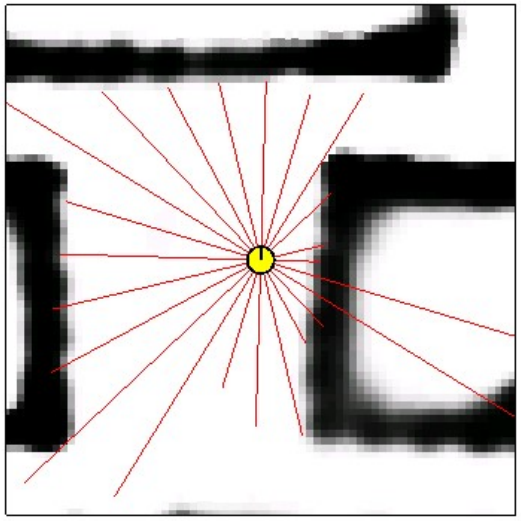
Modello Probabilistico dei Sensori

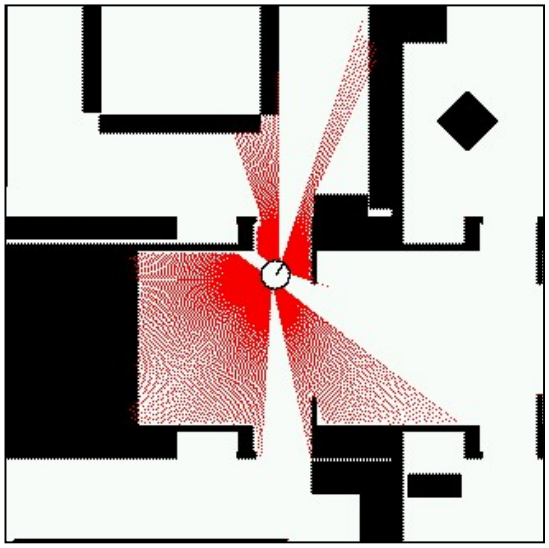
- Contatto: Bumpers
- Interni:
  - Accelerometro
  - Giroscopio
  - Bussola
- Prossimità:
  - Sonar
  - Radar
  - Laser range-finders
  - Infrarossi
- Visuali: Telecamere
- Satellitari: GPS

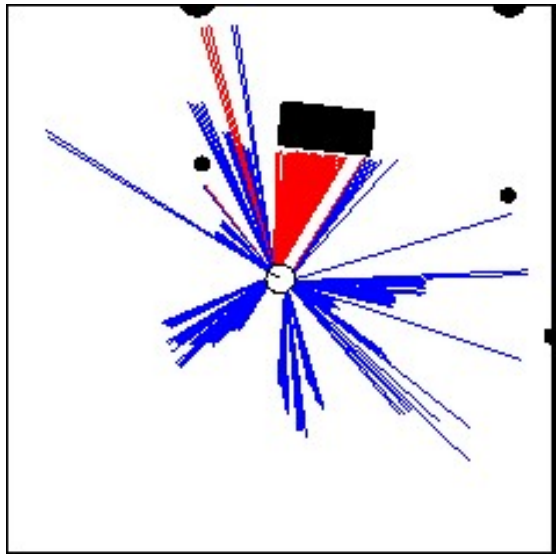
Sensori di Prossimità

- Il task è determinare *P(z|x)*, i.e., la probabilità della misura *z* assumendo il robot in posizione *x*.

- **Domanda**: da dove vengono le probabilità?
- **Approaccio**: si caratterizza il modello del sensore.







Modello Beam-based

- Una scansione *z* consiste di *K* misure.

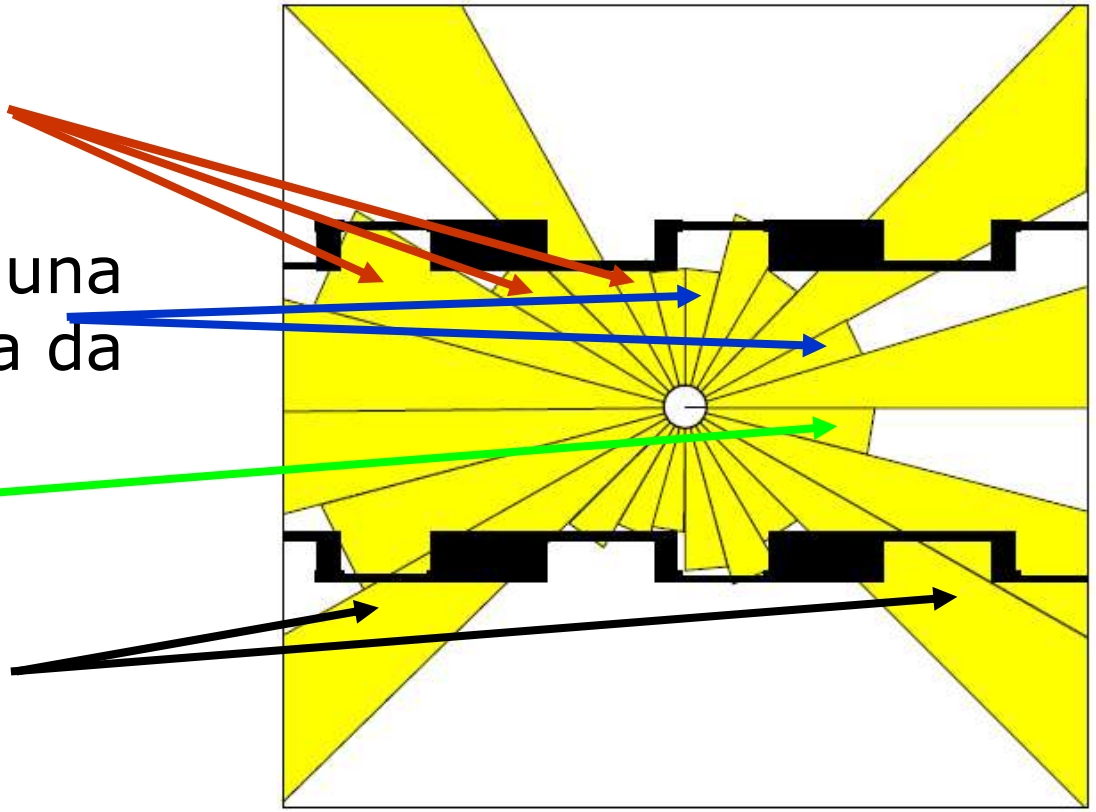
$$z = \{z_1, z_2, \ldots, z_K\}$$

- Ogni misura è indipendente data la posizione del robot.

