3. The Data Association Problem

Outline

1. **Introduction:**
   Why data association is important in SLAM, why it's difficult

2. **Data association in continuous SLAM**

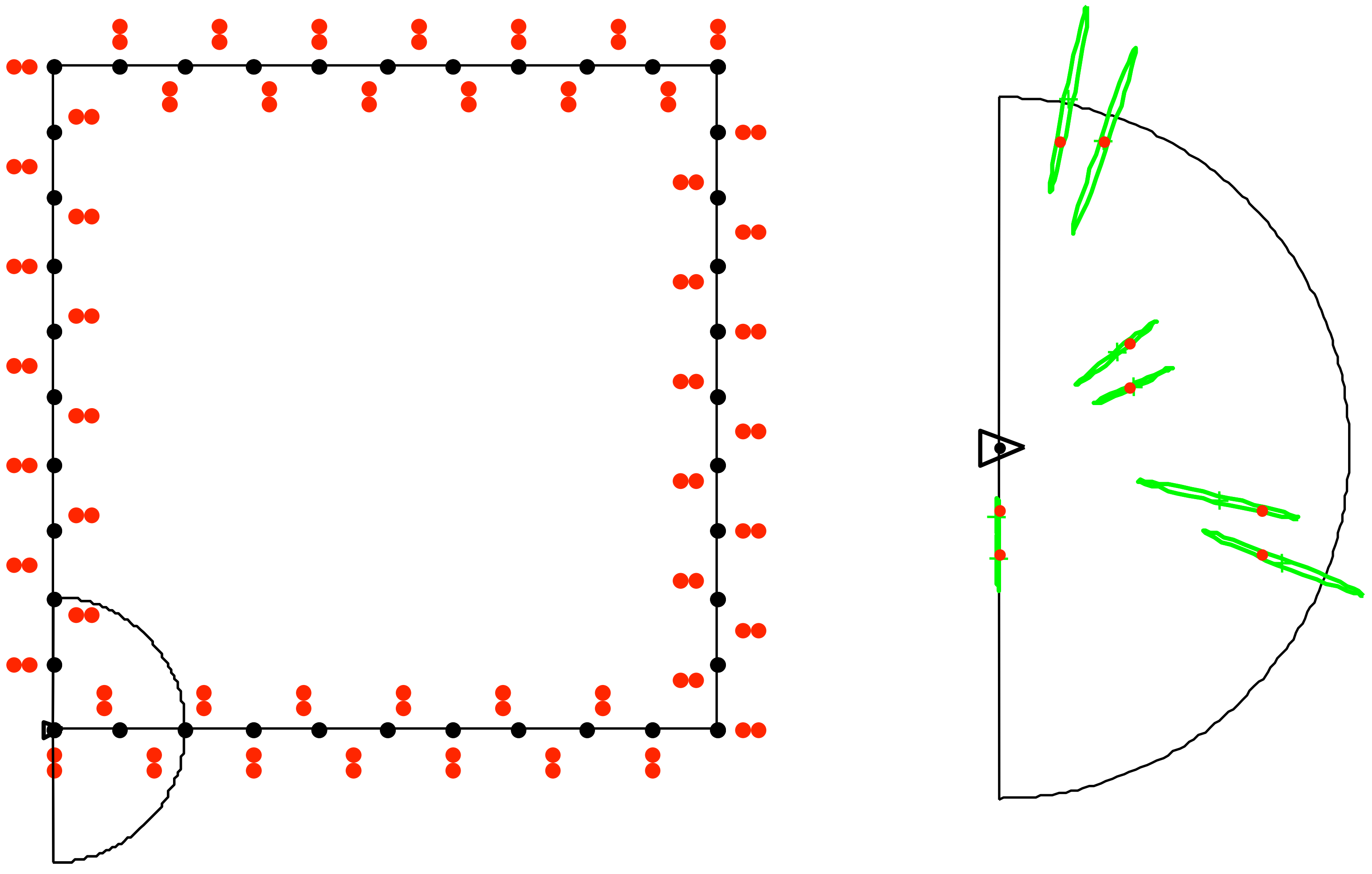
3. **The loop closing problem**

4. **The global localization problem**

5. **Appendix**

1. Introduction

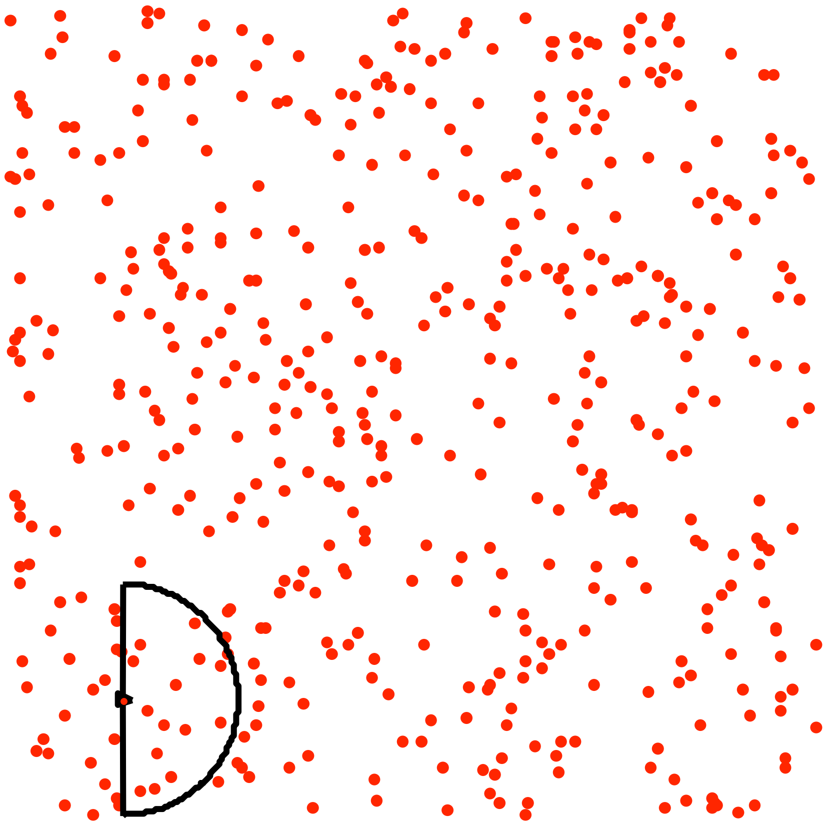
Example: SLAM in a cloister



» Red dots: environment features (columns)
» Black line: robot trajectory
» Black semicircle: sensor range

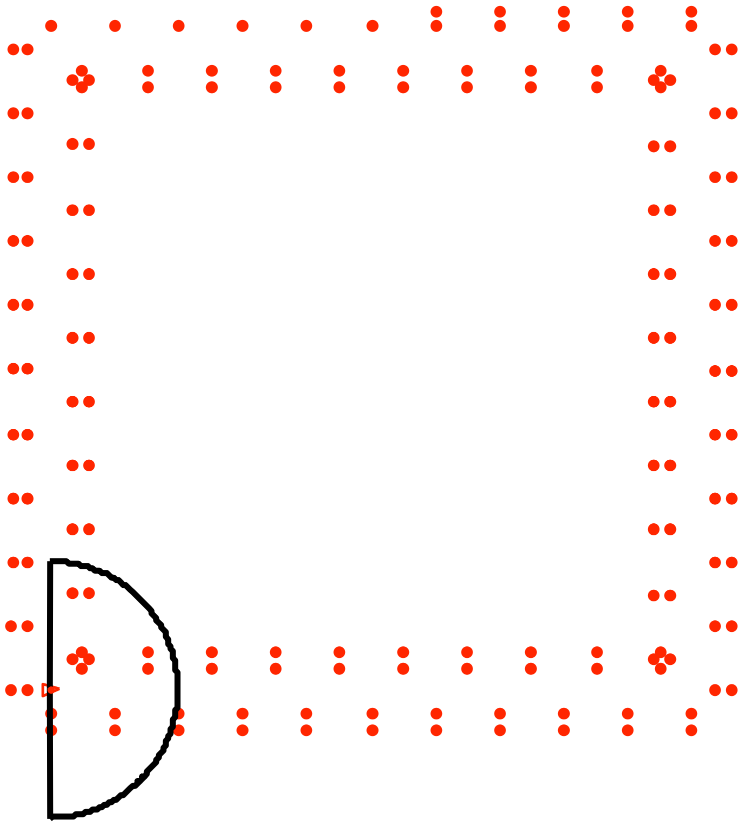
Without loss of generality...

- Enviroment to be mapped has more or less uniform density of features



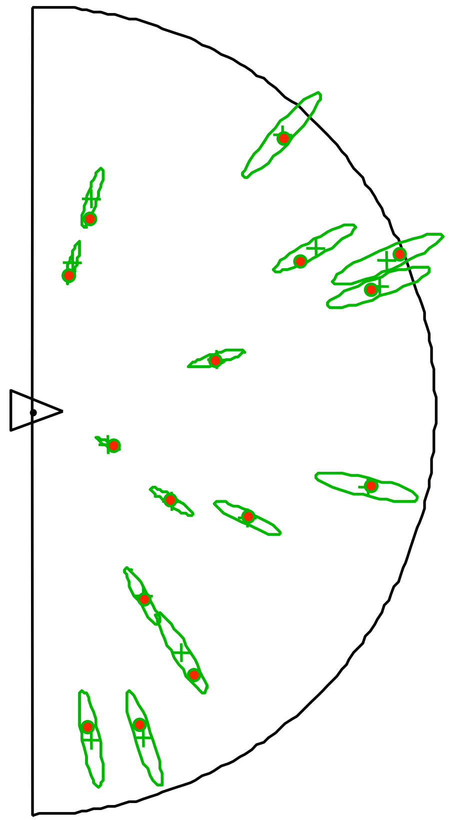
Without loss of generality…

- Enviroment to be mapped has more or less uniform density of features



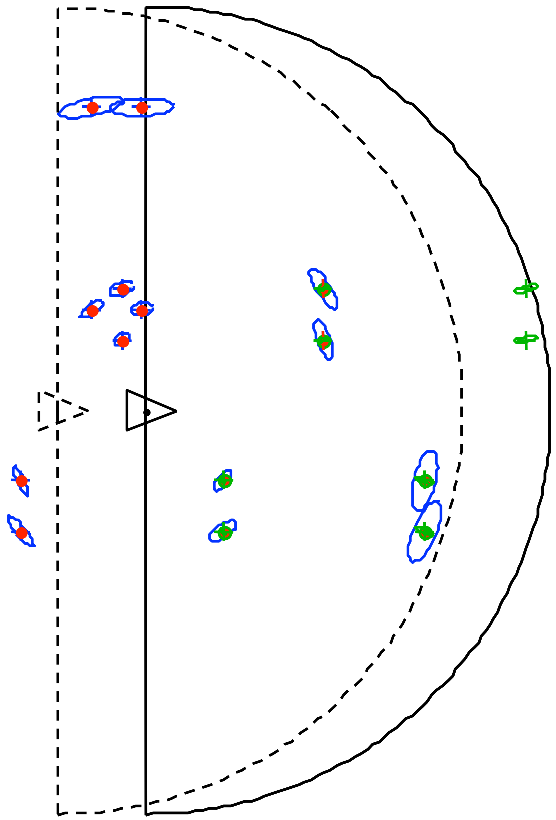
Without loss of generality…

- Onboard range and bearing sensor obtains ***m*** measurements



Without loss of generality…

- Vehicle performs an exploratory trajectory, re-observing ***r*** features, and seeing ***s* = *m* – *r*** new features.



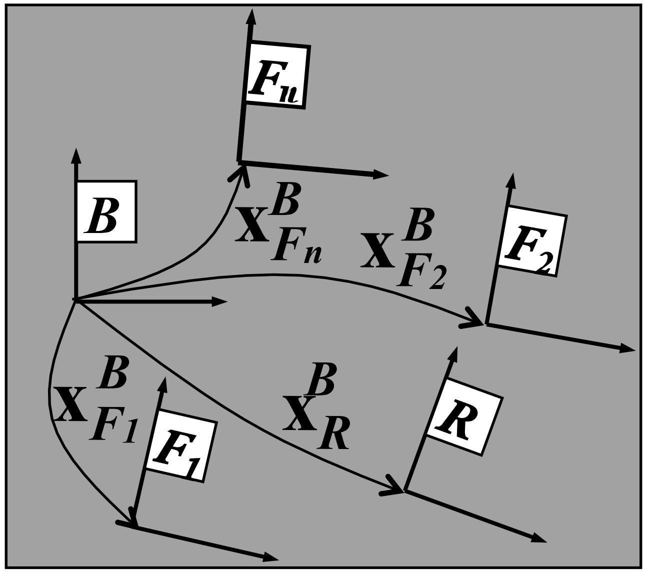
The basic EKF-SLAM algorithm

- Environment information related to a set of elements:

$$\mathcal{F} = \{B,\, R,\, F_1,\, \ldots,\, F_n\}$$

- represented by a **stochastic map**:

$$\mathcal{M}^B = \left(\hat{\mathbf{x}}^B, \mathbf{P}^B\right)$$



$$\hat{\mathbf{x}}^B = \begin{bmatrix} \hat{\mathbf{x}}^B_R \\ \vdots \\ \hat{\mathbf{x}}^B_{F_n} \end{bmatrix}$$

$$\mathbf{P}^B = \begin{bmatrix} \mathbf{P}^B_{RR} & \cdots & \mathbf{P}^B_{RF_n} \\ \vdots & \ddots & \vdots \\ \mathbf{P}^B_{F_nR} & \cdots & \mathbf{P}^B_{F_nF_n} \end{bmatrix}$$

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF-SLAM

Algorithm 1 SLAM:

$\mathbf{x}_0^B = \mathbf{0}$; $\mathbf{P}_0^B = \mathbf{0}$ {*Map initialization*}

$[\mathbf{z}_0, \mathbf{R}_0]$ = get_measurements

$[\mathbf{x}_0^B, \mathbf{P}_0^B]$ = add_new_features($\mathbf{x}_0^B, \mathbf{P}_0^B, \mathbf{z}_0, \mathbf{R}_0$)

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

$[\mathbf{x}_{R_k}^{R_{k-1}}, \mathbf{Q}_k]$ = get_odometry

$[\mathbf{x}_{k|k-1}^B, \mathbf{P}_{k|k-1}^B]$ = compute_motion($\mathbf{x}_{k-1}^B, \mathbf{P}_{k-1}^B, \mathbf{x}_{R_k}^{R_{k-1}}, \mathbf{Q}_k$) {*EKF prediction*}

$[\mathbf{z}_k, \mathbf{R}_k]$ = get_measurements

$\mathcal{H}_k$ = data_association($\mathbf{x}_{k|k-1}^B, \mathbf{P}_{k|k-1}^B, \mathbf{z}_k, \mathbf{R}_k$)

$[\mathbf{x}_k^B, \mathbf{P}_k^B]$ = update_map($\mathbf{x}_{k|k-1}^B, \mathbf{P}_{k|k-1}^B, \mathbf{z}_k, \mathbf{R}_k, \mathcal{H}_k$) {*EKF update*}

$[\mathbf{x}_k^B, \mathbf{P}_k^B]$ = add_new_features($\mathbf{x}_k^B, \mathbf{P}_k^B, \mathbf{z}_k, \mathbf{R}_k, \mathcal{H}_k$)

end for

Map Features in 2D

Points:

$$\mathbf{x}_P^B = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

$$\mathbf{x}_P^A = \mathbf{x}_B^A \oplus \mathbf{x}_P^B = \begin{bmatrix} x_1 + x_2 \cos\phi_1 - y_2 \sin\phi_1 \\ y_1 + x_2 \sin\phi_1 + y_2 \cos\phi_1 \end{bmatrix}$$

$$J_{1\oplus}\{\mathbf{x}_B^A, \mathbf{x}_P^B\} = \begin{bmatrix} 1 & 0 & -x_2 \sin\phi_1 - y_2 \cos\phi_1 \\ 0 & 1 & x_2 \cos\phi_1 - y_2 \sin\phi_1 \end{bmatrix}$$

$$J_{2\oplus}\{\mathbf{x}_B^A, \mathbf{x}_P^B\} = \begin{bmatrix} \cos\phi_1 & -\sin\phi_1 \\ \sin\phi_1 & \cos\phi_1 \end{bmatrix}$$

Lines:

$$\mathbf{x}_L^B = \begin{bmatrix} \rho_2 \\ \theta_2 \end{bmatrix}$$

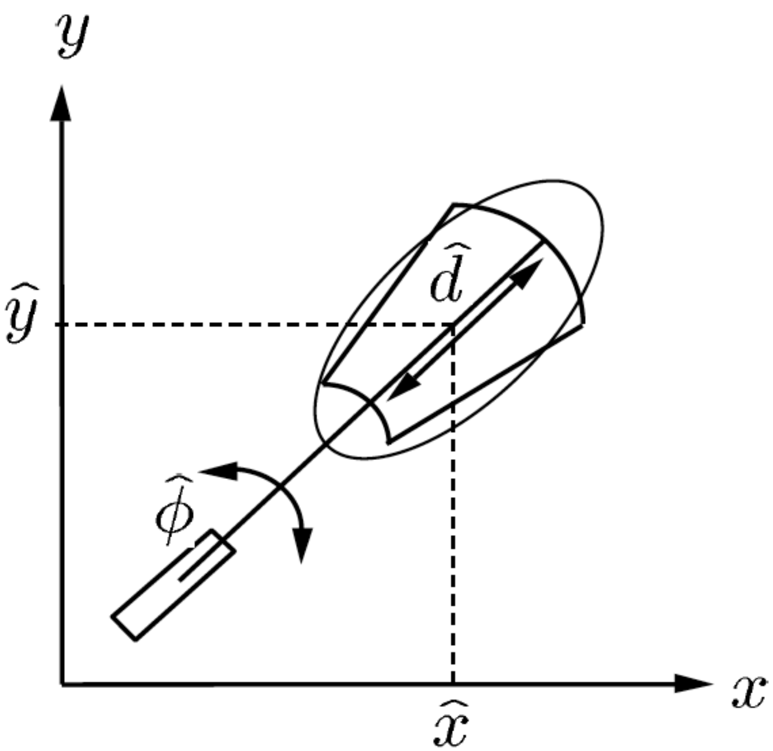
$$\mathbf{x}_L^A = \mathbf{x}_B^A \oplus \mathbf{x}_L^B = \begin{bmatrix} x_1 \cos(\phi_1 + \theta_2) + y_1 \sin(\phi_1 + \theta_2) + \rho_2 \\ \phi_1 + \theta_2 \end{bmatrix}$$

$$J_{1\oplus}\{\mathbf{x}_B^A, \mathbf{x}_L^B\} = \begin{bmatrix} \cos(\phi_1 + \theta_2) & \sin(\phi_1 + \theta_2) & -x_1 \sin(\phi_1 + \theta_2) + y_1 \cos(\phi_1 + \theta_2) \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_{2\oplus}\{\mathbf{x}_B^A, \mathbf{x}_L^B\} = \begin{bmatrix} 1 & -x_1 \sin(\phi_1 + \theta_2) + y_1 \cos(\phi_1 + \theta_2) \\ 0 & 1 \end{bmatrix}$$

Sensor measurements

- In polar coordinates:



$$
\begin{aligned}
\hat{\mathbf{p}} &= (\hat{d},\, \hat{\phi})^T \\
P_{\mathbf{p}} &= \mathtt{diag}(\sigma_d^2,\, \sigma_\phi^2)
\end{aligned}
$$

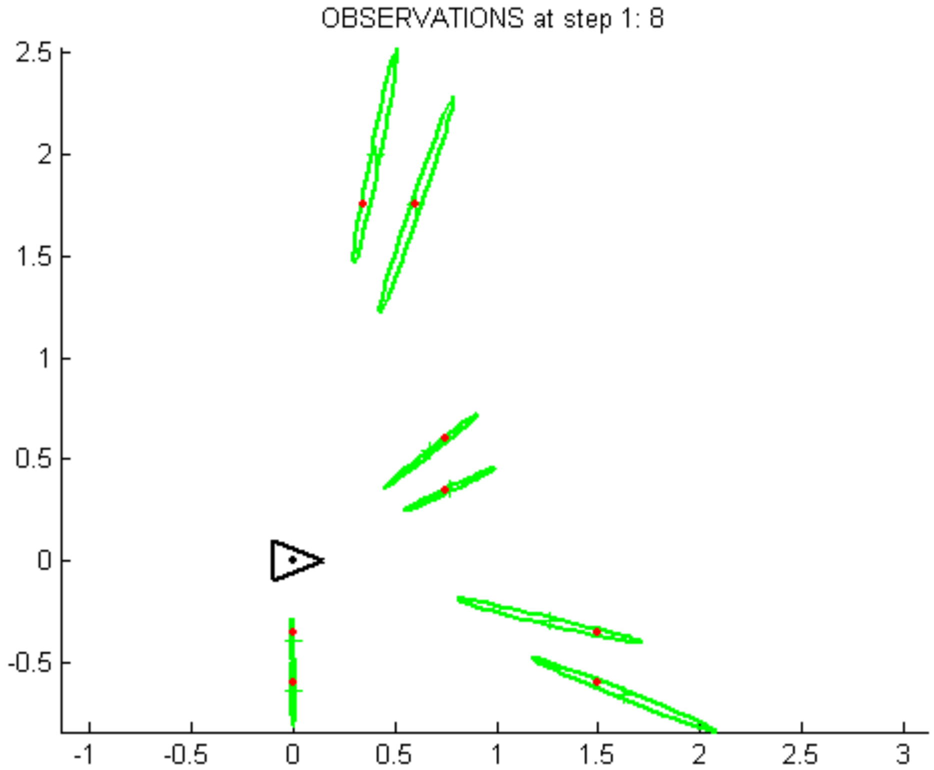
- In cartesian coordinates:

$$
\begin{aligned}
\hat{x} &= \hat{d} \cos \hat{\phi} \\
\hat{y} &= \hat{d} \sin \hat{\phi}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{x} &= f(\mathbf{p}) \\
P_{\mathbf{x}} &\simeq J\, P_{\mathbf{p}}\, J^T \\
J &= \begin{bmatrix} \frac{\partial x}{\partial d} & \frac{\partial x}{\partial \phi} \\ \frac{\partial y}{\partial d} & \frac{\partial y}{\partial \phi} \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\hat{\mathbf{x}} &= (\hat{x},\, \hat{y})^T \\
P_{\mathbf{x}} &= \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}
\end{aligned}
$$

The basic EKF SLAM Algorithm



Sensor measurements

EKF-SLAM: Observations

Observations at instant k:

$$\mathbf{z}_{k,i} \quad \text{with } i = 1 \ldots s$$

Measurement equation:

$$\begin{aligned} \mathbf{z}_k &= \mathbf{h}_k(\mathbf{x}_k^B) + \mathbf{w}_k \\ \mathbf{h}_k &= \begin{bmatrix} \mathbf{h}_{1j_1} \\ \mathbf{h}_{2j_2} \\ \vdots \\ \mathbf{h}_{sj_s} \end{bmatrix} \end{aligned}$$

Sensor model (white noise):

$$\begin{aligned} E[\mathbf{w}_k] &= \mathbf{0} \\ E[\mathbf{w}_k \mathbf{w}_j^T] &= \delta_{kj} \mathbf{R}_k \\ E[\mathbf{w}_k \mathbf{v}_j^T] &= \mathbf{0} \end{aligned}$$

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: add new features

$$\mathbf{x}_k^B = \begin{pmatrix} \mathbf{x}_{R_k}^B \\ \mathbf{x}_{F_{1,k}}^B \\ \vdots \\ \mathbf{x}_{F_{n,k}}^B \end{pmatrix} \Rightarrow \mathbf{x}_{k+}^B = \begin{pmatrix} \mathbf{x}_{R_k}^B \\ \mathbf{x}_{F_{1,k}}^B \\ \vdots \\ \mathbf{x}_{F_{n,k}}^B \\ \mathbf{x}_{F_{n+1,k}}^B \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{R_k}^B \\ \mathbf{x}_{F_{1,k}}^B \\ \vdots \\ \mathbf{x}_{F_{n,k}}^B \\ \mathbf{x}_{R_k}^B \oplus \mathbf{z}_i \end{pmatrix}$$

