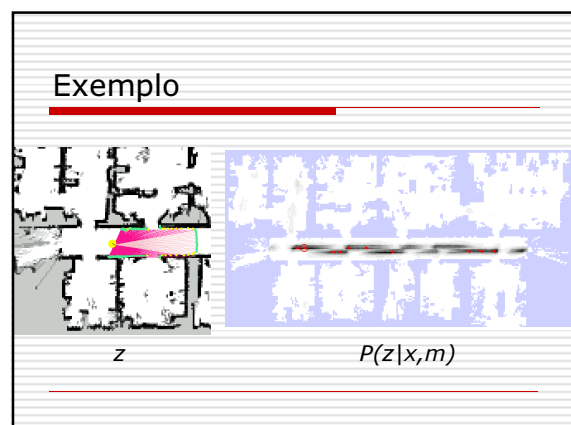
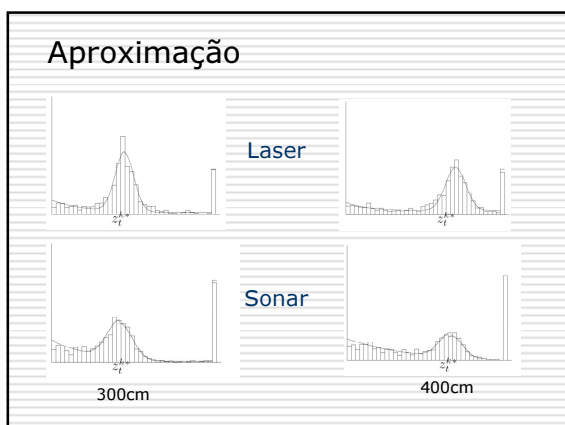
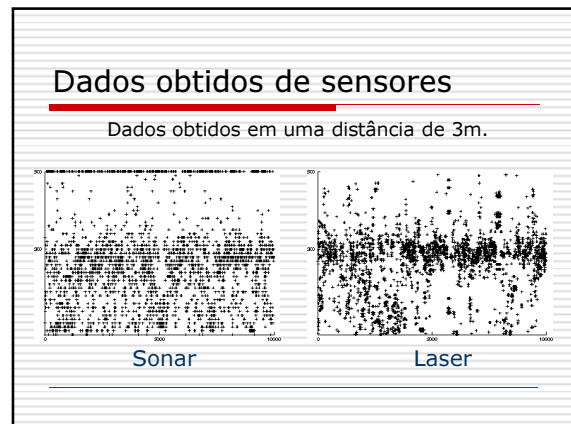
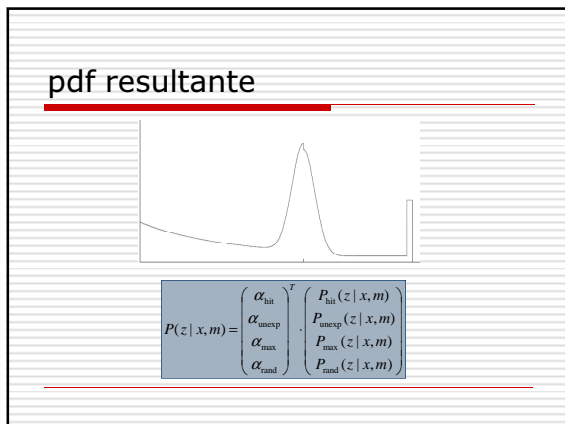