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

Tipici Errori

1. Beams riflessi da ostacoli
2. Beams riflessi da una persona o causata da crosstalk
3. Misure random
4. Misure di range massimo

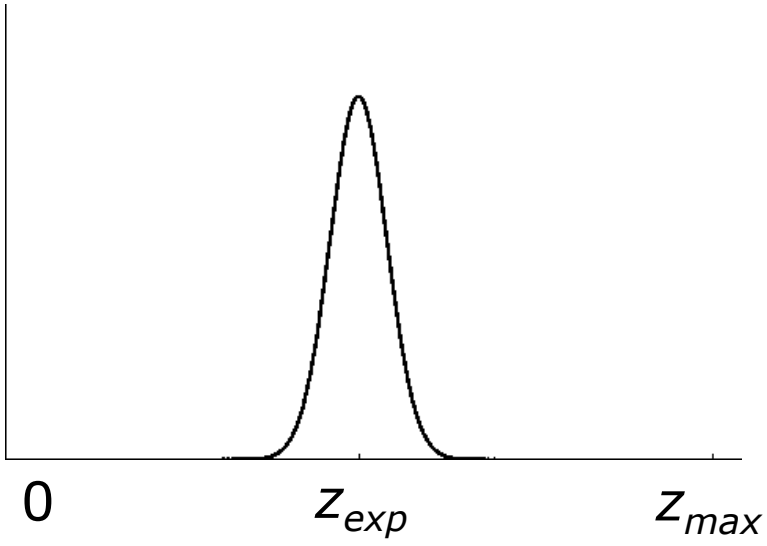


Misure di prossimità

- Misure causate da…
  - Ostacolo noto.
  - Cross-talk.
  - Ostacolo inatteso (e.g. passanti).
  - Ostacolo perso (riflessione totale, vetro, …).
- Rumore dovuto ad incertezza …
  - Nella distanza da un ostacolo noto.
  - Nella posizione di un ostacolo noto.
  - Nella posizione di altri ostacoli.
  - Se l'ostacolo è perso.

