

Robotica Probabilistica

SLAM

Il problema SLAM

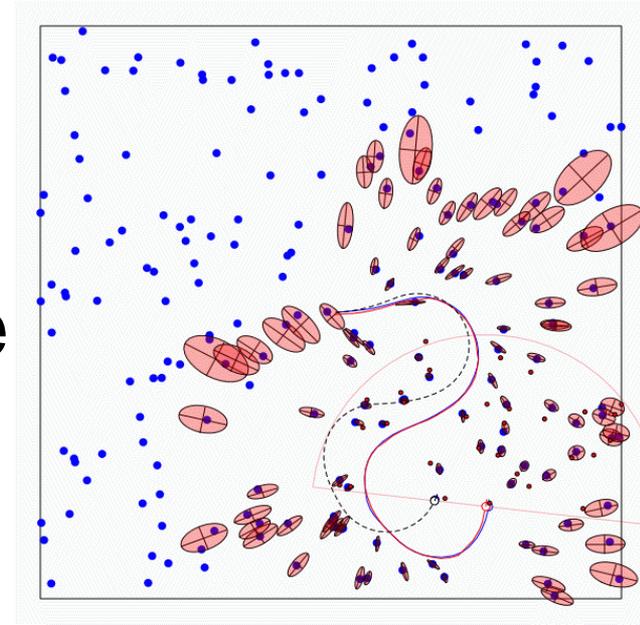
Un robot esplora un ambiente sconosciuto e statico.

Dato:

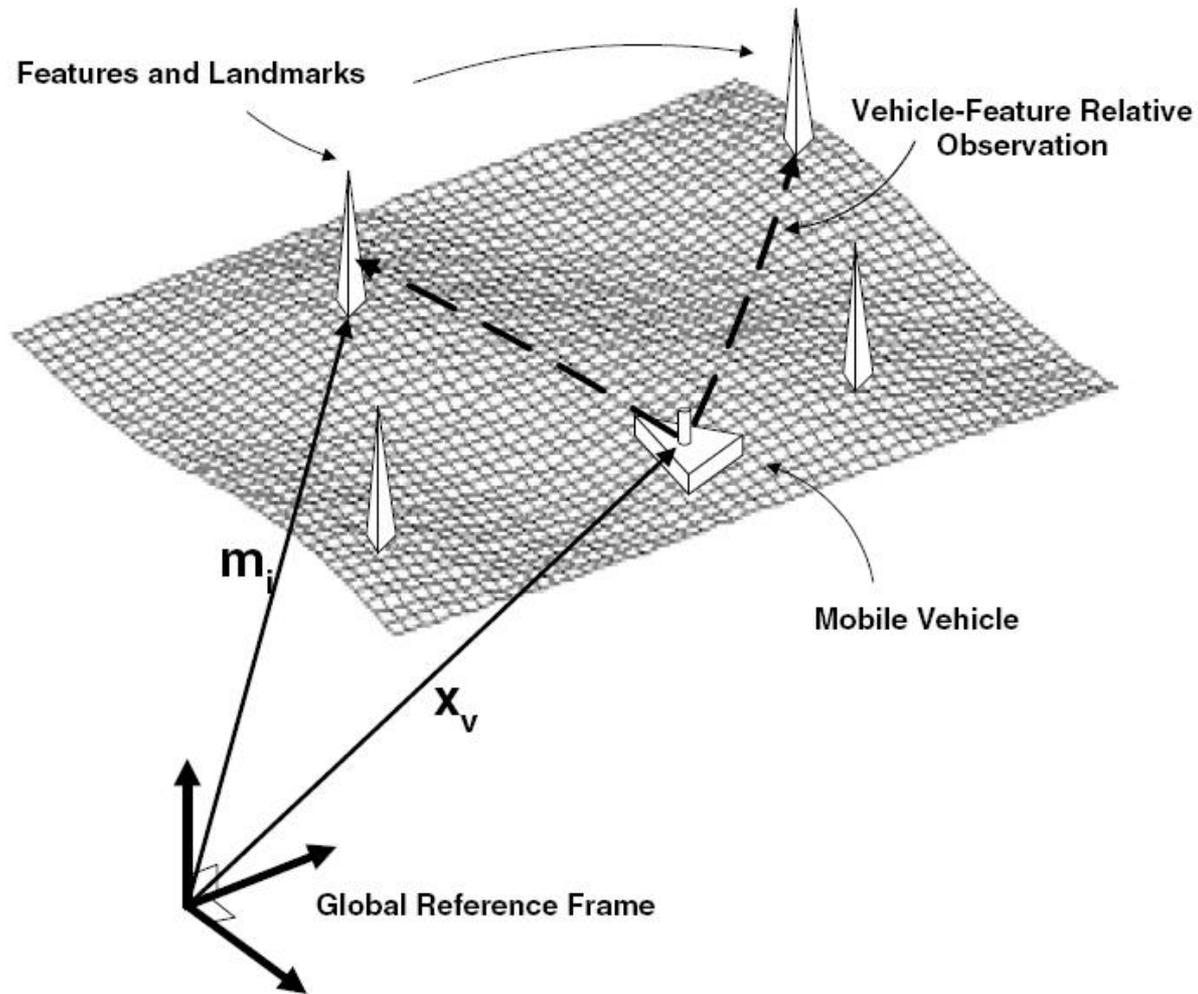
- Il controllo del robot
- Osservazione di features vicine

Stima:

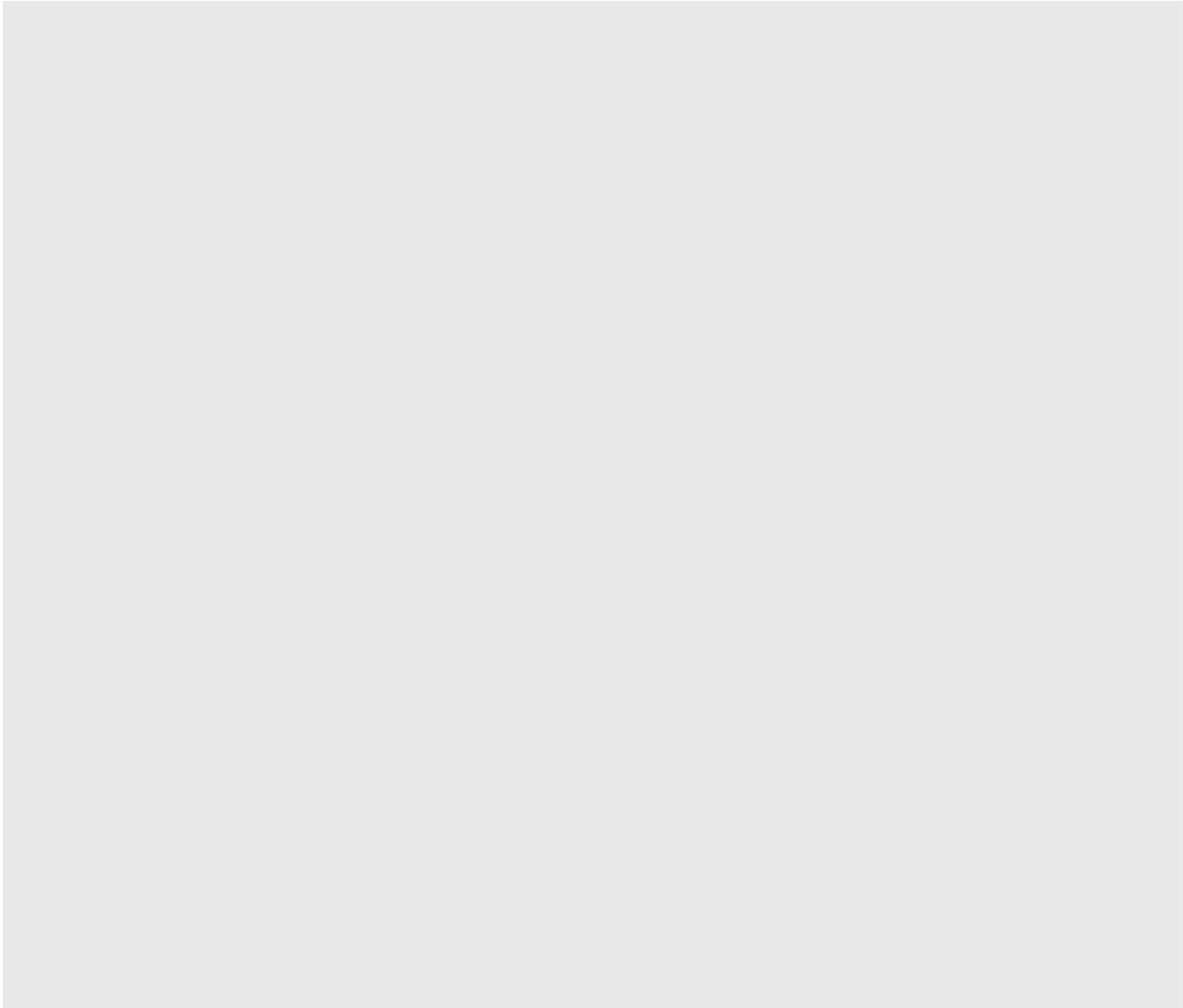
- Mappa delle features
- Percorso del robot



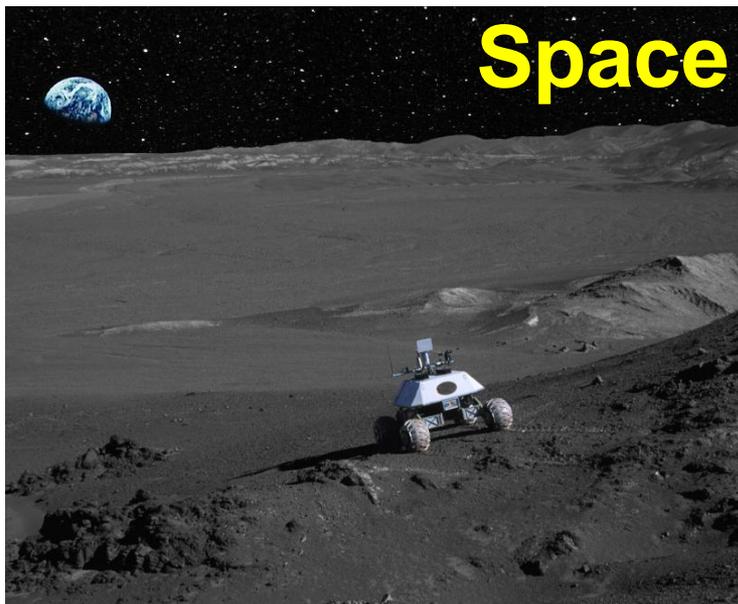
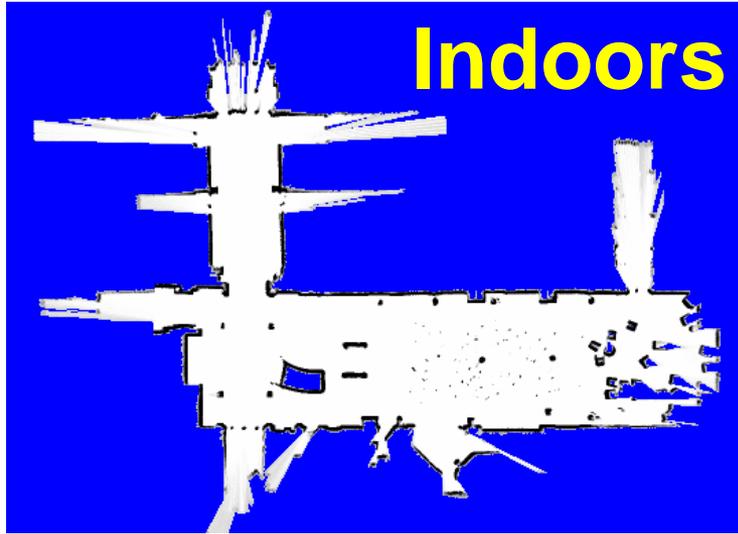
Struttura del problema di Landmark-based SLAM



Mapping with Raw Odometry

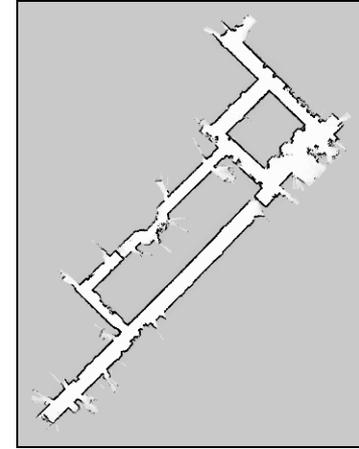


Applicazioni dello SLAM



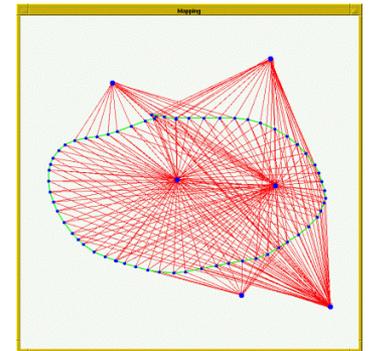
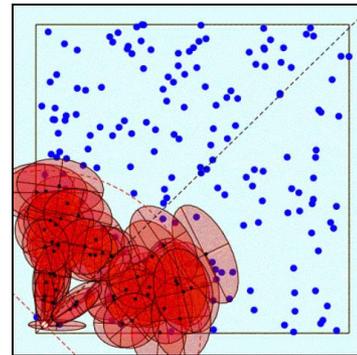
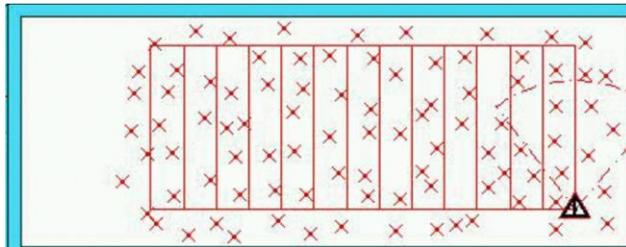
Rappresentazioni

- Grid maps o scans



[Lu & Milios, 97; Gutmann, 98; Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01;...]