Linearization:

$$\mathbf{x}_{k+}^B \simeq \hat{\mathbf{x}}_{k+}^B + \mathbf{F}_k(\mathbf{x}_k^B - \hat{\mathbf{x}}_k^B) + \mathbf{G}_k(\mathbf{z}_i - \hat{\mathbf{z}}_i)$$

$$\mathbf{P}_{k+}^B = \mathbf{F}_k \mathbf{P}_k^B \mathbf{F}_k^T + \mathbf{G}_k \mathbf{R}_k \mathbf{G}_k^T$$

Where:

$$\mathbf{F}_k = \frac{\partial \mathbf{x}_{k+}^B}{\partial \mathbf{x}_k^B} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \\ \mathbf{J}_{1\oplus}\left\{\hat{\mathbf{x}}_{R_k}^B, \hat{\mathbf{z}}_i\right\} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} ; \mathbf{G}_k = \frac{\partial \mathbf{x}_{k+}^B}{\partial \mathbf{z}_i} = \begin{pmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{J}_{2\oplus}\left\{\hat{\mathbf{x}}_{R_k}^B, \hat{\mathbf{z}}_i\right\} \end{pmatrix}$$

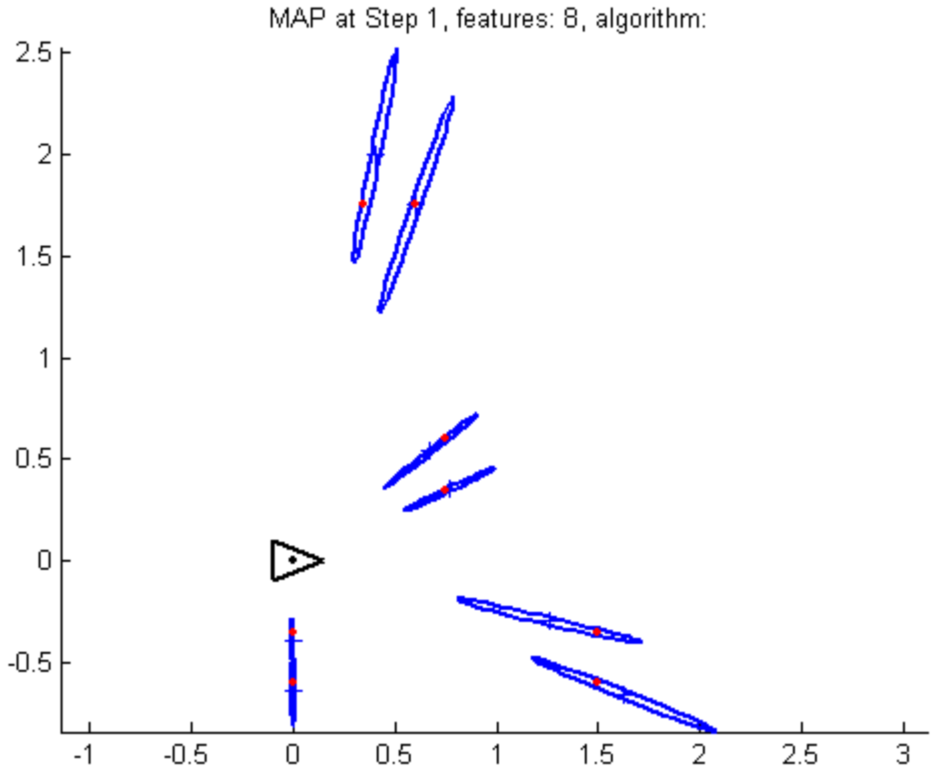
EKF-SLAM: add new features

$$\mathbf{P}_k^B = \begin{pmatrix} \mathbf{P}_R & \mathbf{P}_{RF_1} & \cdots & \mathbf{P}_{RF_n} \\ \mathbf{P}_{RF_1}^T & \mathbf{P}_{F_1} & \cdots & \mathbf{P}_{F_1 F_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{RF_n}^T & \mathbf{P}_{F_1 F_n}^T & \cdots & \mathbf{P}_{F_n} \end{pmatrix}$$

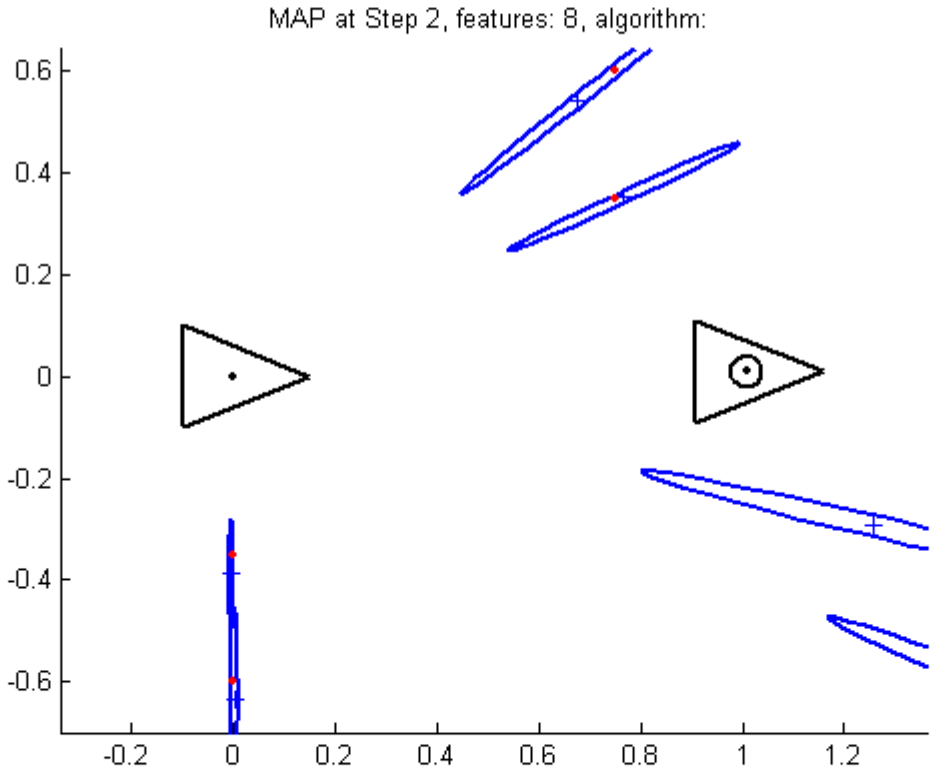
$$\mathbf{P}_{k+}^B = \begin{pmatrix} \mathbf{P}_R & \mathbf{P}_{RF_1} & \cdots & \mathbf{P}_{RF_n} & \mathbf{P}_R \mathbf{J}_{1\oplus}^T \\ \mathbf{P}_{RF_1}^T & \mathbf{P}_{F_1} & \cdots & \mathbf{P}_{F_1 F_n} & \mathbf{P}_{RF_1}^T \mathbf{J}_{1\oplus}^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{P}_{RF_n}^T & \mathbf{P}_{F_1 F_n}^T & \cdots & \mathbf{P}_{F_n} & \mathbf{P}_{RF_n}^T \mathbf{J}_{1\oplus}^T \\ \mathbf{J}_{1\oplus}\mathbf{P}_R & \mathbf{J}_{1\oplus}\mathbf{P}_{RF_1} & \cdots & \mathbf{J}_{1\oplus}\mathbf{P}_{RF_n} & \mathbf{J}_{1\oplus}\mathbf{P}_R \mathbf{J}_{1\oplus}^T + \mathbf{J}_{2\oplus}\mathbf{R}_k \mathbf{J}_{2\oplus}^T \end{pmatrix}$$

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: add new features

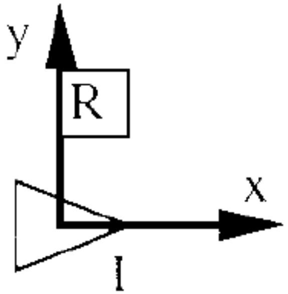


EKF-SLAM: compute robot motion



Vehicle motion in 2D

$$\mathbf{x}_B^A = \begin{bmatrix} x_1 \\ y_1 \\ \phi_1 \end{bmatrix} \qquad \mathbf{x}_C^B = \begin{bmatrix} x_2 \\ y_2 \\ \phi_2 \end{bmatrix}$$



Composition:

$$\mathbf{x}_C^A = \mathbf{x}_B^A \oplus \mathbf{x}_C^B = \begin{bmatrix} x_1 + x_2 \cos\phi_1 - y_2 \sin\phi_1 \\ y_1 + x_2 \sin\phi_1 + y_2 \cos\phi_1 \\ \phi_1 + \phi_2 \end{bmatrix}$$

$$\mathbf{x}_{R_k}^B = \mathbf{x}_{R_{k-1}}^B \oplus \mathbf{x}_{R_k}^{R_{k-1}}$$

Inversion:

$$\mathbf{x}_A^B = \ominus \mathbf{x}_B^A = \begin{bmatrix} -x_1 \cos\phi_1 - y_1 \sin\phi_1 \\ x_1 \sin\phi_1 - y_1 \cos\phi_1 \\ -\phi_1 \end{bmatrix}$$

Odometry in 2D

Jacobians:

$$J_{1\oplus}\{\mathbf{x}_B^A, \mathbf{x}_C^B\} = \left.\frac{\partial\left(\mathbf{x}_B^A \oplus \mathbf{x}_C^B\right)}{\partial \mathbf{x}_B^A}\right|_{(\hat{\mathbf{x}}_B^A, \hat{\mathbf{x}}_C^B)} = \begin{bmatrix} 1 & 0 & -x_2 \sin\phi_1 - y_2 \cos\phi_1 \\ 0 & 1 & x_2 \cos\phi_1 - y_2 \sin\phi_1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_{2\oplus}\{\mathbf{x}_B^A, \mathbf{x}_C^B\} = \left.\frac{\partial\left(\mathbf{x}_B^A \oplus \mathbf{x}_C^B\right)}{\partial \mathbf{x}_C^B}\right|_{(\hat{\mathbf{x}}_B^A, \hat{\mathbf{x}}_C^B)} = \begin{bmatrix} \cos\phi_1 & -\sin\phi_1 & 0 \\ \sin\phi_1 & \cos\phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_{\ominus}\{\mathbf{x}_B^A\} = \left.\frac{\partial\left(\ominus\mathbf{x}_B^A\right)}{\partial \mathbf{x}_B^A}\right|_{(\hat{\mathbf{x}}_B^A)} = \begin{bmatrix} -\cos\phi_1 & -\sin\phi_1 & -x_1 \sin\phi_1 - y_1 \cos\phi_1 \\ \sin\phi_1 & -\cos\phi_1 & x_1 \cos\phi_1 + y_1 \sin\phi_1 \\ 0 & 0 & -1 \end{bmatrix}$$

EKF-SLAM: compute robot motion

$$\mathbf{x}^B_{R_k} = \mathbf{x}^B_{R_{k-1}} \oplus \mathbf{x}^{R_{k-1}}_{R_k}$$

Odometry model (white noise):

$$\begin{aligned} \mathbf{x}^{R_{k-1}}_{R_k} &= \hat{\mathbf{x}}^{R_{k-1}}_{R_k} + \mathbf{v}_k \\ E[\mathbf{v}_k] &= \mathbf{0} \\ E[\mathbf{v}_k \mathbf{v}_j^T] &= \delta_{kj} \mathbf{Q}_k \end{aligned}$$

EKF prediction:

$$\hat{\mathbf{x}}^B_{k|k-1} = \begin{bmatrix} \hat{\mathbf{x}}^B_{R_{k-1}} \oplus \hat{\mathbf{x}}^{R_{k-1}}_{R_k} \\ \hat{\mathbf{x}}^B_{F_{1,k-1}} \\ \vdots \\ \hat{\mathbf{x}}^B_{F_{m,k-1}} \end{bmatrix}$$

$$\mathbf{P}^B_{k|k-1} = \mathbf{F}_k \mathbf{P}^B_{k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T$$

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{J}_{1\oplus}\left\{\hat{\mathbf{x}}^B_{R_{k-1}}, \hat{\mathbf{x}}^{R_{k-1}}_{R_k}\right\} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & & \vdots \\ \vdots & & \ddots & \\ \mathbf{0} & \cdots & & \mathbf{I} \end{bmatrix}$$

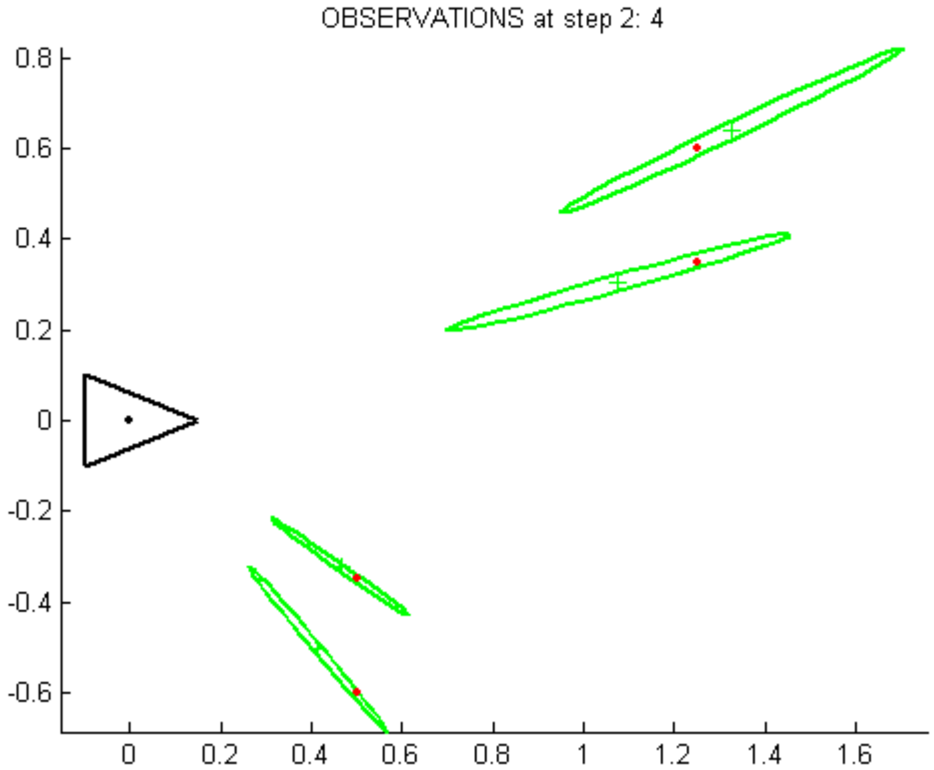
$$\mathbf{G}_k = \begin{bmatrix} \mathbf{J}_{2\oplus}\left\{\hat{\mathbf{x}}^B_{R_{k-1}}, \hat{\mathbf{x}}^{R_{k-1}}_{R_k}\right\} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

EKF-SLAM: compute robot motion

$$\mathbf{P}^B_{k-1|k-1} = \begin{pmatrix} \mathbf{P}_R & \mathbf{P}_{RF_1} & \cdots & \mathbf{P}_{RF_n} \\ \mathbf{P}^T_{RF_1} & \mathbf{P}_{F_1} & \cdots & \mathbf{P}_{F_1F_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^T_{RF_n} & \mathbf{P}^T_{F_1F_n} & \cdots & \mathbf{P}_{F_n} \end{pmatrix}$$

$$\mathbf{P}^B_{k|k-1} = \begin{pmatrix} \mathbf{J}_{1\oplus}\mathbf{P}_R\mathbf{J}^T_{1\oplus} + \mathbf{J}_{2\oplus}\mathbf{Q}_k\mathbf{J}^T_{2\oplus} & \mathbf{J}_{1\oplus}\mathbf{P}_{RF_1} & \cdots & \mathbf{J}_{1\oplus}\mathbf{P}_{RF_n} \\ \mathbf{J}^T_{1\oplus}\mathbf{P}^T_{RF_1} & \mathbf{P}_{F_1} & \cdots & \mathbf{P}_{F_1F_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{J}^T_{1\oplus}\mathbf{P}^T_{RF_n} & \mathbf{P}^T_{F_1F_n} & \cdots & \mathbf{P}_{F_n} \end{pmatrix}$$

EKF-SLAM: Observations



EKF-SLAM: Observations

Observations at instant k:

$$\mathbf{z}_{k,i} \quad \text{with } i = 1 \ldots s$$

Association Hypothesis (obs. i with map feature j_i) :

$$\mathcal{H}_k = [j_1, j_2, \cdots, j_s]$$

Measurement equation:

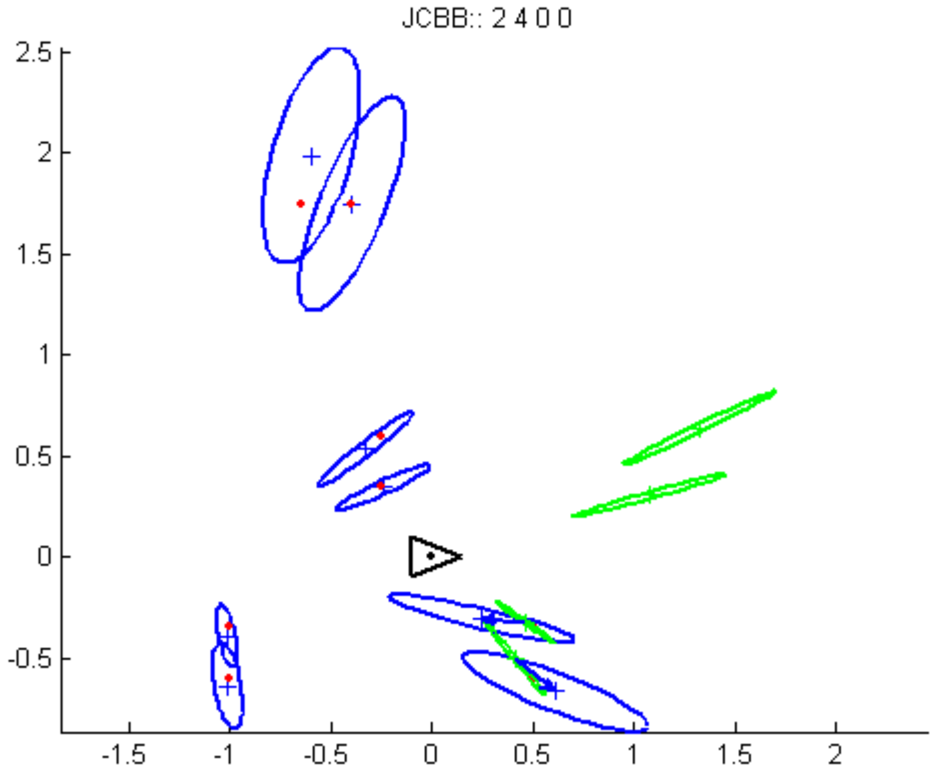
$$\begin{aligned} \mathbf{z}_k &= \mathbf{h}_k(\mathbf{x}_k^B) + \mathbf{w}_k \\ \mathbf{h}_k &= \begin{bmatrix} \mathbf{h}_{1j_1} \\ \mathbf{h}_{2j_2} \\ \vdots \\ \mathbf{h}_{sj_s} \end{bmatrix} \end{aligned}$$

Sensor model (white noise):

$$\begin{aligned} E[\mathbf{w}_k] &= \mathbf{0} \\ E[\mathbf{w}_k \mathbf{w}_j^T] &= \delta_{kj} \mathbf{R}_k \\ E[\mathbf{w}_k \mathbf{v}_j^T] &= \mathbf{0} \end{aligned}$$

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: Data association



EKF-SLAM: Observations

Linearization:

$$
\begin{aligned}
\mathbf{z}_k &\simeq \mathbf{h}_k(\hat{\mathbf{x}}^B_{k|k-1}) + \mathbf{H}_k(\mathbf{x}^B_k - \hat{\mathbf{x}}^B_{k|k-1}) \\
\mathbf{H}_k &= \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}^B_k}\right|_{(\hat{\mathbf{x}}^B_{k|k-1})} = \begin{pmatrix} \mathbf{H}_R & \mathbf{0} & \cdots & \mathbf{H}_F \cdots & \mathbf{0} \end{pmatrix} \\
\mathbf{H}_R &= \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}^B_{R_k}}\right|_{(\hat{\mathbf{x}}^B_{k|k-1})} \; ; \; \mathbf{H}_F = \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}^B_{F_k}}\right|_{(\hat{\mathbf{x}}^B_{k|k-1})}
\end{aligned}
$$

Data association

Innovation:

$$\begin{aligned} \nu_k &= \mathbf{z}_k - \widehat{\mathbf{z}}_k \\ \mathbf{Cov}(\nu_k) &= \mathbf{H}_k \mathbf{P}_k^B \mathbf{H}_k^T + \mathbf{R}_k^T \end{aligned}$$

Mahalanobis distance:

$$D^2 = \nu_k^T \mathbf{Cov}(\nu_k)^{-1} \nu_k \sim \chi_r^2$$

where r =dim(ν_k)

Hypothesis test:

$D^2 \leq \chi^2_{r,\alpha} \;\Rightarrow\; \mathbf{z}_k$ **compatible with** $\widehat{\mathbf{z}}_k$

where $\alpha = 0.05$ (common)

ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: map update

State update:

$$\hat{\mathbf{x}}_k^B = \hat{\mathbf{x}}_{k|k-1}^B + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1}^B))$$

Covariance update:

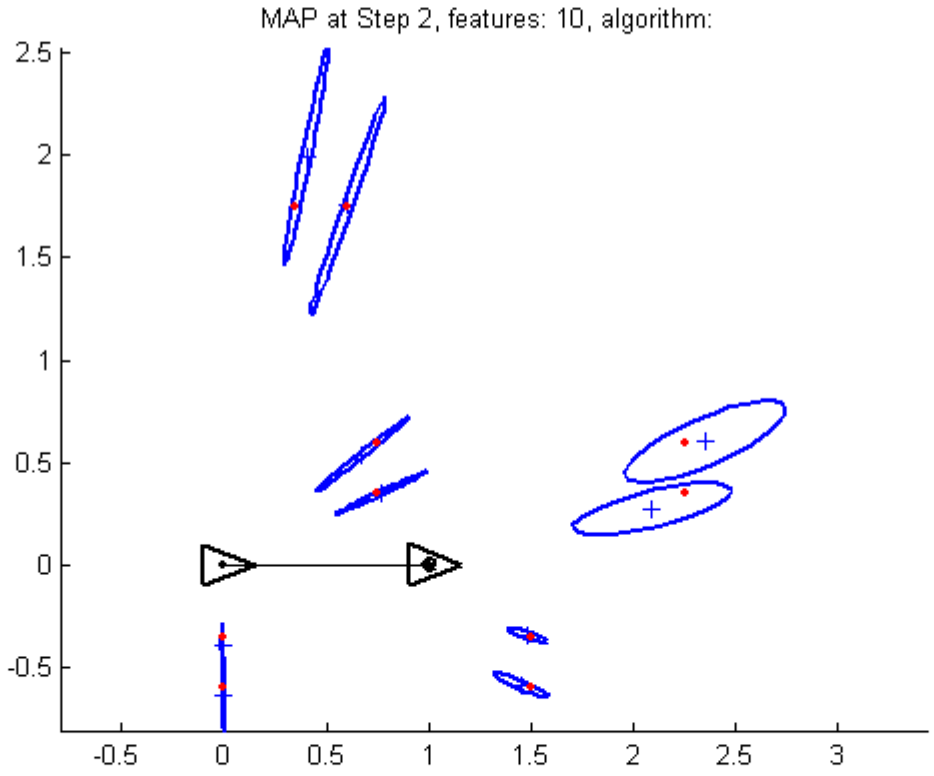
$$\mathbf{P}_k^B = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}^B$$

Filter gain:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}^B\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|k-1}^B\mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: map update



The Data Association Problem

- ***n*** map features: $\mathcal{F} = \{F_1 \ldots F_n\}$

- ***m*** sensor measurements: $\mathcal{E} = \{E_1 \ldots E_m\}$

- Data association should return a hypothesis that associates each observation E_i with a feature F_{ji}