Modello Beam-based

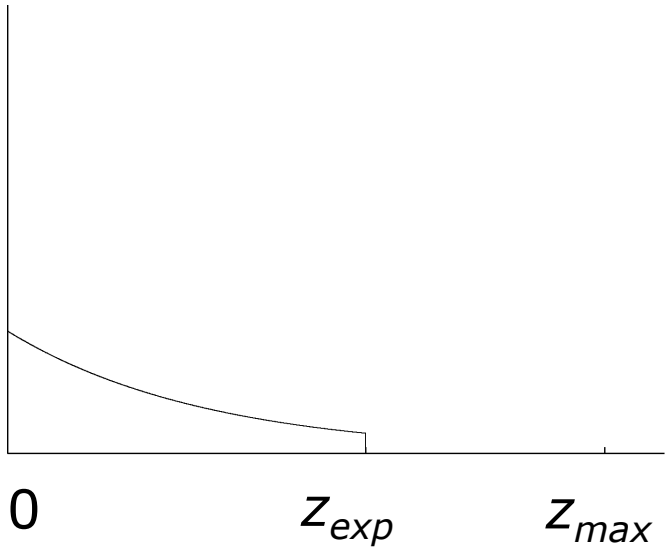
Rumore di Misura



Gaussiana

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

Ostacoli inattesi

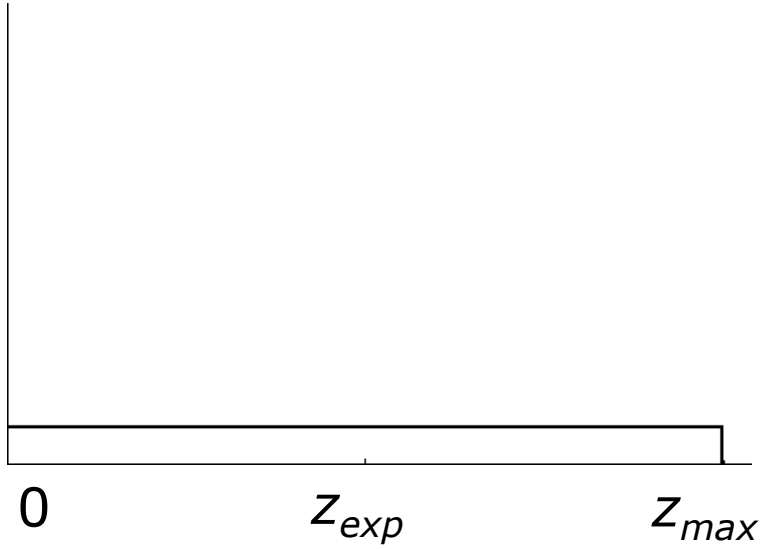


Attenuazione exp

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

Modello Beam-based

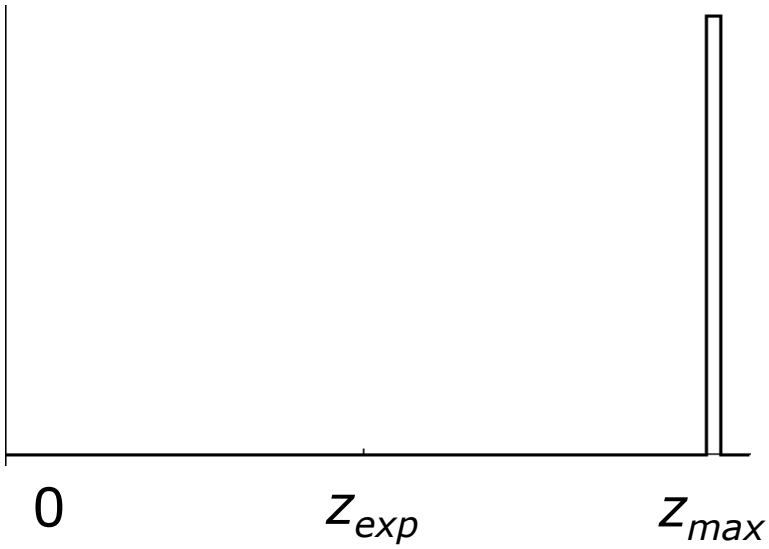
Misure random



uniforme

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

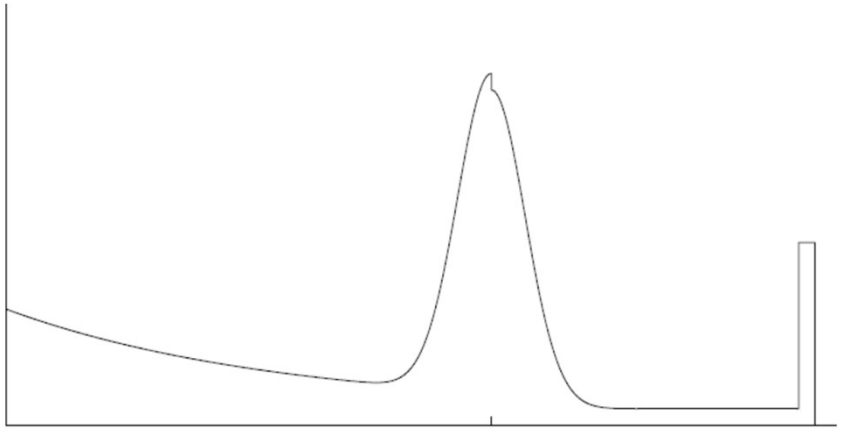
Range massimo



Dirac

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

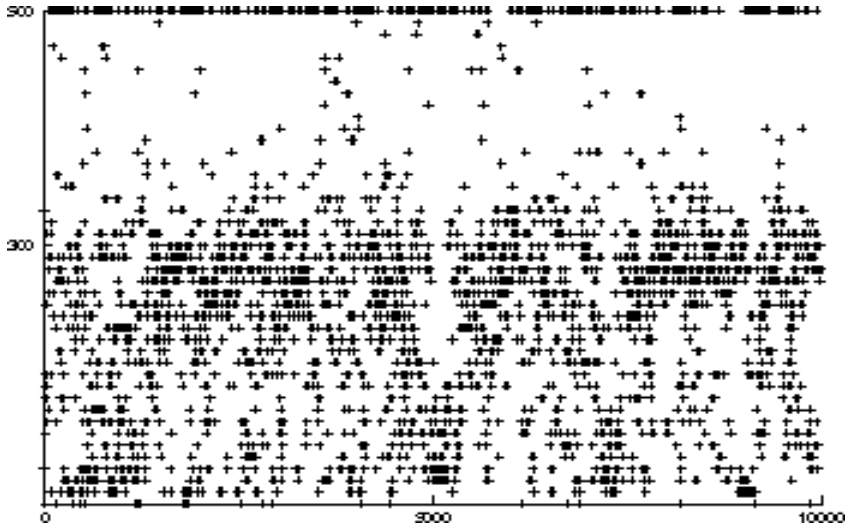
Mistura risultante



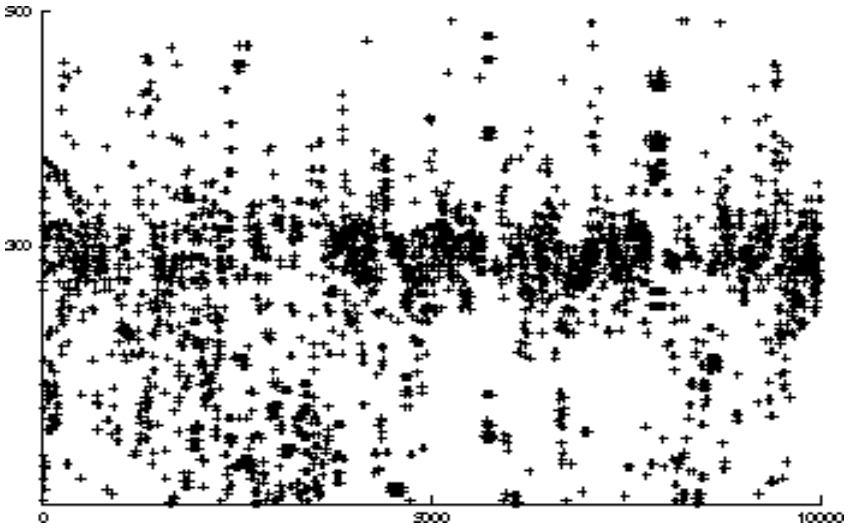
$$P(z \mid x,m) = \begin{pmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{unexp}} \\ \alpha_{\text{max}} \\ \alpha_{\text{rand}} \end{pmatrix}^{T} \cdot \begin{pmatrix} P_{\text{hit}}(z \mid x,m) \\ P_{\text{unexp}}(z \mid x,m) \\ P_{\text{max}}(z \mid x,m) \\ P_{\text{rand}}(z \mid x,m) \end{pmatrix}$$

Come determinare i parameteri del modello?

Raw Sensor Data



Sonar



Laser

Errori più evidenti per i sonar

Approssimazione

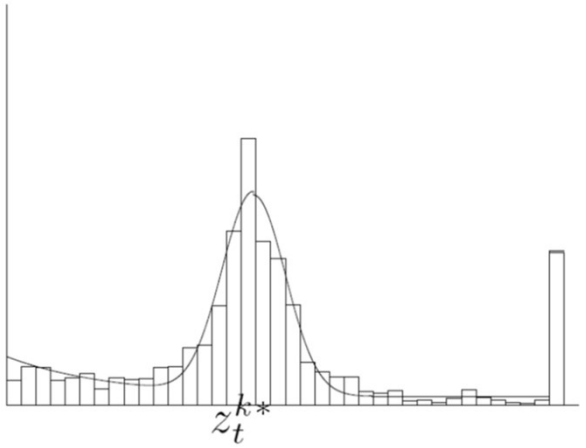
- Massimizza log likelihood del dato

$$p(Z \mid X, m, \Theta),$$

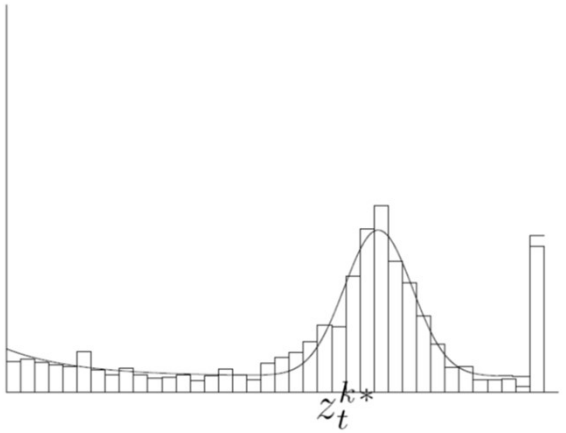
- Ricerca nello spazio di n-1 parametri Θ.
  - Hill climbing
  - Gradient descent
  - Genetic algorithms
  - …
- Calcolo deterministico del n-th parametro per soddisfare vincoli di normalizzazione.