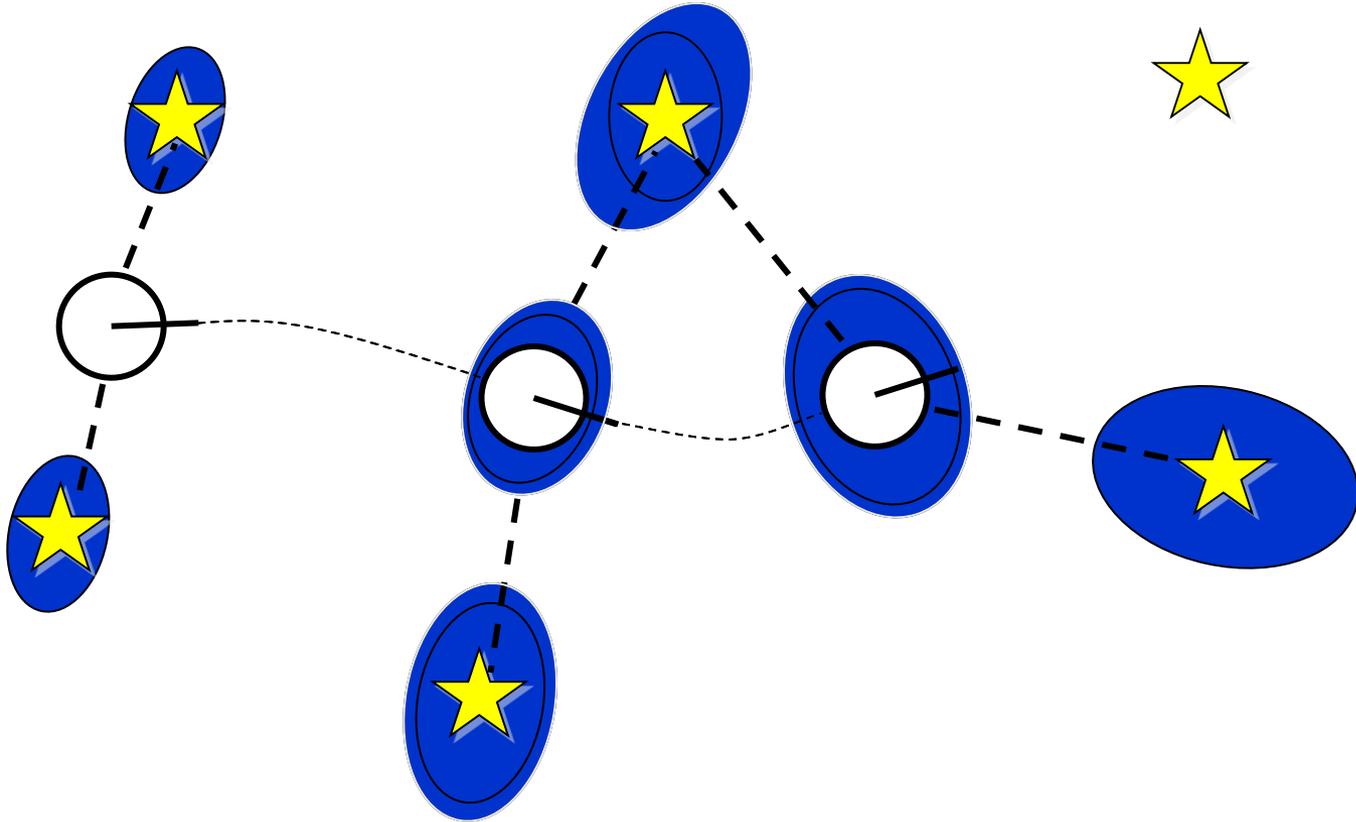
- Landmark-based



[Leonard et al., 98; Castelanos et al., 99; Dissanayake et al., 2001; Montemerlo et al., 2002;...]

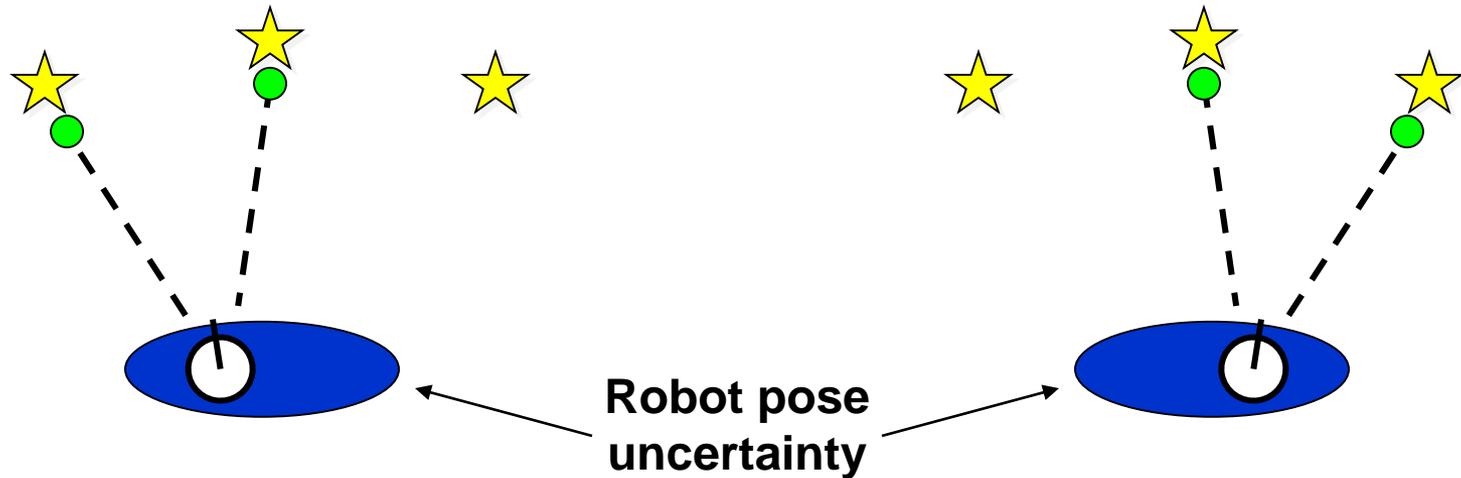
Perché SLAM è difficile?

SLAM: posizione e mappa del robot **sconosciuti**



Errori di percorso sono correlati agli errori nella mappa

Perché SLAM è difficile?



- Nel mondo reale, il mapping tra osservazioni e landmarks è sconosciuto
- Associazioni sbagliate possono avere conseguenze catastrofiche
- Errori di posa correlati a associazioni di dati

SLAM:

Simultaneous Localization and Mapping

- Full SLAM: Stima di percorso e mappa

$$p(x_{1:t}, m | z_{1:t}, u_{1:t})$$

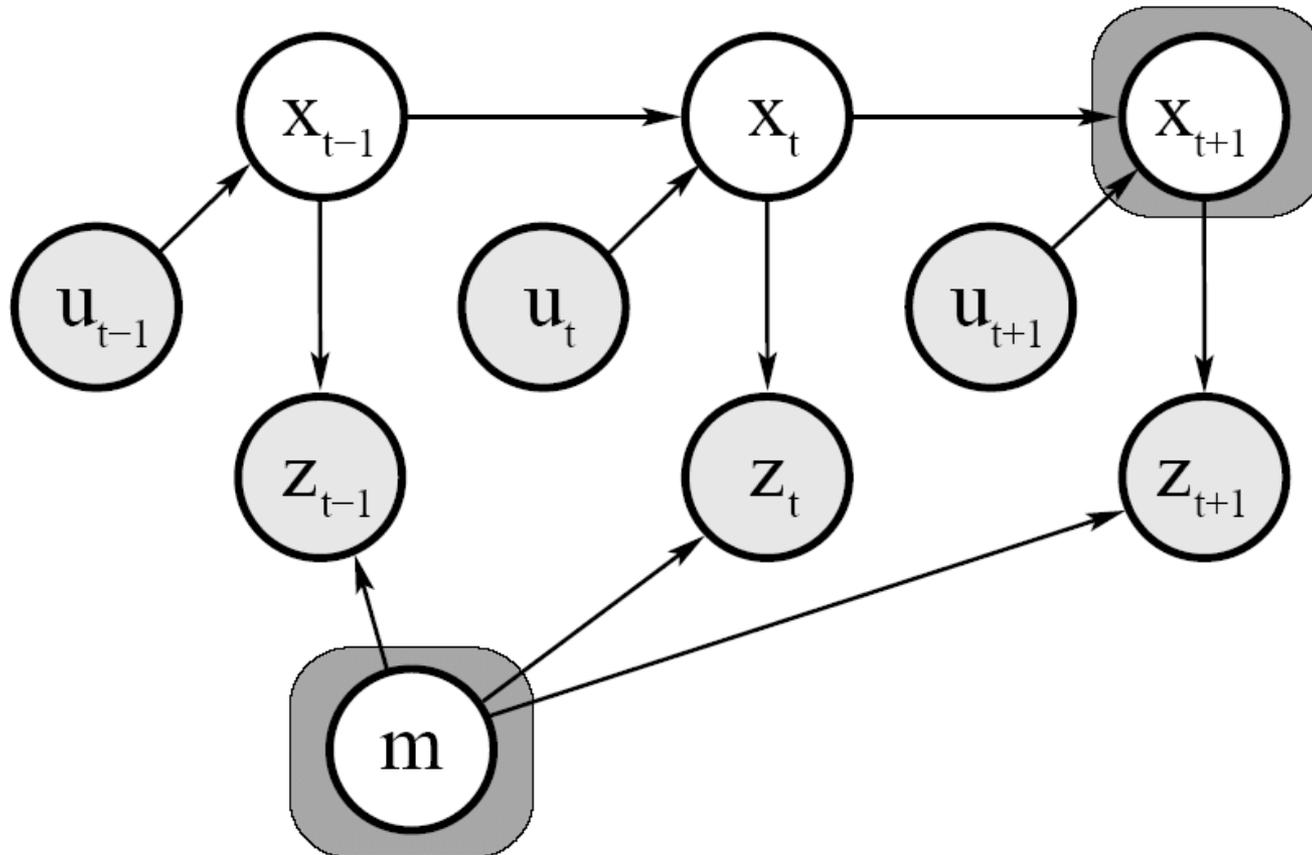
- Online SLAM:

$$p(x_t, m | z_{1:t}, u_{1:t}) = \int \int \dots \int p(x_{1:t}, m | z_{1:t}, u_{1:t}) dx_1 dx_2 \dots dx_{t-1}$$