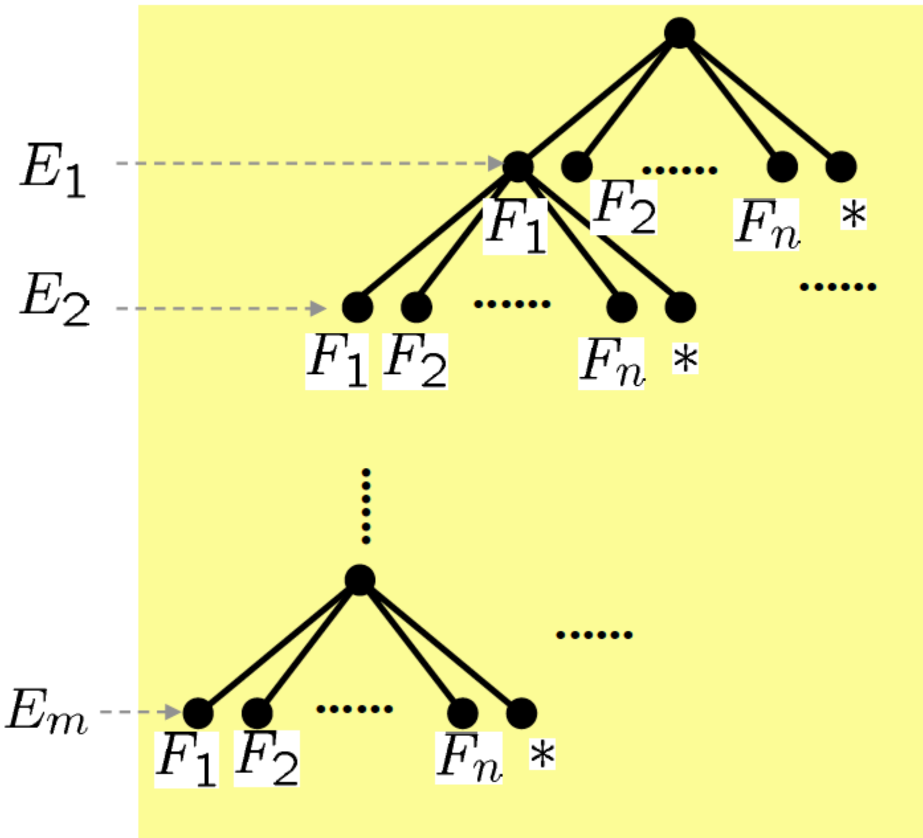
$$\mathcal{H}_m = [j_1 \ldots j_i \ldots j_m]$$

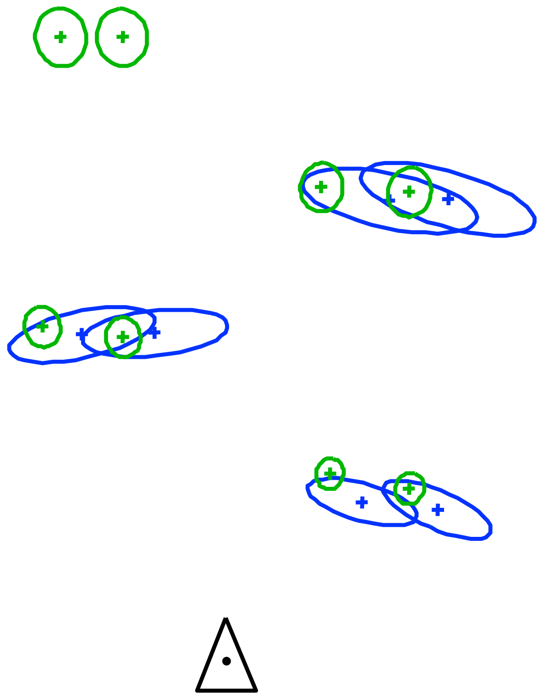
$$E_i \leftarrow j_i \rightarrow F_{j_i}$$

- Non matched observations: $j_i = 0$

The Correspondence Space

Interpretation tree
(Grimson et al. 87):



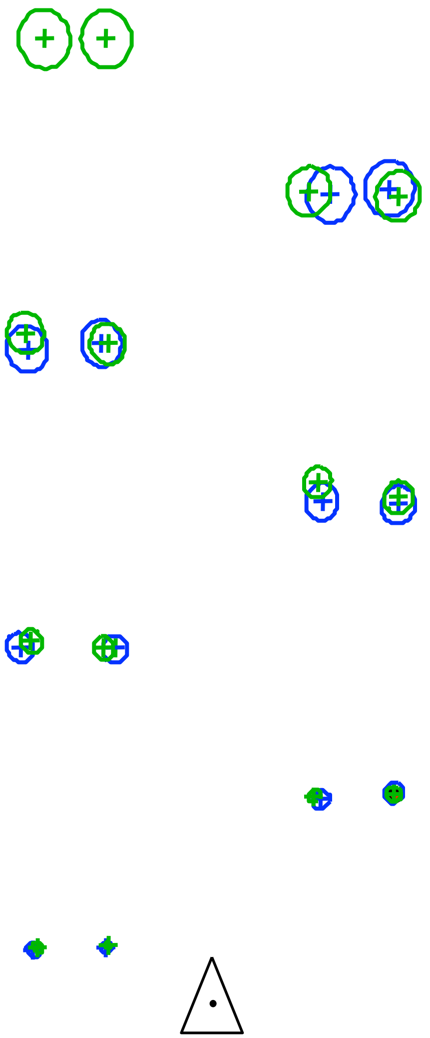


Green points: measurements
Blue Points: predicted features

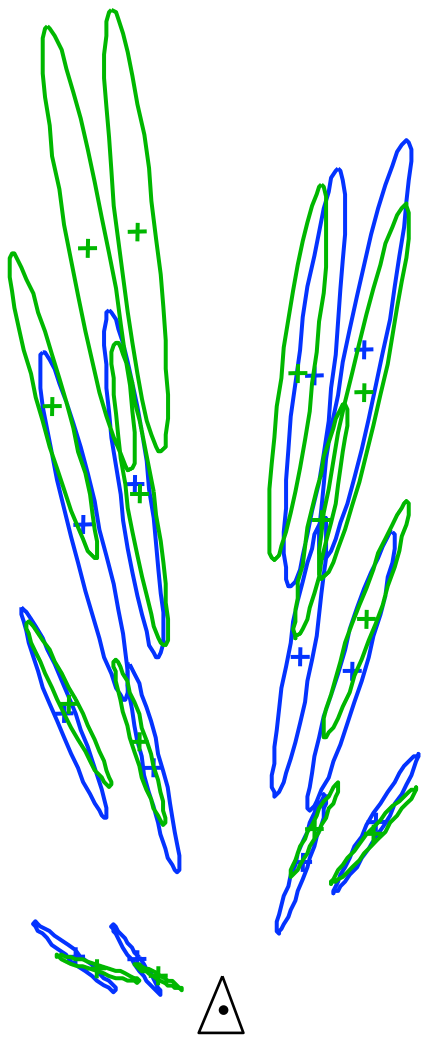
$(n+1)^m$ possible hypotheses

Why data association is difficult

- Low sensor error

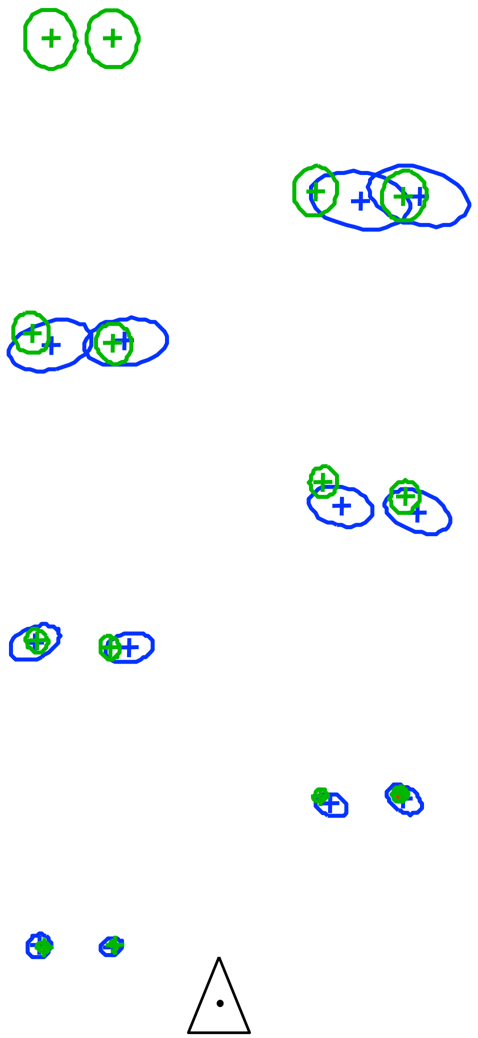


- High sensor error

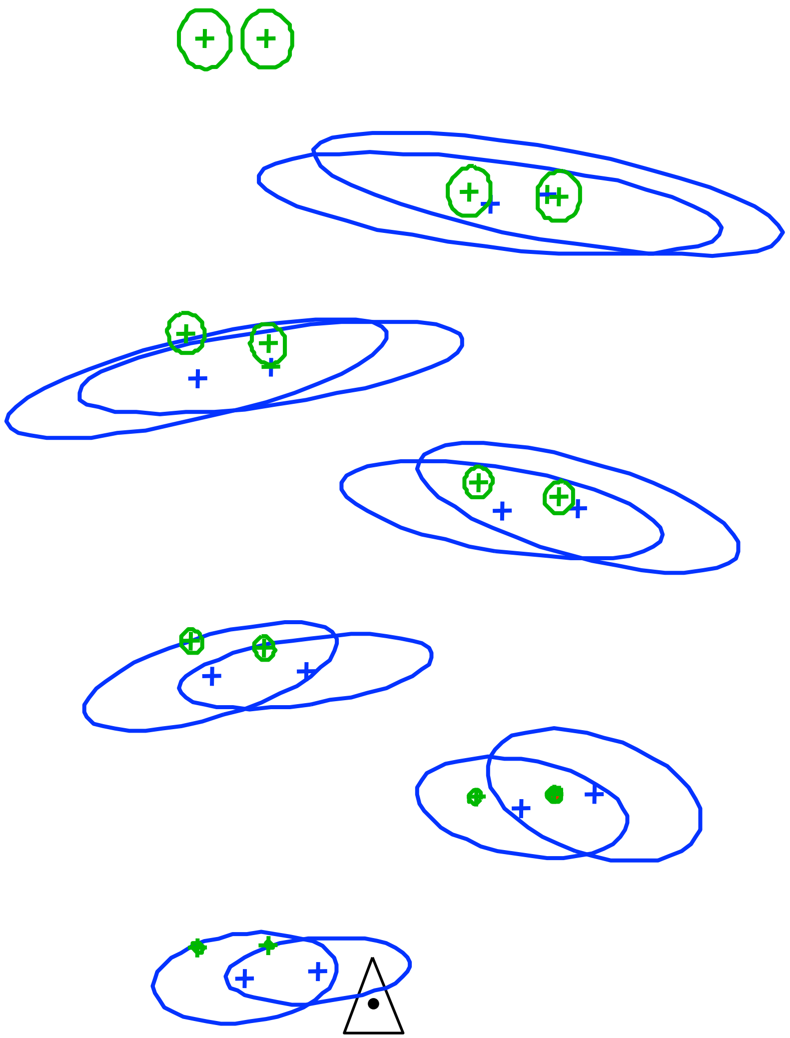


Why data association is difficult

- Low odometry error

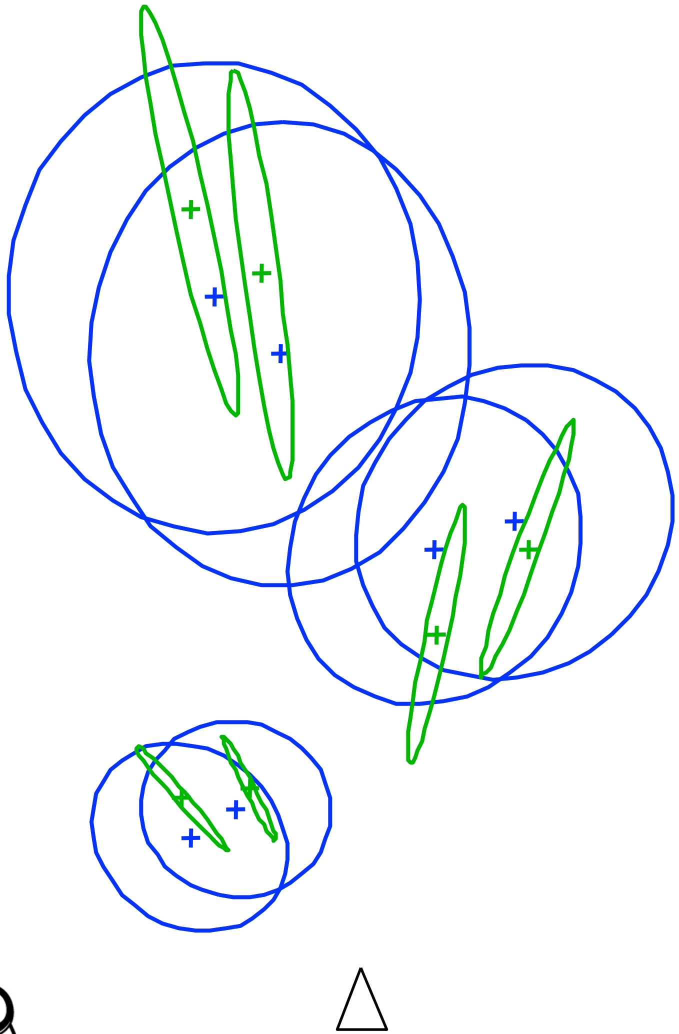


- High odometry error

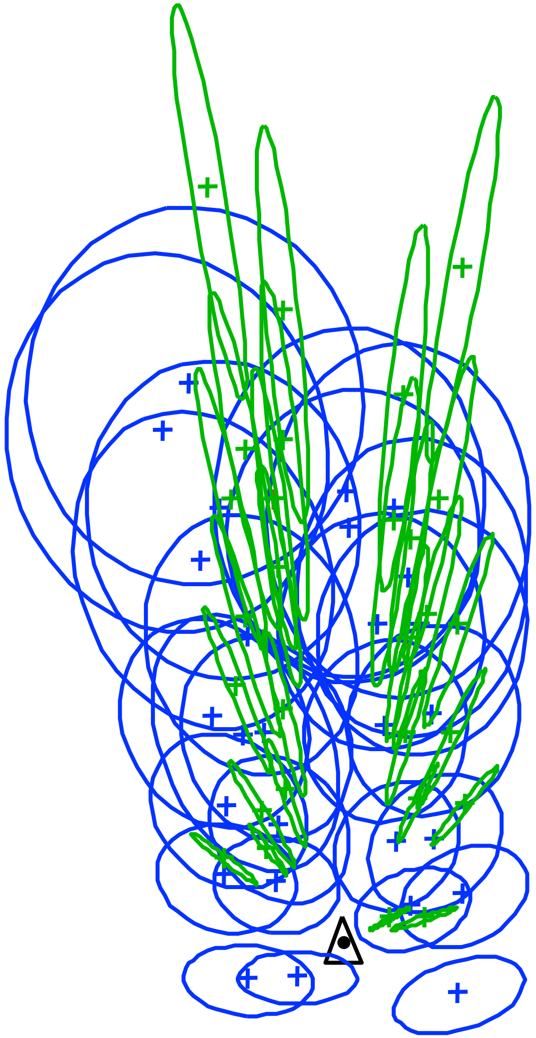


Why data association is difficult

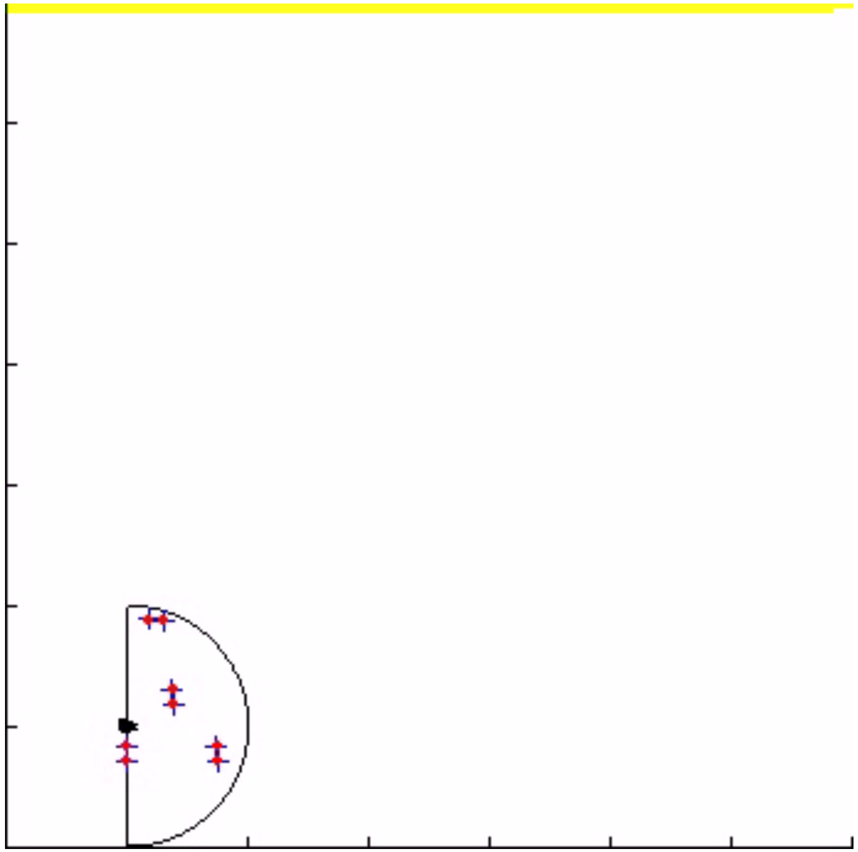
- Low feature density



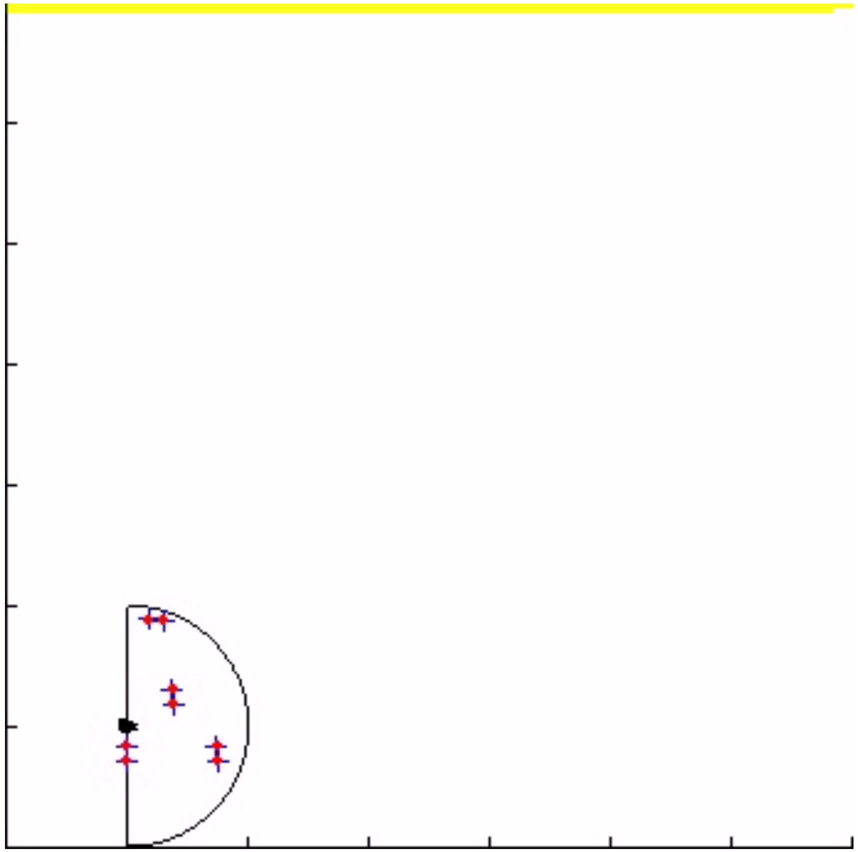
- High feature density



How important is data association?

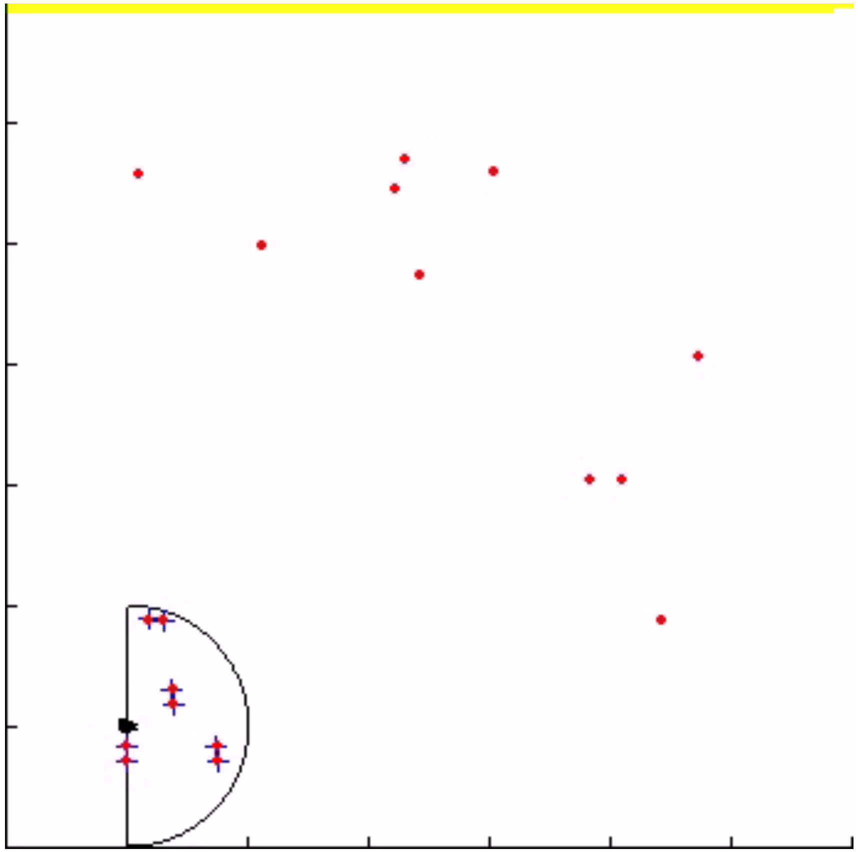


A good algorithm

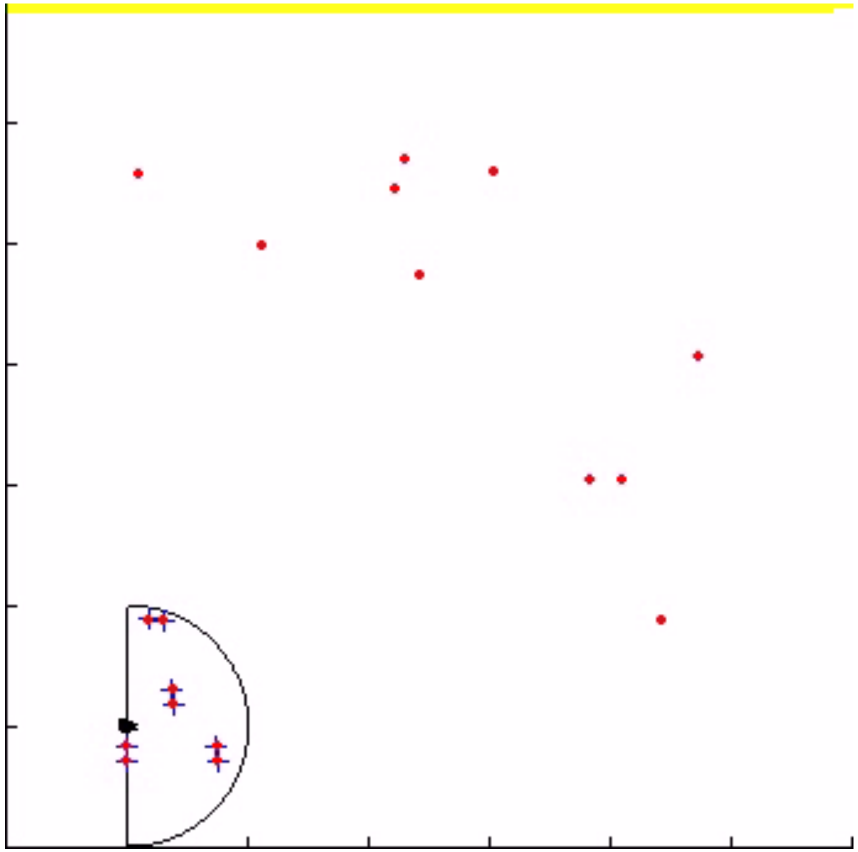


A **bad** algorithm

Why it's difficult?