Max range



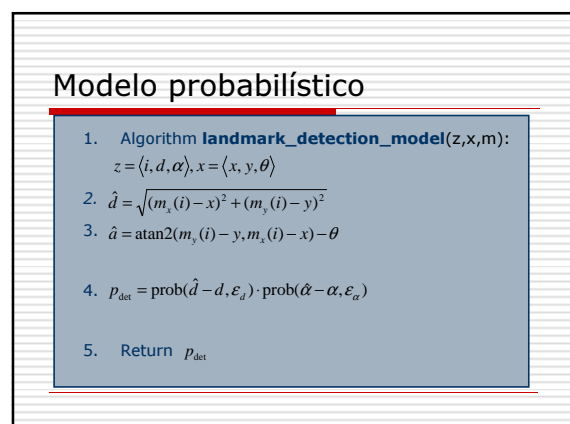
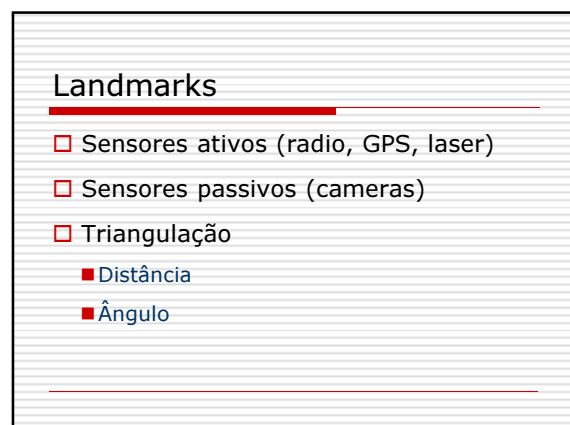
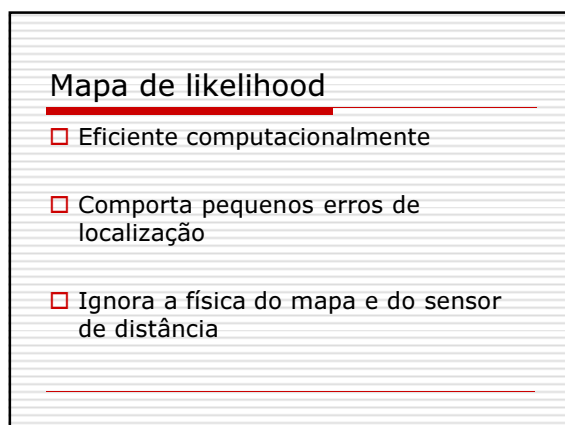
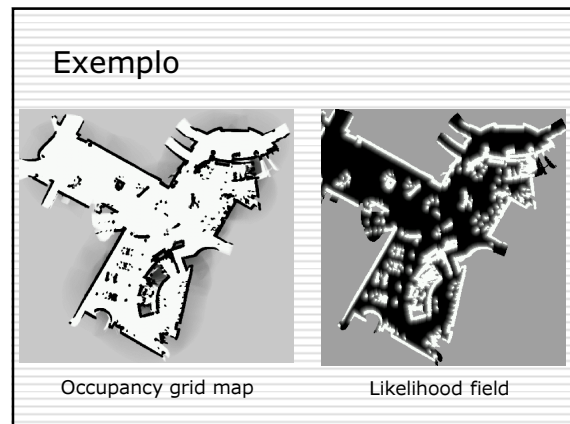
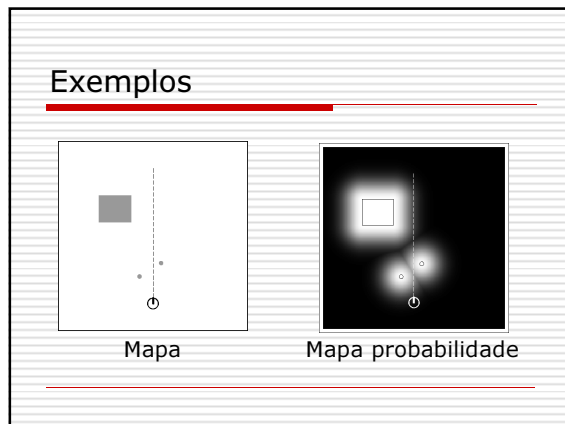
$$P_{\text{rand}}(z | x, m) = \eta \frac{1}{z_{\text{max}}}$$

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

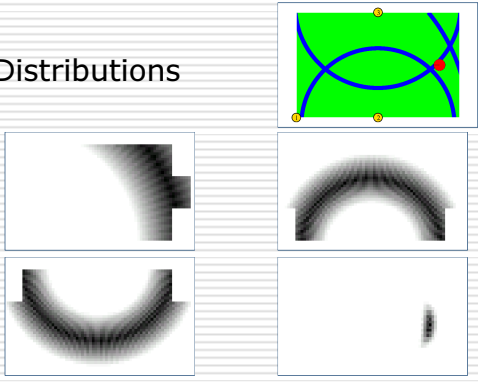


- ### Modelo de raio-x
- Alto custo computacional
  - Não tolera pequenos erros de localização

- ### 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



Distributions



The 'Distributions' section contains five sub-plots. The top-right plot is a green square with two overlapping blue circles and four orange dots at their intersections. The other four plots are grayscale images: a quarter-circle, a semi-circle, a full circle, and a small vertical bar.

---

**Próxima aula**

Localização II

- Monte Carlo
- Filtro de Kalman

---