UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Robótica Móvel

Mapeamento – Occupancy Grid

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

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

VERLab
Visão e Robótica

Introdução

- Mapeamento é uma tarefa fundamental
  - Robôs com pouca/nenhuma supervisão
- Aplicações
  - Localização
  - Planejamento de caminhos
  - Tomada de decisão
  - ...

Mapeamento
Problemas

- Incertezas
  - Extrair informação de sensores ruidosos
  - Erros provenientes da atuação (localização)
- Ambiguidade
  - Estabelecer correspondência entre leituras
  - Loop-Closure (re-identificar local já visitado)
- Solução: representação baseada em probabilidades!

Occupancy Grid

- Algoritmo probabilístico
  - Proposto por Moravec e Elfes (1985)
- Principais considerações
  - Localização do robô é <u>conhecida</u>
  - Ambiente <u>estático</u> (mapa é "adaptável")
- Utiliza uma representação em grid
- Facilita a integração de diferentes sensores

Occupancy Grid

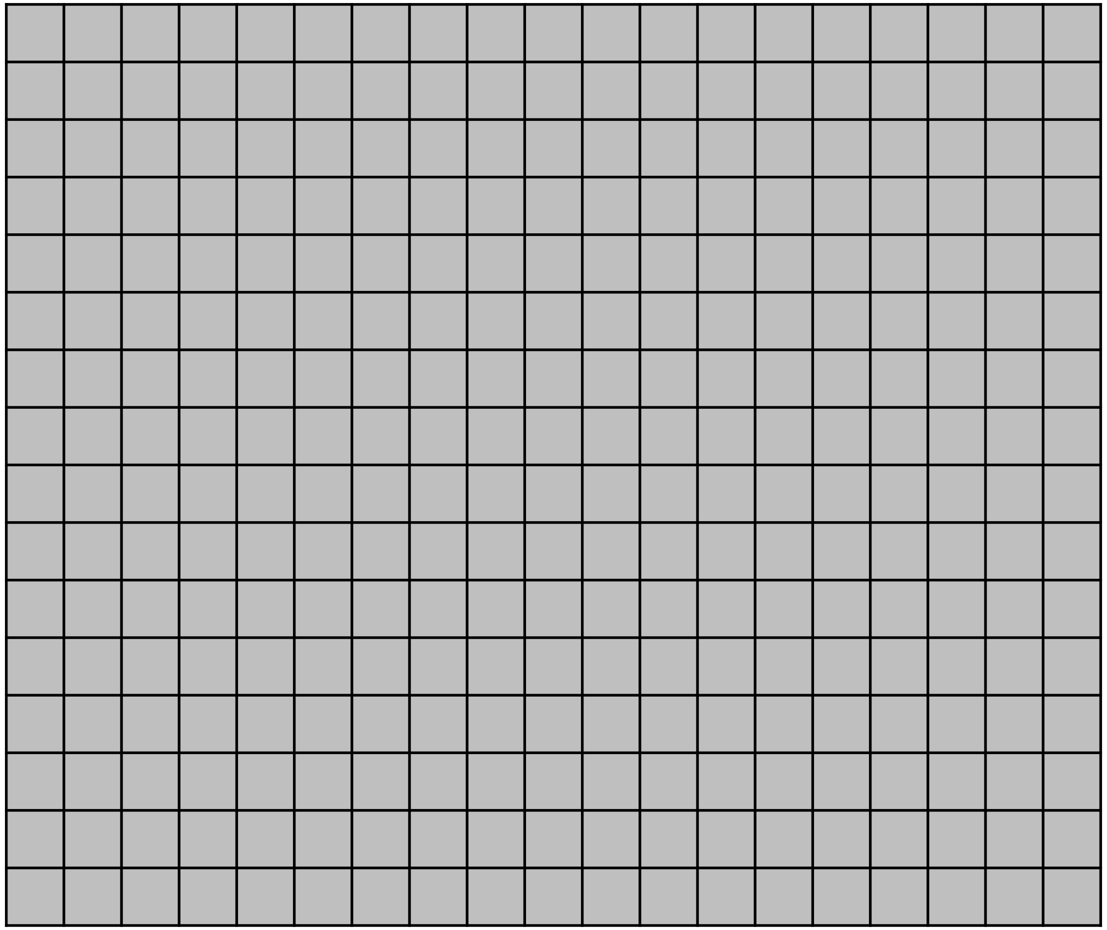
- Cada célula i no grid é uma variável aleatória binária que modela a ocupação (obstáculo) na posição associada à célula
  - $p(m_i) = 0$ (vazio)
  - $p(m_i) = 1$ (ocupado)

A notação compacta $p(m_i)$ é usada para representar a probabilidade da célula estar **ocupada**, ou seja, $p(m_i = 1)$.

- Inicialmente, células com $p(m_i) = 0{,}5$ (desconhecido)
- Probabilidades nas células são <u>independentes</u>
- Mapa é a união das células: $m = \sum_i m_i$ ➔ $p(m) = \prod_i p(m_i)$