A good algorithm



A **bad** algorithm

Importance of Data Association

- EKF update:

Values that depend on $\mathcal{H}_m$

$$\hat{\mathbf{x}}_k^B = \hat{\mathbf{x}}_{k|k-1}^B + \mathbf{K}_k \nu_k$$

$$\mathbf{P}_k^B = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_{k|k-1}^B$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}^B \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1}^B \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

- If the association of E_i with feature F_j is.....

correct:

spurious:

error: $\mathbf{x} - \hat{\mathbf{x}}$

covariance: P

Consistency

Divergence!

2. Data association in continuous SLAM

Individual Compatibility

- Measurement equation for observation E_i and feature F_j

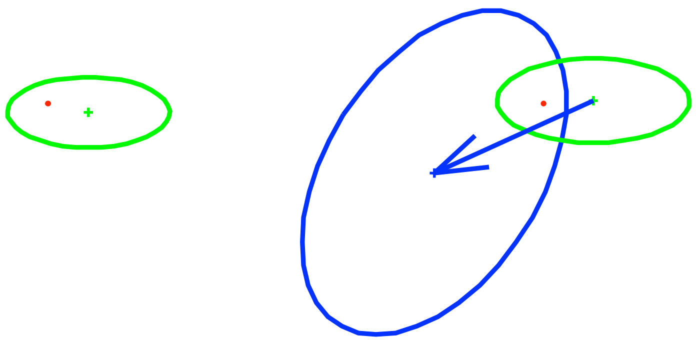
$$\begin{aligned} \mathbf{z}_i &= \mathbf{h}_{ij}(\mathbf{x}^B) + \mathbf{w}_i \\ \mathbf{z}_i &\simeq \mathbf{h}_{ij}(\hat{\mathbf{x}}^B) + \mathbf{H}_{ij}(\mathbf{x}^B - \hat{\mathbf{x}}^B) \end{aligned}$$

$$\begin{aligned} E[\mathbf{w}_i \mathbf{w}_i^T] &= \mathbf{R}_i \\ \mathbf{H}_{ij} &= \left. \frac{\partial \mathbf{h}_{ij}}{\partial \mathbf{x}^B} \right|_{(\hat{\mathbf{x}}^B)} \end{aligned}$$

- E_i and F_j are compatible if:

$$\begin{aligned} D_{ij}^2 &= (\mathbf{z}_i - \mathbf{h}_{ij}(\hat{\mathbf{x}}^B))^T \mathbf{P}_{ij}^{-1} (\mathbf{z}_i - \mathbf{h}_{ij}(\hat{\mathbf{x}}^B)) < \chi_{d,\alpha}^2 \\ \mathbf{P}_{ij} &= \mathbf{H}_{ij} \mathbf{P}^B \mathbf{H}_{ij}^T + \mathbf{R}_i \end{aligned}$$

d = length(z_i)

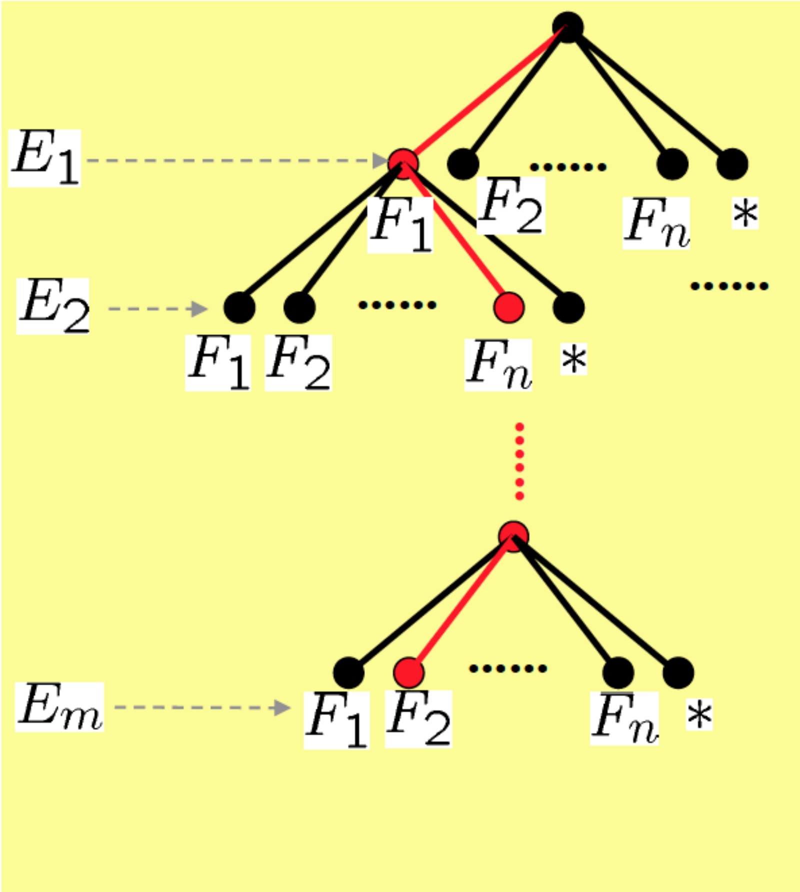


Nearest Neighbor

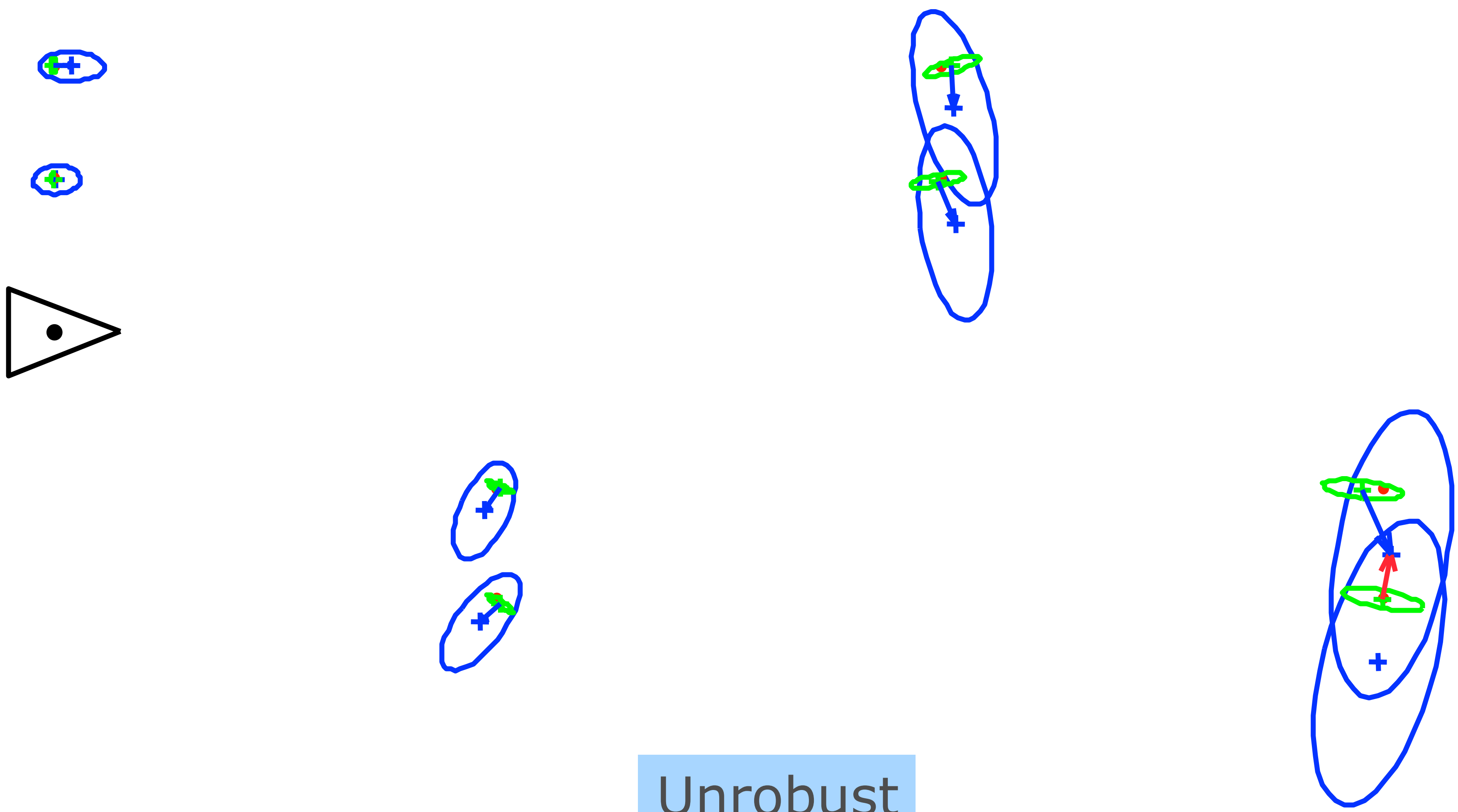
Algorithm 2 Individual Compatibility Nearest Neighbor ICNN $(E_{1\cdots m}, F_{1\cdots n})$

for $i = 1$ to m **do** {measurement E_i}
  $D^2_{\texttt{min}} \leftarrow \texttt{mahalanobis2}(E_i, F_1)$
  nearest $\leftarrow 1$
  for $j = 2$ to n **do** {feature F_j}
    $D^2_{ij} \leftarrow \texttt{mahalanobis2}(E_i, F_j)$
    if $D^2_{ij} < D^2_{\texttt{min}}$ **then**
      nearest $\leftarrow j$
      $D^2_{\texttt{min}} \leftarrow D^2_{ij}$
    end if
  end for
  if $D^2_{\texttt{min}} \leq \chi^2_{d_i, 1-\alpha}$ **then**
    $\mathcal{H}_i \leftarrow$ nearest
  else
    $\mathcal{H}_i \leftarrow 0$
  end if
end for
return $\mathcal{H}$

Greedy algorithm: *O(mn)*



The Fallacy of the Nearest Neighbor



Unrobust

Joint Compatibility

- Given a hypothesis $\mathcal{H} = [j_1, j_2, \cdots, j_s]$
- Joint measurement equation

$$\mathbf{z}_{\mathcal{H}} = \mathbf{h}_{\mathcal{H}}(\mathbf{x}^B) + \mathbf{w}_{\mathcal{H}}$$

$$\mathbf{h}_{\mathcal{H}} = \begin{bmatrix} \mathbf{h}_{1j_1} \\ \mathbf{h}_{2j_2} \\ \vdots \\ \mathbf{h}_{sj_s} \end{bmatrix}$$

- The joint hypothesis is compatible if:

$$D^2_{\mathcal{H}} = (\mathbf{z}_{\mathcal{H}} - \mathbf{h}_{\mathcal{H}}(\hat{\mathbf{x}}^B))^T C_{\mathcal{H}}^{-1} (\mathbf{z}_{\mathcal{H}} - \mathbf{h}_{\mathcal{H}}(\hat{\mathbf{x}}^B)) < \chi^2_{d,\alpha}$$

$$C_{\mathcal{H}} = \mathbf{H}_{\mathcal{H}} \mathbf{P}^B \mathbf{H}_{\mathcal{H}}^T + \mathbf{R}_{\mathcal{H}}$$

d = length(z)

Joint Compatibility Branch and Bound

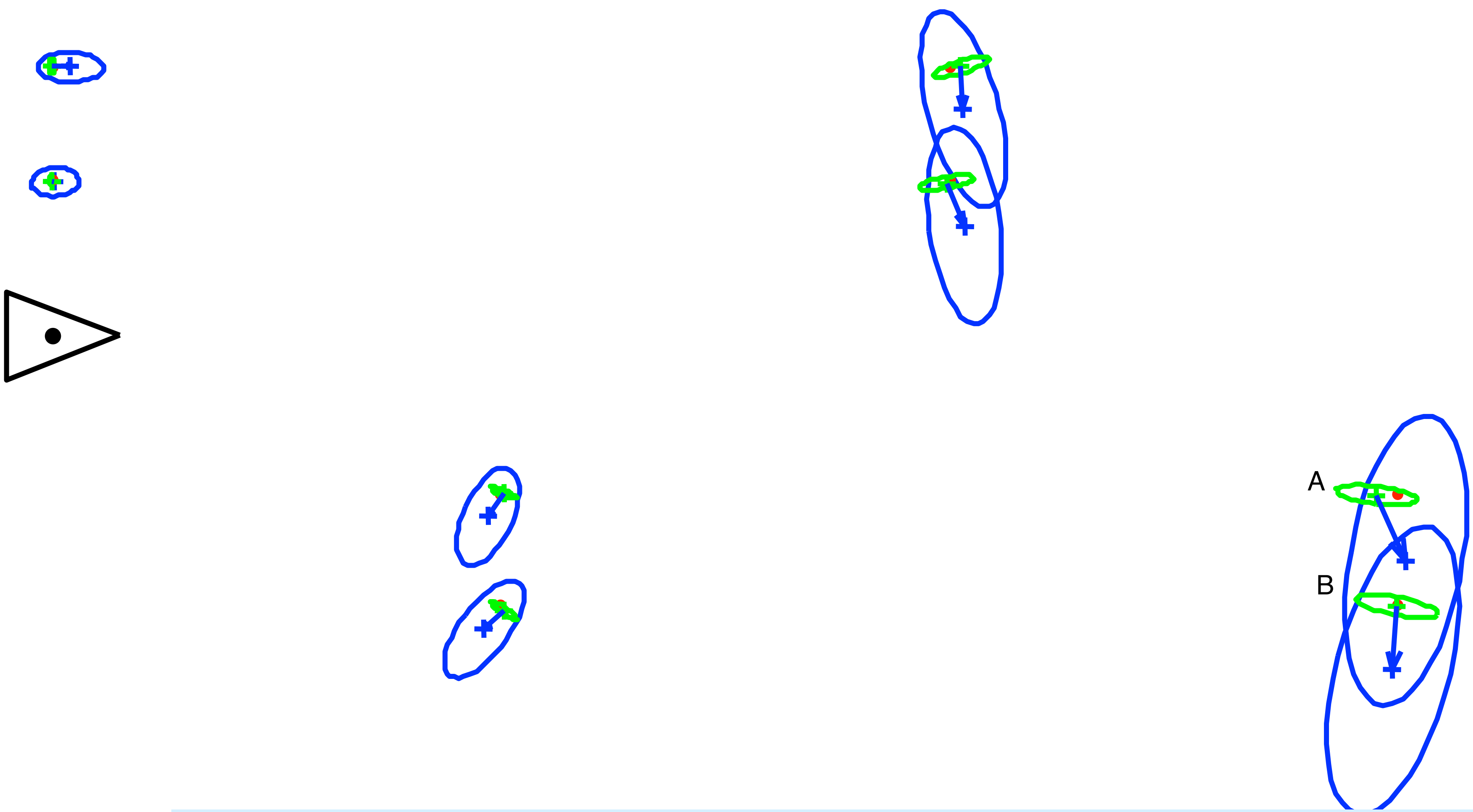
- Find the largest hypothesis with **jointly consistent** pairings

```
procedure JCBB (H, i):  -- find pairings for observation E_i

if i > m -- leaf node?
   if pairings(H) > pairings(Best)
      Best = H
   fi
else
   for j in {1...n}
      if individual_compatibility(i, j) and then
         joint_compatibility(H, i, j)
         JCBB([H j], i + 1) -- pairing (E_i, F_j) accepted
      fi
   rof
   if pairings(H) + m - i> pairings(Best) -- can do better?
      JCBB([H 0], i + 1) -- star node, E_i not paired
   fi
fi
```

Selects the largest set of pairings where there is **consensus**

The Fallacy of the Nearest Neighbor

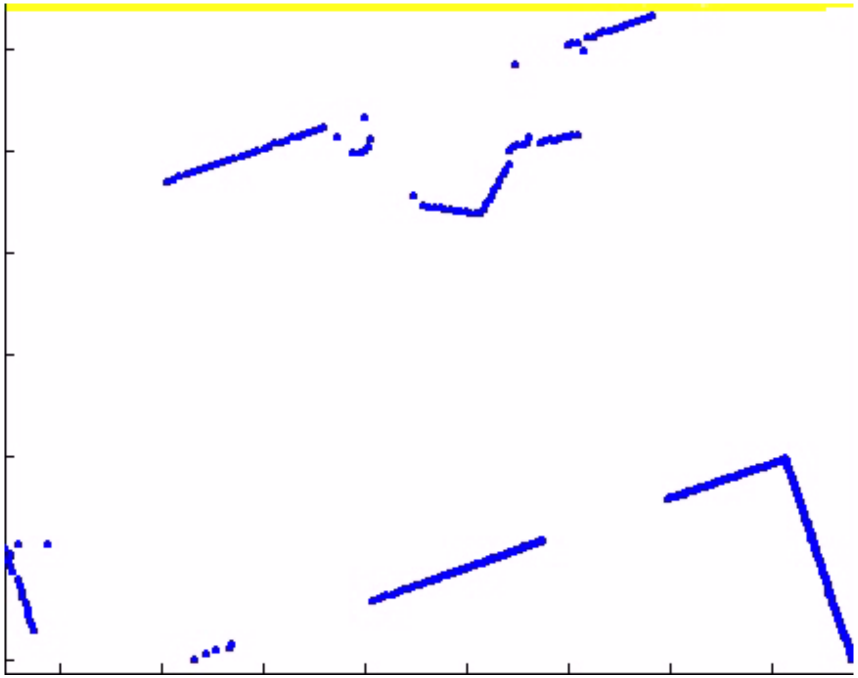


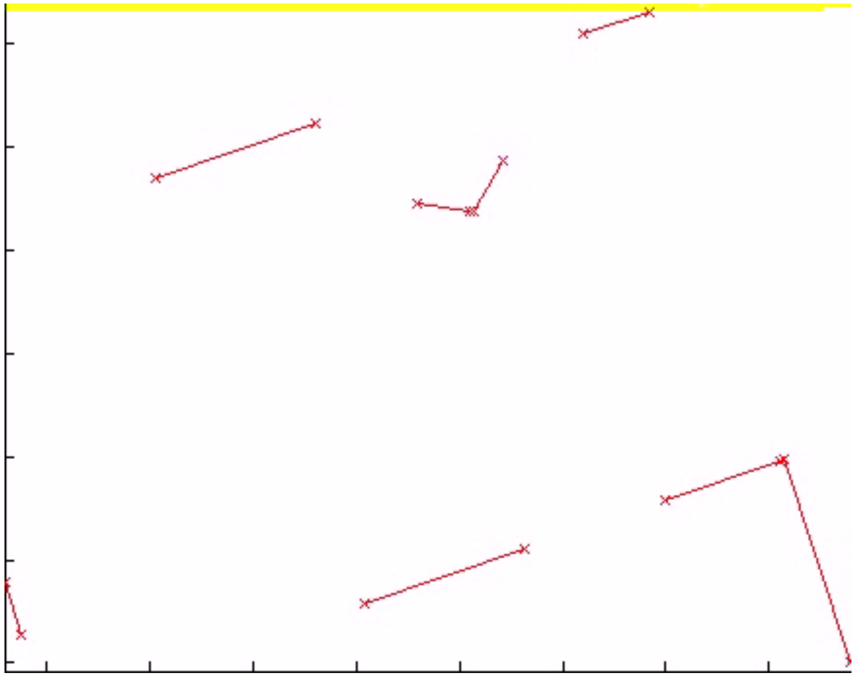
J. Neira, J.D. Tardós. **Data Association in Stochastic Mapping using the Joint Compatibility Test** IEEE Trans. Robotics and Automation, Vol. 17, No. 6, Dec 2001, pp 890 –897