Integrazioni fatte una alla volta

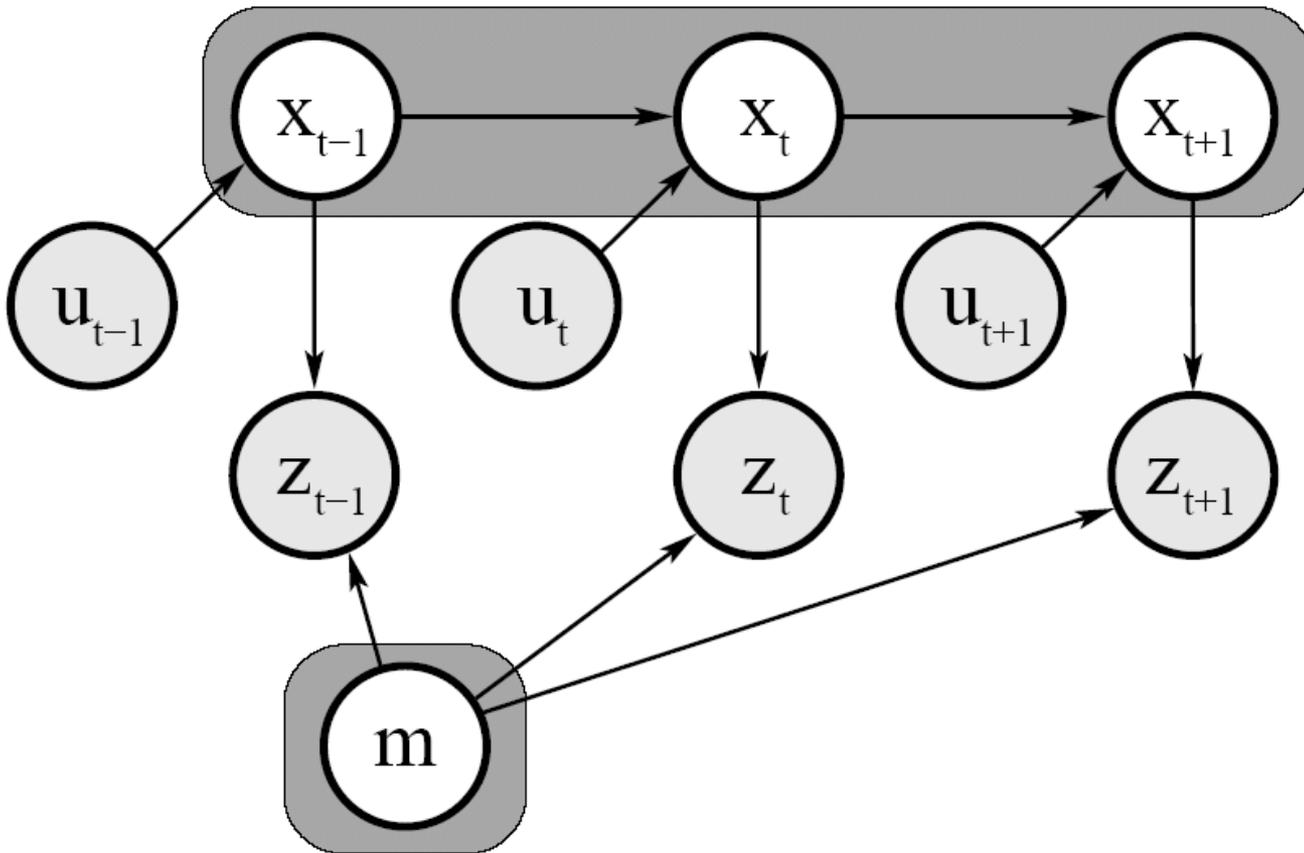
Stima della posa e mappa più recente

Graphical Model di Online SLAM:



$$p(x_t, m | z_{1:t}, u_{1:t}) = \int \int \dots \int p(x_{1:t}, m | z_{1:t}, u_{1:t}) dx_1 dx_2 \dots dx_{t-1}$$

Graphical Model of Full SLAM:



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

Tecniche per la generazione di mappe consistenti

- Scan matching
- EKF SLAM
- Fast-SLAM
- Probabilistic mapping with a single map and a posterior about poses
Mapping + Localization
- Graph-SLAM, SEIFs

Scan Matching

Massimizza il likelihood della i-sima posa e mappa rispetto alla (i-1)-sima posa e mappa.

$$\hat{x}_t = \arg \max_{x_t} \left\{ p(z_t | x_t, \hat{m}^{[t-1]}) \cdot p(x_t | u_{t-1}, \hat{x}_{t-1}) \right\}$$

current measurement

map constructed so far

robot motion

Calcola la mappa $\hat{m}^{[t]}$ secondo il "mapping con posa nota" basata sulle pose e le osservazioni.

Scan Matching

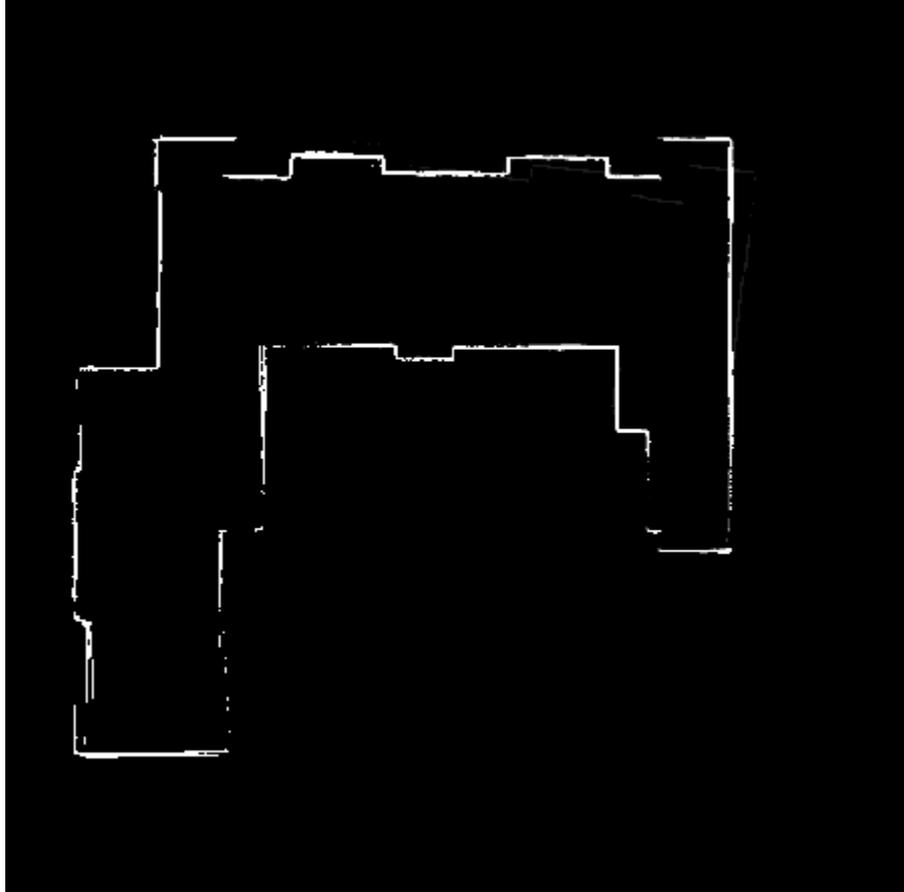
Tecnica di scan matching.

- Non necessita di landmark
- E' robusta per piccole modifiche nell'ambiente
- Può lavorare in ogni tipo di ambiente chiuso

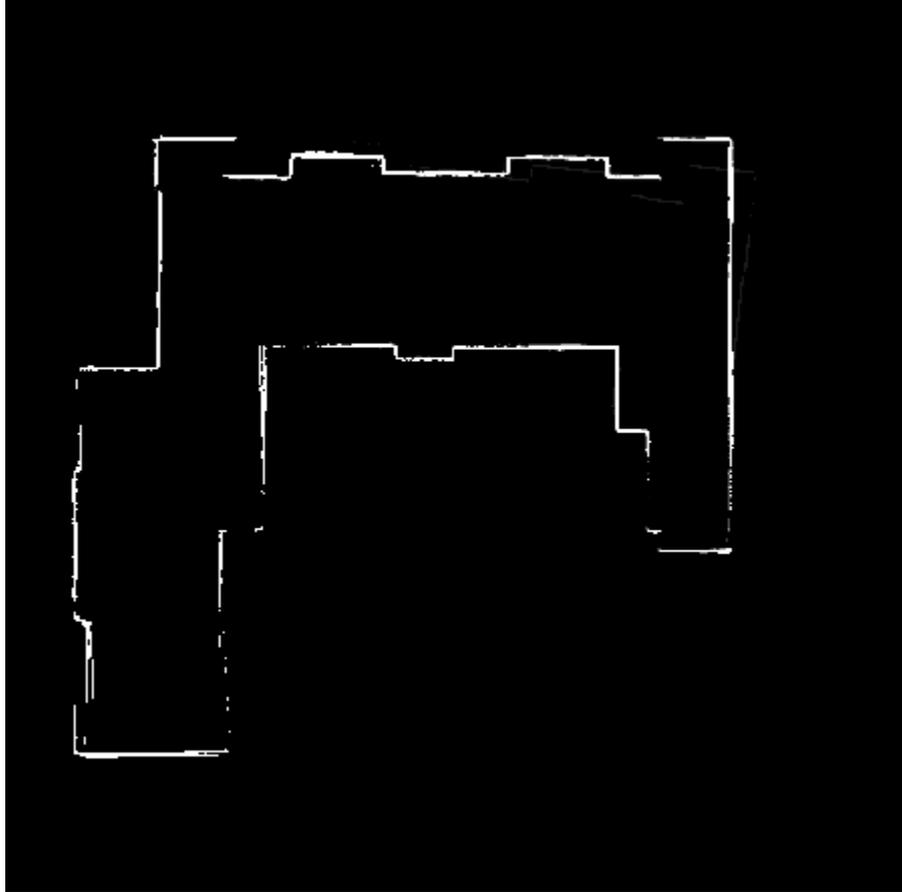
Scan Matching

- La tecnica si basa sul confronto tra immagini.
- Date due immagini, attraverso un algoritmo di scan matching si calcola la rotazione e la traslazione tra queste.
- Questi valori usati per correggere i dati odometrici e risolvere quindi il problema dello SLAM.

Calcolo della rotazione



Calcolo della traslazione



Algoritmo del Filtro di Kalman

1. Algorithm **Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):
2. Prediction:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$
5. Correction:
6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$
9. Return μ_t, Σ_t

(E)KF-SLAM

- Primo algoritmo di SLAM
- Mappe Feature-based: pochi landmark (< 1000) e non troppo ambigui
- Rumore Gaussiano: non eccessivo
- Solo con misure di landmark

(E)KF-SLAM (corrispondenze note)

- Stima la posizione del robot e quella dei landmark
- Si assumono corrispondenze note
- Si aggiungono le coordinate dei landmark nello state vector

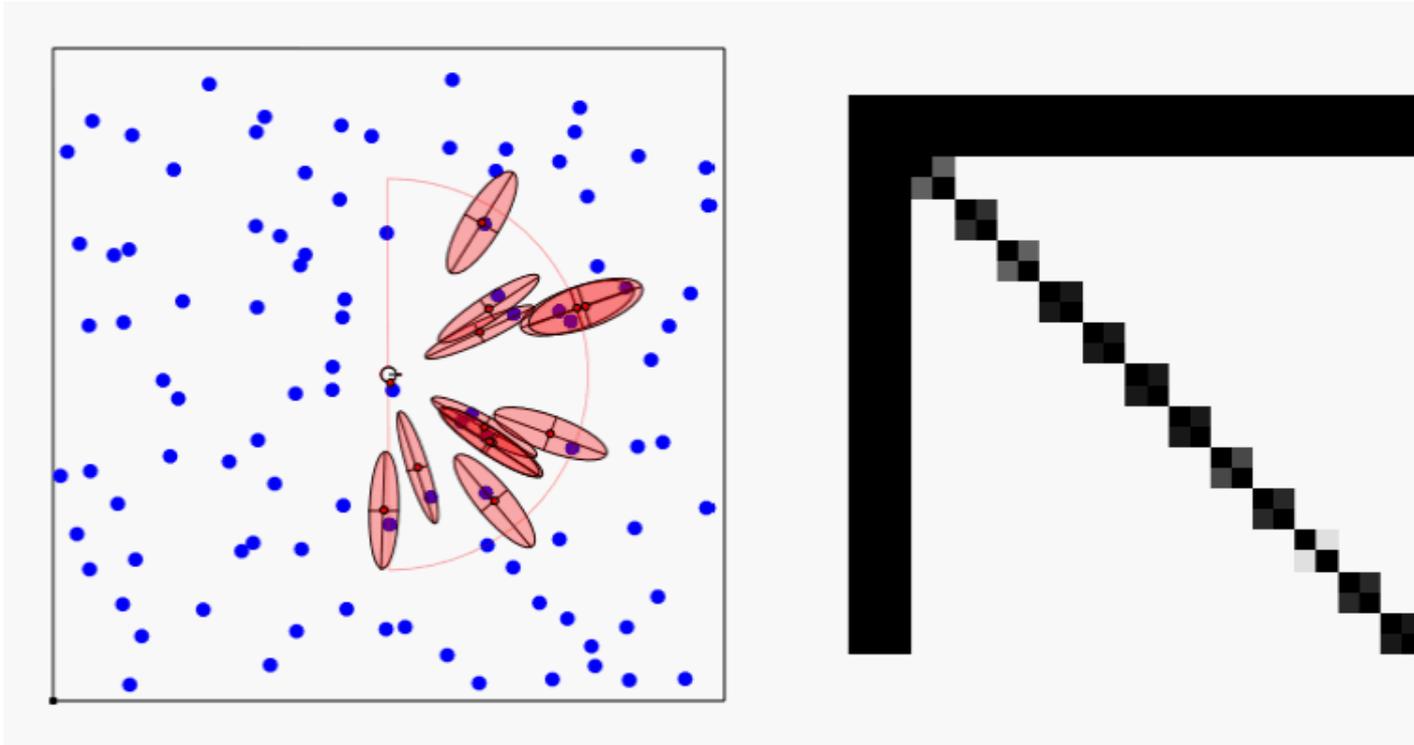
(E)KF-SLAM

- Mappa con N landmark: (3+2N)-Gaussiana

$$Bel(x_t, m_t) = \left(\begin{array}{c} x \\ y \\ \theta \\ l_1 \\ l_2 \\ \vdots \\ l_N \end{array} \right), \left(\begin{array}{ccc|cccc} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xl_1} & \sigma_{xl_2} & \cdots & \sigma_{xl_N} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} & \sigma_{yl_1} & \sigma_{yl_2} & \cdots & \sigma_{yl_N} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 & \sigma_{\theta l_1} & \sigma_{\theta l_2} & \cdots & \sigma_{\theta l_N} \\ \hline \sigma_{xl_1} & \sigma_{yl_1} & \sigma_{\theta l_1} & \sigma_{l_1}^2 & \sigma_{l_1 l_2} & \cdots & \sigma_{l_1 l_N} \\ \sigma_{xl_2} & \sigma_{yl_2} & \sigma_{\theta l_2} & \sigma_{l_1 l_2} & \sigma_{l_2}^2 & \cdots & \sigma_{l_2 l_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{xl_N} & \sigma_{yl_N} & \sigma_{\theta l_N} & \sigma_{l_1 l_N} & \sigma_{l_2 l_N} & \cdots & \sigma_{l_N}^2 \end{array} \right)$$