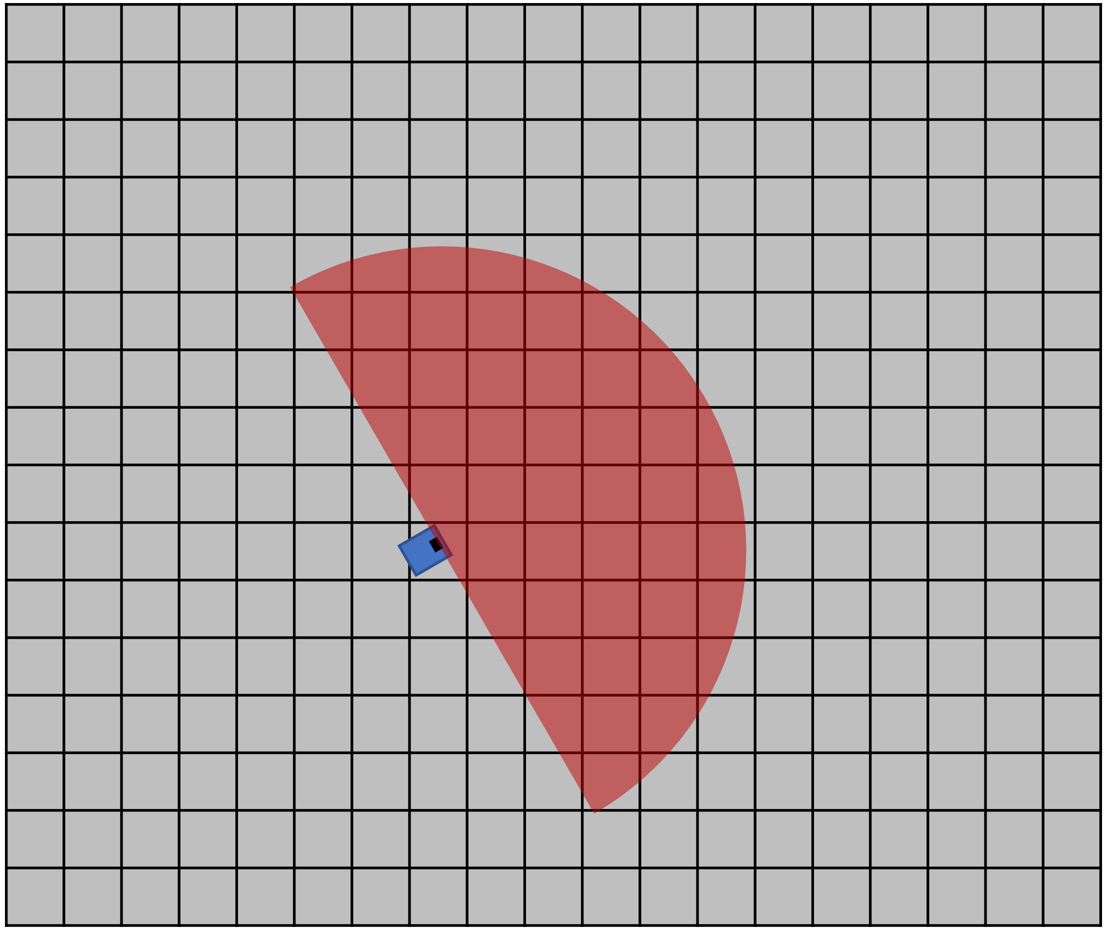
Occupancy Grid

Representação em tons de cinza facilita a visualização.

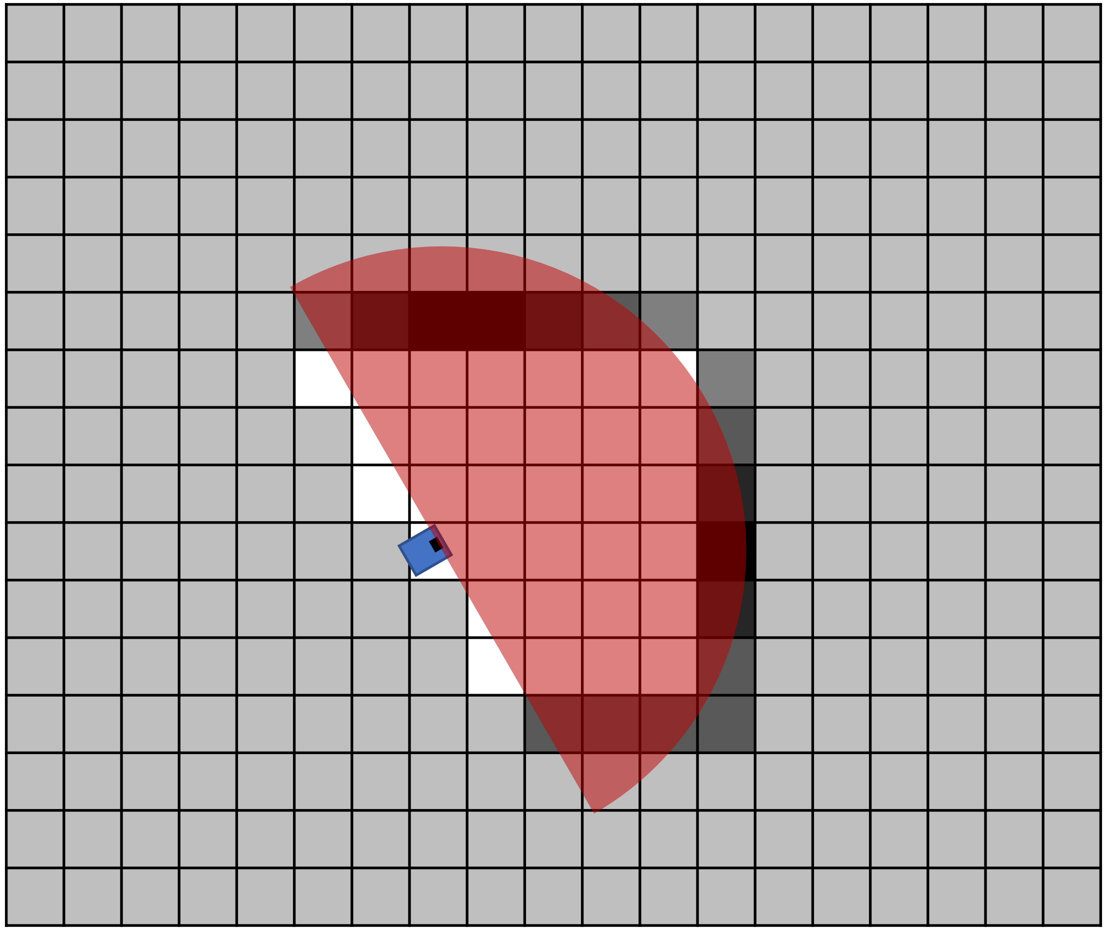


Occupancy Grid

Assume-se que posição do robô no ambiente é conhecida.