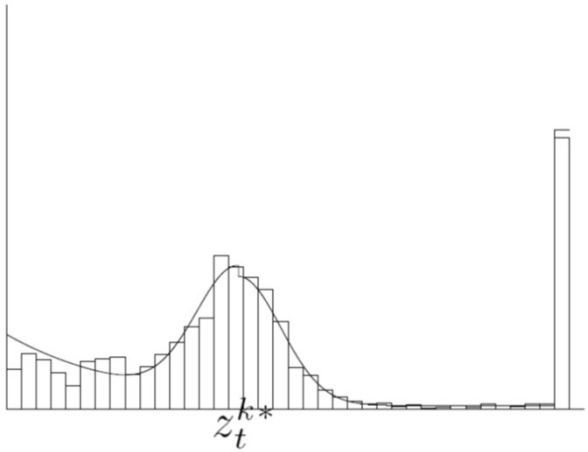
Approssimazione del modello

Misure vicine più precise, laser più precisi, molte imprecisioni

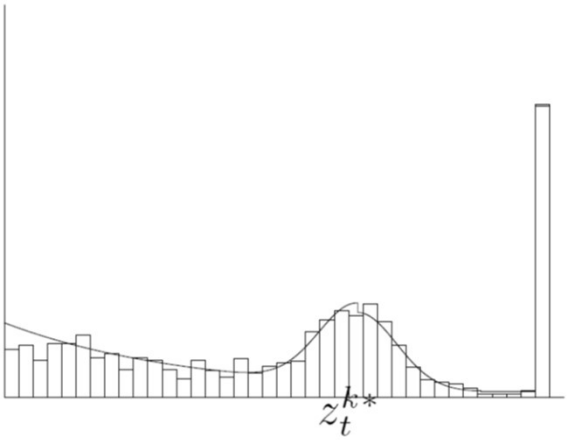


Laser





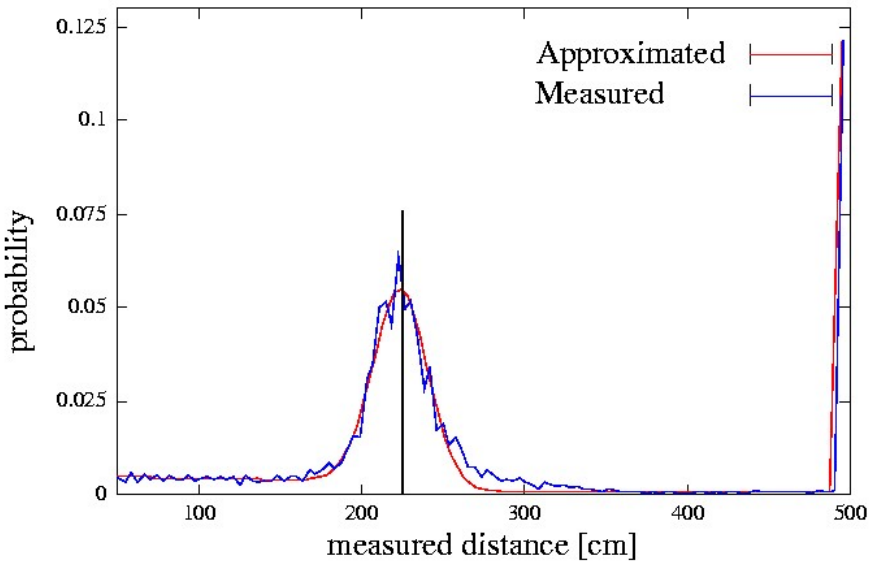
Sonar



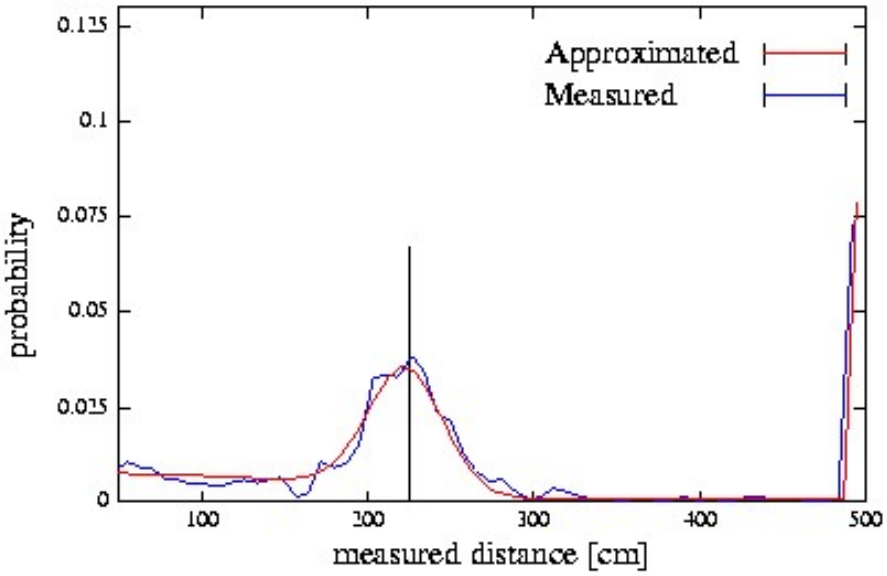
Modello discreto dei sensori di prossimità

- Instead of densities, consider discrete steps along the sensor beam.
- Consider dependencies between different cases.

$$P(d_i \mid l) = 1 - (1 - (1 - \sum_{j<i} P_u(d_j))\ c_d\ P_m(d_i \mid l))) \cdot (1 - (1 - \sum_{j<i} P(d_j))\ c_r)$$