- Può gestire centinaia dimensioni

Soluzione Classica – EKF



Blue path = true path **Red path** = estimated path **Black path** = odometry

- Approssima il post di SLAM con una Gaussiana ad alta dimensione [Smith & Cheesman, 1986] ...
- **Associazione di dati con singola ipotesi**

(E) KF-SLAM

- Posa: robot+ landmark

$$\begin{aligned}
 y_t &= \begin{pmatrix} x_t \\ m \end{pmatrix} \\
 &= (x \ y \ \theta \ m_{1,x} \ m_{1,y} \ s_1 \ m_{2,x} \ m_{2,y} \ s_2 \ \dots \ m_{N,x} \ m_{N,y} \ s_N)^T
 \end{aligned} \tag{10.7}$$

- Situazione Iniziale: Valore Medio e Covarianza

$$\begin{aligned}
 \mu_0 &= (0 \ 0 \ 0 \ \dots \ 0)^T \\
 \Sigma_0 &= \begin{pmatrix} 0 & 0 & 0 & \infty & \dots & \infty \\ 0 & 0 & 0 & \infty & \dots & \infty \\ 0 & 0 & 0 & \infty & \dots & \infty \\ \infty & \infty & \infty & \infty & \dots & \infty \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \infty & \infty & \infty & \infty & \dots & \infty \end{pmatrix}
 \end{aligned}$$

- Movimento:

$$y_t = y_{t-1} + \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \theta + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \theta - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t + \gamma_t \Delta t \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

(E)KF-SLAM

Movimento compatto:

$$y_t = y_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \theta + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \theta - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t + \gamma_t \Delta t \end{pmatrix}$$

Matrice con dim $3N + N$

$$F_x = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & \underbrace{0 & \dots & 0}_{3N \text{ columns}} \end{pmatrix}$$

Introducendo il rumore:

$$y_t = \underbrace{y_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \theta + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \theta - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \omega \Delta t \end{pmatrix}}_{g(u_t, y_{t-1})} + \mathcal{N}(0, F_x^T R_t F_x)$$

(E)KF-SLAM

Applicando l'approssimazione di Taylor

$$g(u_t, y_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t (y_{t-1} - \mu_{t-1})$$

Con G_t Jacobiano di g in u_t

Riduzione dello Jacobiano

$$G_t = I + F_x^T g_t F_x$$

$$g_t = \begin{pmatrix} 0 & 0 & \frac{v_t}{\omega_t} \cos \mu_{t-1, \theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1, \theta} + \omega_t \Delta t) \\ 0 & 0 & \frac{v_t}{\omega_t} \sin \mu_{t-1, \theta} - \frac{v_t}{\omega_t} \sin(\mu_{t-1, \theta} + \omega_t \Delta t) \\ 0 & 0 & 0 \end{pmatrix}$$

(E)KF-SLAM

Update del Moto

1: **Algorithm EKF_SLAM_known_correspondences**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, c_t$):

2:
$$F_x = \begin{pmatrix} 1 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & \underbrace{0 \dots 0}_{2N} \end{pmatrix}$$

3:
$$\bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \mu_{t-1, \theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1, \theta} + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \mu_{t-1, \theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1, \theta} + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}$$

4:
$$G_t = I + F_x^T \begin{pmatrix} 0 & 0 & \frac{v_t}{\omega_t} \cos \mu_{t-1, \theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1, \theta} + \omega_t \Delta t) \\ 0 & 0 & \frac{v_t}{\omega_t} \sin \mu_{t-1, \theta} - \frac{v_t}{\omega_t} \sin(\mu_{t-1, \theta} + \omega_t \Delta t) \\ 0 & 0 & 0 \end{pmatrix} F_x$$

5:
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + F_x^T R_t F_x$$

Occorre l'update di misura

(E)KF-SLAM

- Dato il modello di misura

$$z_t^i = \underbrace{\begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta \\ m_{j,s} \end{pmatrix}}_{h(y_t, j)} + \mathcal{N}\left(0, \underbrace{\begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_\phi & 0 \\ 0 & 0 & \sigma_s \end{pmatrix}}_{Q_t}\right) \quad (10)$$

- Dove: x, y, θ posa del robot, i indice del landmark osservato al tempo t , $j = c_{i,t}$ landmark corrispondente
- Applicando l'approssimazione lineare

$$h(y_t, j) \approx h(\bar{\mu}_t, j) + H_t^i (y_t - \bar{\mu}_t)$$

- Con $H_{i,t}$ derivata su tutto lo spazio di stato, ma dipende solo dalla posa del robot e del landmark

$$H_t^i = h_t^i F_{x,j} \quad h_t^i = \begin{pmatrix} \frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q_t}} & \frac{y_t - \bar{\mu}_{t,y}}{\sqrt{q_t}} & 0 & \frac{\bar{\mu}_{t,x} - m_{j,x}}{\sqrt{q_t}} & \frac{\bar{\mu}_{t,y} - y_t}{\sqrt{q_t}} & 0 \\ \frac{\bar{\mu}_{t,y} - y_t}{q_t} & \frac{m_{j,x} - \bar{\mu}_{t,x}}{q_t} & -1 & \frac{y_t - \bar{\mu}_{t,y}}{q_t} & \frac{\bar{\mu}_{t,x} - m_{j,x}}{q_t} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$q_t = (m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2$$

(E)KF-SLAM

- Calcolo Jacobiano: $H_t^i = h_t^i F_{x,j}$

Jacobiano in μ , rispetto ad x_t ed m_j

$$h_t^i = \begin{pmatrix} \frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q_t}} & \frac{y_t - \bar{\mu}_{t,y}}{\sqrt{q_t}} & 0 & \frac{\bar{\mu}_{t,x} - m_{j,x}}{\sqrt{q_t}} & \frac{\bar{\mu}_{t,y} - y_t}{\sqrt{q_t}} & 0 \\ \frac{\bar{\mu}_{t,y} - y_t}{q_t} & \frac{m_{j,x} - \bar{\mu}_{t,x}}{q_t} & -1 & \frac{y_t - \bar{\mu}_{t,y}}{q_t} & \frac{\bar{\mu}_{t,x} - m_{j,x}}{q_t} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Scalare: $q_t = (m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2$

Matrice per
rimappare
sullo spazio:

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0 \dots 0 & 0 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 & 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & 0 \dots 0 & 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 1 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & \underbrace{0 \dots 0}_{2j-2} & 0 & 0 & 1 & \underbrace{0 \dots 0}_{2N-2j} \end{pmatrix}$$

- Inizializzazione pos di landmark (in pos attesa)

$$\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \\ \bar{\mu}_{j,s} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \\ s_t^i \end{pmatrix} + r_t^i \begin{pmatrix} \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \\ 0 \end{pmatrix}$$

(E)KF-SLAM

• Algoritmo: update di misura

Inizializza la posizione
dei landmark

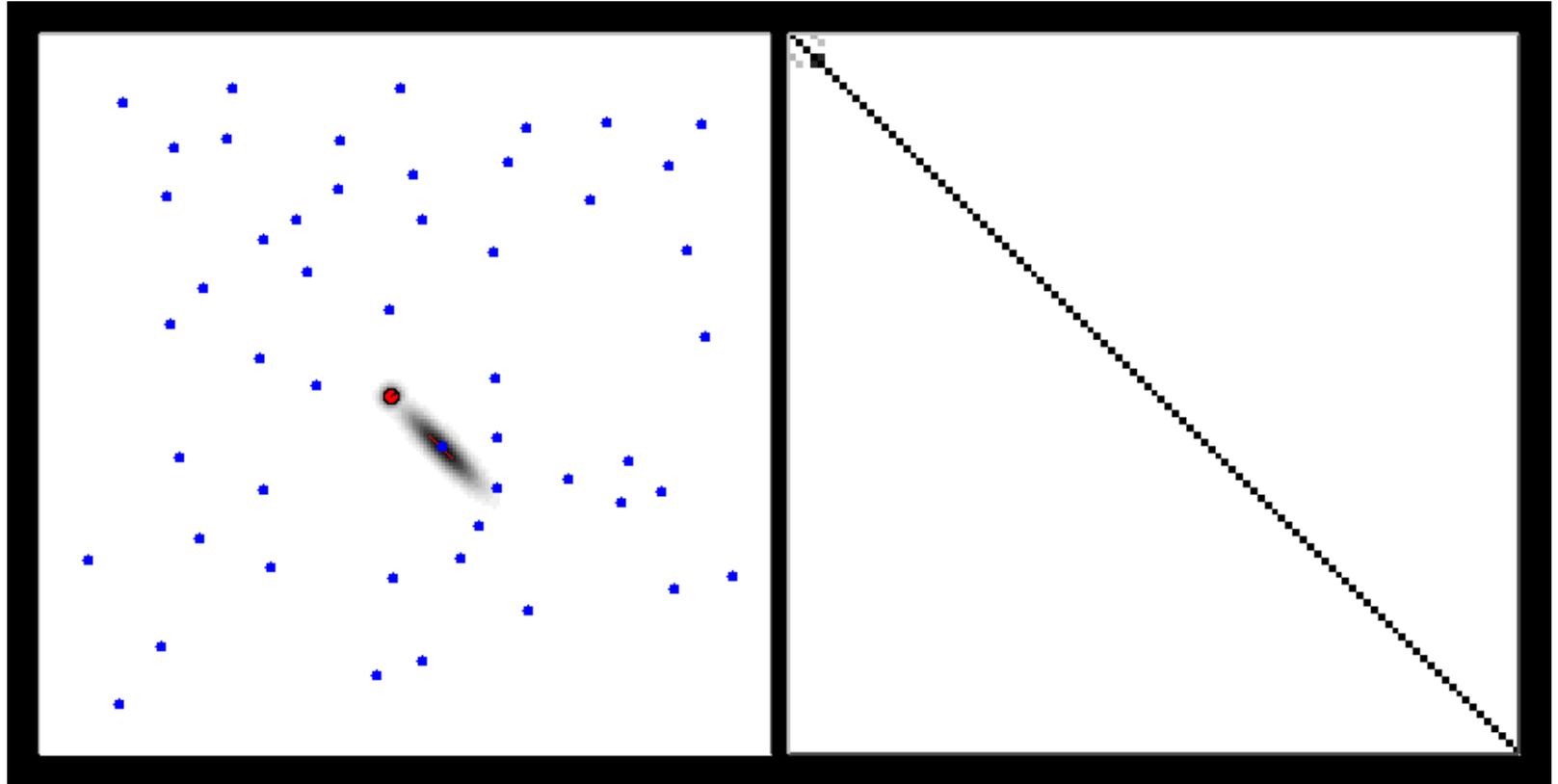
Calcolo misura attesa

Guadagno di Kalman

Matrice non sparsa

```
6:  $Q_t = \begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_\phi & 0 \\ 0 & 0 & \sigma_s \end{pmatrix}$ 
7: for all observed features  $z_t^i = (r_t^i \ \phi_t^i \ s_t^i)^T$  do
8:    $j = c_t^i$ 
9:   if landmark  $j$  never seen before
10:      $\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \\ \bar{\mu}_{j,s} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \\ s_t^i \end{pmatrix} + r_t^i \begin{pmatrix} \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \\ 0 \end{pmatrix}$ 
11:   endif
12:    $\delta = \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix}$ 
13:    $q = \delta^T \delta$ 
14:    $\hat{z}_t^i = \begin{pmatrix} \sqrt{q} \\ \text{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \\ \bar{\mu}_{j,s} \end{pmatrix}$ 
15:    $F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0 \dots 0 & 0 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 & 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & 0 \dots 0 & 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 1 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & \underbrace{0 \dots 0}_{2j-2} & 0 & 0 & 1 & \underbrace{0 \dots 0}_{2N-2j} \end{pmatrix}$ 
16:    $H_t^i = \frac{1}{q} \begin{pmatrix} \sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & -\sqrt{q}\delta_x & \sqrt{q}\delta_y & 0 \\ \delta_y & \delta_x & -1 & -\delta_y & -\delta_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} F_{x,j}$ 
17:    $K_t^i = \bar{\Sigma}_t H_t^{iT} (H_t^i \bar{\Sigma}_t H_t^{iT} + Q_t)^{-1}$ 
18:   endfor
19:    $\mu_t = \bar{\mu}_t + \sum_i K_t^i (z_t^i - \hat{z}_t^i)$ 
20:    $\Sigma_t = (I - \sum_i K_t^i H_t^i) \bar{\Sigma}_t$ 
21:   return  $\mu_t, \Sigma_t$ 
```