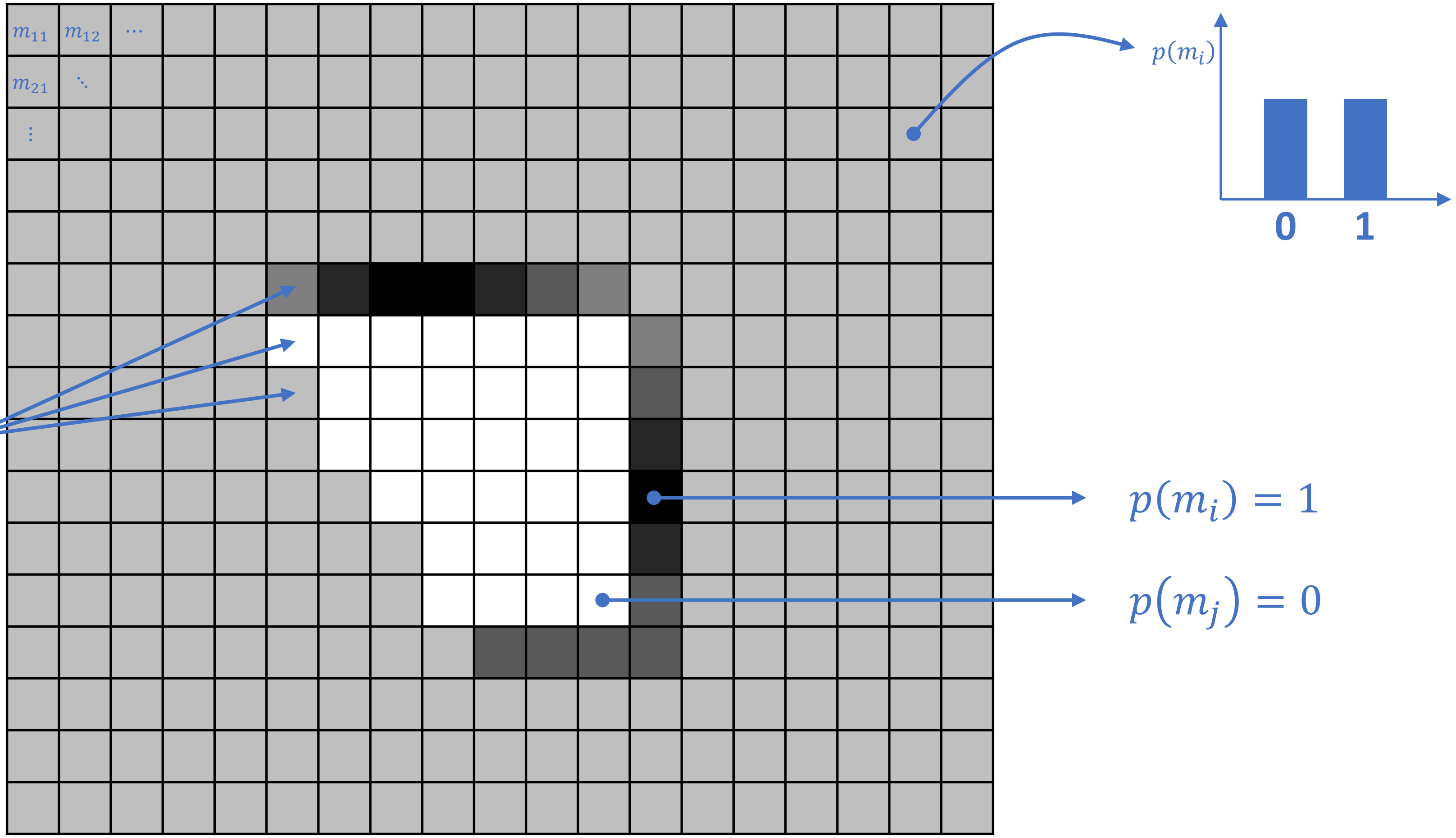
Occupancy Grid



Considerando a incerteza associada à percepção, as celulas são preenchidas com as probabilidades de estarem ocupadas.

Occupancy Grid

Assumimos que não existe correlação entre as células!



Occupancy Grid

- O objetivo é determinar a probabilidade a posteriori

$$p(m \mid z_{1:t}, x_{1:t})$$

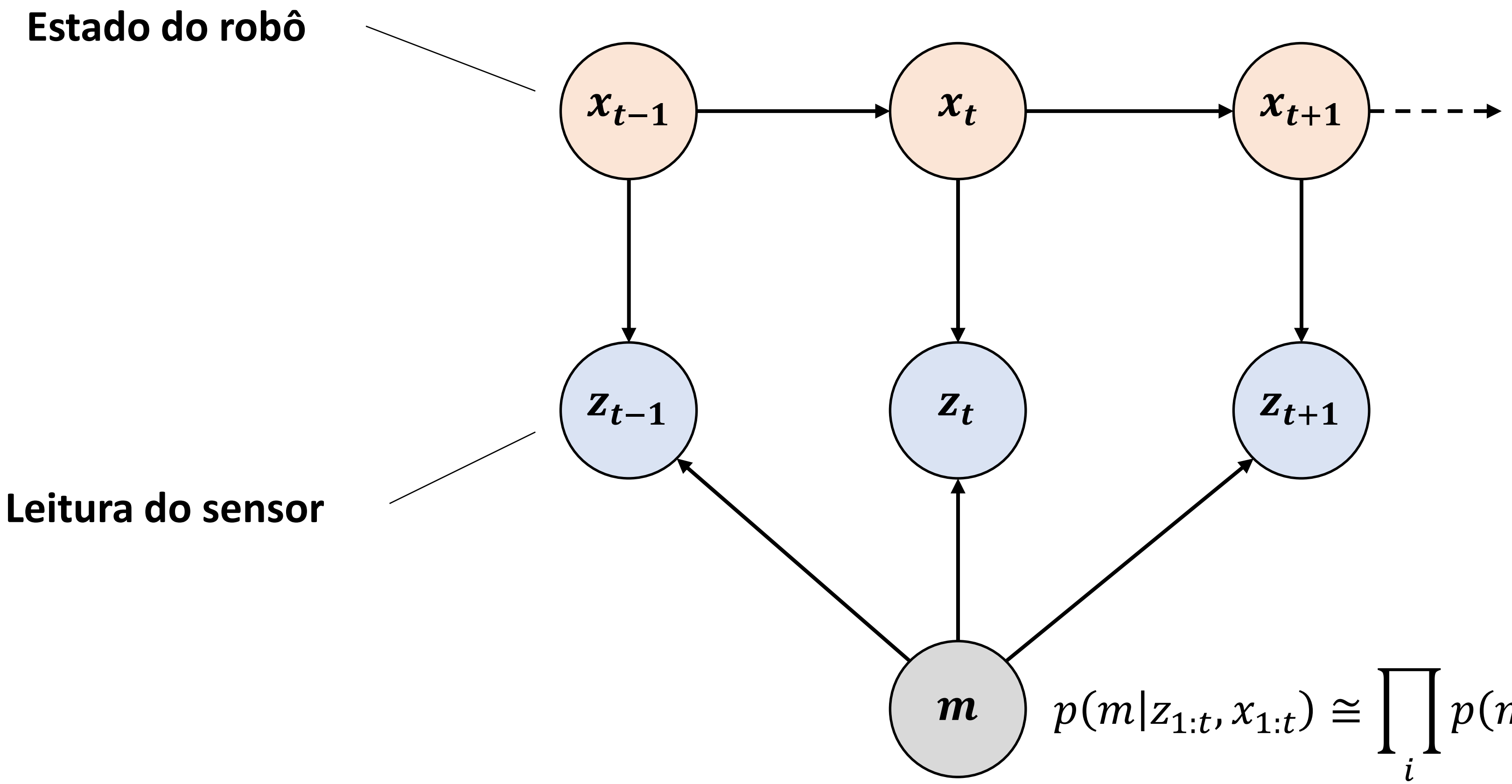
onde $z_{1:t}$ são as observações e $x_{1:t}$ o caminho do robô

- Se torna rapidamente intratável com o tamanho do mapa
  - Um grid 100×100 possui 2^{10000} mapas binários possíveis
- Para resolver fazemos uma aproximação

$$p(m \mid z_{1:t}, x_{1:t}) \cong \prod_i p(m_i \mid z_{1:t}, x_{1:t})$$

probabilidade marginal

Occupancy Grid



$$p(m|z_{1:t}, x_{1:t}) \cong \prod_i p(m_i|z_{1:t}, x_{1:t})$$

Occupancy Grid

- Mas como calcular $p(m_i \mid z_{1:t}, x_{1:t})$?

- Teorema de Bayes Condicional

$$p(A \mid B, C) = \frac{p(B \mid A, C)p(A \mid C)}{p(B \mid C)}$$

Occupancy Grid

Bayes $$p(m_i|z_{1:t},x_{1:t}) = \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t})p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})}$$

Medição atual só depende do estado atual e da célula de mapa

Estado atual sem a medição atual não fornece nenhuma informação adicional sobre a ocupação da célula

Se C é independente de A dado B, então C fornece nenhuma informação extra sobre A depois de sabermos B

$$p(A \mid B, C) = p(A \mid B)$$

Markov $$p(m_i|z_{1:t},x_{1:t}) = \frac{\boxed{p(z_t \mid m_i, x_t)}\,p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})}$$

Probabilistic Robotics (Cap. 2.4.4): The Markov assumption postulates that past and future data are independent if one knows the current state x_t.