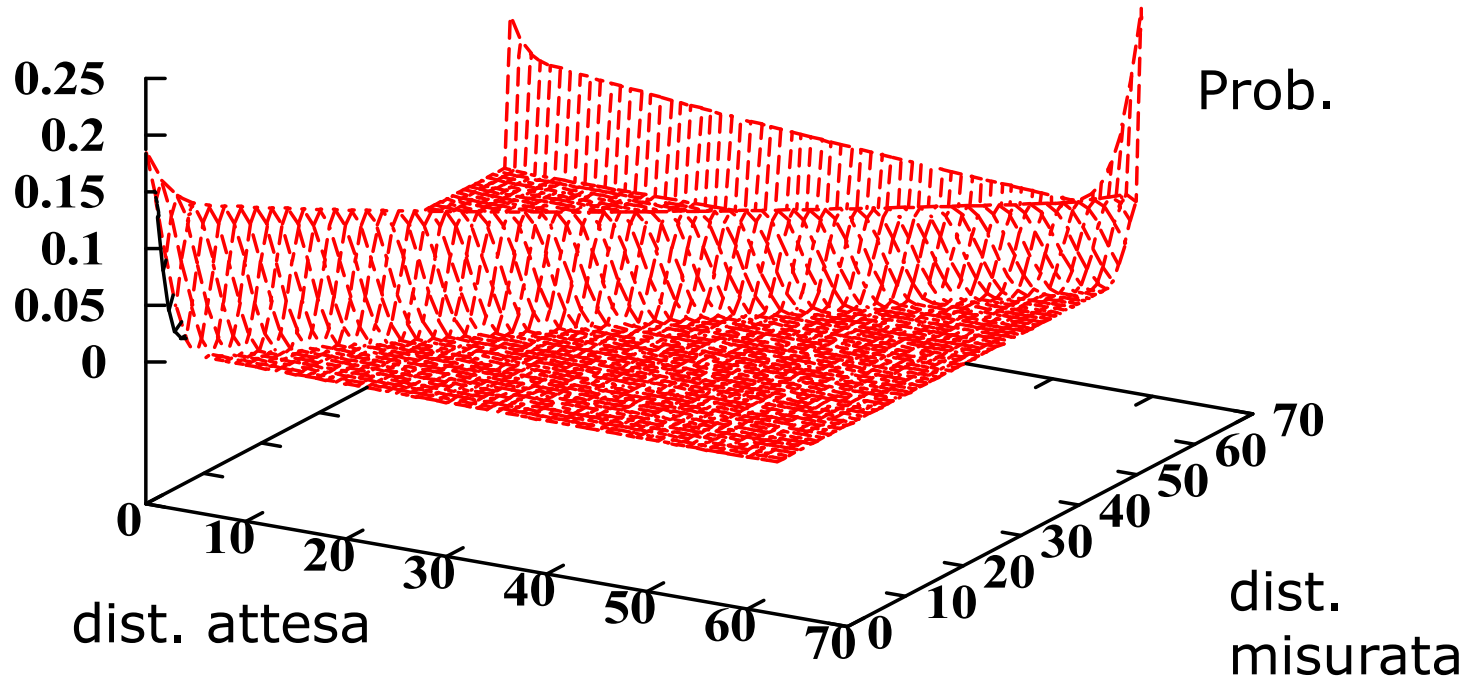


Laser sensor

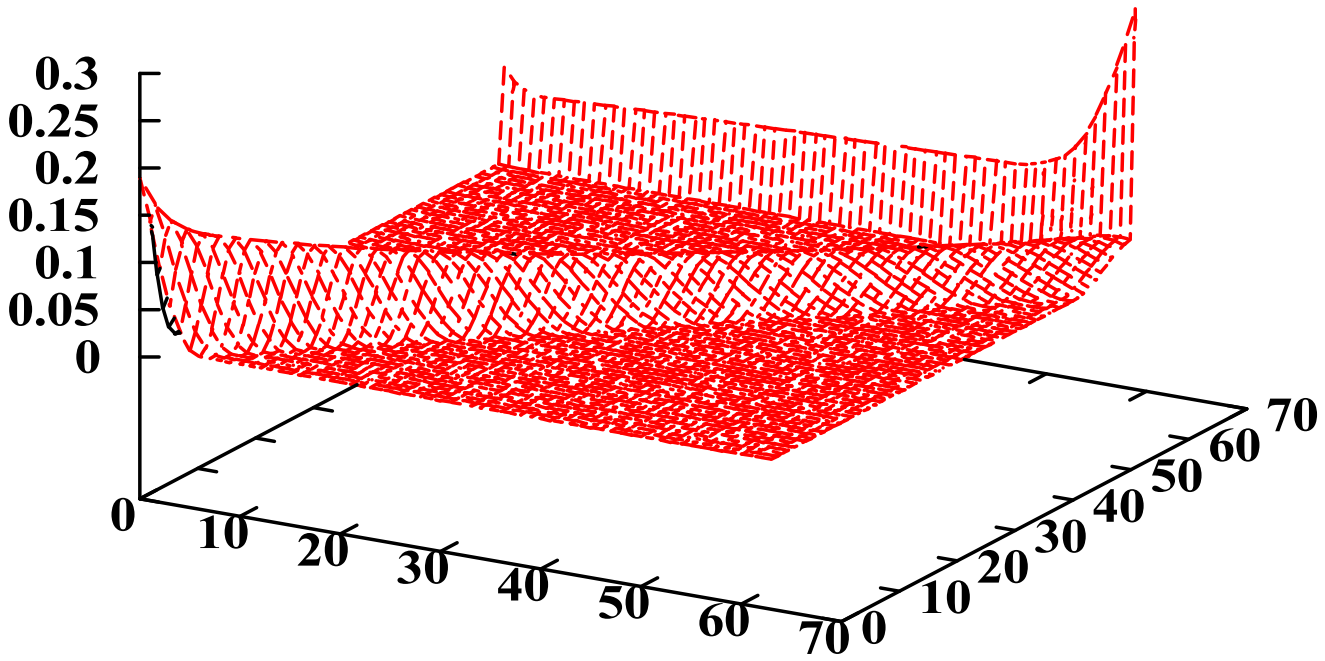


Sonar sensor

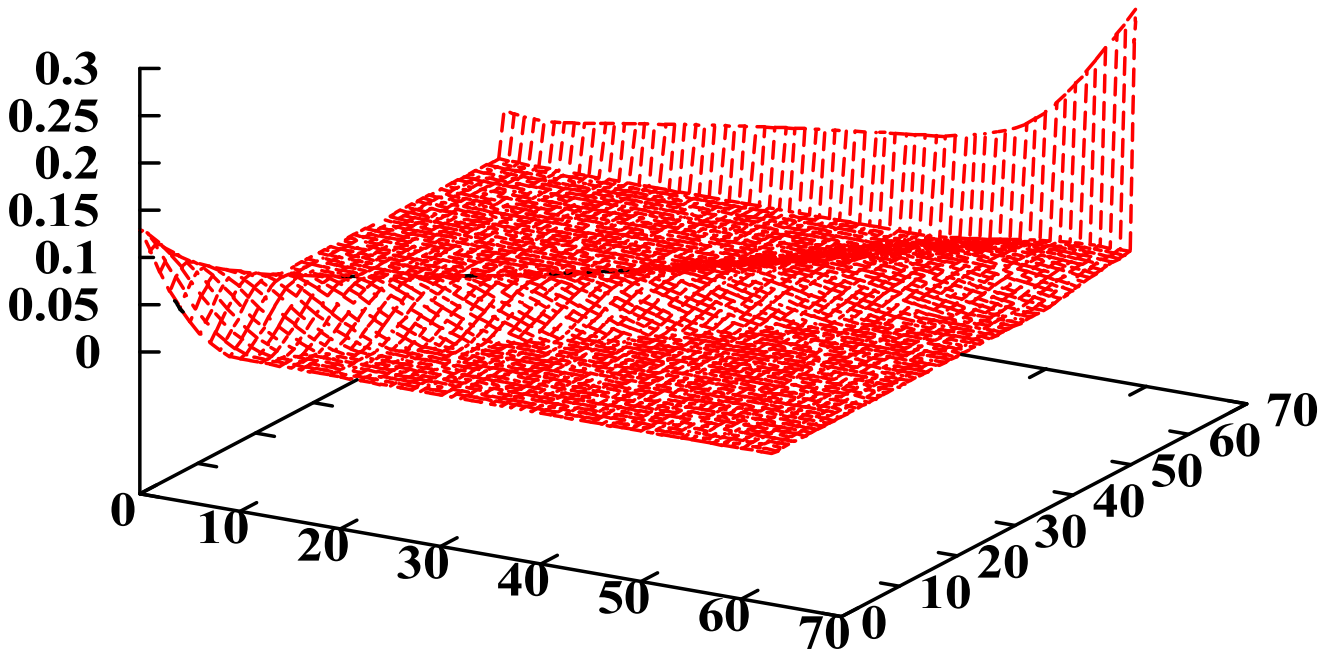
Influenza dell'angolo rispetto all'ostacolo