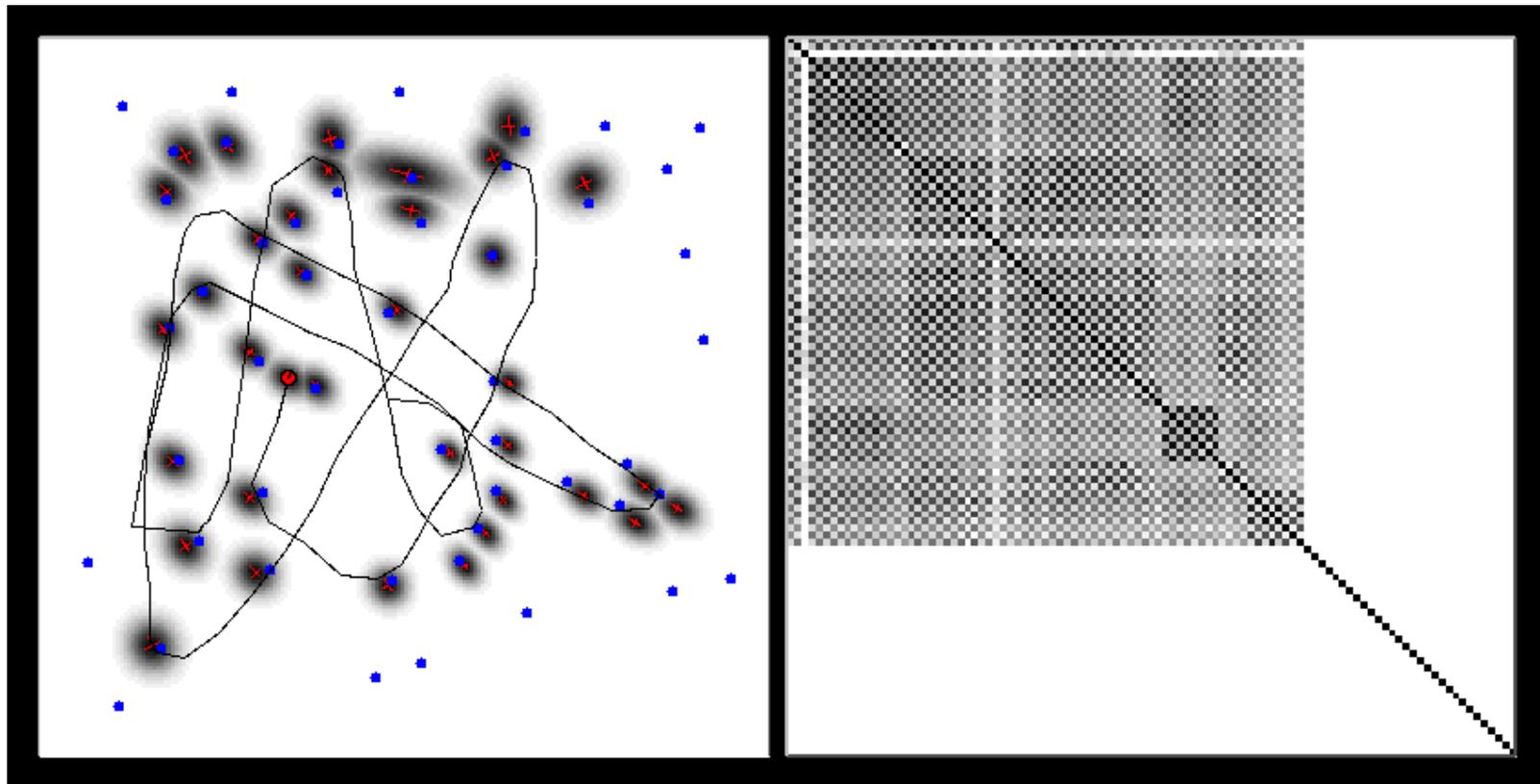
EKF-SLAM



Mappa

Matrice di Correlazione:
covarianza normalizzata

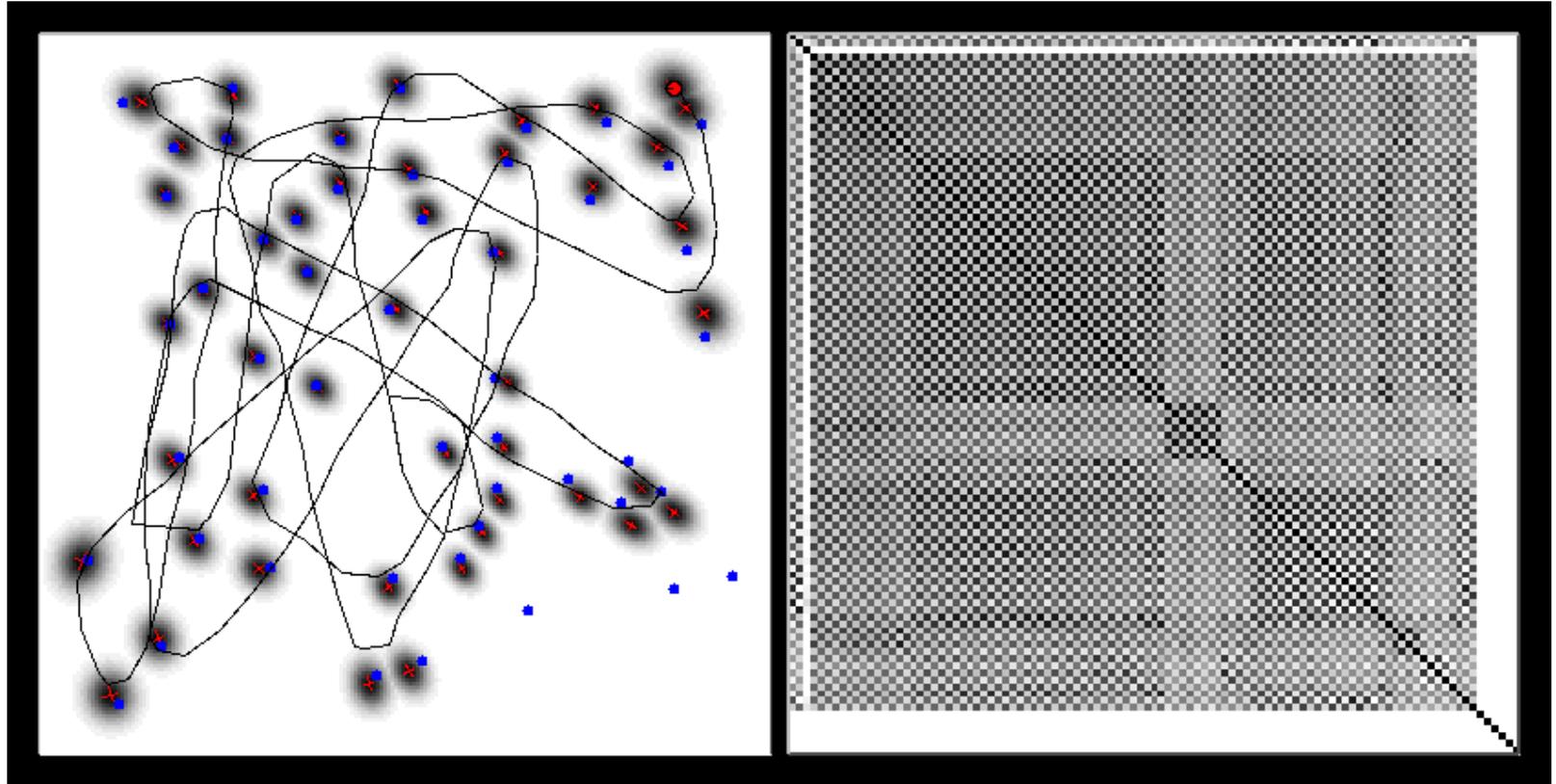
EKF-SLAM



Mappa

Matrice di Correlazione

EKF-SLAM



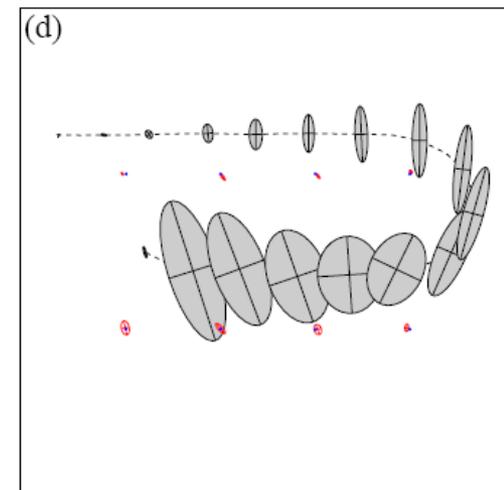
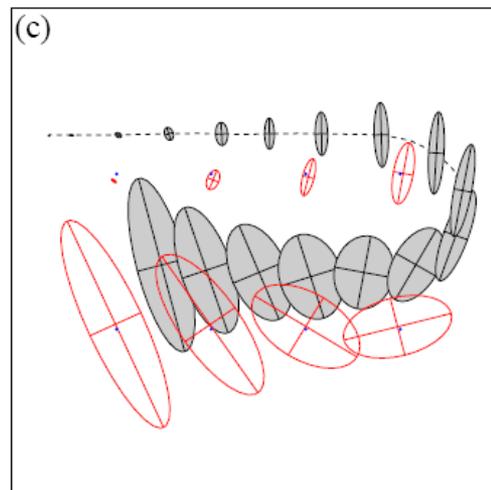
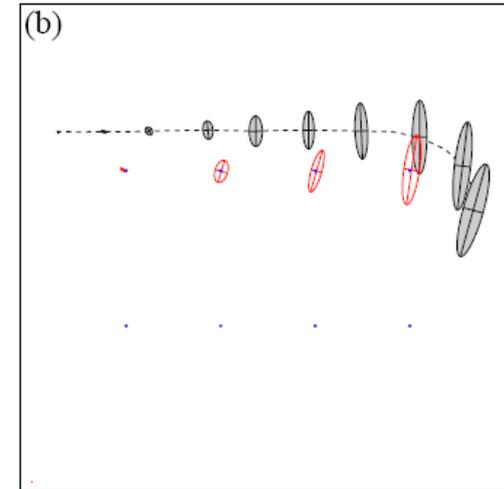
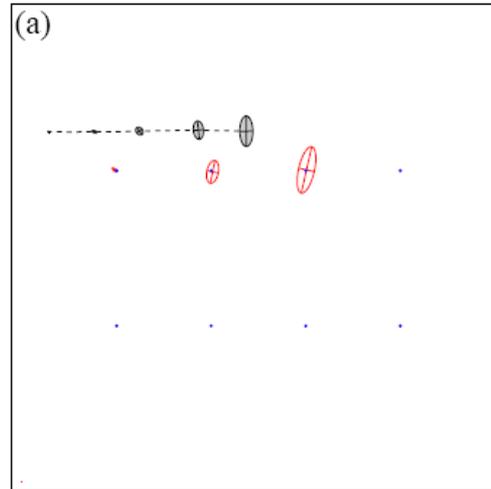
Mappa

Matrice di Correlazione

Dopo qualche iterazione la mappa è correlata

(E)KF-SLAM

- Incertezza aumenta (a-c) su posa e landmark;
- Appena rivede un landmark già visto (d), riduce l'incertezza;
- Riduce l'incertezza di tutte le posizioni;
- Matrice di cov lega la posizione del robot e le altre posizioni.



Propert  del KF-SLAM (Caso Lineare) [Dissanayake et al., 2001]

Teorema:

Il determinante di ogni sottomatrice della matrice di covarianza della mappa decresce monotonamente per successive osservazioni.

Teorema:

Al limite la stima dei landmark diventa totalmente correlata.

(E)KF-SLAM

(corrispondenze non note)

- Se corrispondenze non note occorre uno stimatore incrementale ML (maximum likelihood) per stabilire tali corrispondenze
- L'algoritmo precedente può essere adattato
- Numero di landmark corrente N_t dimensione della mappa.