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

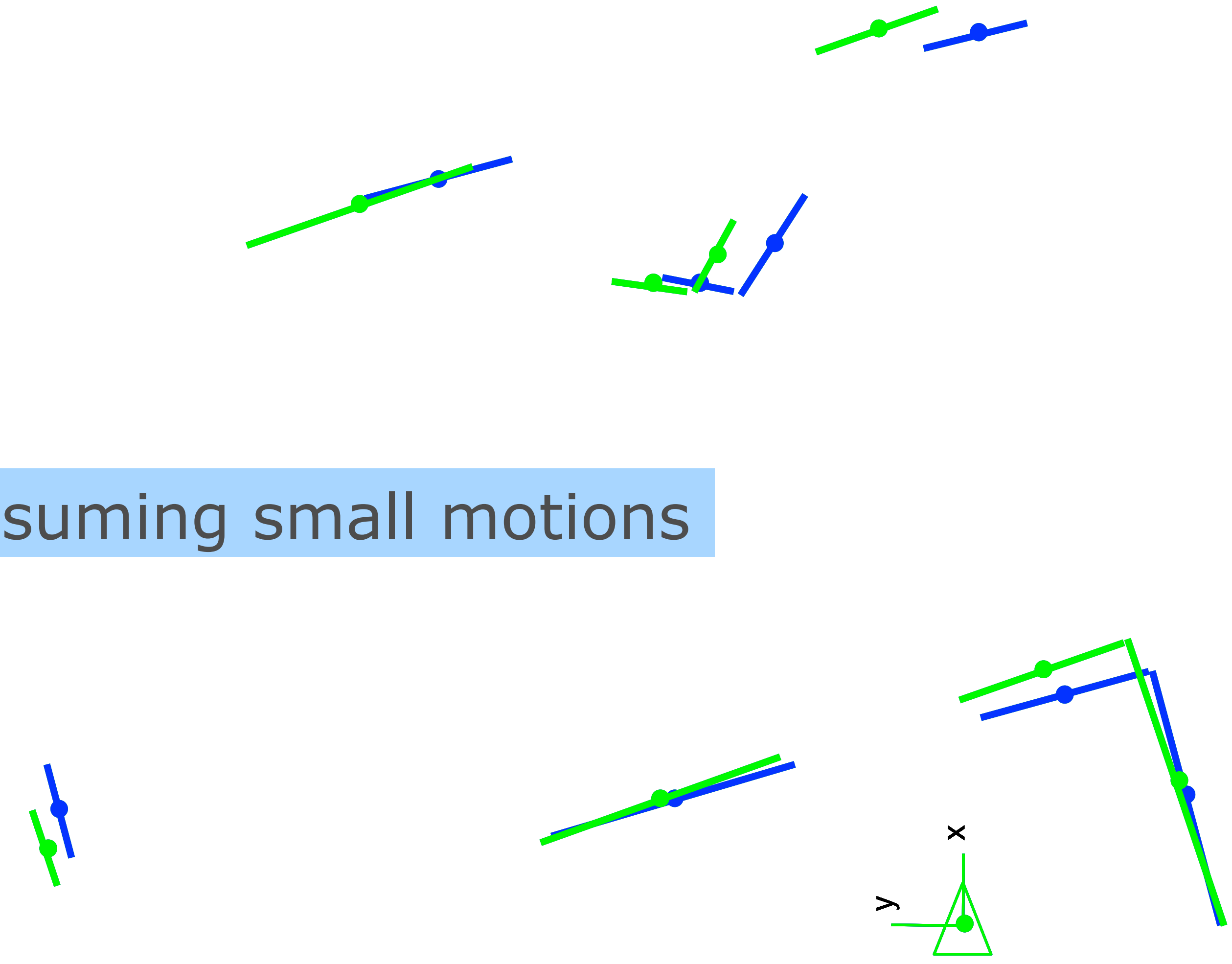
SLAM without odometry

- No estimation of the vehicle motion



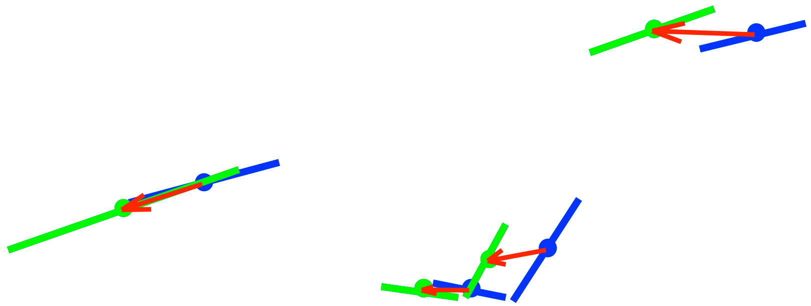


SLAM without odometry

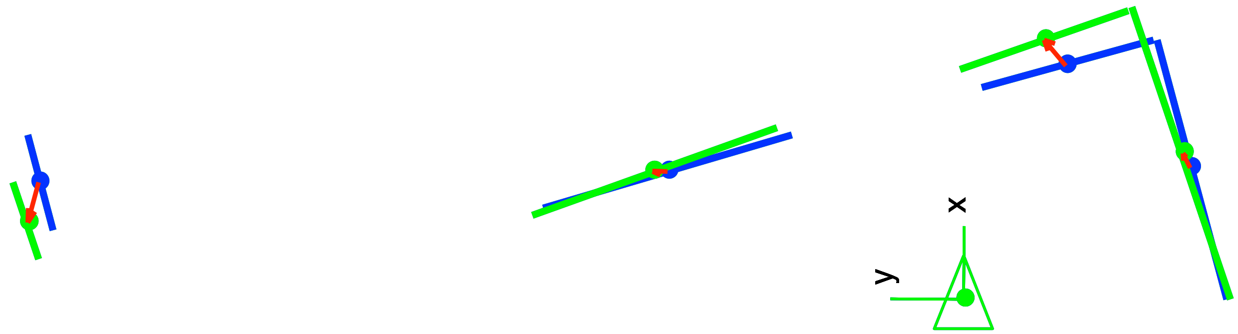


Assuming small motions

SLAM without odometry

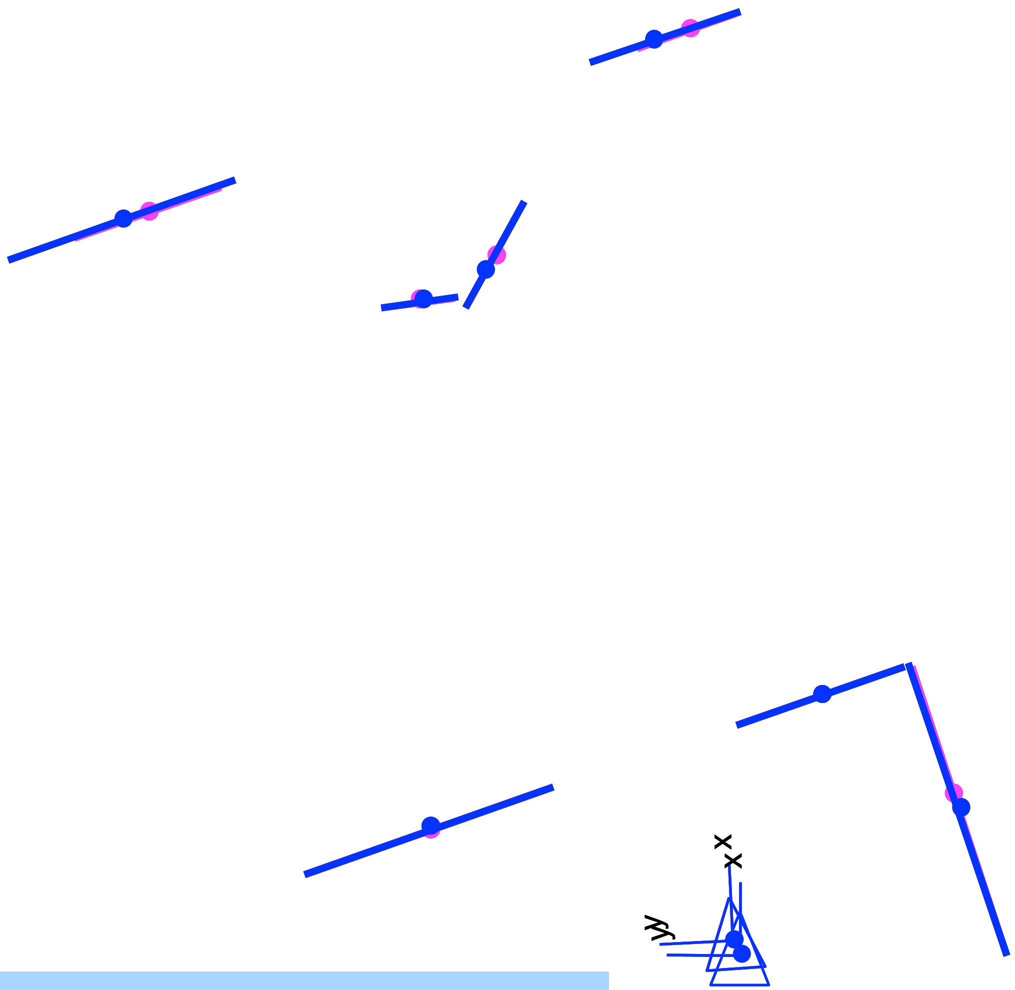


Data association using Joint Compatibility



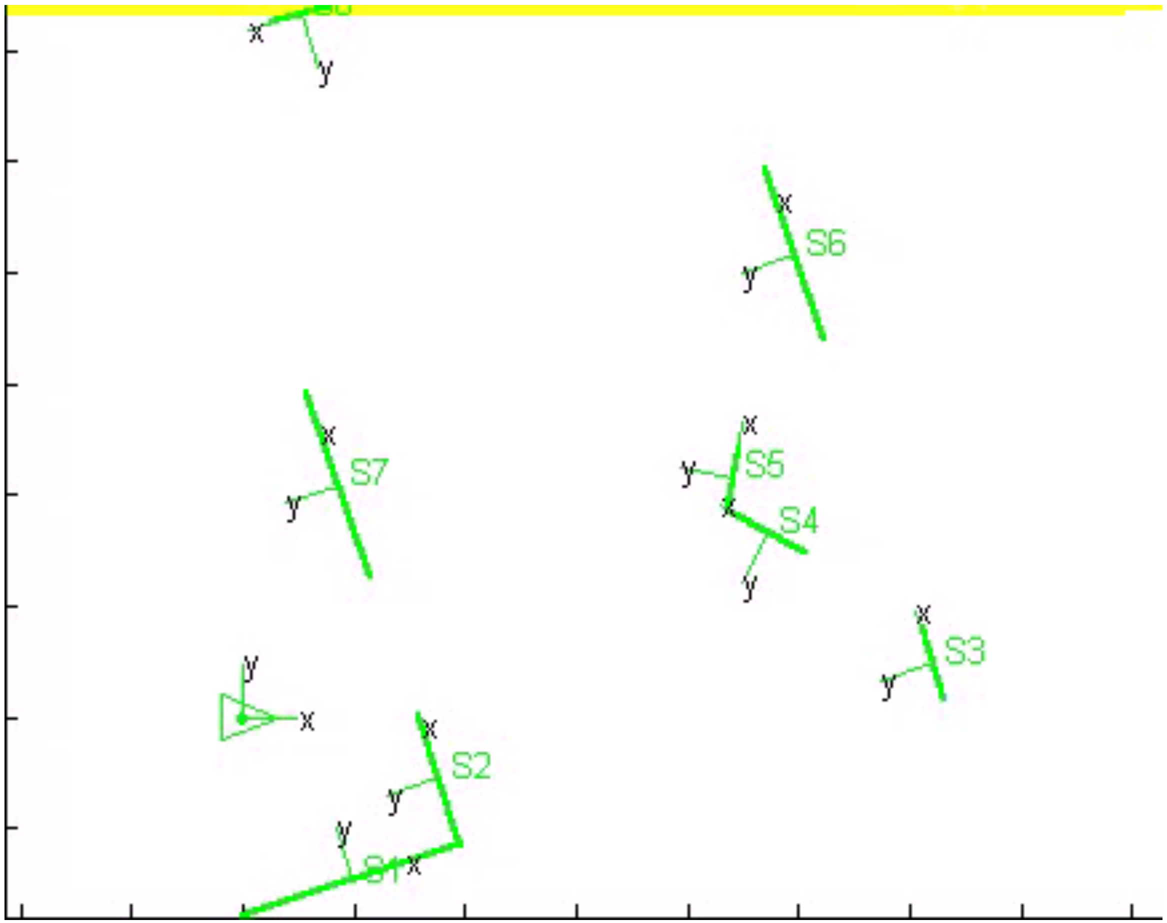
ROBOTICS
UNIVERSIDAD DE ZARAGOZA

SLAM without odometry

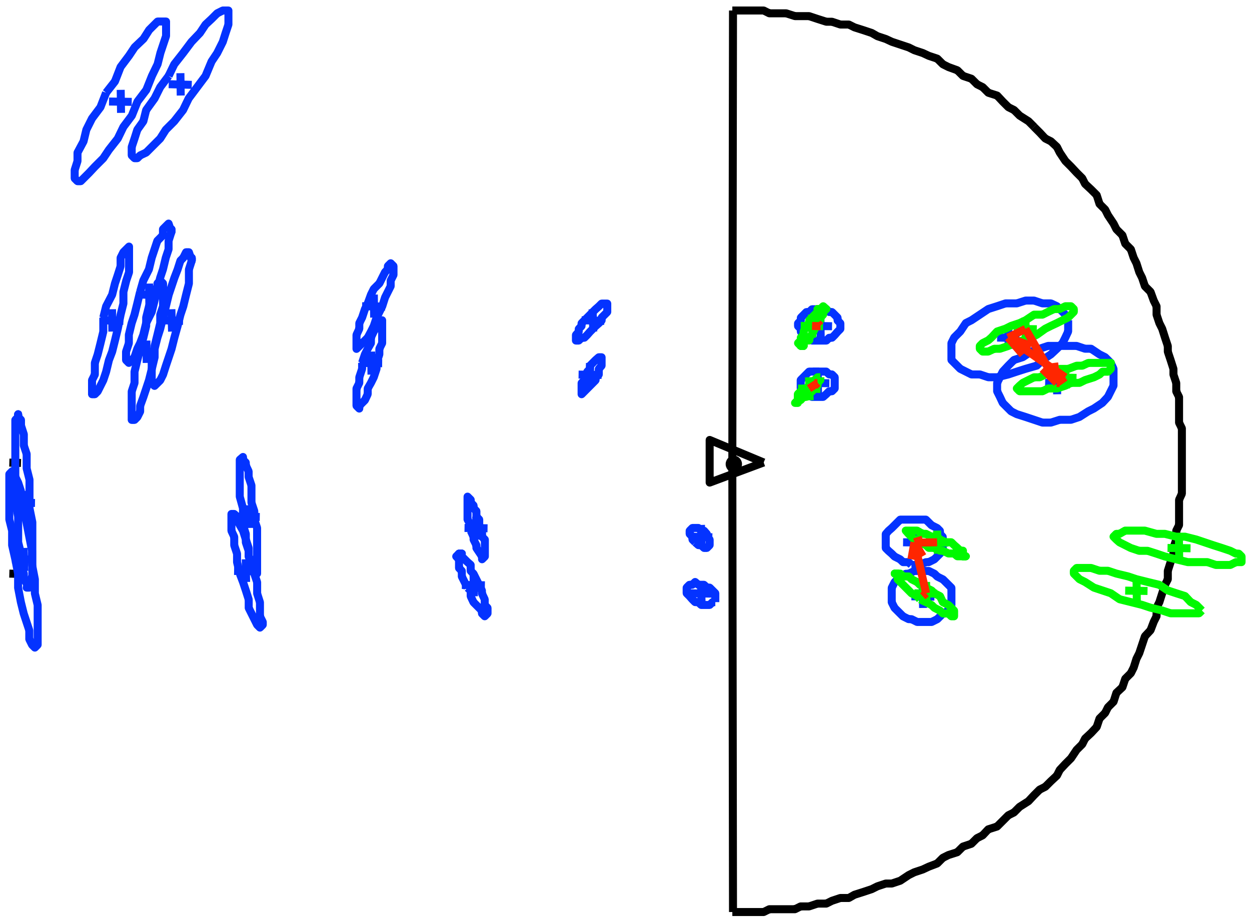


Vehicle motion estimation

SLAM without odometry

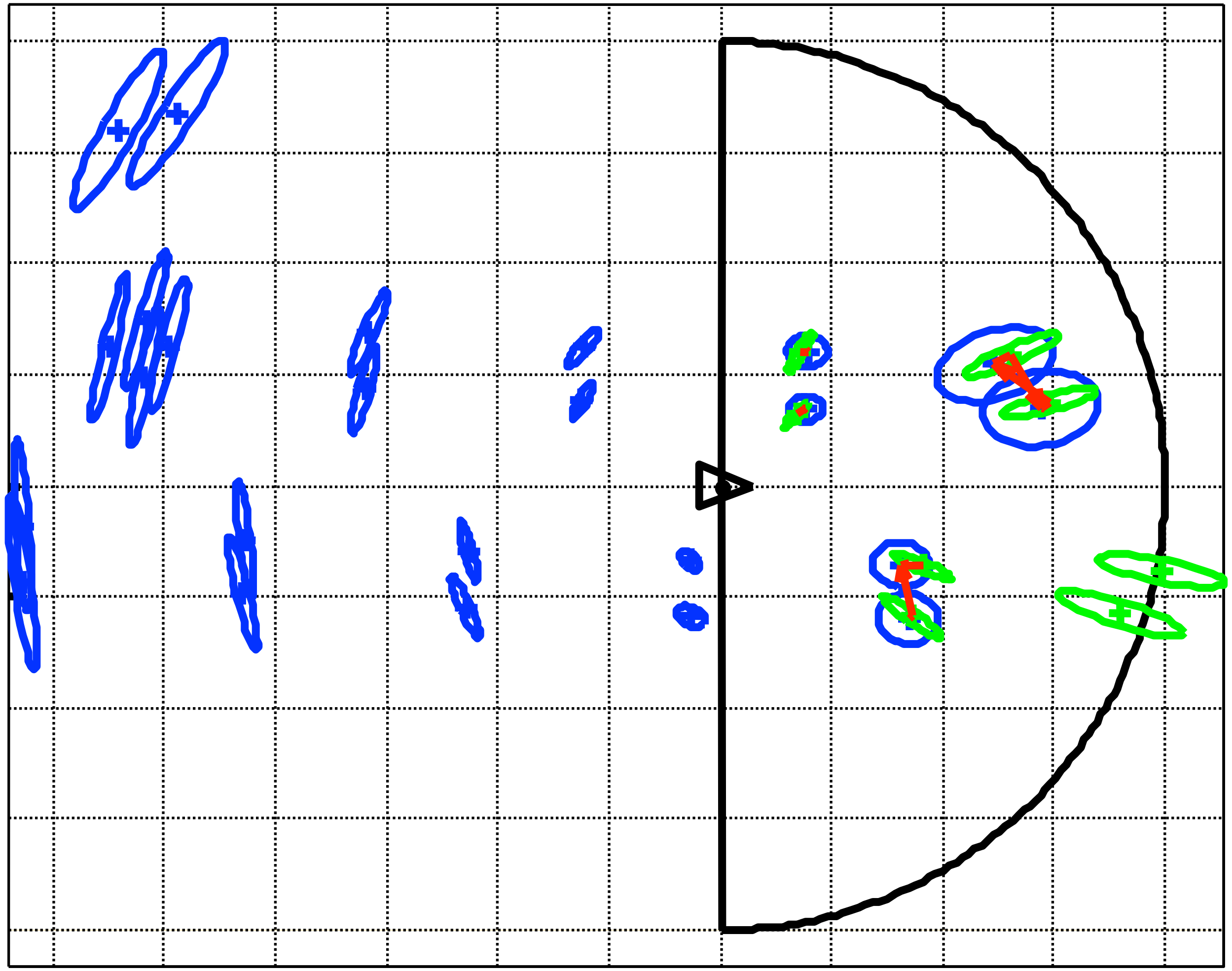


Continuous data association



Individual compatibility is $O(nm) = O(n)$

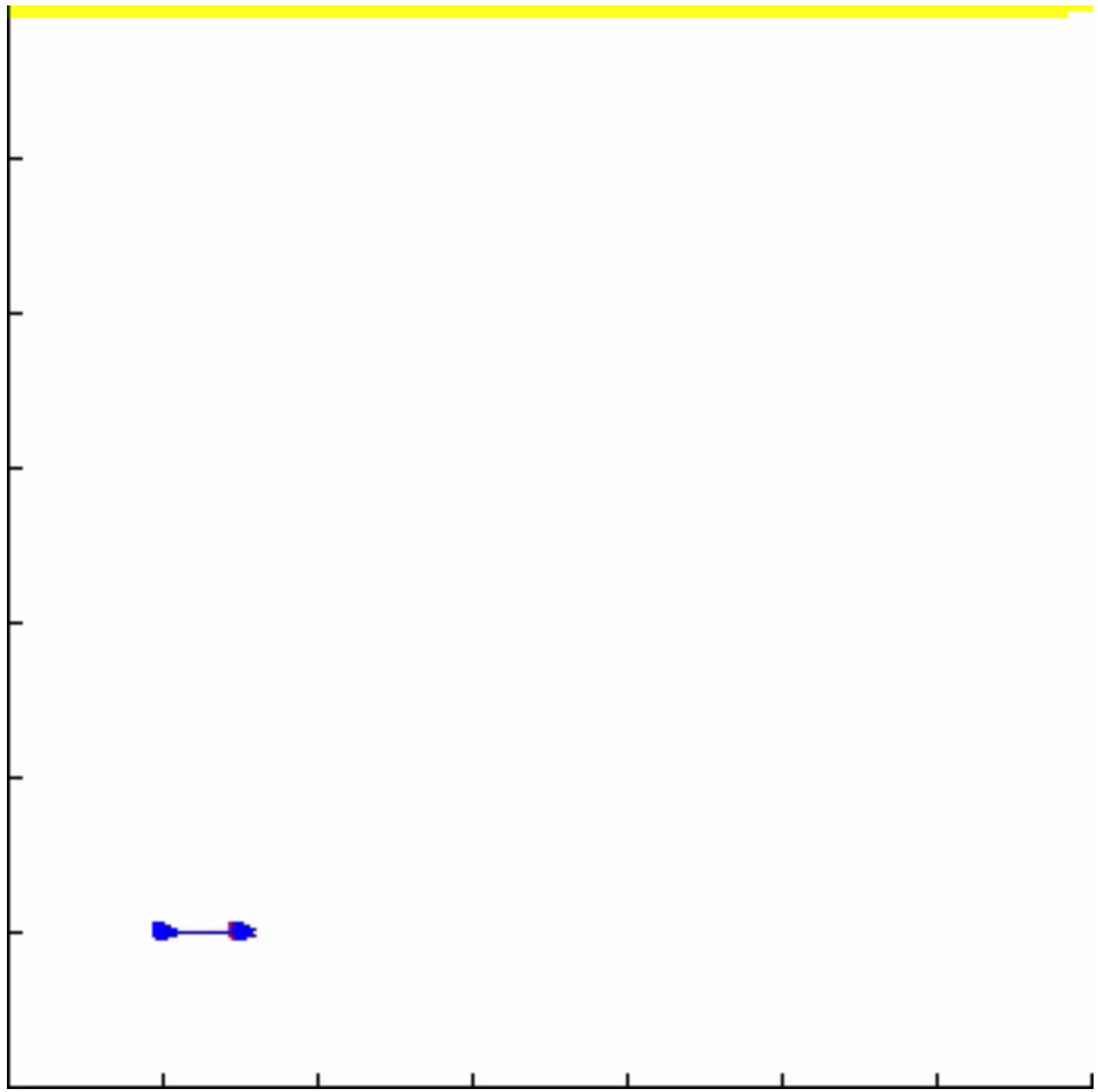