Influenza dell'angolo rispetto all'ostacolo



Influenza dell'angolo rispetto all'ostacolo



Riassunto del Modello Beam

- Assume indipendenza tra beams.
- Modella cause fisiche:
  - Misto di densità per queste cause.
  - Assume independenza tra le cause. Problemi?
- Implementazione:
  - Apprende parameteri su dati reali.
  - Modelli differenti per differenti angoli con cui il beam colpisce l'ostacolo.
  - Determina la distanza attesa con ray-tracing.
  - Distanze attese possono essere pre-processate.

Modello scan-based

- Il modello beam-based è …
  - non graduale con piccoli ostacoli e su bordi.
  - non efficiente.

- Idea: le posizioni finali dei beam direttamente proiettate nella mappa globale.

- Occorre: posa del robot, del sensore, misura

$$\begin{pmatrix} x_{z_t^k} \\ y_{z_t^k} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_{k,\text{sens}} \\ y_{k,\text{sens}} \end{pmatrix} + z_t^k \begin{pmatrix} \cos(\theta + \theta_{k,\text{sens}}) \\ \sin(\theta + \theta_{k,\text{sens}}) \end{pmatrix}$$

Modello probabilistico scan-based

- Probabilità è una mistura di 3
  - **Misura**: Gaussiana con valore medio a distanza dell'ostacolo più vicino (calcolata da likelihood field)

$$p_{\text{hit}}(z_t^k \mid x_t, m) \quad = \quad \varepsilon_{\sigma_{\text{hit}}^2}(dist^2)$$

  - **Misure Random**: Uniforme
  - **Fallimento**: piccola uniforme per misure max range.

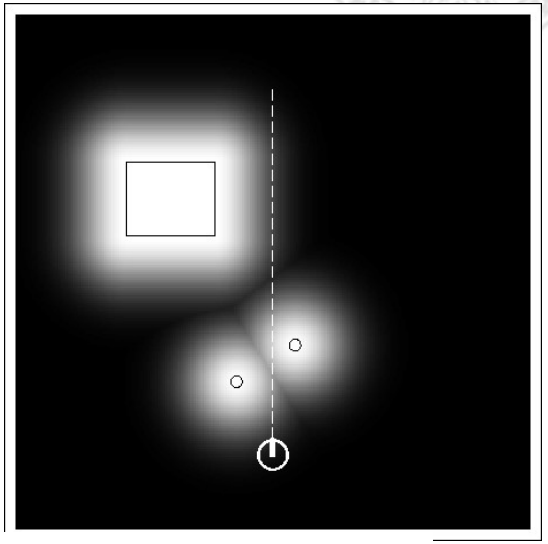
$$z_{\text{hit}} \cdot p_{\text{hit}} + z_{\text{rand}} \cdot p_{\text{rand}} + z_{\text{max}} \cdot p_{\text{max}}$$

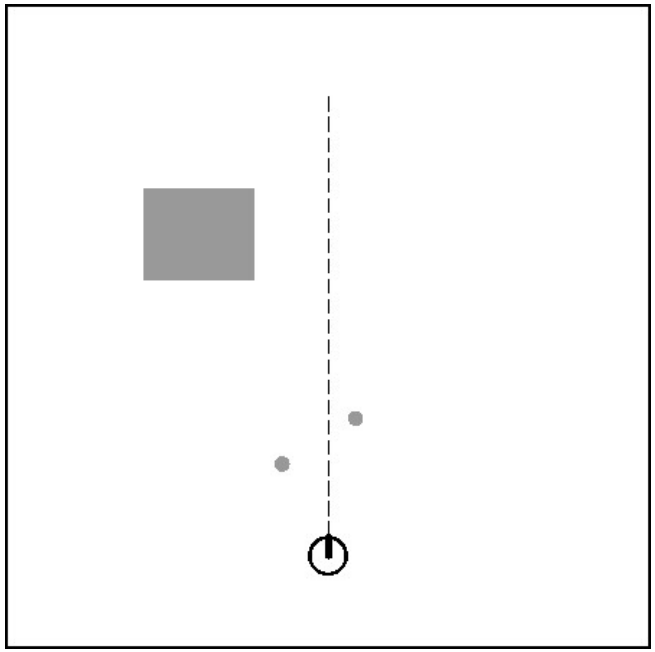
- Si assume independenza tra componenti diverse.