(E)KF-SLAM

- Update del moto è lo stesso
- Update misura diverso:
 - Si ipotizza nuovo landmark e si inizializza
 - Si effettuano update su tutti i landmark
 - Si verifica la creazione di un nuovo landmark: se misura Mahalanobis da tutti i landmark dista più di una soglia
 - Se nuovo landmark, la mappa viene estesa e si calcola nuovo valor medio e covarianza

EKF-SLAM

- Corrispondenze non note

1: **Algorithm EKF-SLAM**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, N_{t-1}$):

2: $N_t = N_{t-1}$

3:
$$F_x = \begin{pmatrix} 1 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & 0 \dots 0 \end{pmatrix}$$

4:
$$\bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \mu_{t-1, \theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1, \theta} + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \mu_{t-1, \theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1, \theta} + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}$$

5:
$$G_t = I + F_x^T \begin{pmatrix} 0 & 0 & \frac{v_t}{\omega_t} \cos \mu_{t-1, \theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1, \theta} + \omega_t \Delta t) \\ 0 & 0 & \frac{v_t}{\omega_t} \sin \mu_{t-1, \theta} - \frac{v_t}{\omega_t} \sin(\mu_{t-1, \theta} + \omega_t \Delta t) \\ 0 & 0 & 0 \end{pmatrix} F_x$$

6:
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + F_x^T R_t F_x$$

7:
$$Q_t = \begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_\phi & 0 \\ 0 & 0 & \sigma_s \end{pmatrix}$$

8: for all observed features $z_t^i = (r_t^i \ \phi_t^i \ s_t^i)^T$ do

9:
$$\begin{pmatrix} \bar{\mu}_{N_t+1, x} \\ \bar{\mu}_{N_t+1, y} \\ \bar{\mu}_{N_t+1, s} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t, x} \\ \bar{\mu}_{t, y} \\ s_t^i \end{pmatrix} + r_t^i \begin{pmatrix} \cos(\phi_t^i + \bar{\mu}_{t, \theta}) \\ \sin(\phi_t^i + \bar{\mu}_{t, \theta}) \\ 0 \end{pmatrix}$$

10: for $k = 1$ to $N_t + 1$ do

11:
$$\delta_k = \begin{pmatrix} \delta_{k, x} \\ \delta_{k, y} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{k, x} - \bar{\mu}_{t, x} \\ \bar{\mu}_{k, y} - \bar{\mu}_{t, y} \end{pmatrix}$$

12:
$$q_k = \delta_k^T \delta_k$$

Ipotesi di nuovo landmark e inizializzazione

Aggiornamento landmark

EKF-SLAM

- Corrispondenze non note

```

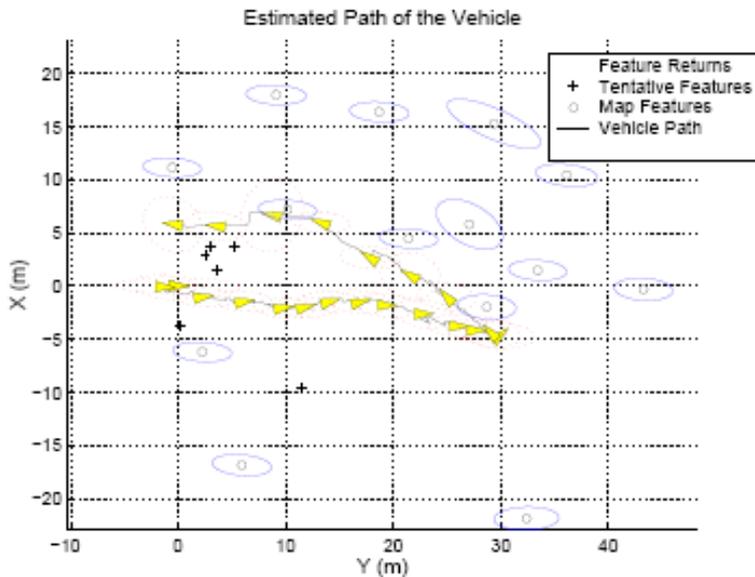
13:    $\hat{z}_t^k = \begin{pmatrix} \sqrt{q_k} \\ \text{atan2}(\delta_{k,y}, \delta_{k,x}) - \bar{\mu}_{t,\theta} \\ \bar{\mu}_{k,s} \end{pmatrix}$ 
aggiornamento landmark con 14:    $F_{x,k} = \begin{pmatrix} 1 & 0 & 0 & 0 \dots 0 & 0 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 & 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & 0 \dots 0 & 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 1 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 0 & 1 & 0 \dots 0 \end{pmatrix}$ 
ML
Distanza 15:    $H_t^k = \frac{1}{q_k} \begin{pmatrix} \sqrt{q_k} \delta_{k,x} & -\sqrt{q_k} \delta_{k,y} & 0 & -\sqrt{q_k} \delta_{k,x} & \sqrt{q_k} \delta_{k,y} & 0 \\ \delta_{k,y} & \delta_{k,x} & -1 & -\delta_{k,y} & -\delta_{k,x} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} F_{x,k}$ 
Mahalanobis 16:    $\Psi_k = H_t^k \bar{\Sigma}_t [H_t^k]^T + Q_t$ 
17:    $\pi_k = (z_t^i - \hat{z}_t^k)^T \Psi_k^{-1} (z_t^i - \hat{z}_t^k)$  Misura di dissimilarità
18:   endfor
19:    $\pi_{N_t+1} = \alpha$ 
20:    $j(i) = \underset{k}{\text{argmin}} \pi_k$ 
21:    $N_t = \max\{N_t, j(i)\}$ 
Eventuale 22:    $K_t^i = \bar{\Sigma}_t [H_t^{j(i)}]^T \Psi_{j(i)}^{-1}$ 
incremento dim 23:   endfor
della matrice 24:    $\mu_t = \bar{\mu}_t + \sum_i K_t^i (z_t^i - \hat{z}_t^{j(i)})$ 
25:    $\Sigma_t = (I - \sum_i K_t^i H_t^{j(i)}) \bar{\Sigma}_t$ 
26:   return  $\mu_t, \Sigma_t$ 

```

Dominio Sottomarino

Oberon, Univ. Sydney

Pencil sonar, oggetti verticali come landmarks



EKF-SLAM per il percorso e le features

Victoria Park Data Set



http://www-personal.acfr.usyd.edu.au/nebot/publications/slam/IJRR_slam.htm

[courtesy by E. Nebot]

Victoria Park Data Set Vehicle

Dati raccolti con una macchina: Laser RF, velocità, GPS



[courtesy by E. Nebot]

Acquisizione Dati

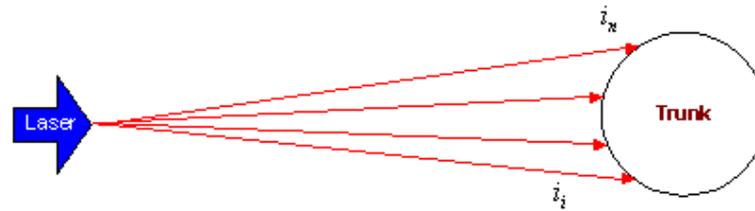
Landmark:
tronchi
alberi



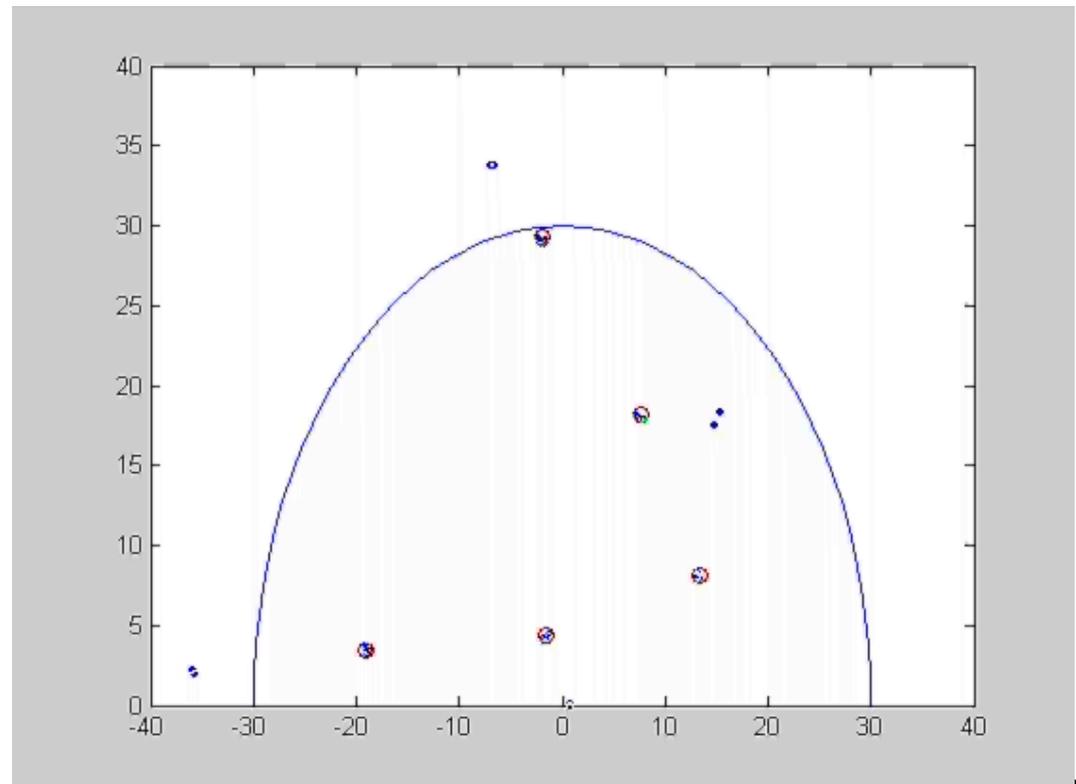
[courtesy by E. Nebot]

Landmark

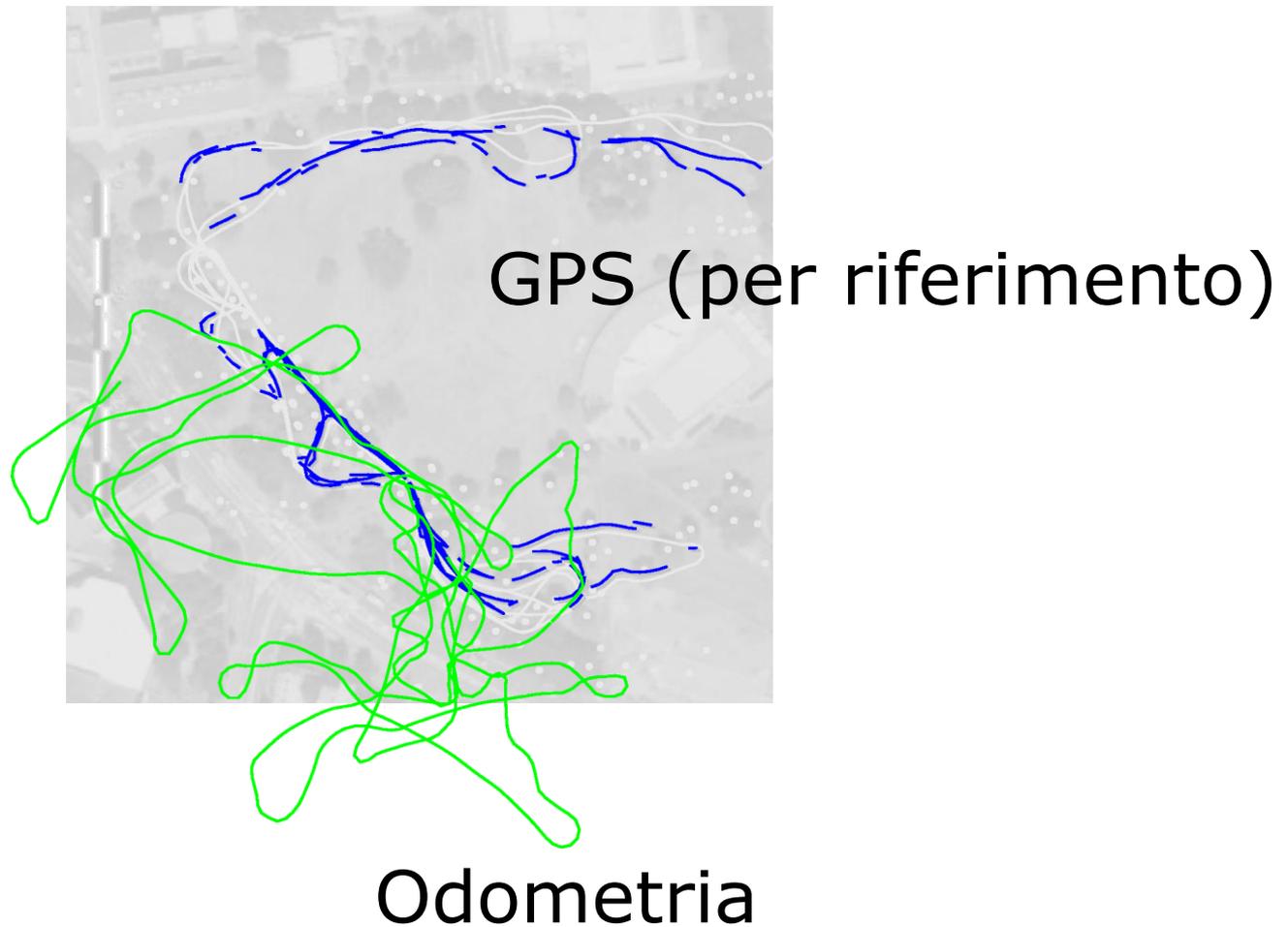
The circumference correspondent to the estimated Diameter and the centre of the object based on processing the Kalman Filter.