Occupancy Grid

Bayes

$$p(z_t \mid m_i, x_t) = \frac{p(m_i \mid z_t, x_t)p(z_t \mid x_t)}{p(m_i \mid x_t)}$$

$$p(m_i|z_{1:t}, x_{1:t}) = \frac{p(m_i \mid z_t, x_t)p(\, z_t \mid x_t)p(m_i \,|z_{1:t-1}, x_{1:t-1})}{p(m_i \mid x_t)p(z_t \mid z_{1:t-1}, x_{1:t})}$$

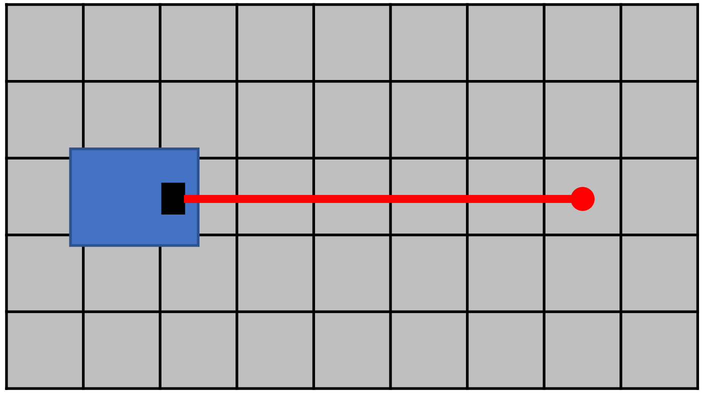
Estado atual sem a medição atual
não fornece nenhuma informação

Markov

$$p(m_i|z_{1:t}, x_{1:t}) = \frac{p(m_i \mid z_t, x_t)p(\, z_t \mid x_t)p(m_i \,|z_{1:t-1}, x_{1:t-1})}{p(m_i)p(z_t \mid z_{1:t-1}, x_{1:t})}$$

Occupancy Grid

Prior

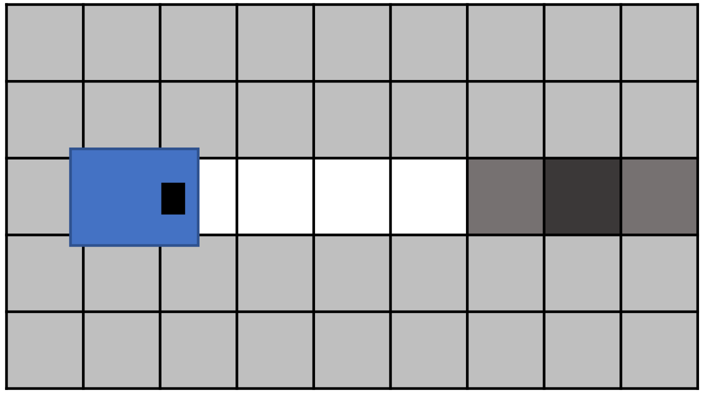


$p(m_i \mid x_t)$

Modelo de Observação

$p(z_t \mid m_i, x_t)$

Posterior



$p(m_i \mid z_t, x_t)$

Occupancy Grid

- A manipulação direta de probabilidades pode ser difícil

$$p(m_i|z_{1:t}, x_{1:t}) = \frac{p(m_i \mid z_t, x_t)p(\ z_t \mid x_t)p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i)p(z_t \mid z_{1:t-1}, x_{1:t})}$$

- Uma forma de simplificar é usar a razão das probabilidades
  - Essa também é uma maneira de associar os eventos (occ, free)

$$p(\neg m_i|z_{1:t}, x_{1:t}) = \frac{p(\neg m_i \mid z_t, x_t)p(\ z_t \mid x_t)p(\neg m_i \mid z_{1:t-1}, x_{1:t-1})}{p(\neg m_i)p(z_t \mid z_{1:t-1}, x_{1:t})}$$

Occupancy Grid

- Odds
  - Domínio: $[0, +\infty)$
  - Multiplicação das probabilidades cresce cada vez mais

$$o(A) = \frac{p(A)}{p(\neg A)} = \frac{p(A)}{1 - p(A)}$$

$$p(A) = \frac{o(A)}{1 + o(A)} = \frac{1}{1 + 1/o(A)}$$

Occupancy Grid

$$\frac{p(m_i|z_{1:t},x_{1:t})}{p(\neg m_i|z_{1:t},x_{1:t})} = \frac{\dfrac{p(m_i \mid z_t,x_t)\cancel{p(z_t \mid x_t)}p(m_i \mid z_{1:t-1},x_{1:t-1})}{p(m_i)\cancel{p(z_t \mid z_{1:t-1},x_{1:t})}}}{\dfrac{p(\neg m_i \mid z_t,x_t)\cancel{p(z_t \mid x_t)}p(\neg m_i \mid z_{1:t-1},x_{1:t-1})}{p(\neg m_i)\cancel{p(z_t \mid z_{1:t-1},x_{1:t})}}}$$

$$= \frac{p(m_i|z_t,x_t)p(m_i|z_{1:t-1},x_{1:t-1})p(\neg m_i)}{p(\neg m_i \mid z_t,x_t)p(\neg m_i|z_{1:t-1},x_{1:t-1})p(m_i)}$$

$$\frac{p(m_i|z_{1:t},x_{1:t})}{1-p(m_i|z_{1:t},x_{1:t})} = \underbrace{\frac{p(m_i|z_t,x_t)}{1-p(m_i|z_t,x_t)}}_{\textbf{Utiliza } z_t}\underbrace{\frac{p(m_i|z_{1:t-1},x_{1:t-1})}{1-p(m_i|z_{1:t-1},x_{1:t-1})}}_{p(\textbf{occ})\textbf{ em } t-1}\underbrace{\frac{1-p(m_i)}{p(m_i)}}_{\textbf{Prior}}$$

Mapeamento
Probabilidade

- Log-odds
  - Domínio: $(-\infty, +\infty)$
  - Utilizado por razões de eficiência
  - Instabilidade numérica com probabilidades próximas de 0 ou 1
    - Evita problemas de truncamento pela multiplicação de probabilidades

$$l(\mathrm{A}) = \log(o(A)) = \log\left(\frac{p(A)}{p(\neg A)}\right)$$

https://towardsdatascience.com/https-towardsdatascience-com-what-and-why-of-log-odds-64ba988bf704

Occupancy Grid

- Representando os dois lados da equação com log-odds

$$\frac{p(m_i|z_{1:t},x_{1:t})}{1-p(m_i|z_{1:t},x_{1:t})} = \frac{p(m_i|z_t,x_t)}{1-p(m_i|z_t,x_t)} \frac{p(m_i|z_{1:t-1},x_{1:t-1})}{1-p(m_i|z_{1:t-1},x_{1:t-1})} \frac{1-p(m_i)}{p(m_i)}$$

$$l_{t,i} = l(m_i|z_{1:t},x_{1:t}) = \log\left(\frac{p(m_i|z_{1:t},x_{1:t})}{1-p(m_i|z_{1:t},x_{1:t})}\right)$$