Likelihood Field

Likelihood field

Bianco dove è verosimile la misura





Map *m*

P(z|x,m)

Distribuzione risultante, data la posizione del robot

Likelihood Field

Algorithm likelihood_field_range_finder_model(z_t, x_t, m):

$q = 1$

for all k *do*

  if $z_t^k \neq z_{\max}$

    $x_{z_t^k} = x + x_{k,\text{sens}} \cos\theta - y_{k,\text{sens}} \sin\theta + z_t^k \cos(\theta + \theta_{k,\text{sens}})$

    $y_{z_t^k} = y + y_{k,\text{sens}} \cos\theta + x_{k,\text{sens}} \sin\theta + z_t^k \sin(\theta + \theta_{k,\text{sens}})$

    $dist^2 = \min_{x',y'} \left\{ (x_{z_t^k} - x')^2 + (y_{z_t^k} - y')^2 \;\middle|\; \langle x', y' \rangle \text{ occupied in } m \right\}$

    $q = q \cdot \left(z_{\text{hit}} \cdot \text{prob}(dist^2, \sigma_{\text{hit}}^2) + \frac{z_{\text{random}}}{z_{\max}} \right)$

return q

Ricerca ostacolo più vicino

Calcolo del likelihood di misura usando la distanza dall'ostacolo più vicino

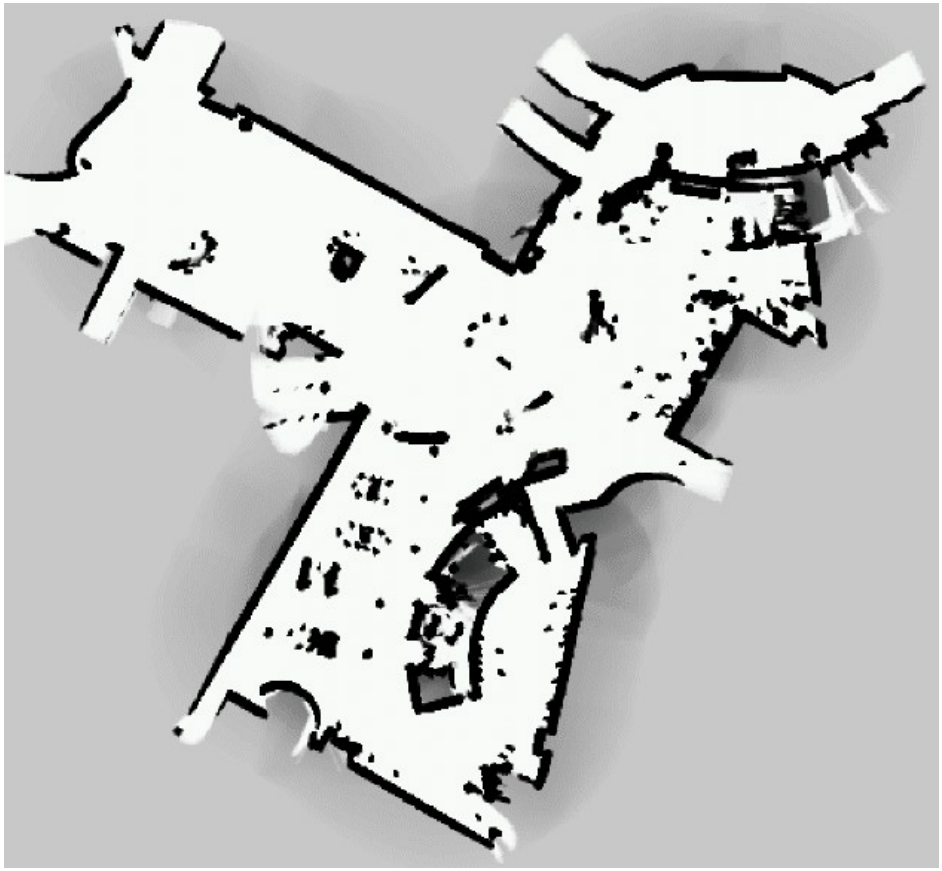
Proprietà dello scan-based

- Altamente efficiente, usa solo 2D.
- Smooth rispetto ai piccoli cambiamenti di posizione del robot.
- Permette gradient descent, scan matching.
- Ignora le proprietà fisiche dei beams.

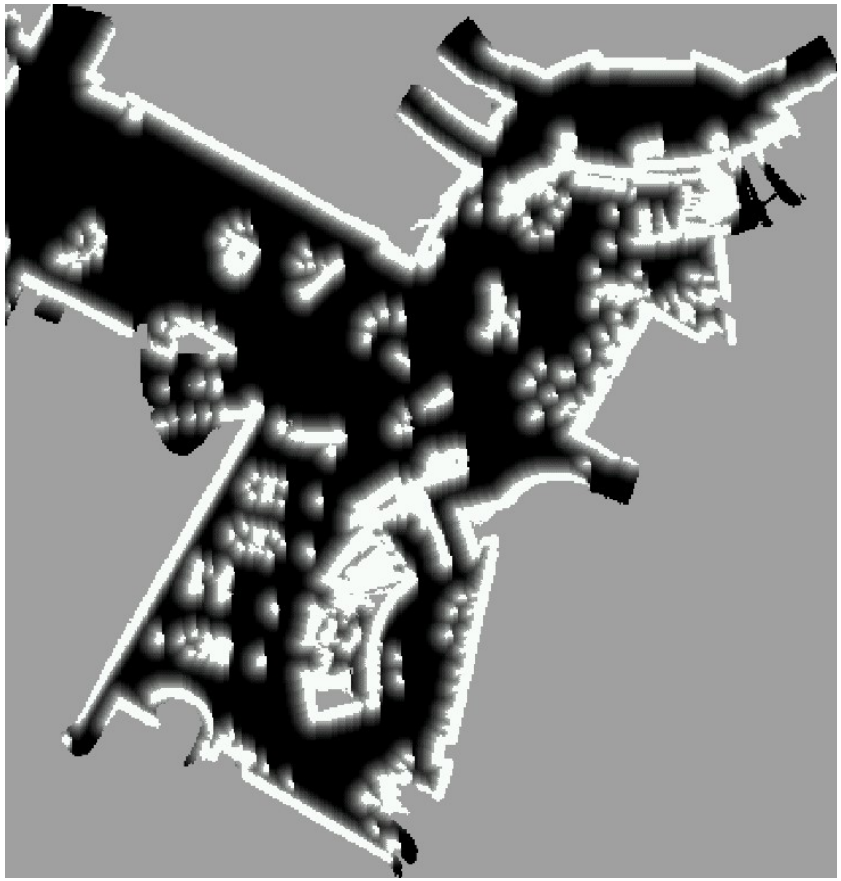
- Problemi:
  - Non modella short readings (ostacoli inattesi)
  - Assume mappa certa (non esplorazione mappa)
  - Può vedere attraverso