The blue dots correspond to the laser returns; the red circle is the position of the centre of the trunk evaluated by the filter

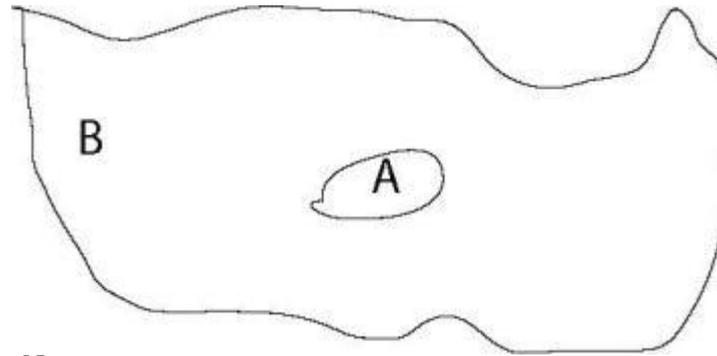


Raw Odometry (no SLAM)



Compressed Filter

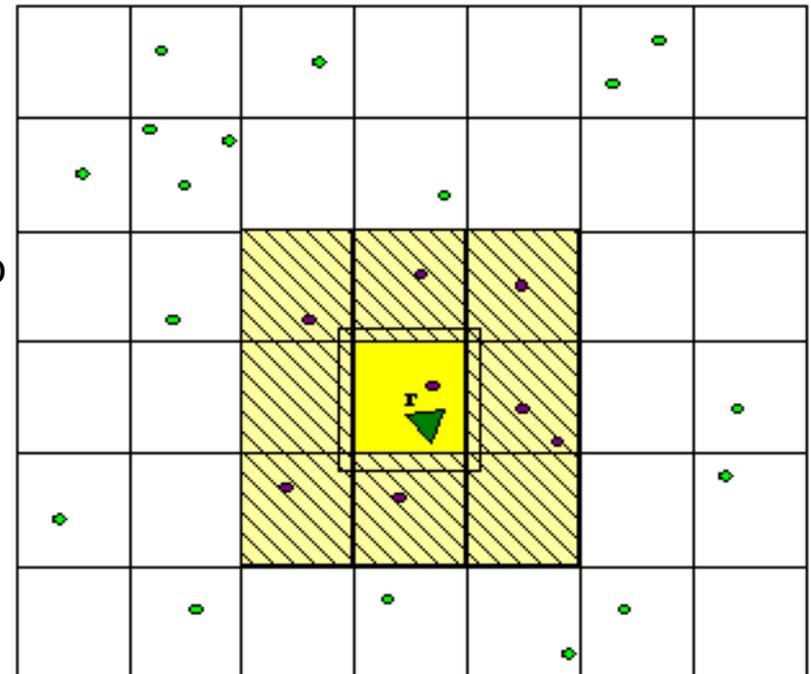
Global and local map



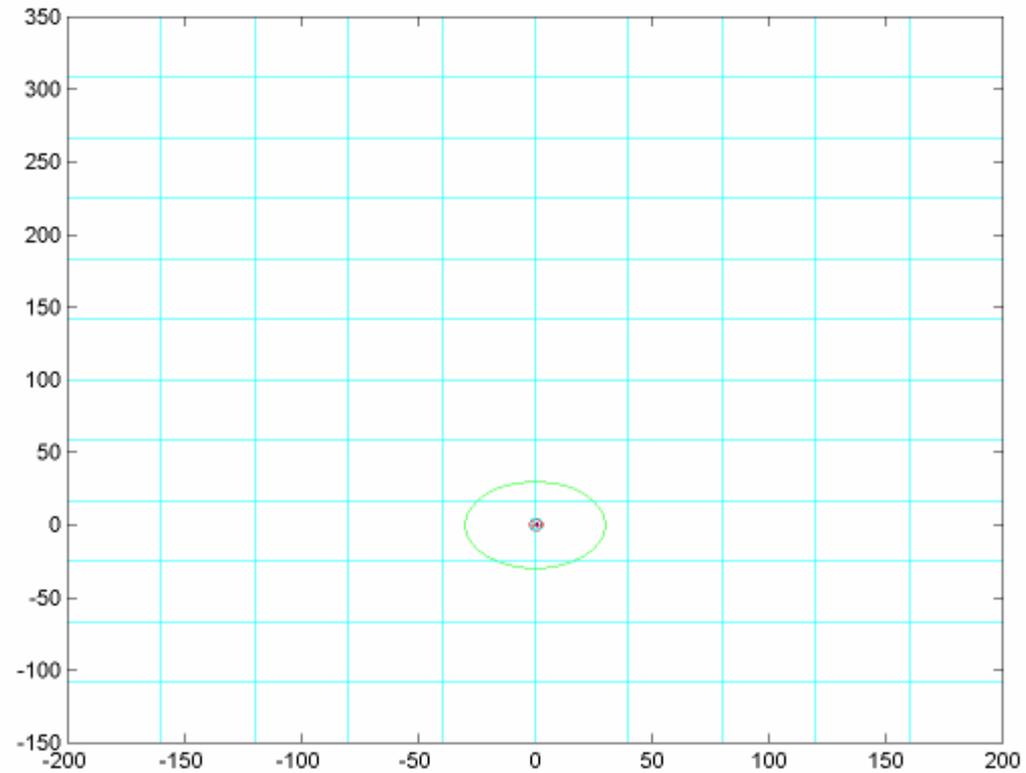
$$X = \begin{bmatrix} X_A \\ X_B \end{bmatrix}, \quad X_A \in \mathbb{R}^{2N_A+3}, \quad X_B \in \mathbb{R}^{2N_B}, \quad X \in \mathbb{R}^{2N+3}, \quad N = N_A + N_B$$

Every time the vehicle moves to a new region, the active state group X_A , changes to those states that belong to the new region r and its adjacent regions.

The active group always includes the vehicle states.

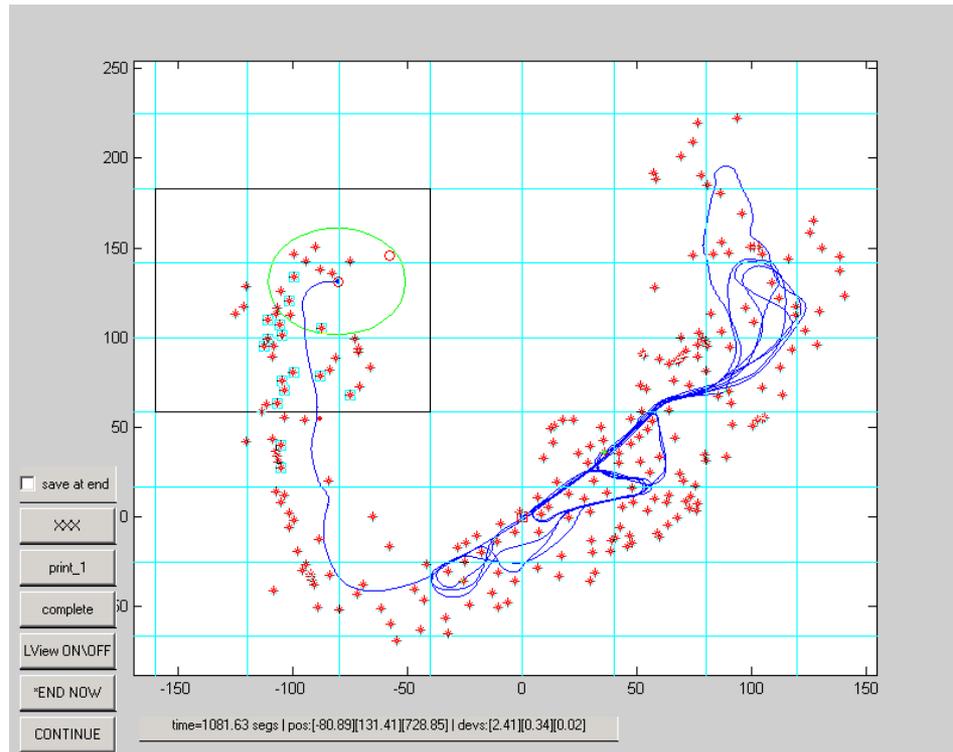


(E)KF-SLAM



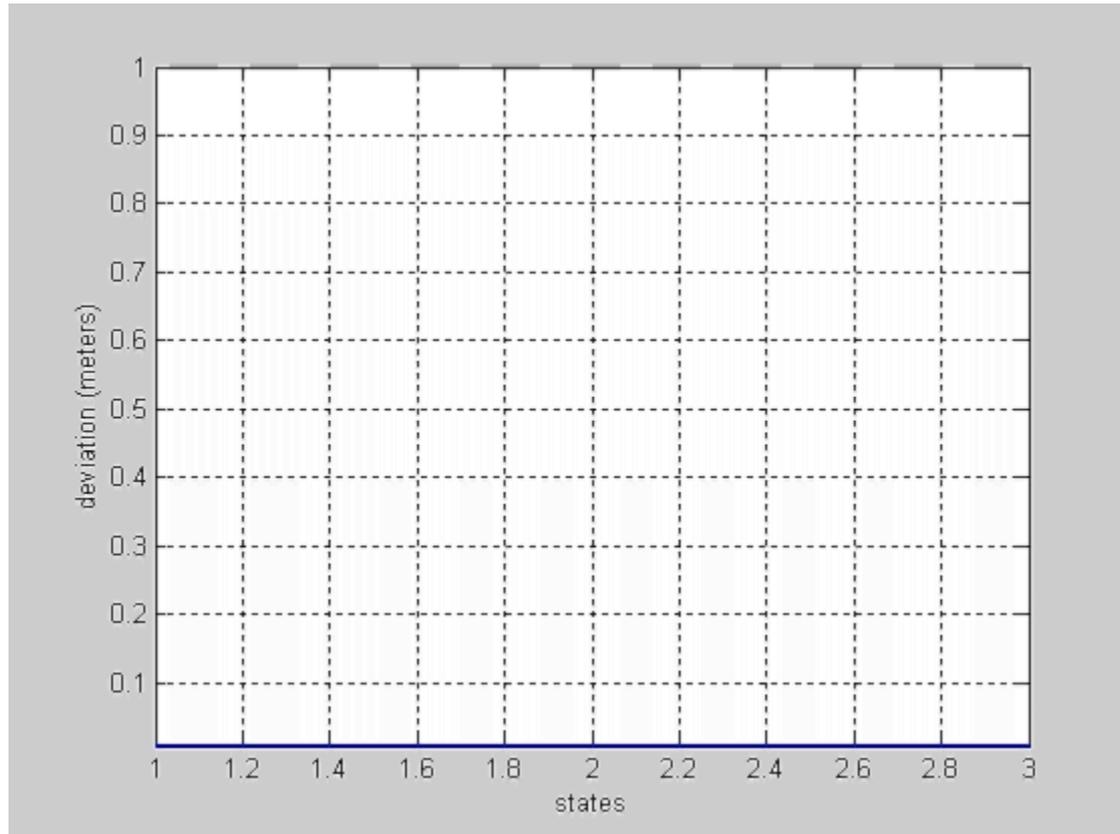
[courtesy by E. Nebot]

Mappa e Traiettoria



[courtesy by E. Nebot]