Map tessellation



Individual compatibility can be *O(1)*

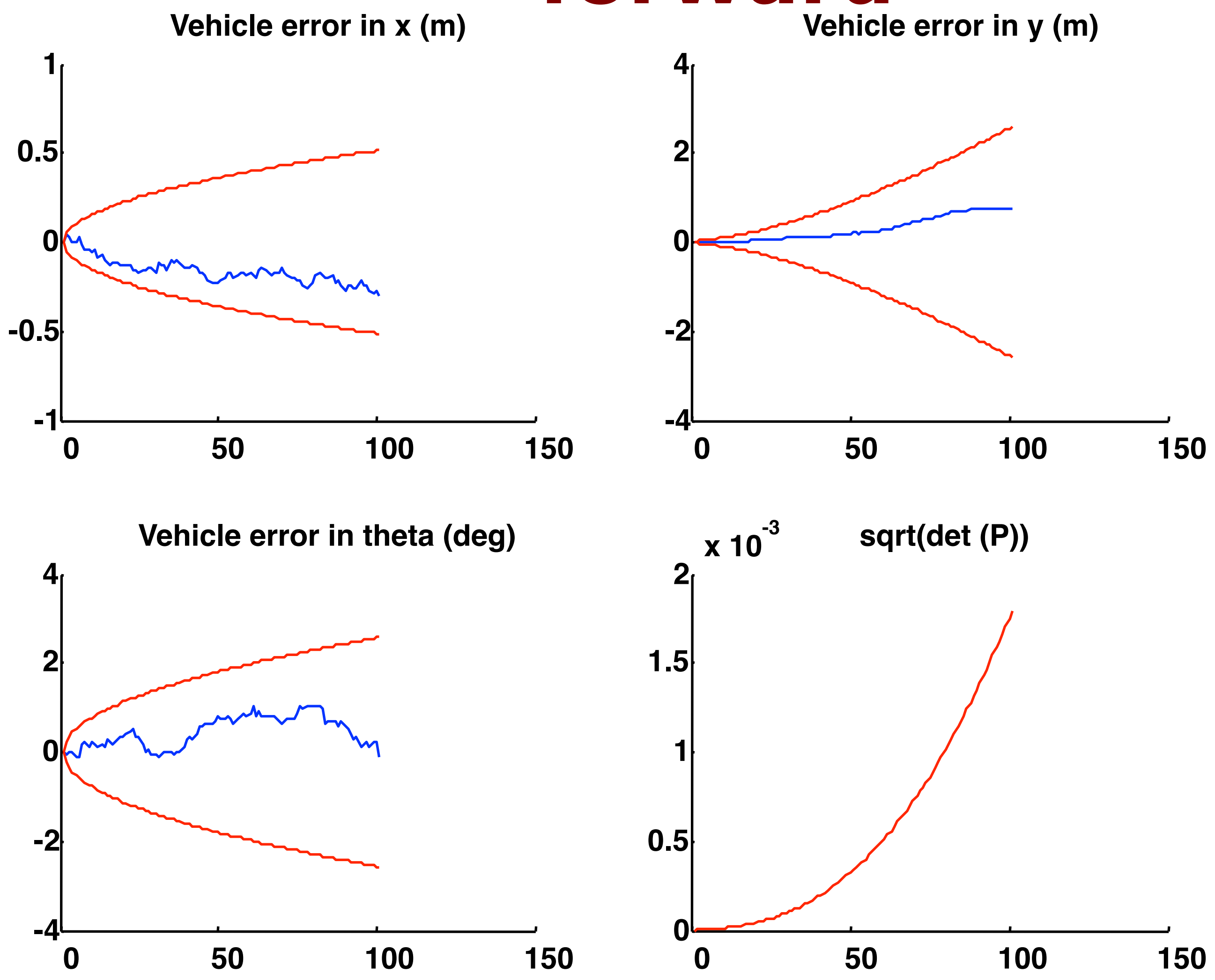
4. The loop closing problem

Why we do SLAM

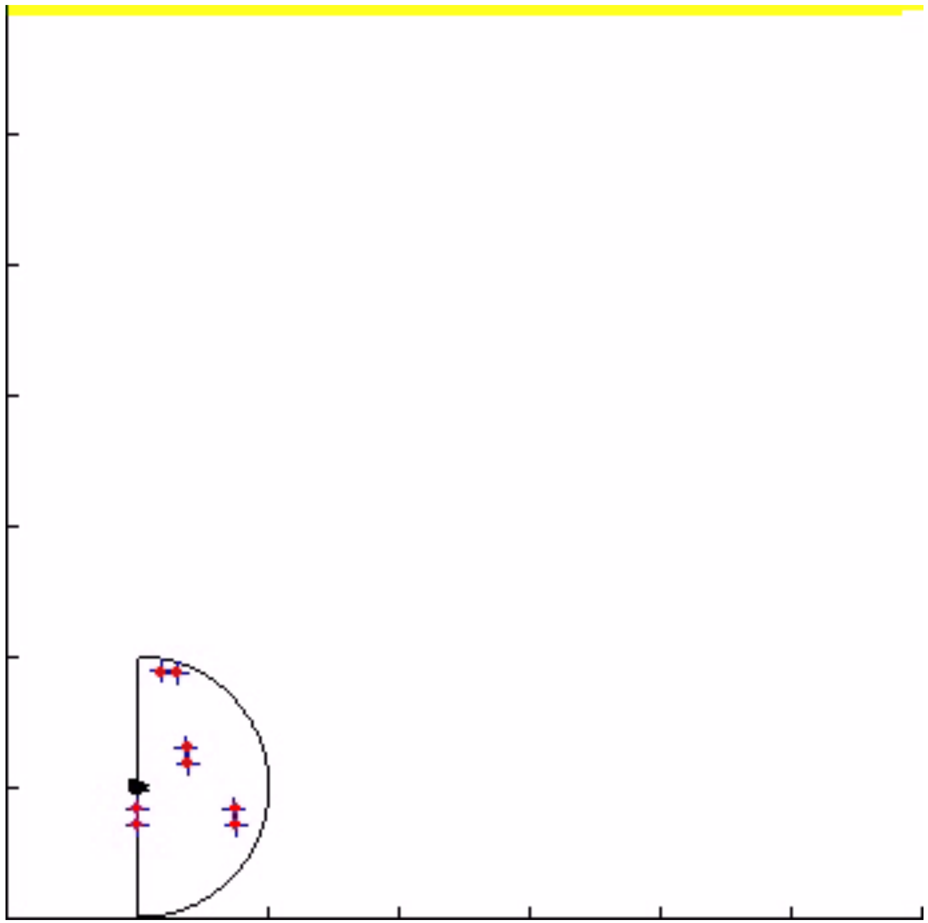


Dead-reckoning drift

Dead-reckoning, moving forward

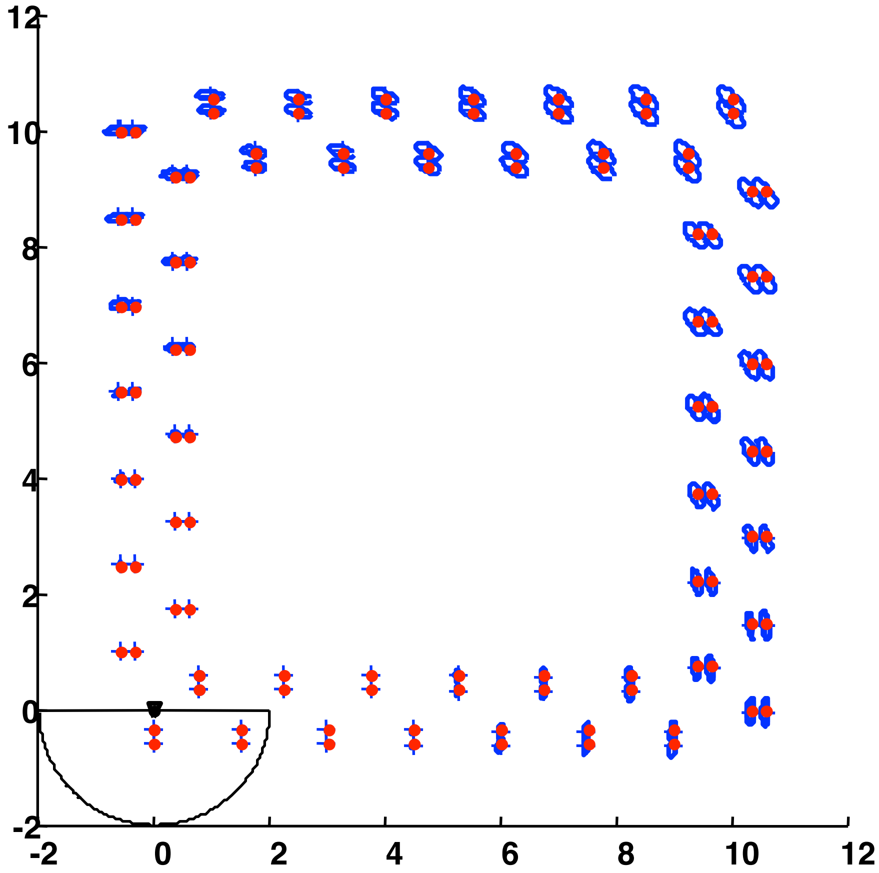


Bad news...



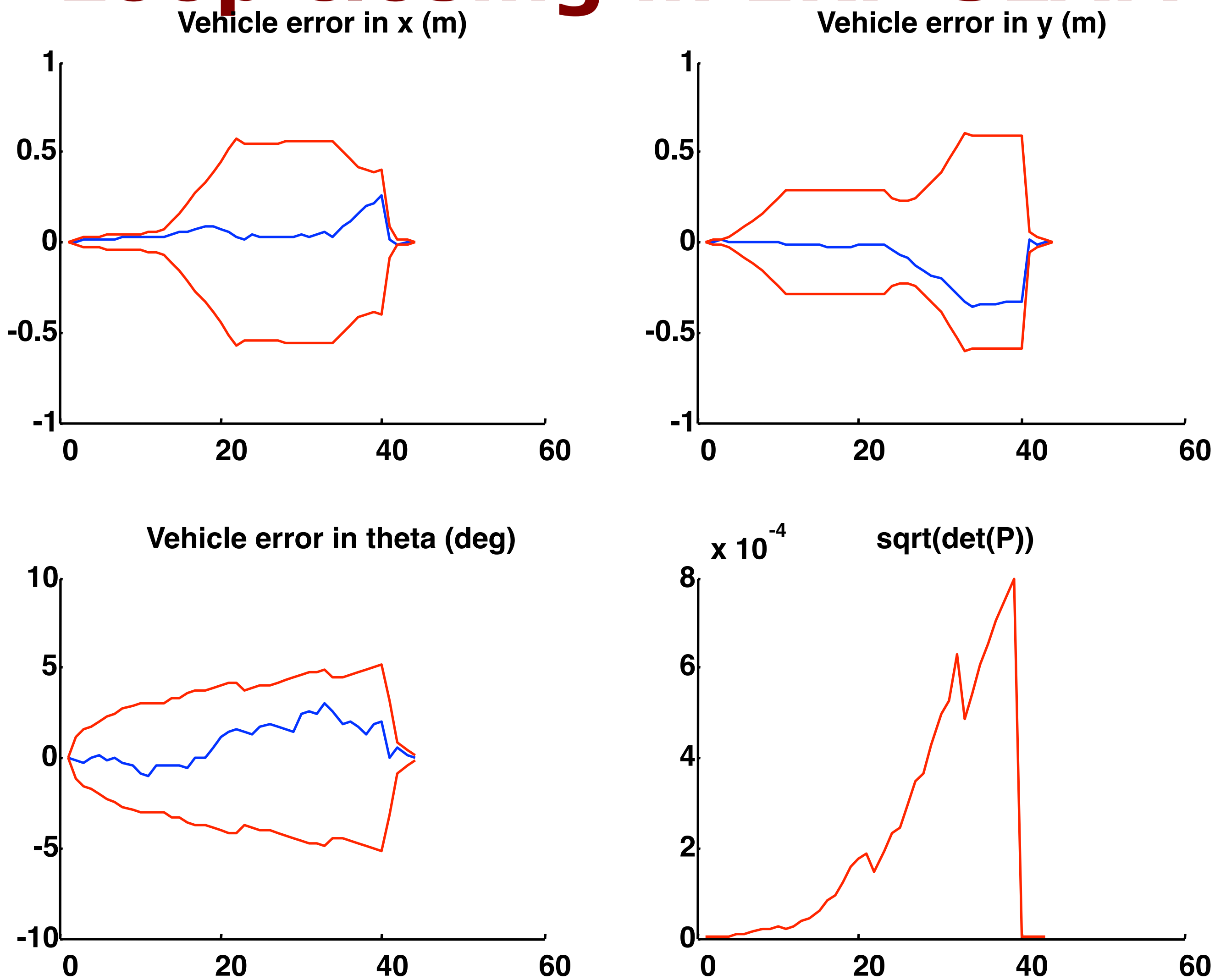
Uncertainty still grows!

Good news!



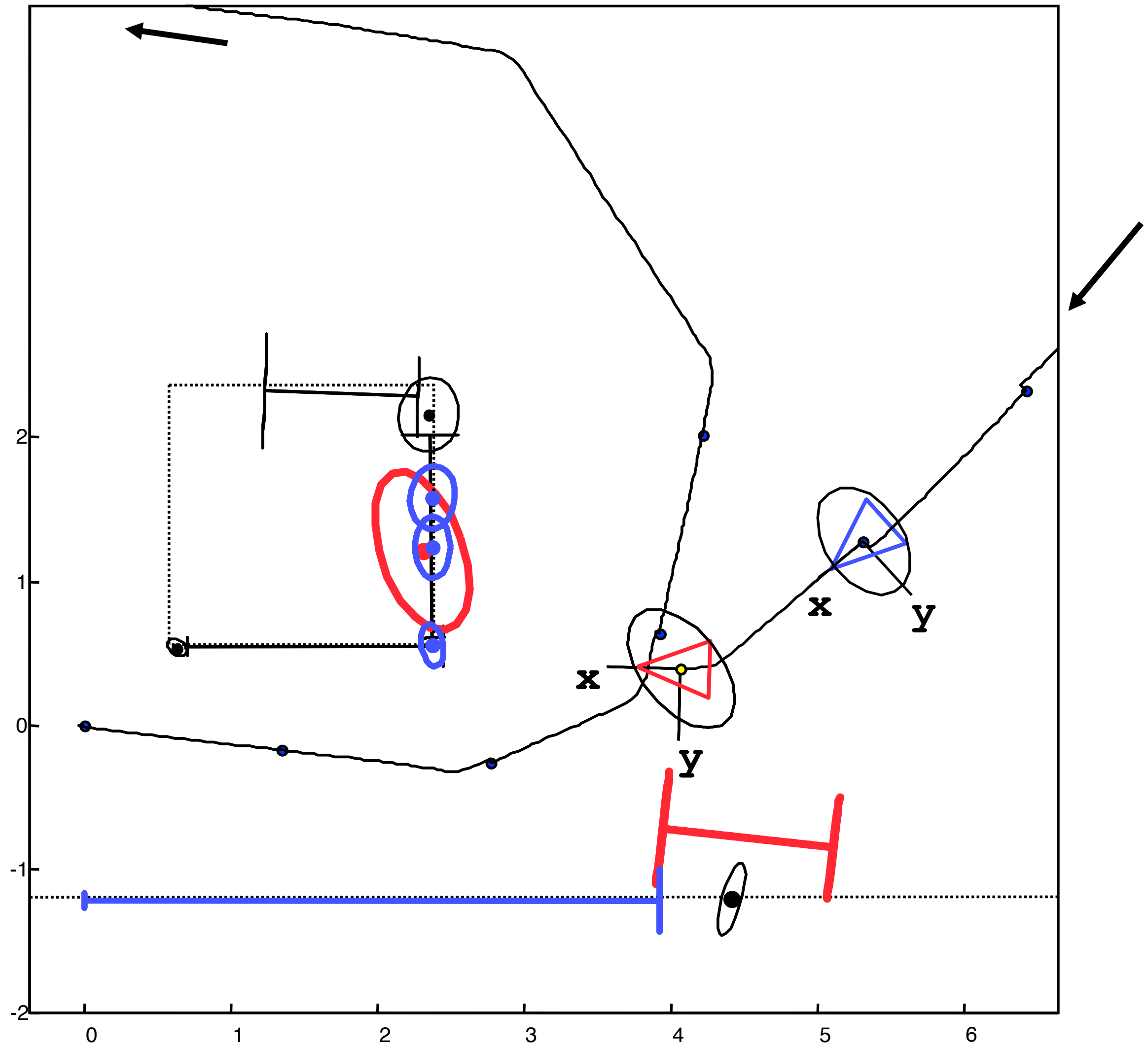
Loop closing reduces uncertainty!

Loop closing in EKF-SLAM



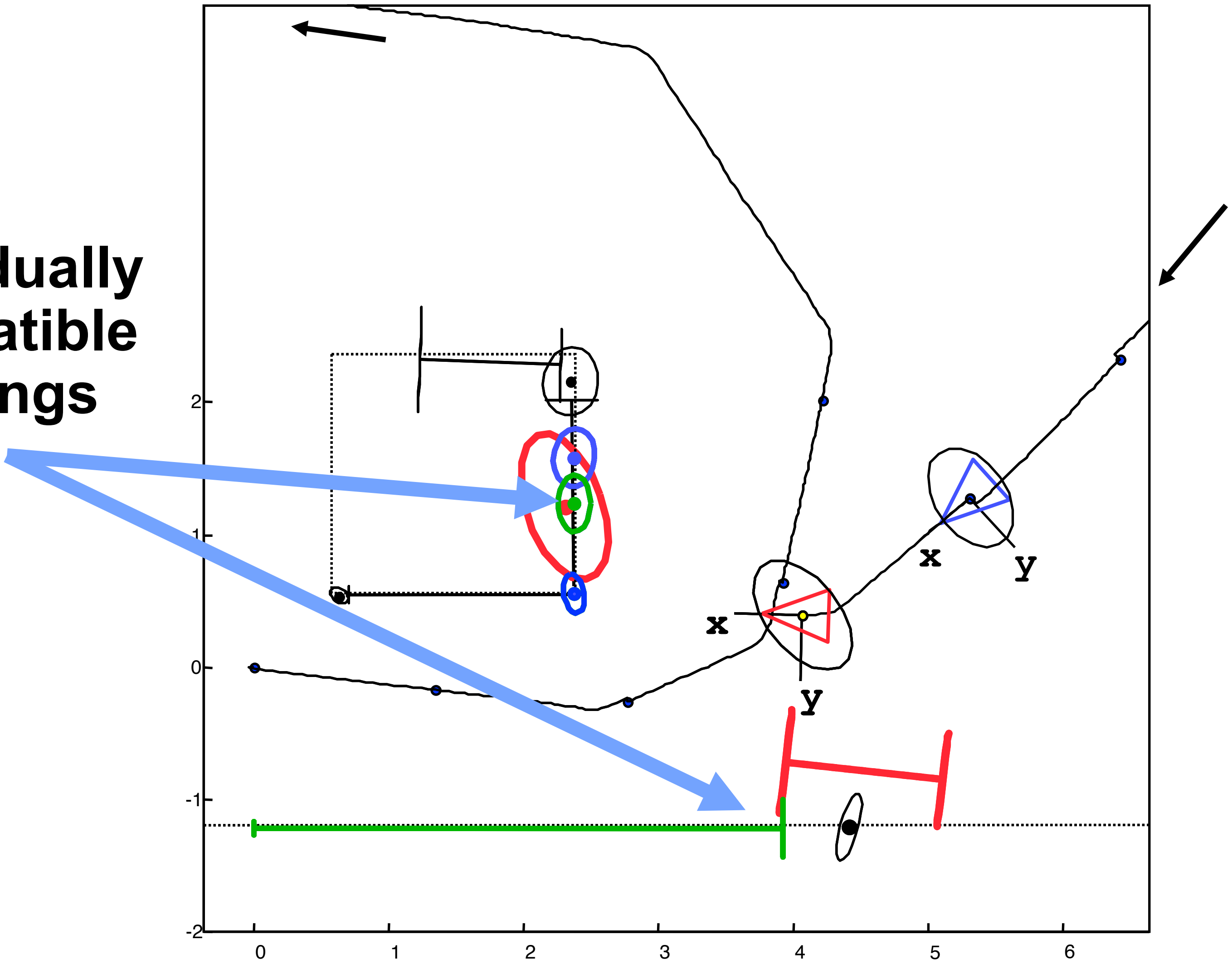
Loop closing reduces uncertainty!

Loop closing



Nearest Neighbor

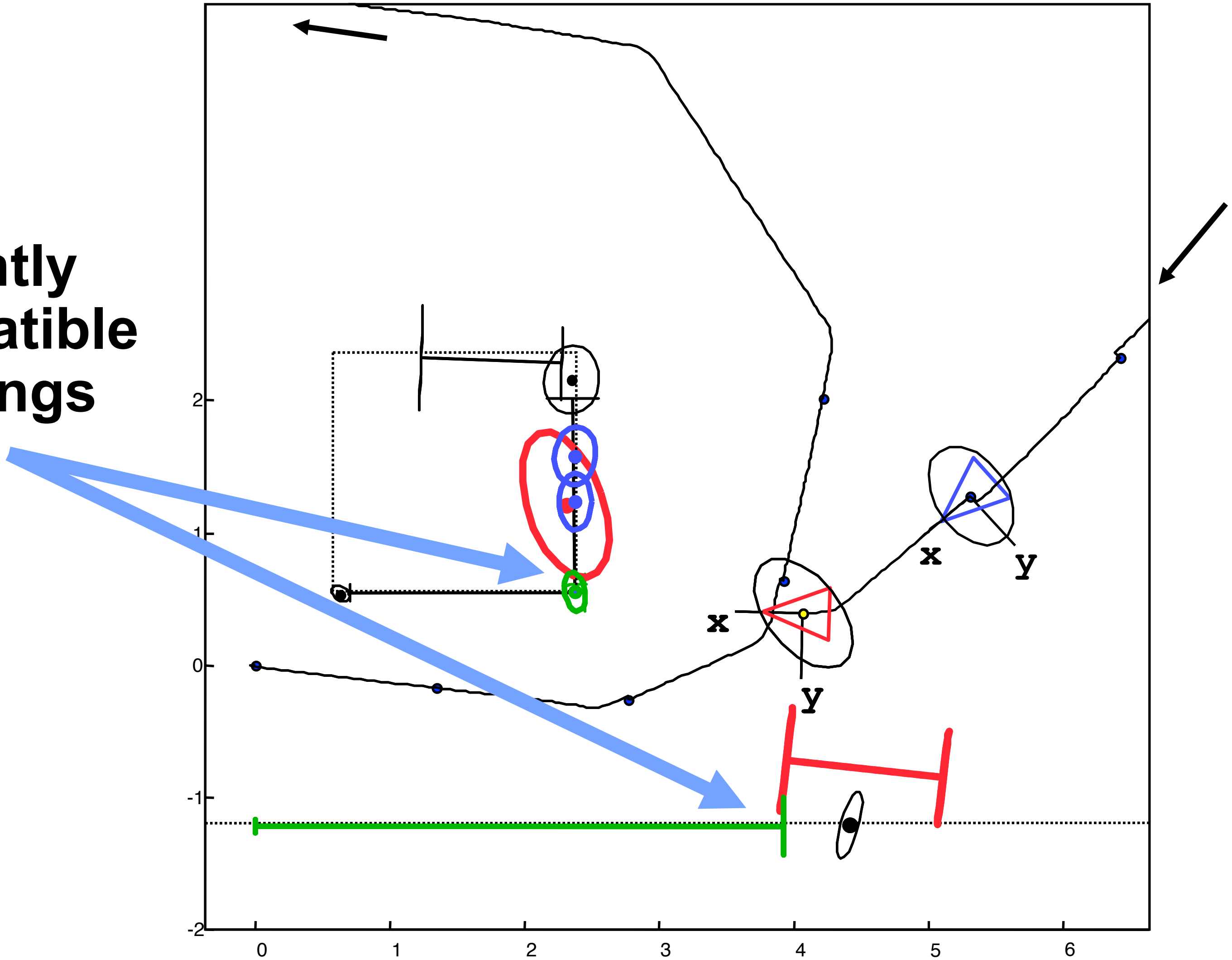
Individually compatible pairings



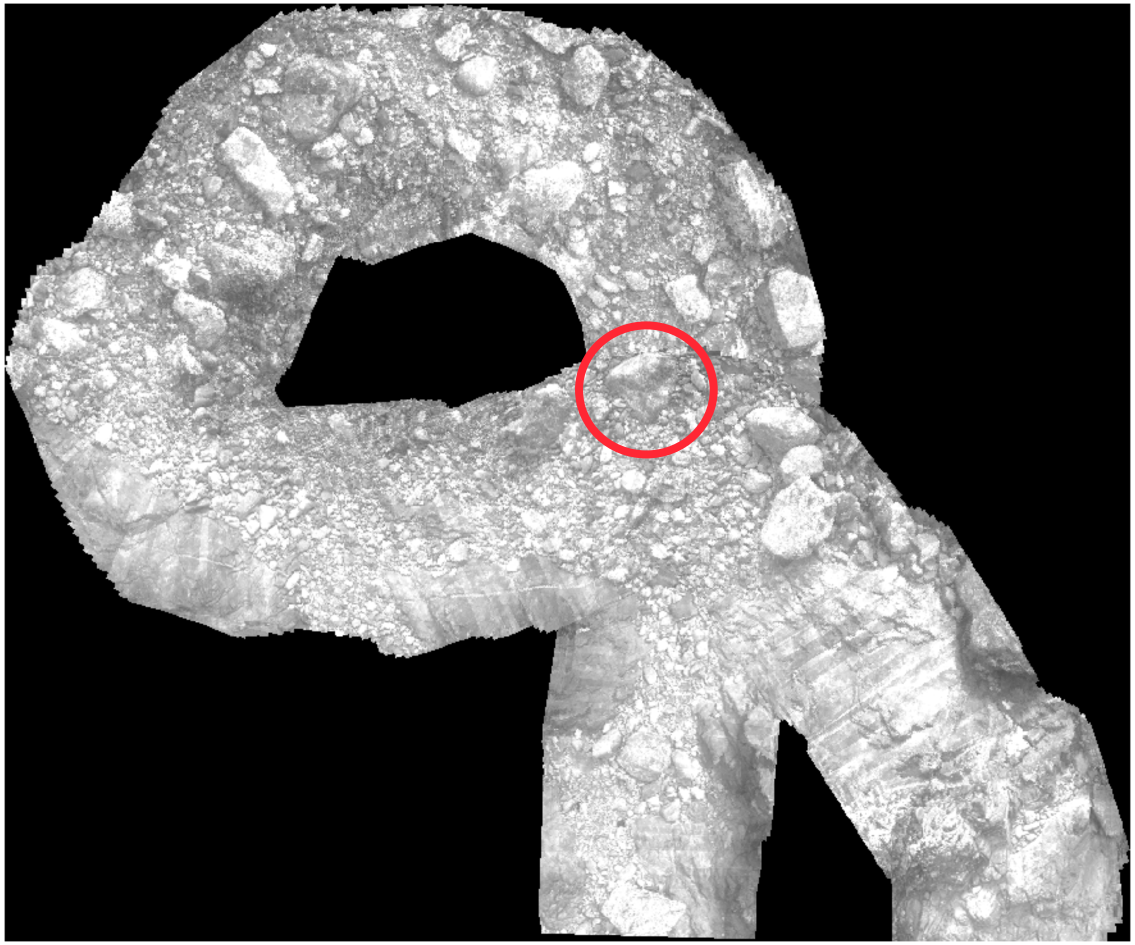
Wrong answer!