Mappe

Esplorazione: Uso combinato di occupancy grid per likelihood field



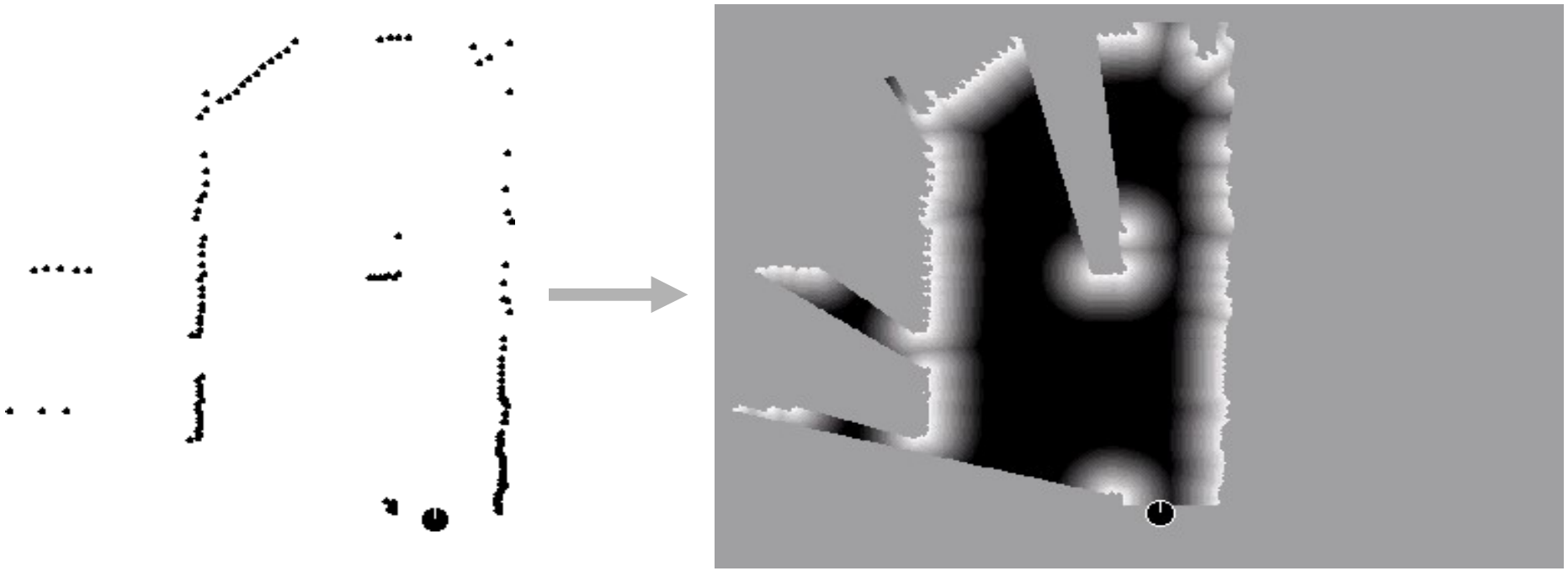
Occupancy grid map



Likelihood field

Scan Matching

- Estrazione del likelihood field dallo scan e uso per combinare scan differenti (mapping)



Altri modelli di prossimità

- **Map matching (sonar,laser):** generate small, local maps from sensor data and match local maps against global model.
- **Scan matching (laser):** map is represented by scan endpoints, match scan into this map.
- **Features (sonar, laser, vision):** Extract features such as doors, hallways from sensor data.

Features

I modelli dei sensori precedente basati su dati grezzi

Si può lavorare su features estratte dall'ambiente

Estrazione di features dal dato *z:* *f(z)*

Per range scans: linee, angoli, minimi locali che corrispondono a corridoi, angoli, oggetti etc.

Misure di Landmark

In molti casi le feature corrispondono ad oggetti nel mondo: landmark

- beacons attivi (*e.g.*, radio, GPS)
- passivi (*e.g.*, visual, retro-reflective)
- Occorre localizzarli nelle coordinate del robot: triangolazione è approccio standard

- Sensori forniscono
  - Distanza r
  - Direzione Φ
  - Segnatura s

Features

Estrazione di features: f(z)

Vettore di features variabile nel tempo:

$$f(z_t) = \{f_t^1, f_t^2, \ldots\} = \{\begin{pmatrix} r_t^1 \\ \phi_t^1 \\ s_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ \phi_t^1 \\ s_t^2 \end{pmatrix}, \ldots\}$$

Assunzione Markov

$$p(f(z_t) \mid x_t, m) = \prod_i p(r_t^i, \phi_t^i, s_t^i \mid x_t, m)$$

Features

Feature-based map: mappa come insieme di features localizzate $\{m_1, \ldots, m_n\}$

Con $m_{i,x}$ $m_{i,y}$ si indica la locazione x,y per la feature i-esima

Legame probabilistico tra feature j nella mappa globale e feature estratta nella mappa locale

$$\begin{pmatrix} r_t^i \\ \phi_t^i \\ s_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta \\ s_j \end{pmatrix} + \begin{pmatrix} \varepsilon_{\sigma_r^2} \\ \varepsilon_{\sigma_\phi^2} \\ \varepsilon_{\sigma_s^2} \end{pmatrix}$$

Likelihood con corrispondenza nota

Assumendo corrispondenza nota tra feature e landmark: $c_{i,t}$ in {1, …, N+1} con $c_{i,t} = j < N+1$, N+1 è il caso di non corrispondenza

Algorithm landmark_model_known_correspondence(f_t^i, c_t^i, x_t, m)**:**

$j = c_t^i$

$\hat{r} = \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2}$

$\hat{\phi} = \text{atan2}(m_{j,y} - y, m_{j,x} - x)$

$q = \text{prob}(r_t^i - \hat{r}, \sigma_r^2) \cdot \text{prob}(\phi_t^i - \hat{\phi}, \sigma_\phi^2) \cdot \text{prob}(s_t^i - s_j, \sigma_s^2)$

return q

Calcolo del likelihood della misurazione f_{it} da x_t data la mappa m e la corrispondenza c_{it}

Modello a landmark

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. $\hat{a} = \text{atan2}(m_y(i) - y, m_x(i) - x) - \theta$

4. $p_{\text{det}} = \text{prob}(\hat{d} - d, \varepsilon_d) \cdot \text{prob}(\hat{\alpha} - \alpha, \varepsilon_\alpha)$

5. Return $z_{\text{det}} p_{\text{det}} + z_{\text{fp}} P_{\text{uniform}}(z \mid x, m)$

Riassunto modello dei sensori

- Modellare esplicitamente l'incertezza del sensing.

- In molti casi i modelli possono essere trovati con l'approccio seguente:
  1. Determinare modelli parametrici di rumore.
  2. Analizzare la sorgente di rumore.
  3. Aggiungere rumore ai parameteri (eventualmente un mix di densità di rumore).
  4. Apprendimento (e verifica) di parameteri del modello nel dato.
  5. Likelihood delle misure è dato da "confrontando probabilisticamente" l'attuale con le misure attese.

- Questo vale anche per il modello di moto.