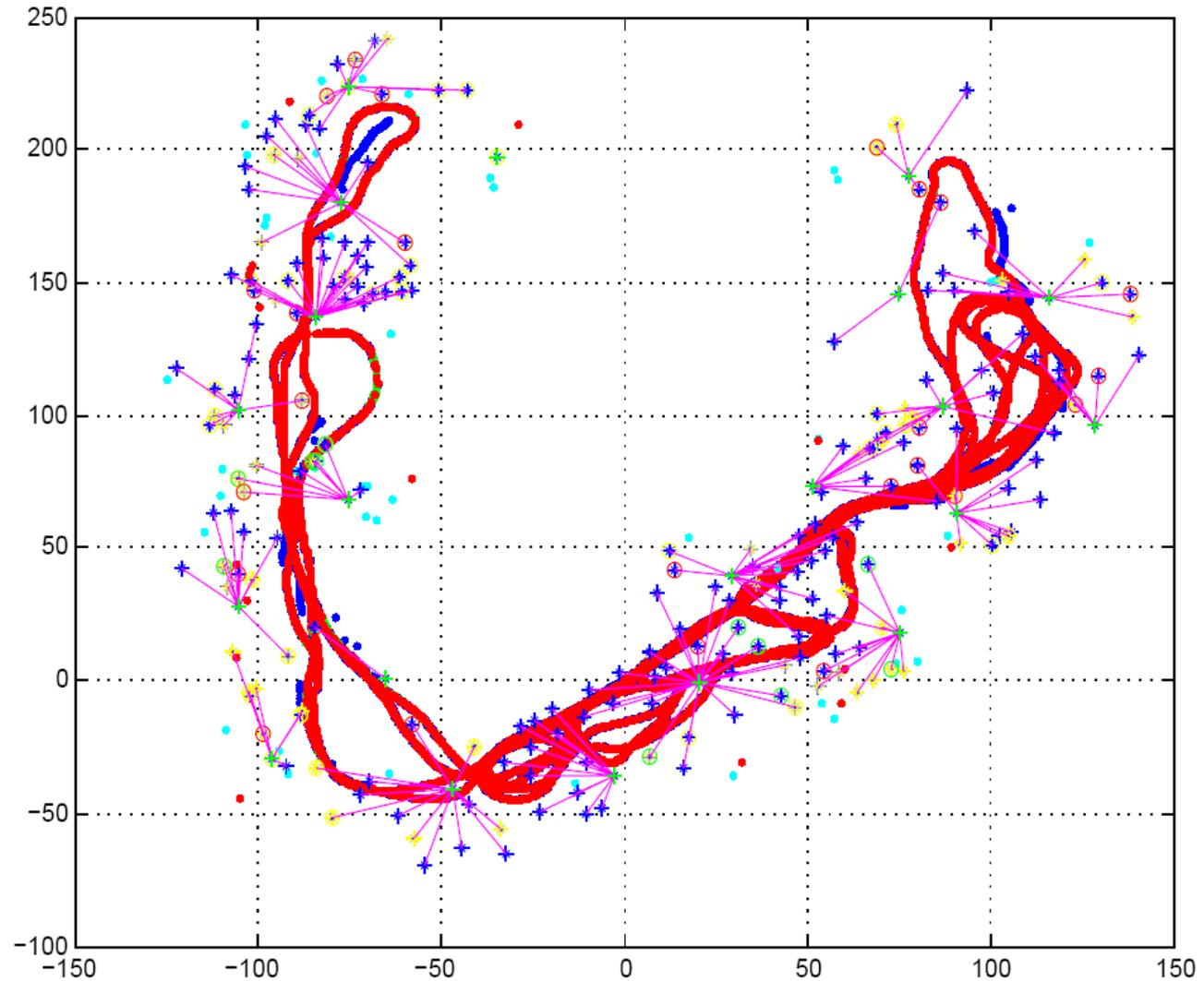
Landmark Covariance



[courtesy by E. Nebot]

Traiettoria Stimata

SLAM algorithm presented uses a compressed filter that reduce the standard SLAM computational requirement making possible the application of this algorithm in environments with more than 10000 landmarks.



[courtesy by E. Nebot]

Large-scale SLAM

- What is the problem?
 - Many applications require persistent, long-term autonomy in large environments
- Why is it difficult?
 - Linear Gaussian case:
 - Computational complexity in maintaining the cross correlations
 - Full covariance SLAM solution runs in $O(n^2)$
 - General non-linear non-Gaussian case:
 - “curse of dimensionality” (Bellman, 1960s)

SLAM and Covariance

Covariance Matrix in the Linear Gaussian SLAM solution

- Pros :
 - Enables us to transfer information from one part of the environment to the other parts
 - Prevents overconfidence and divergence
- Cons :
 - Increases computational complexity



EKF SLAM

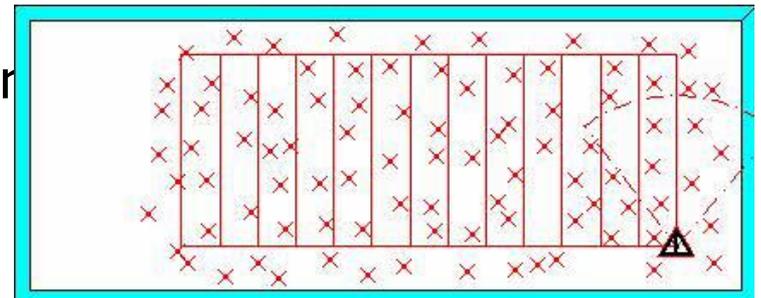
64 hurdles, 4 tennis courts



[courtesy by John Leonard]

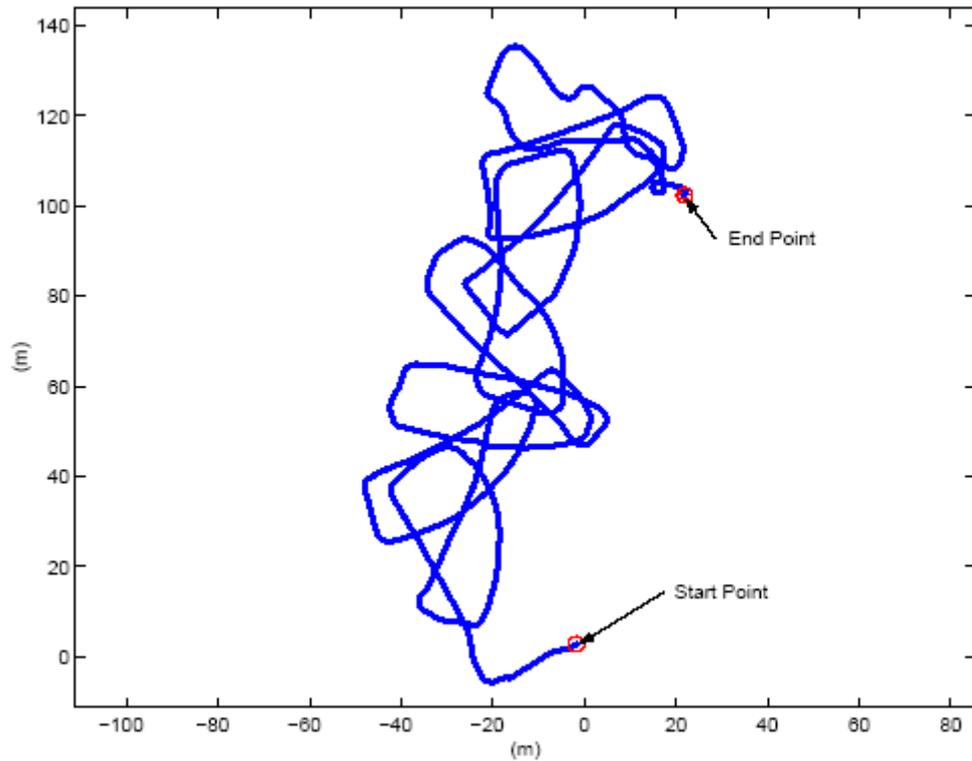
Large-scale SLAM

- Decoupled Stochastic Mapping
 - Globally referenced multiple overlapping submaps
 - Map transition techniques
 - Cross-map relocation (no spatial convergence)
 - Cross-map updating $O(1)$
 - Performance
 - Runs in
 - Needs empirical test to verify
 - Example

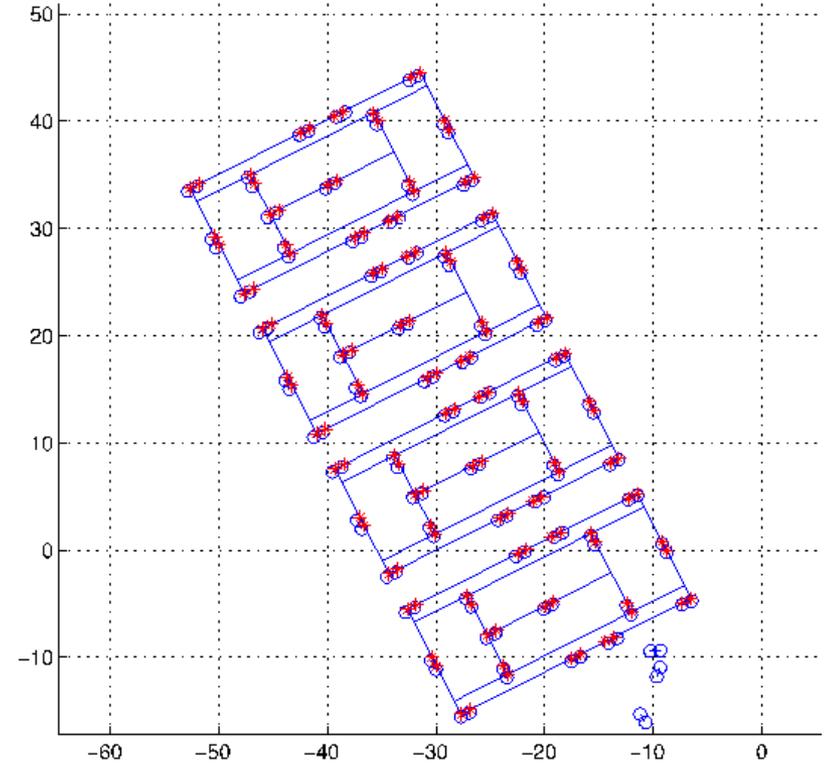


EKF SLAM Application

Odometry Profile of the Robot Locations



odometry



estimated trajectory

Approssimazioni per SLAM

- Local submaps

[Leonard et al.99, Bosse et al. 02, Newman et al. 03]

- Sparse links (correlations)

[Lu & Milios 97, Guivant & Nebot 01]

- Sparse extended information filters

[Frese et al. 01, Thrun et al. 02]

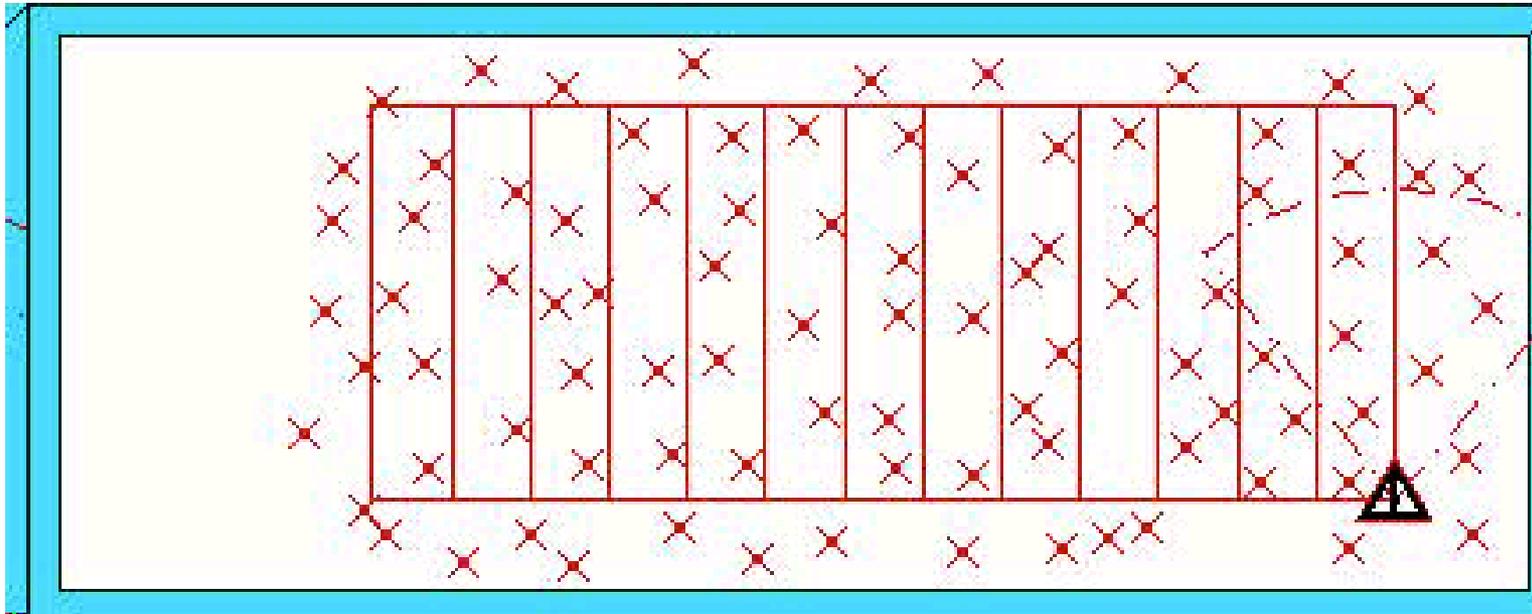
- Thin junction tree filters

[Paskin 03]

- Rao-Blackwellisation (FastSLAM)

[Murphy 99, Montemerlo et al. 02, Eliazar et al. 03, Haehnel et al. 03]

Sub-maps for EKF SLAM



EKF-SLAM: Complessità

- Costo per passo: $O(n^2)$, quadratica nel numero di landmarks: $O(n^2)$
- Costo totale per costruire una mappa con n landmarks: $O(n^3)$
- Memoria: $O(n^2)$

Approcci permettono EKF-SLAM
 $O(n^{1.5})$ / $O(n^{2.5})$ / $O(n)$

EKF-SLAM Sommario

- Quadratica nel numero dei landmarks: $O(n^2)$
- Risultato di convergenza nel caso lineare.
- Può **divergere** per grosse nonlinearità!
- Applicata con successo in ambienti di grandi dimensioni.
- Approssimazione riduce la complessità computazionale.