Occupancy Grid

- A regra de atualização se baseia apenas na soma de termos

$$l(m_i|z_{1:t}, x_{1:t}) = \underbrace{l(m_i|z_{t-1}, x_{t-1})}_{\text{recursive term}} + \underbrace{l(m_i|z_t, x_t)}_{\text{inverse sensor model}} - \underbrace{l(m_i)}_{\text{prior}}$$

- De maneira ainda mais compacta

$$l_{t,i} = l_{t-1,i} + \text{inverse\_sense\_model}(m_i, x_t, z_t) - l_0$$

Occupancy Grid

- Pode-se recuperar a probabilidade a partir do log-odds

$$l_{t,i} = l(m_i|z_{1:t}, x_{1:t}) = \log\left(\frac{p(m_i|z_{1:t}, x_{1:t})}{1 - p(m_i|z_{1:t}, x_{1:t})}\right)$$

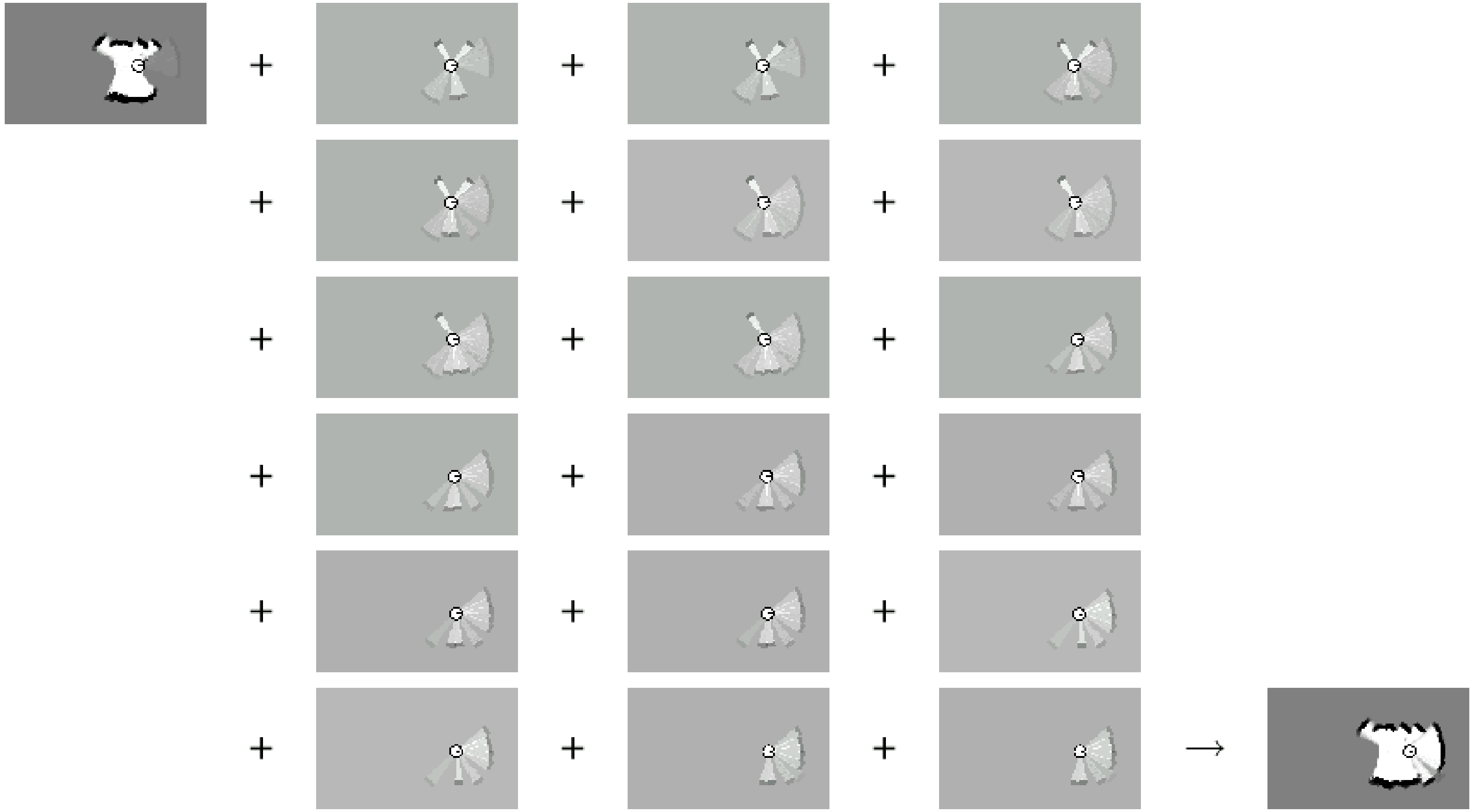
$$p(m_i|z_{1:t}, x_{1:t}) = 1 - \frac{1}{1 + \exp(l_{t,i})}$$

Occupancy Grid

Algorithm occupancy_grid_mapping($\{l_{t-1,i}\}, x_t, z_t$**):**
    for all cells $\mathbf{m}_i$ do
        if $\mathbf{m}_i$ in perceptual field of z_t then
            $l_{t,i} = l_{t-1,i} + \textbf{inverse\_sensor\_model}(\mathbf{m}_i, x_t, z_t) - l_0$
        else
            $l_{t,i} = l_{t-1,i}$
        endif
    endfor
    return $\{l_{t,i}\}$