Joint Compatibility at work

Jointly compatible pairings

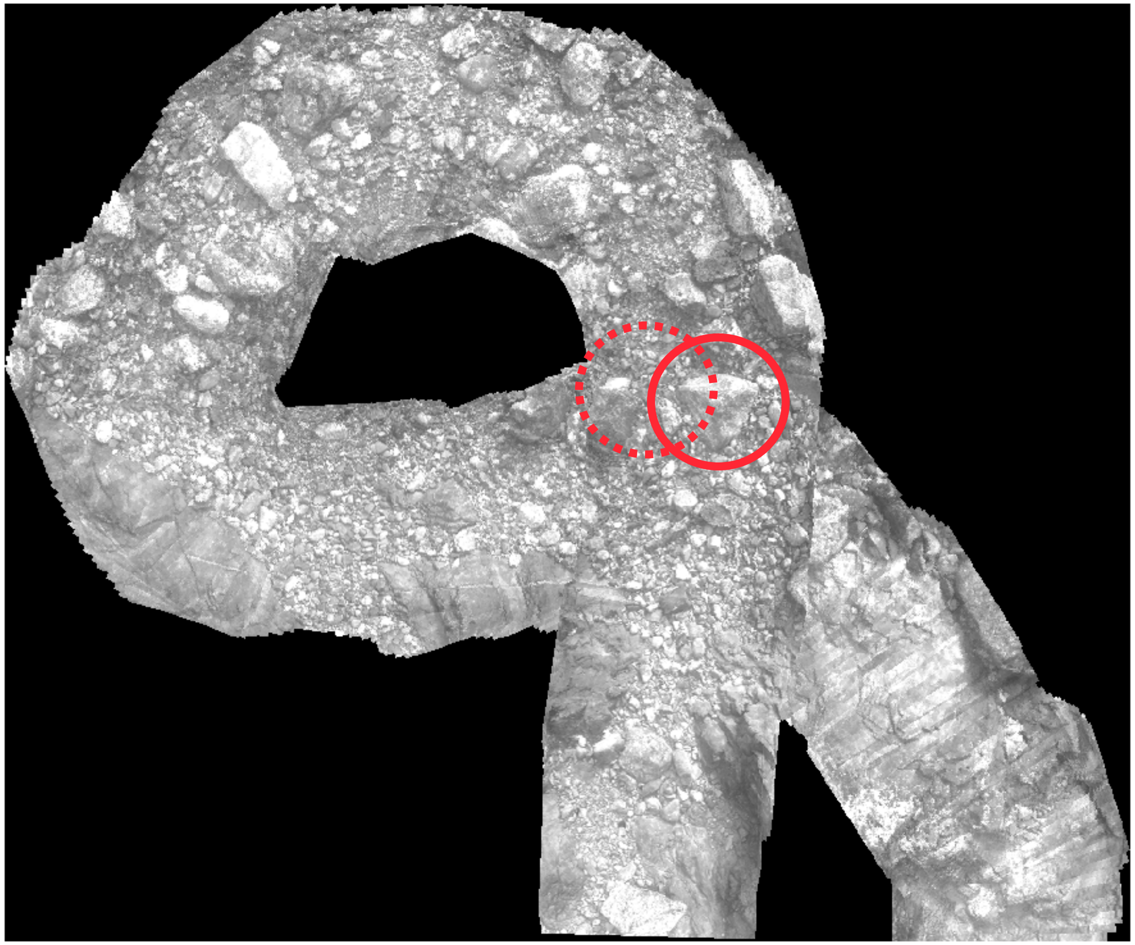


Loop closing in mosaicing use first



Joint work with R. García, University of Girona

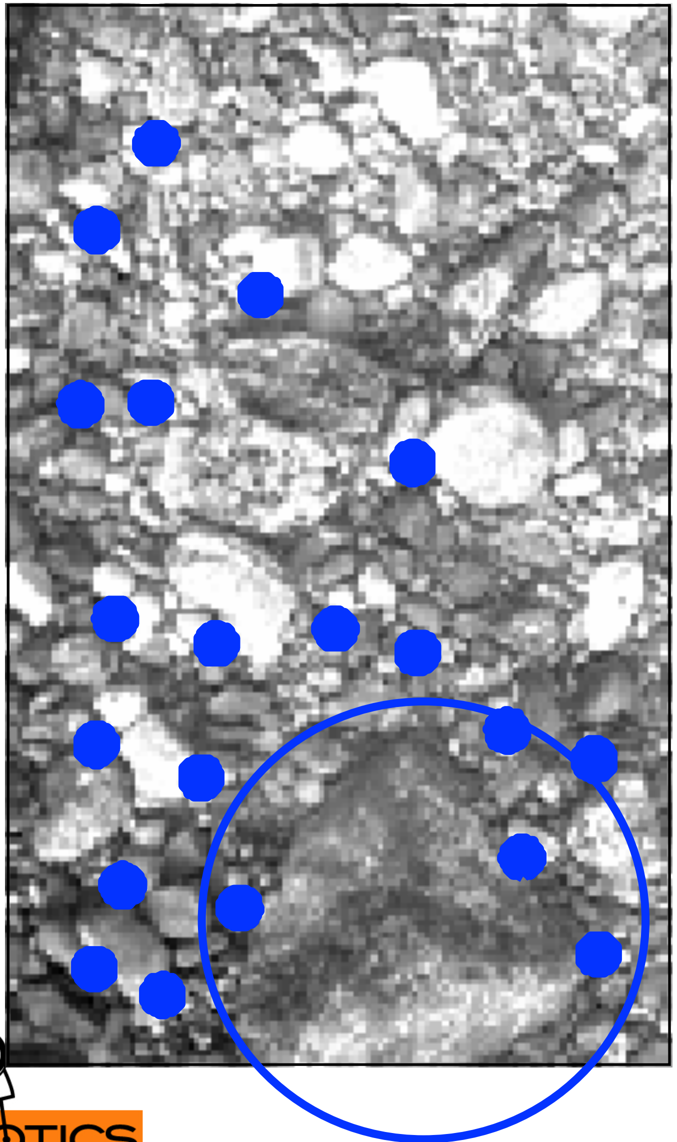
Loop closing in mosaicing: use Last



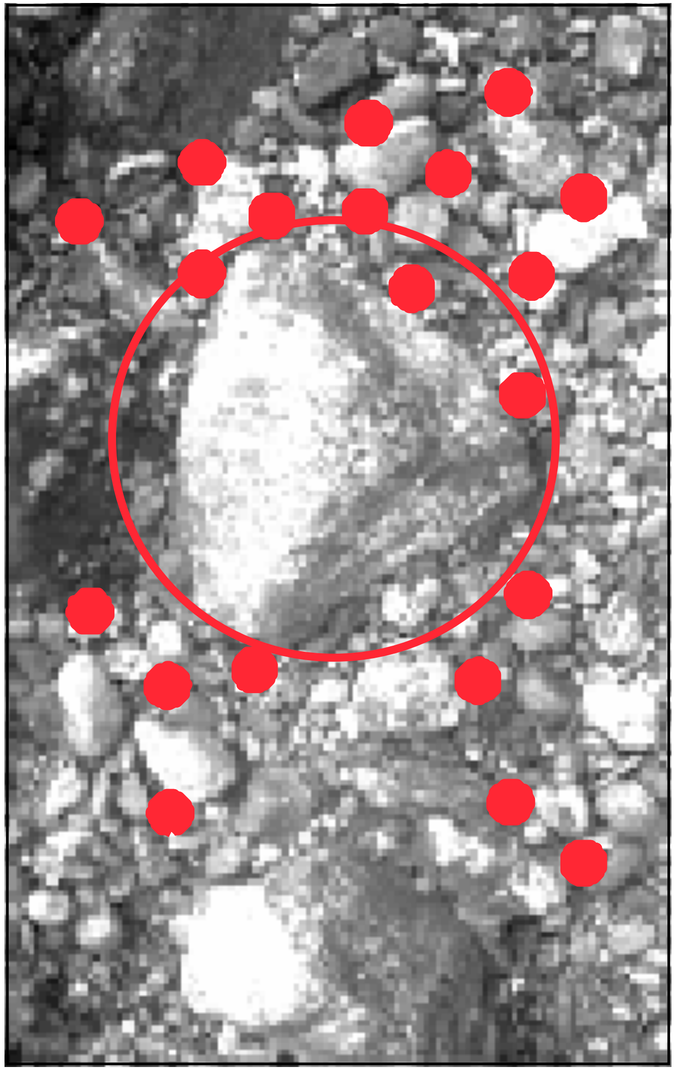
Sequential mosaicing is a form of odometry

The loop closing problem

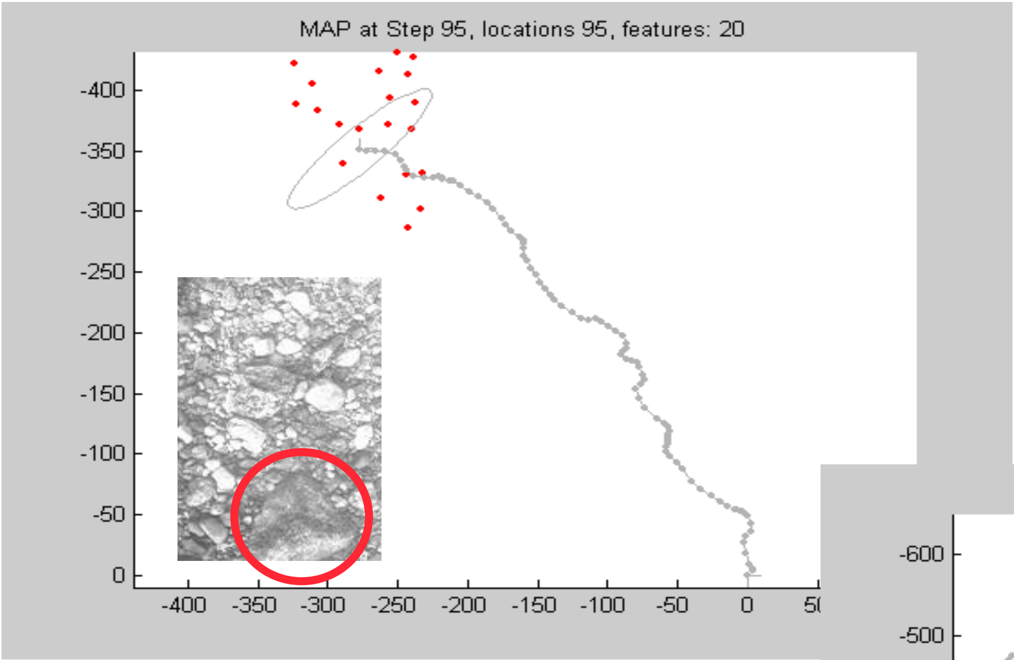
- Loop beginning

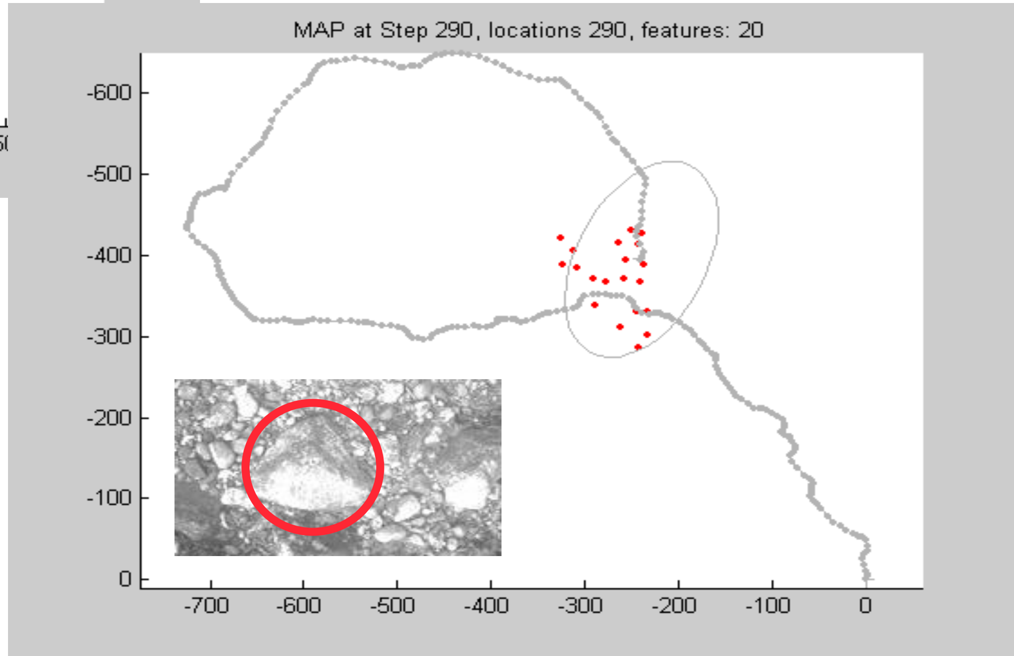


- Loop end

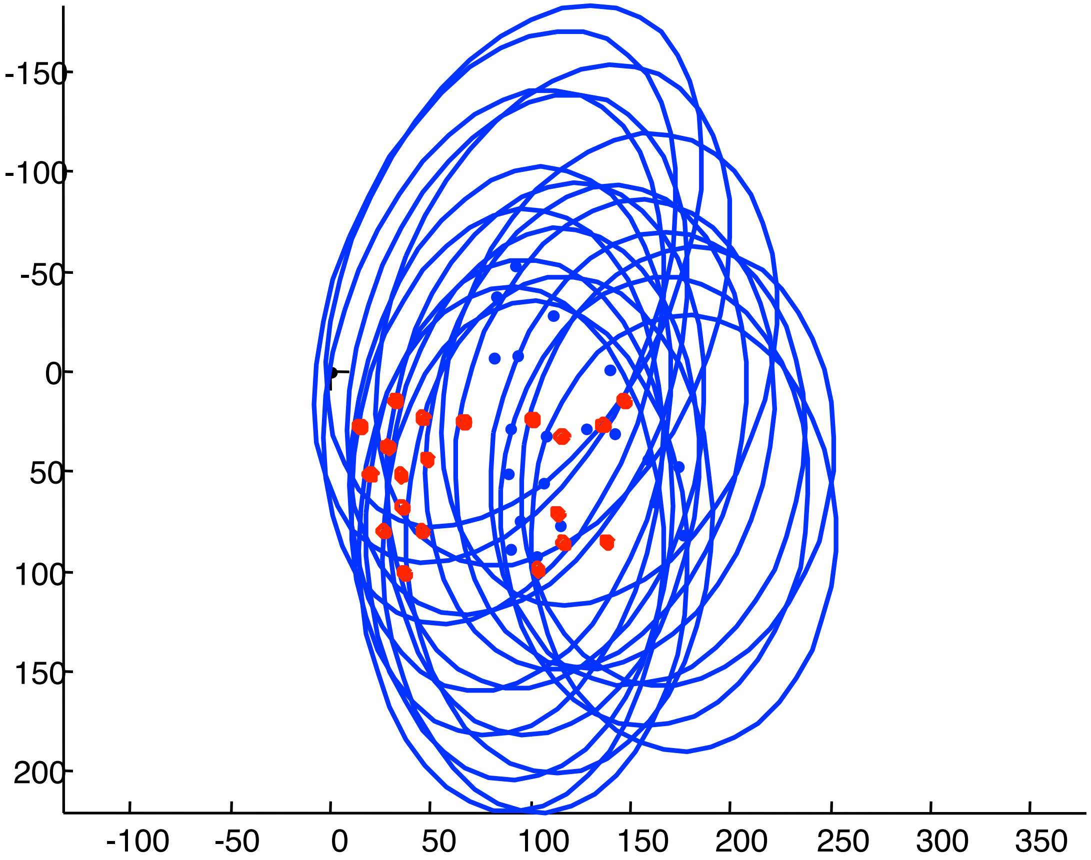


The loop closing problem





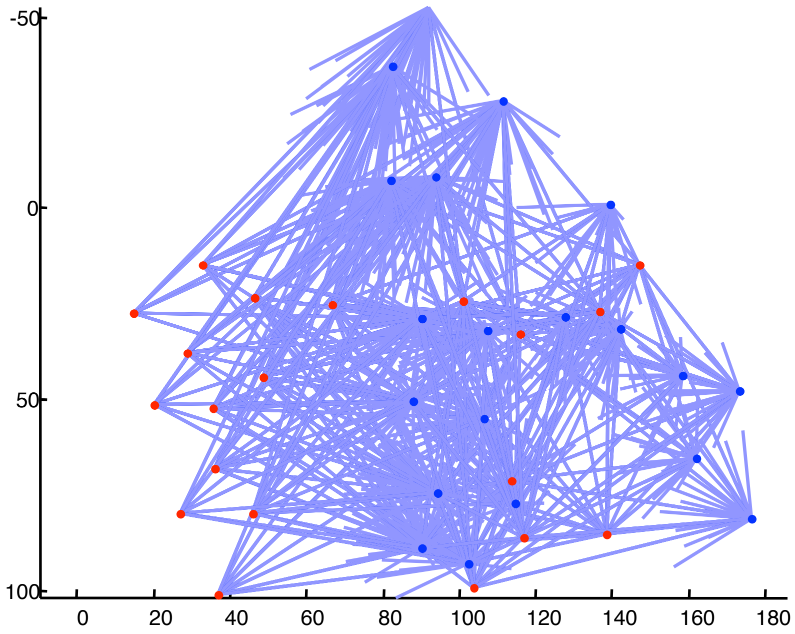
The loop closing problem



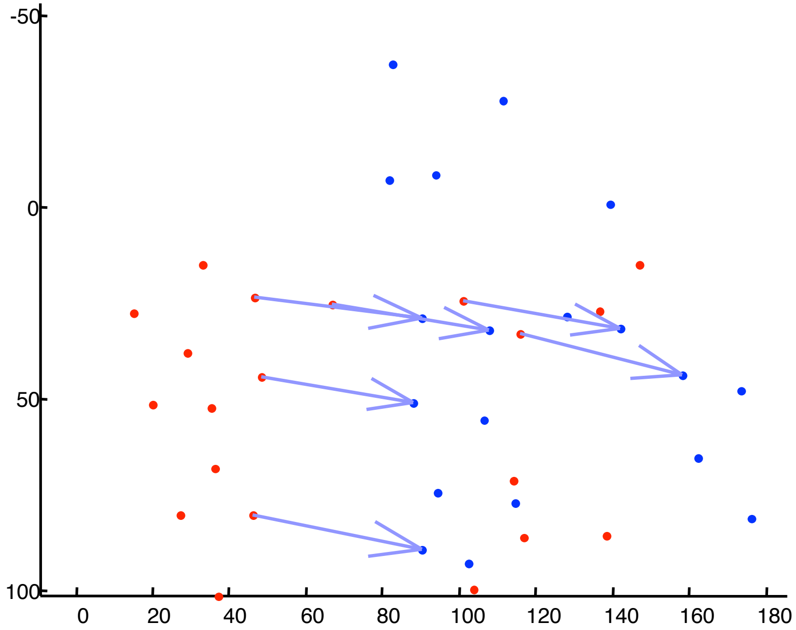
Measurements (red) and predicted features (blue)