Fonte: *Probabilistic Robotics*

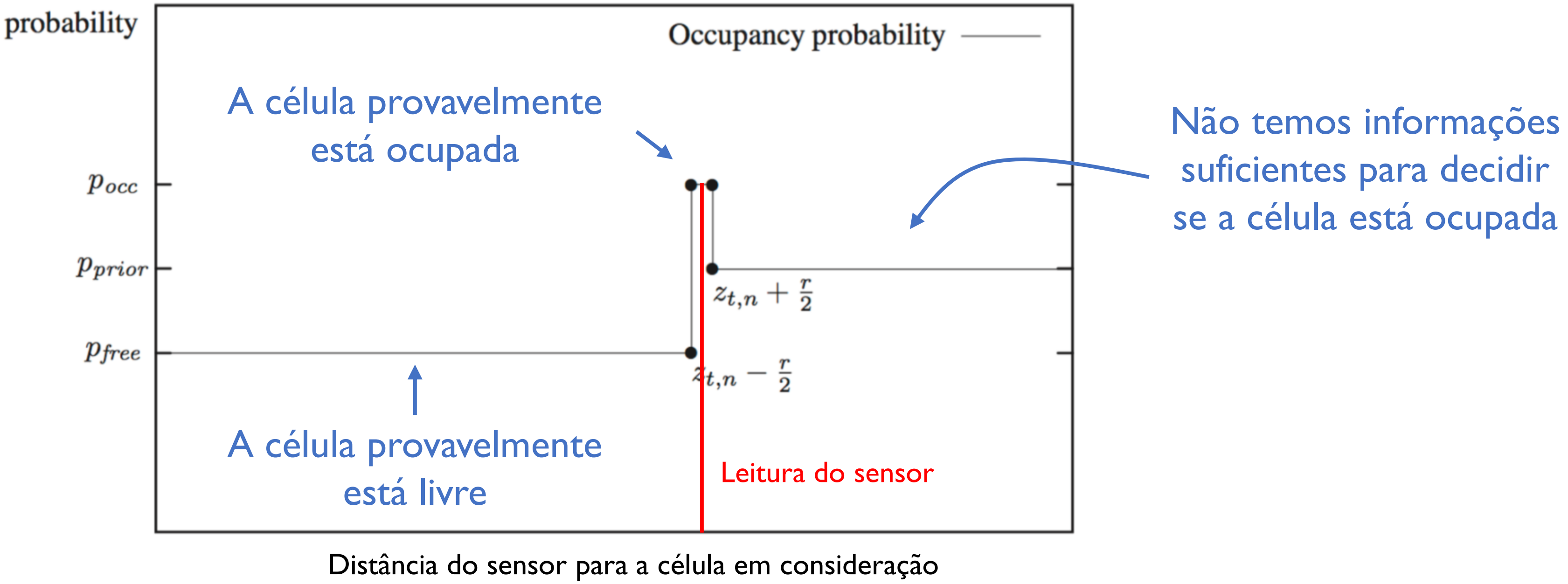
Occupancy Grid



Fonte: Occupancy Grid Mapping (Cyrill Stachniss)

Occupancy Grid
Inverse Sensor Model (Laser)

- O que a nova informação acrescenta em relação ao prior
  - Dada a medição, é mais provável que esteja ocupado ou livre?



Fonte: Occupancy Grid Mapping (Cyrill Stachniss)

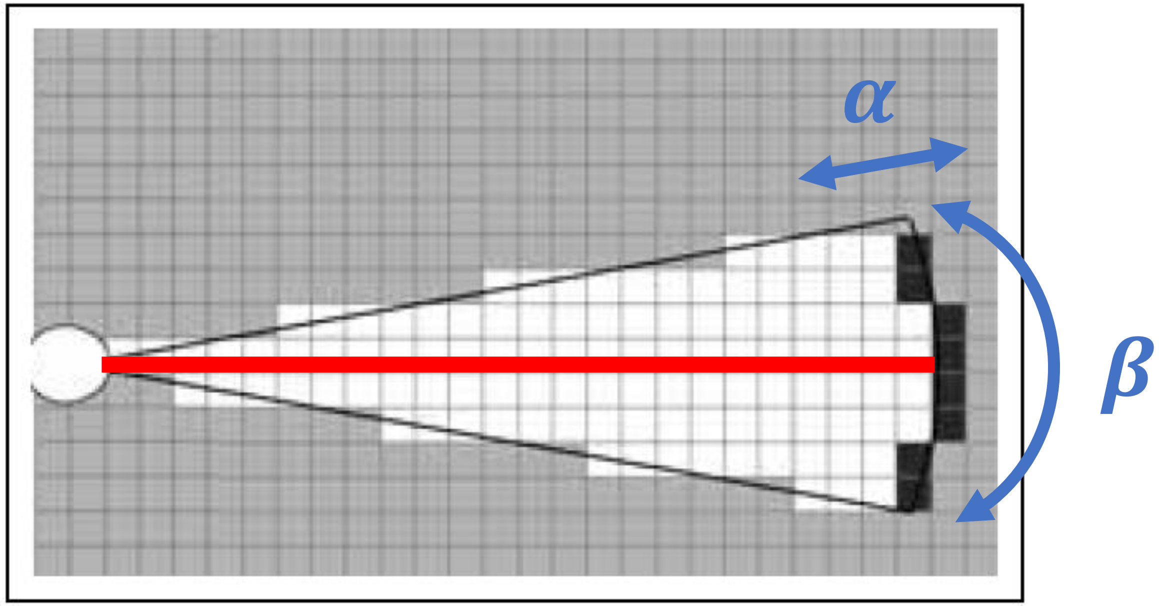
Occupancy Grid
Inverse Sensor Model

Algorithm inverse_range_sensor_model(i, x_t, z_t):
  Let x_i, y_i be the center-of-mass of $\mathbf{m}_i$
  $r = \sqrt{(x_i - x)^2 + (y_i - y)^2}$
  $\phi = \text{atan2}(y_i - y, x_i - x) - \theta$
  $k = \text{argmin}_j \, |\phi - \theta_{j,\text{sens}}|$
  if $r > \min(z_{\max}, z_t^k + \alpha/2)$ or $|\phi - \theta_{k,\text{sens}}| > \beta/2$ then
    return l_0
  if $z_t^k < z_{\max}$ and $|r - z_{\max}| < \alpha/2$
    return l_{occ}
  if $r \leq z_t^k$
    return l_{free}
  endif

Atenção: esse código é para sonares (α, β), deve ser ligeiramente adaptado para lasers.

Fonte: *Probabilistic Robotics (Cap. 9)*

Occupancy Grid
Inverse Sensor Model



Fonte: *Probabilistic Robotics (Cap. 9)*

Occupancy Grid
Beam Models of Range Finders

- Probabilidades dependem do modelo do sensor
- Geralmente a medição é composta por leituras individuais
  - $z_t = \{z_t^1, \dots, z_t^K\}$
  - Aproximado pelo produto das probabilidades individuais

$$p(z_t \mid m, x_t) = \prod_{k=1}^{K} p\left(z_t^k \mid m, x_t\right)$$

- Modelo mais realista para $p(z_t \mid m, x_t)$
  - Considera ruídos e incertezas associadas à medição

Occupancy Grid

Beam Models of Range Finders

- Termos considerados
  - Ruído na medição
  - Objetos inesperados
  - Falhas
  - Aleatoriedades

$$p(z_t^k \mid x_t, m) = \begin{pmatrix} z_{\text{hit}} \\ z_{\text{short}} \\ z_{\text{max}} \\ z_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} p_{\text{hit}}(z_t^k \mid x_t, m) \\ p_{\text{short}}(z_t^k \mid x_t, m) \\ p_{\text{max}}(z_t^k \mid x_t, m) \\ p_{\text{rand}}(z_t^k \mid x_t, m) \end{pmatrix}$$

Algorithm beam_range_finder_model(z_t, x_t, m**):**

$q = 1$

for $k = 1$ *to* K *do*

  compute z_t^{k*} *for the measurement* z_t^k *using ray casting*

  $p = z_{\text{hit}} \cdot p_{\text{hit}}(z_t^k \mid x_t, m) + z_{\text{short}} \cdot p_{\text{short}}(z_t^k \mid x_t, m)$

    $+ z_{\text{max}} \cdot p_{\text{max}}(z_t^k \mid x_t, m) + z_{\text{rand}} \cdot p_{\text{rand}}(z_t^k \mid x_t, m)$

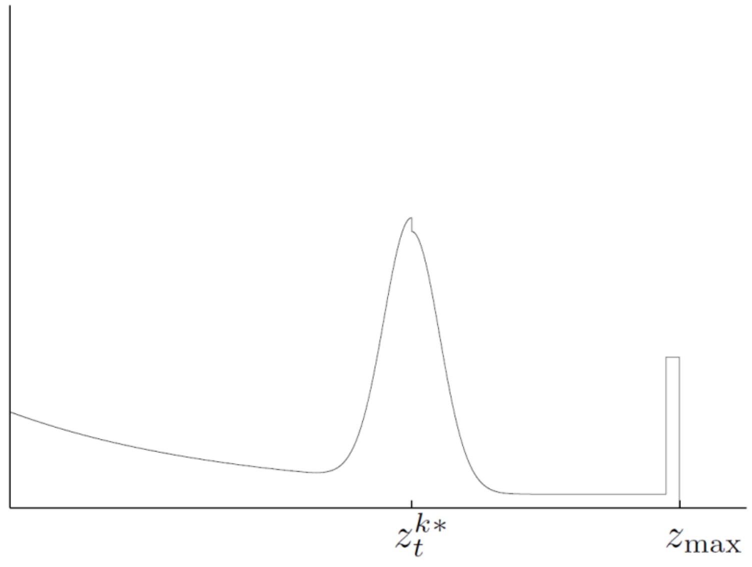
  $q = q \cdot p$

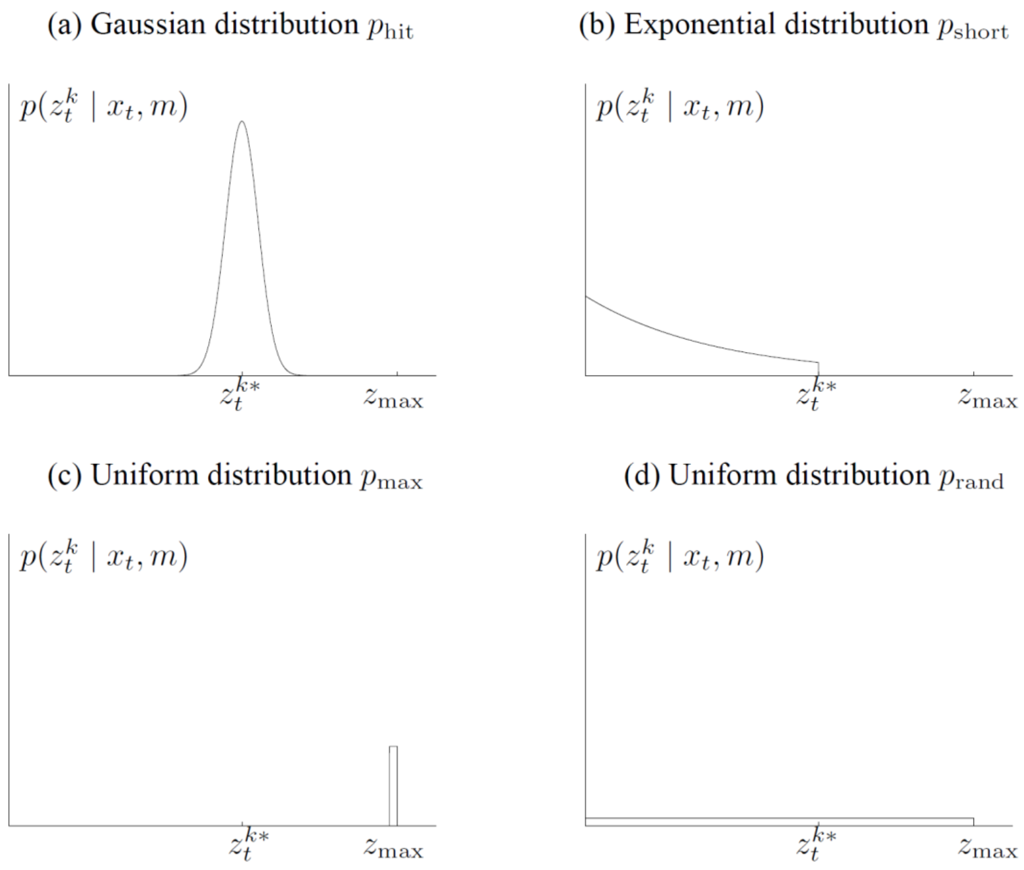
return q

Fonte: *Probabilistic Robotics (Cap. 6)*

Occupancy Grid

Beam Models of Range Finders





Fonte: *Probabilistic Robotics (Cap. 6)*

Occupancy Grid

Beam Models of Range Finders

- Modelo simplificado

$$p(z_t \mid m, x_t) = \frac{1 + S_{occ}^{z} - S_{free}^{z}}{2}$$

Distância para o sensor   Distância leitura   Precisão

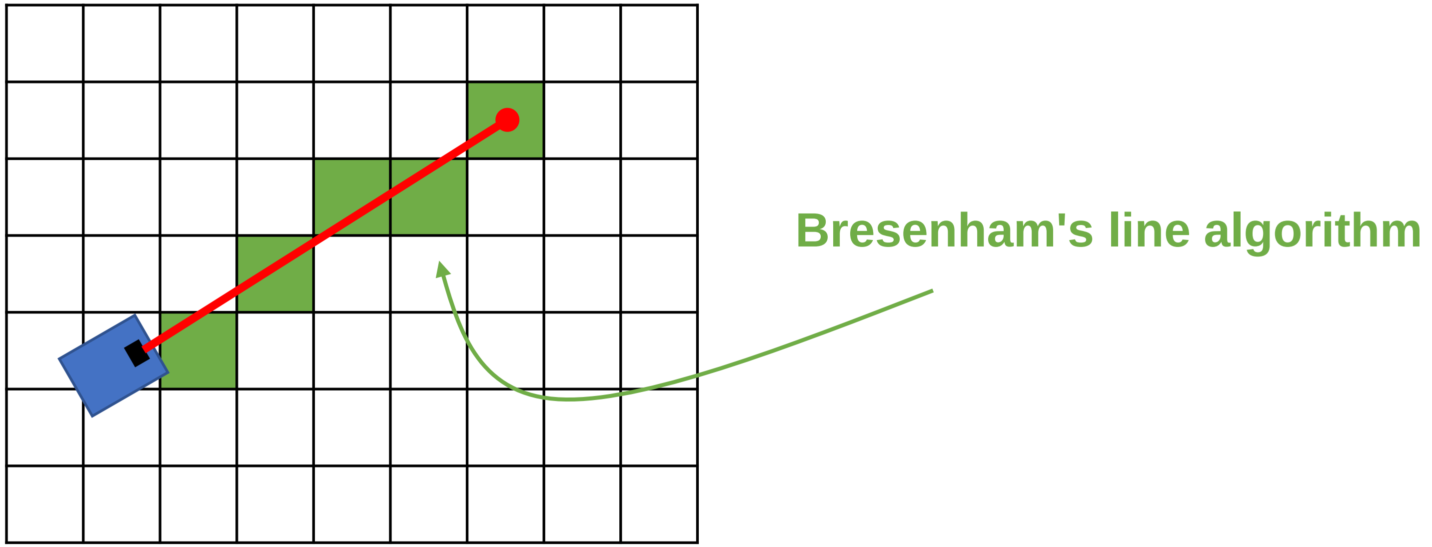
$$S_{occ}^{z} = \begin{cases} 1 & \text{se } d \in [r - \epsilon, r + \epsilon] \\ 0 & \text{caso contrário} \end{cases}$$

$$S_{free}^{z} = \begin{cases} 1 & \text{se } d \in [0, r - \epsilon) \\ 0 & \text{caso contrário} \end{cases}$$

- Outra alternativa
  - $p_{occ} \gg 0{,}5$: se a célula está perto da distância de medição
  - $p_{occ} \ll 0{,}5$: se a célula está longe (entre o robô e a medição)
  - Lembrando que $p_{free} = 1 - p_{occ}$

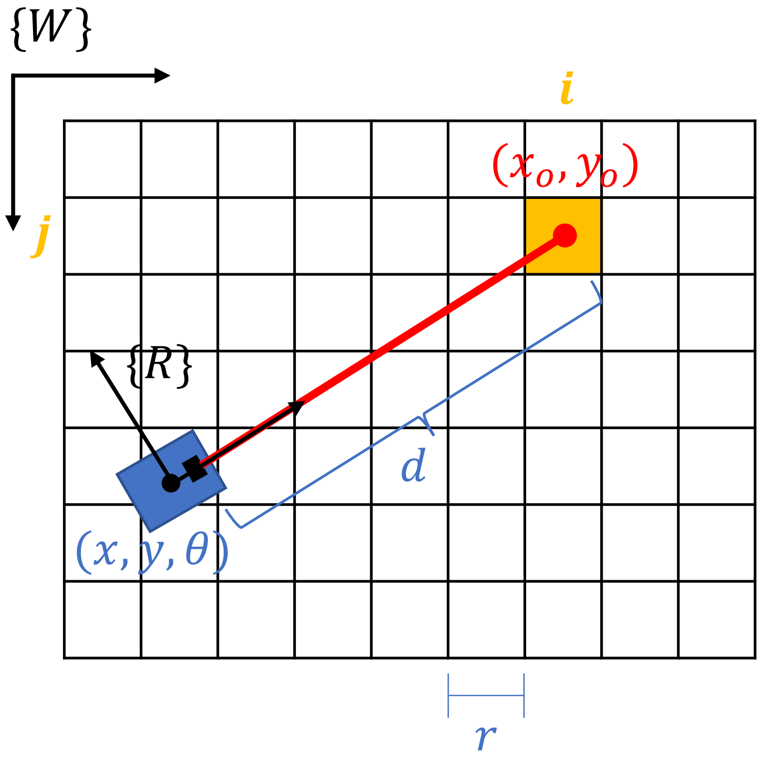
Occupancy Grid

- Precisamos percorrer o mapa todo a cada nova leitura?
  - Não!
  - Atualizar apenas as células associadas às leituras



https://en.wikipedia.org/wiki/Bresenham's_line_algorithm

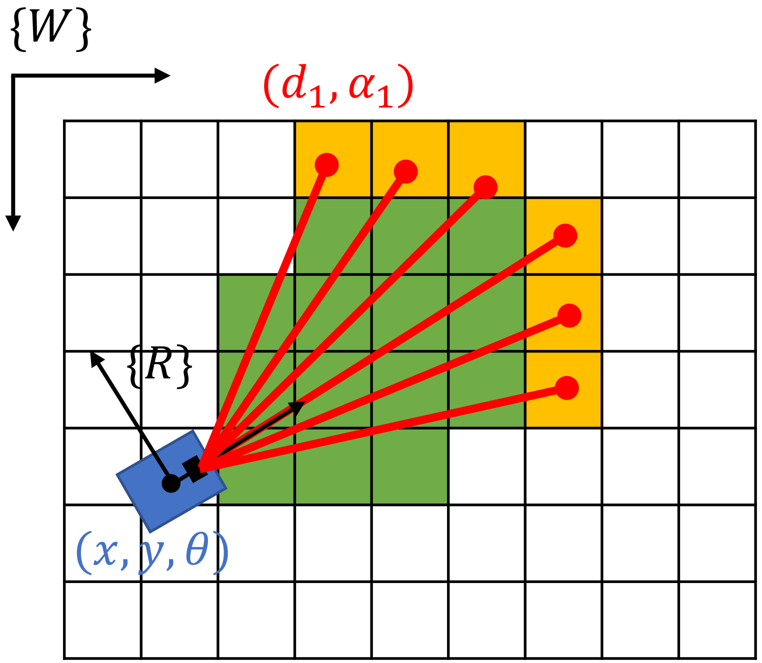
Occupancy Grid



$$\begin{bmatrix} x_o \\ y_o \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} d \\ 0 \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} i \\ j \end{bmatrix} = \text{floor}\left(\frac{1}{r}\begin{bmatrix} x_o \\ y_o \end{bmatrix}\right)$$

Occupancy Grid



$$\begin{bmatrix} x_o^k \\ y_o^k \end{bmatrix} = \begin{bmatrix} d_k \cos(\theta + \alpha_k) \\ -d_k \sin(\theta + \alpha_k) \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} i^k \\ j^k \end{bmatrix} = \text{floor}\left(\frac{1}{r}\begin{bmatrix} x_o^k \\ y_o^k \end{bmatrix}\right)$$

Occupancy Grid

https://youtu.be/aEeS8hDnnYg

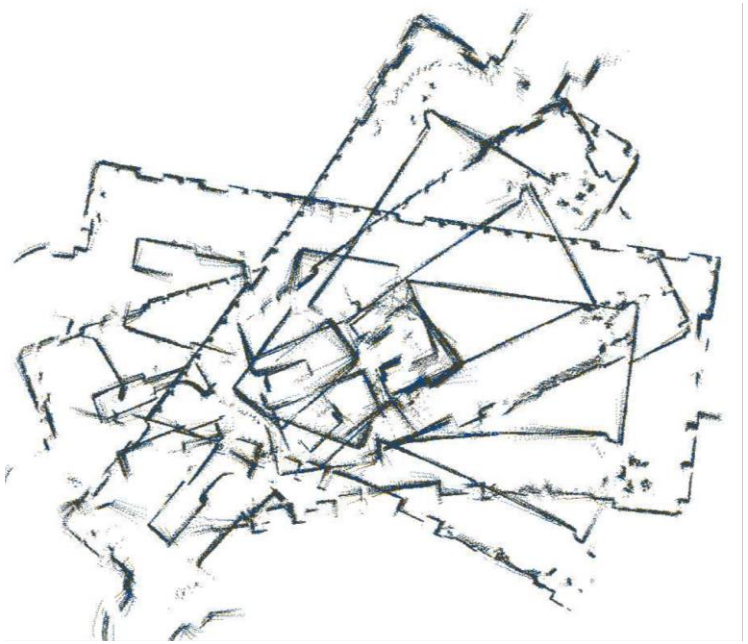
Considerações finais
Histograma

- Contador
  - hits(x,y): Números de vezes que o sensor identificou um obstáculo na célula (x,y)
  - misses(x,y): Números de vezes que o sensor **não** identificou um obstáculo na célula (x,y)

$$p(m_i) = \frac{hits(x, y)}{hits(x, y) + misses(x, y)}$$

Considerações finais

- Convergência com o tempo
  - Probabilidades acumuladas (ocupado/vazio)
- Decisões baseadas em um limiar
  - $p(m_i) = 0{,}45$, está ocupado ou não?
- Problemas da utilização de um grid
  - Resolução, memória, processamento, ...
- A localização também possui incerteza!



Fonte: *Probabilistic Robotics*