The loop closing problem

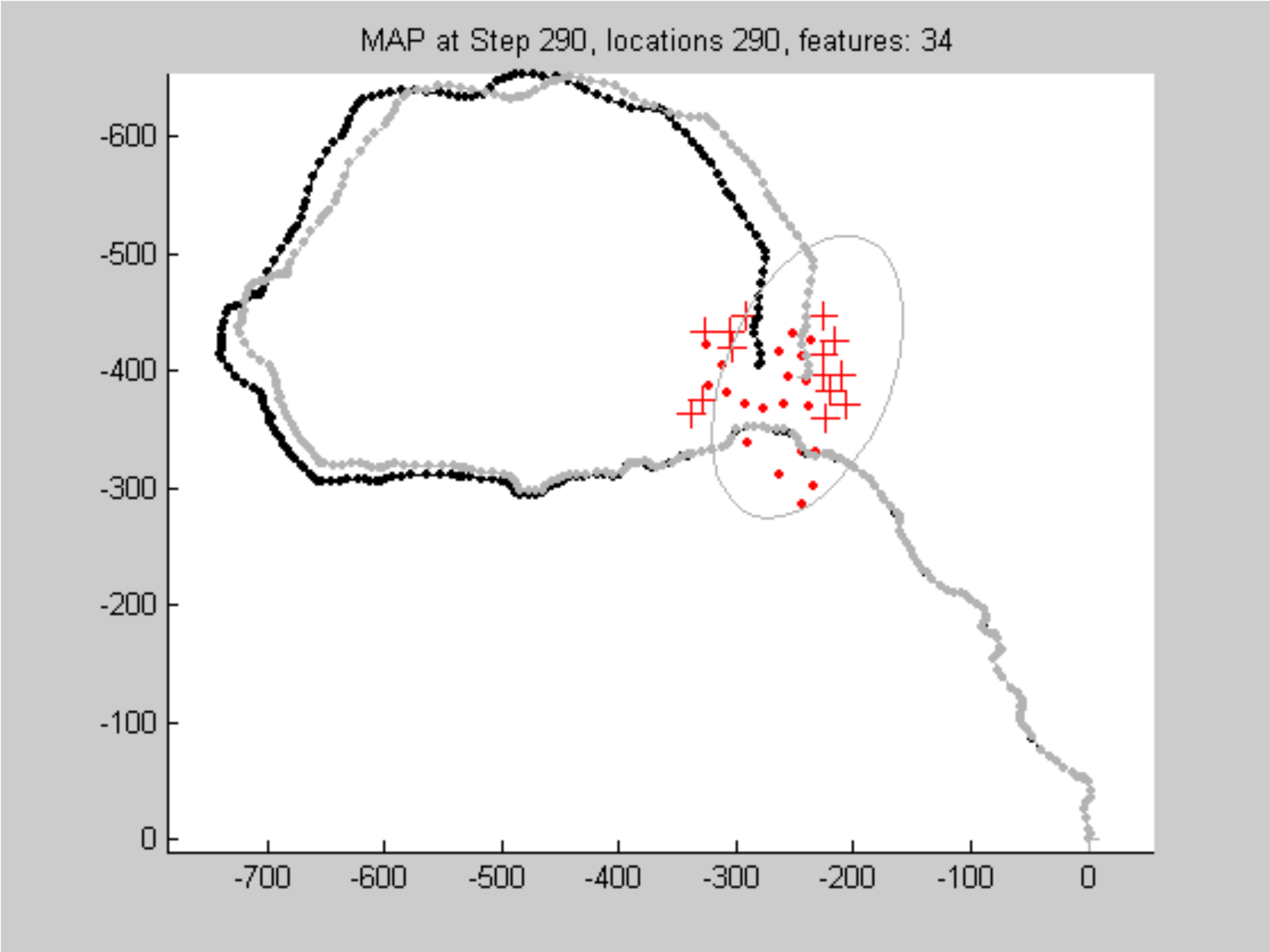
- Individual compatibility



- Joint Compatibility



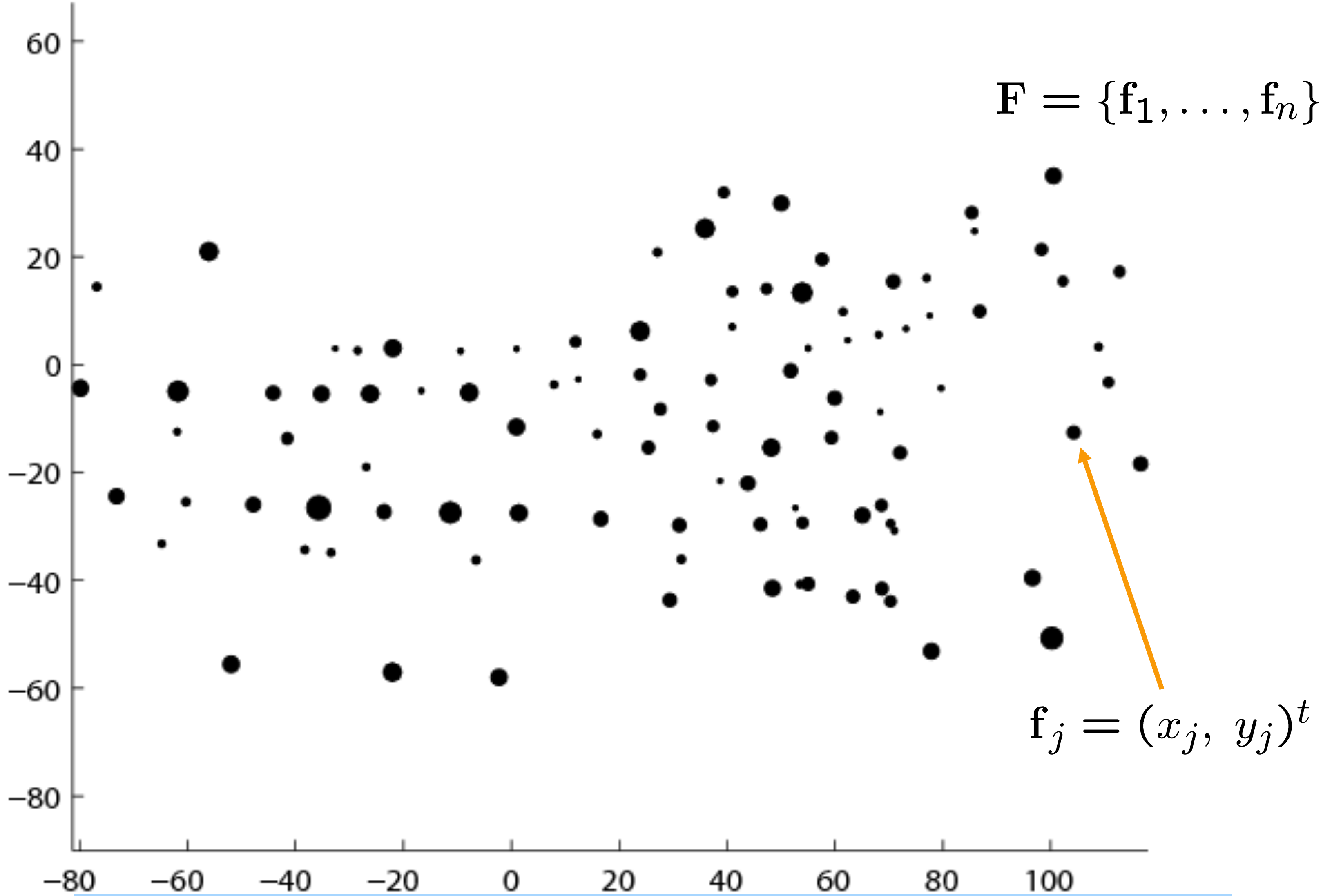
The loop closing problem



4. The Global Localization problem

Global Localization

- Robot placed in a previously mapped environment

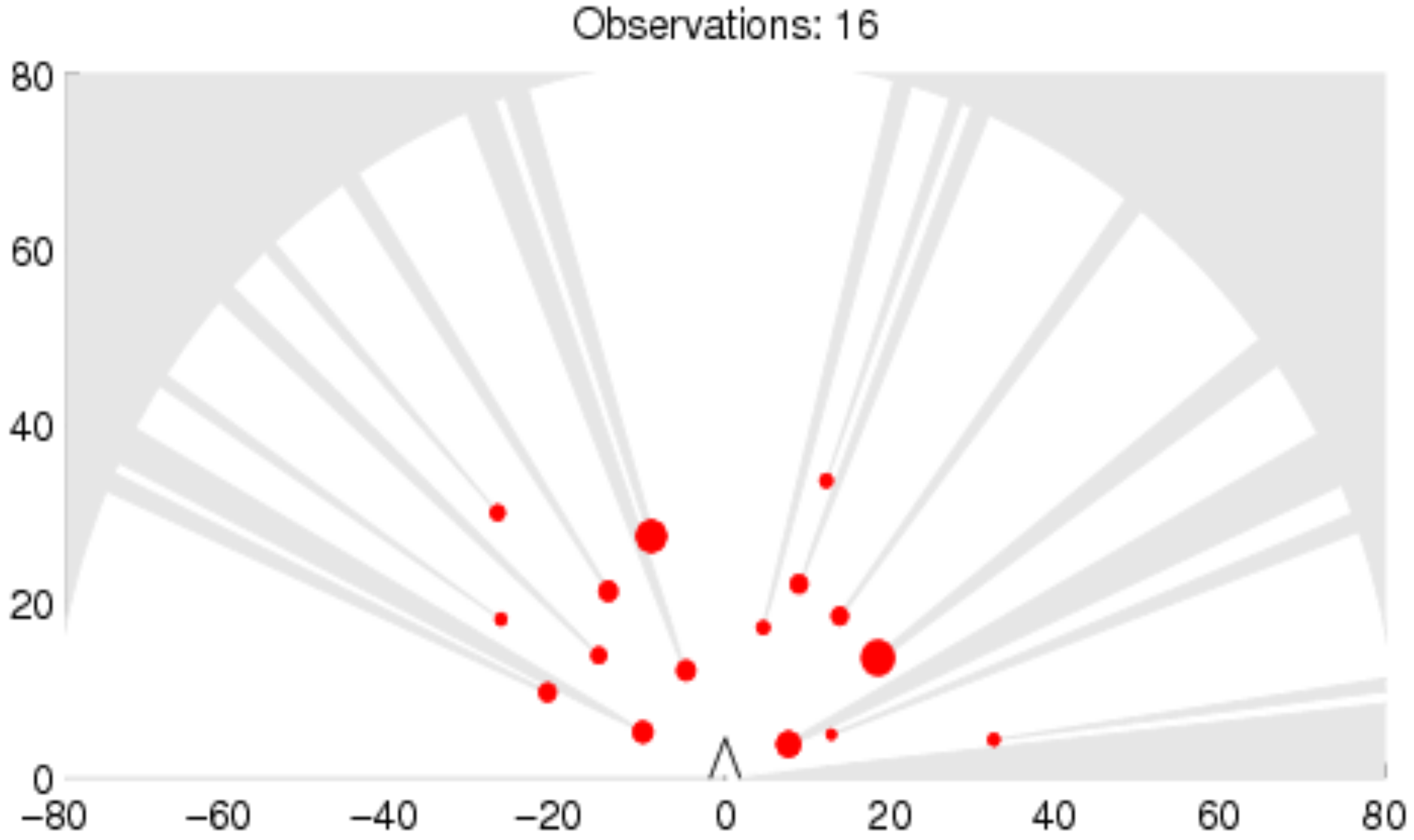


http://www.acfr.usyd.edu.au/homepages/academic/enebot/victoria_park.htm

Problem Definition

- On-board sensor obtains *m* measurements:

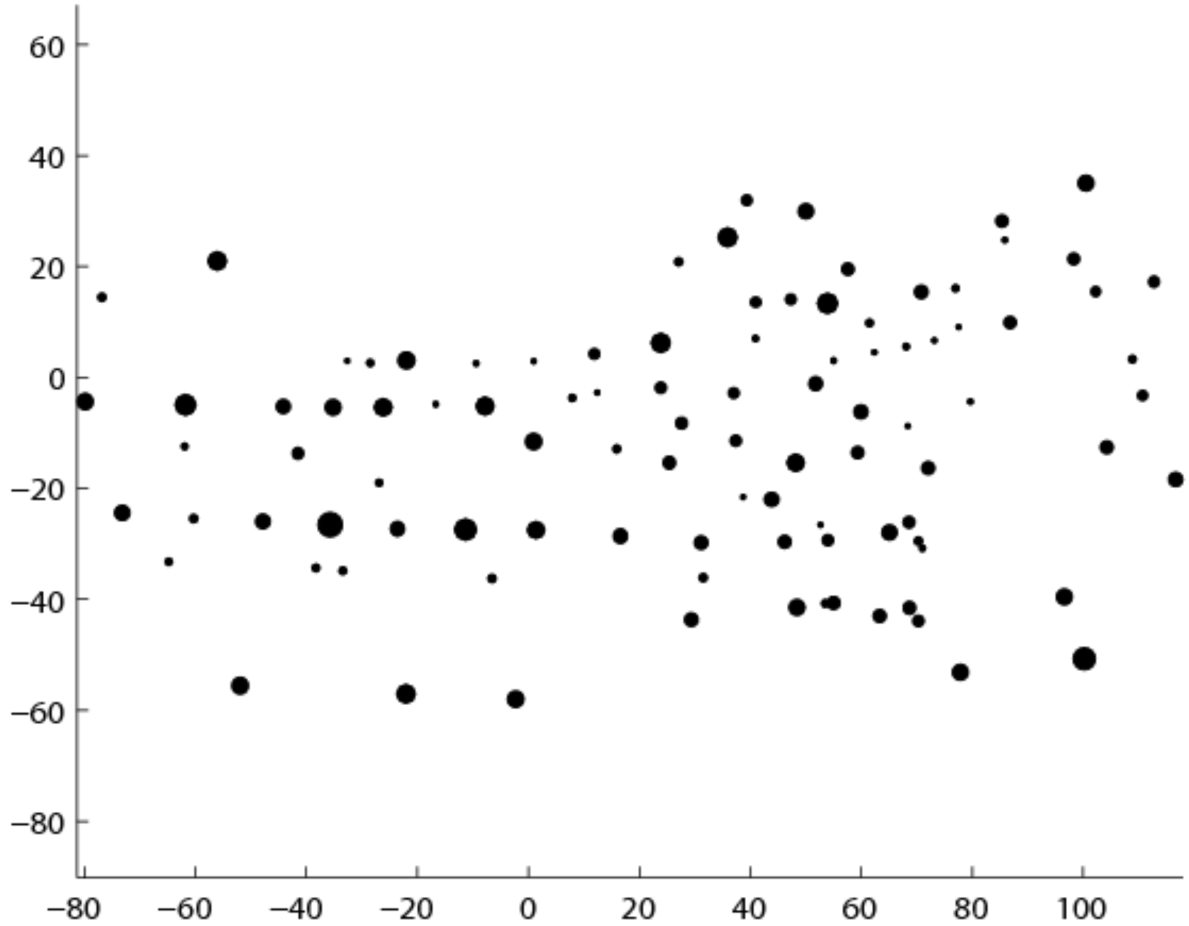
$$\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$$

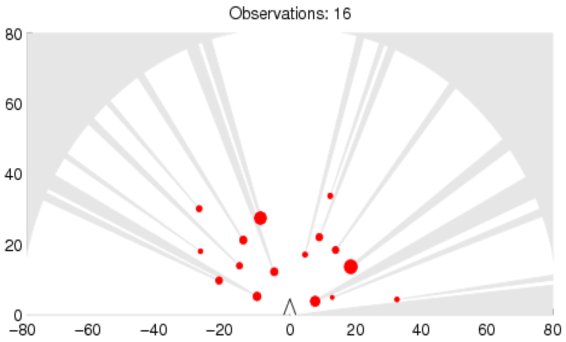


http://www.acfr.usyd.edu.au/homepages/academic/enebot/victoria_park.htm

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Problem Definition





- Twofold question:
  - Is the vehicle in the map?
  - If so, where?

http://www.acfr.usyd.edu.au/homepages/academic/enebot/victoria_park.htm

Global Localization algorithms

- Correspondence space
  - Consider consistent combinations of measurement-feature pairings.
    - » Branch and Bound (Grimson, 1990)
    - » Maximum Clique (Bailey et. al. 2000)
    - » Random Sampling (Neira et. al. 2003)

- Configuration space
  - Consider different vehicle location hypotheses.
    - » Monte Carlo Localization (Fox et. al. 1999)
    - » Markov Localization (Fox et. al. 1998)

In correspondence space

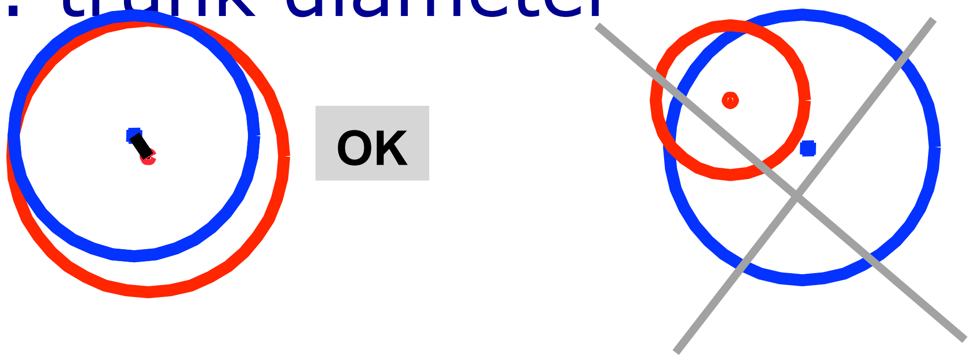
No vehicle location

- **Unary constraints:**

$$p_{ij} = (E_i, F_j)?$$

depend on a single matching (size, color,...)

–Trees: trunk diameter



–Walls: length, corners: angle....

61 714 354 176 000 valid hypotheses

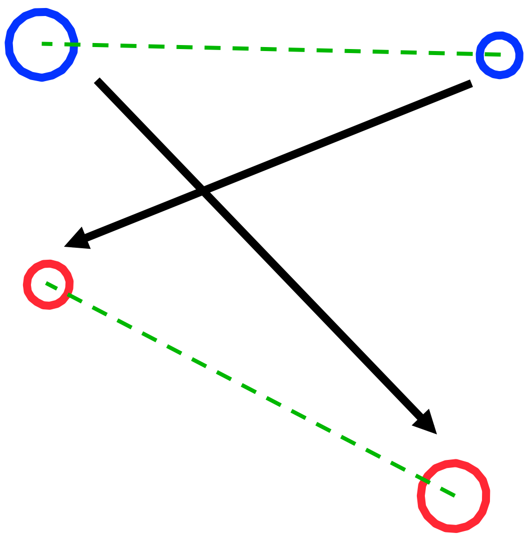
m constraints

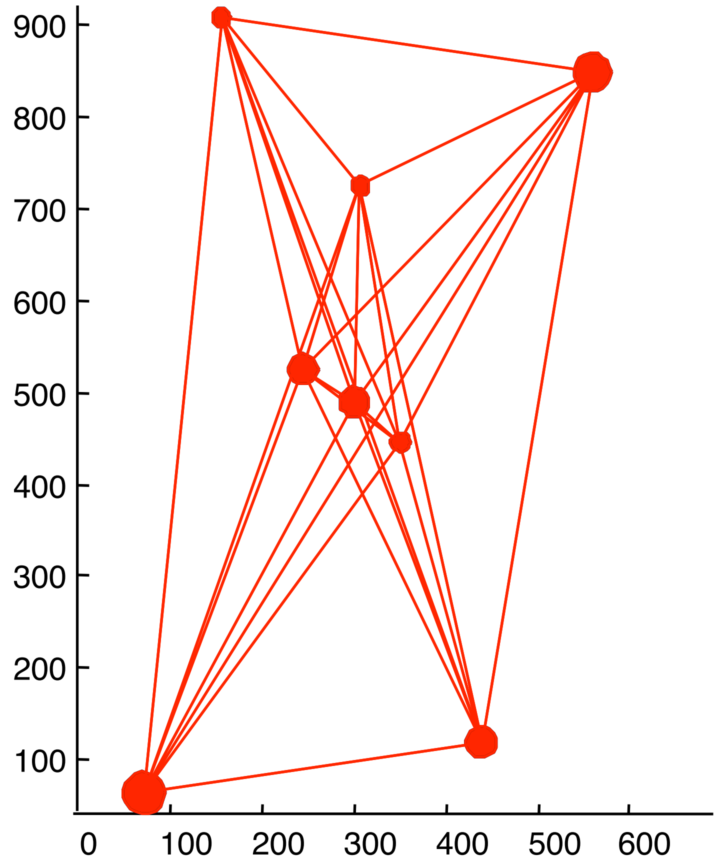
No vehicle location

- **Binary constraints:**

$$p_{ij} = (E_i, F_j)?$$
$$p_{kl} = (E_k, F_l)?$$

distances between points:

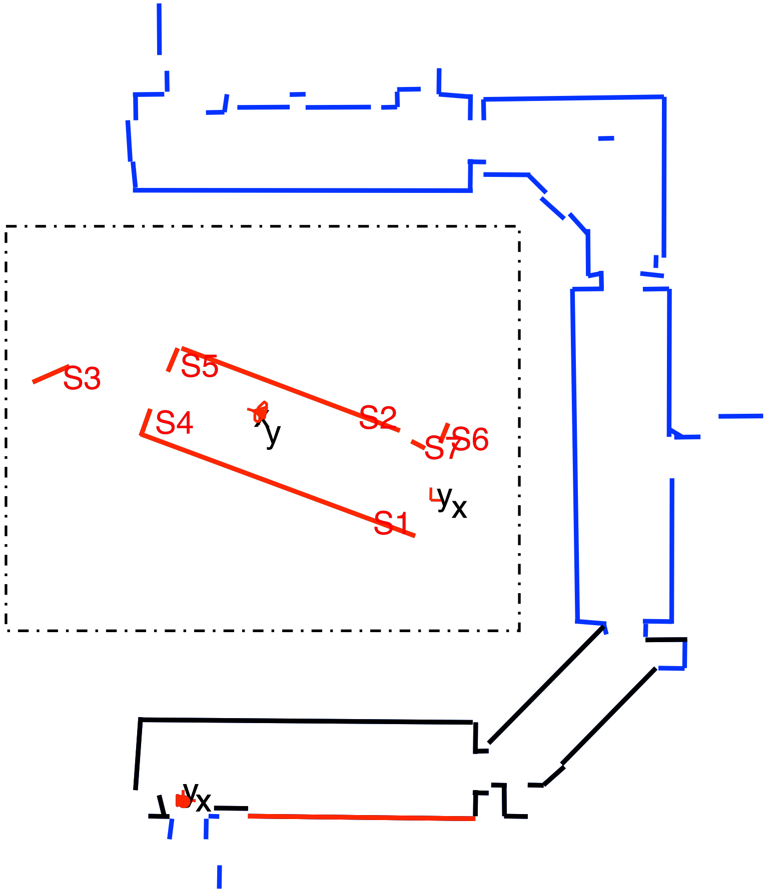


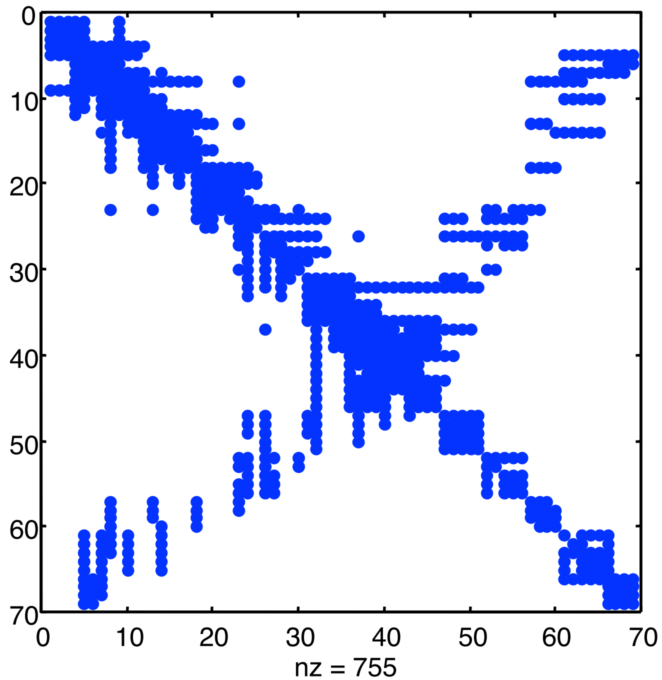


$$\frac{m(m-1)}{2} \text{constraints}$$

Locality

- Features F_i and F_j may belong to the same hypotesis iff they are 'close enough'.





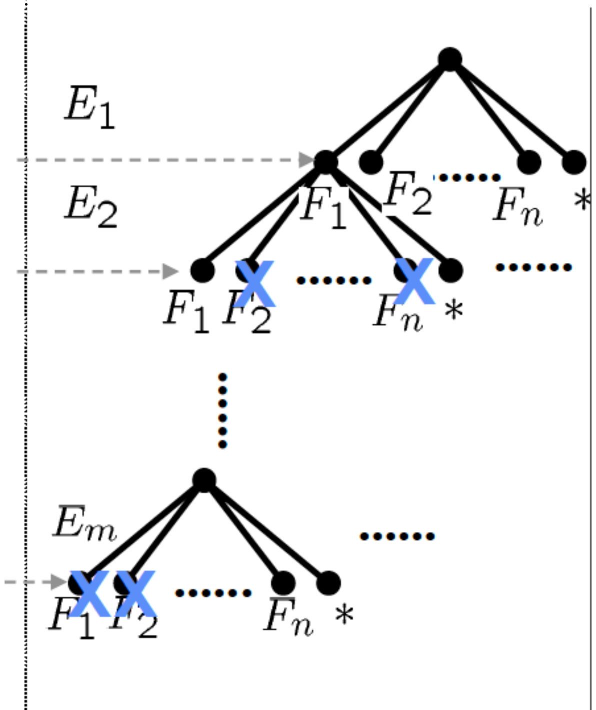
Covisibility matrix

Limit search in the map to subsets of **covisible** features
Locality makes search **linear** with the global map size

Algorithm 1: Geometric Constraints Branch and Bound (Grimson, 1990)

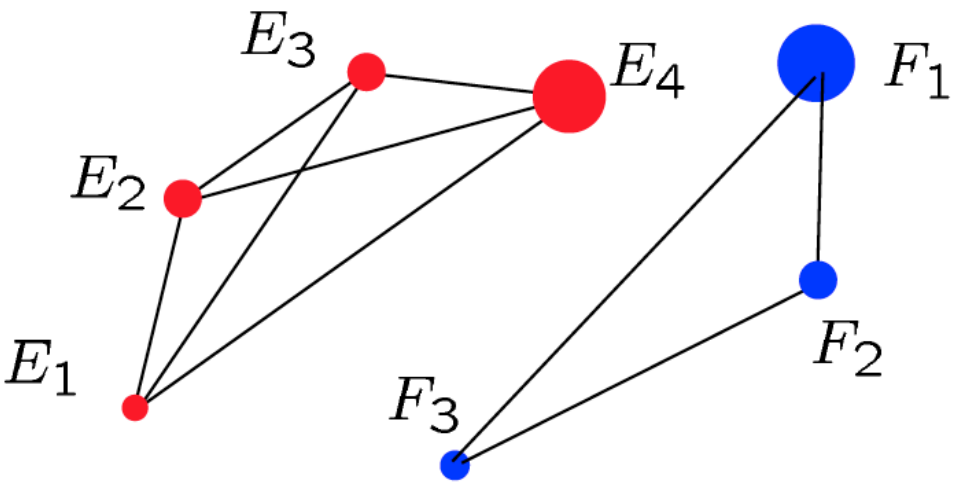
procedure GCBB (H, i):

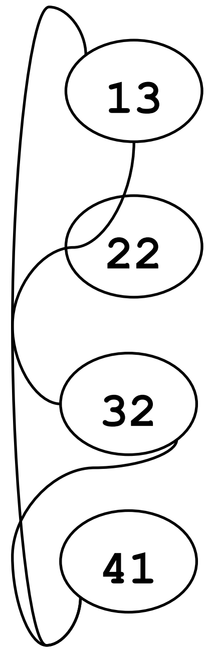
```
if i > m -- leaf node?
   if pairings(H) > pairings(Best) -- did better?
      estimate_location(H)
      if joint_compatibility(H)
         Best = H
      fi
   fi
else
   for j in {1...n}
      if unary(i, j) ∧ binary(i, j, H)
         GCBB([H j], i + 1) -- (E_i, F_j) accepted
      fi
   rof
   if pairings(H) + m - i > pairings(Best)
      GCBB([H 0], i + 1) -- try star node
   fi
fi
```



Algorithm 2: Maximum Clique

- All unary and binary constraints can be precomputed
- Build a compatibility graph where:
  - Nodes represent unary compatible pairings
  - Arcs represent pairs of binary compatible pairings

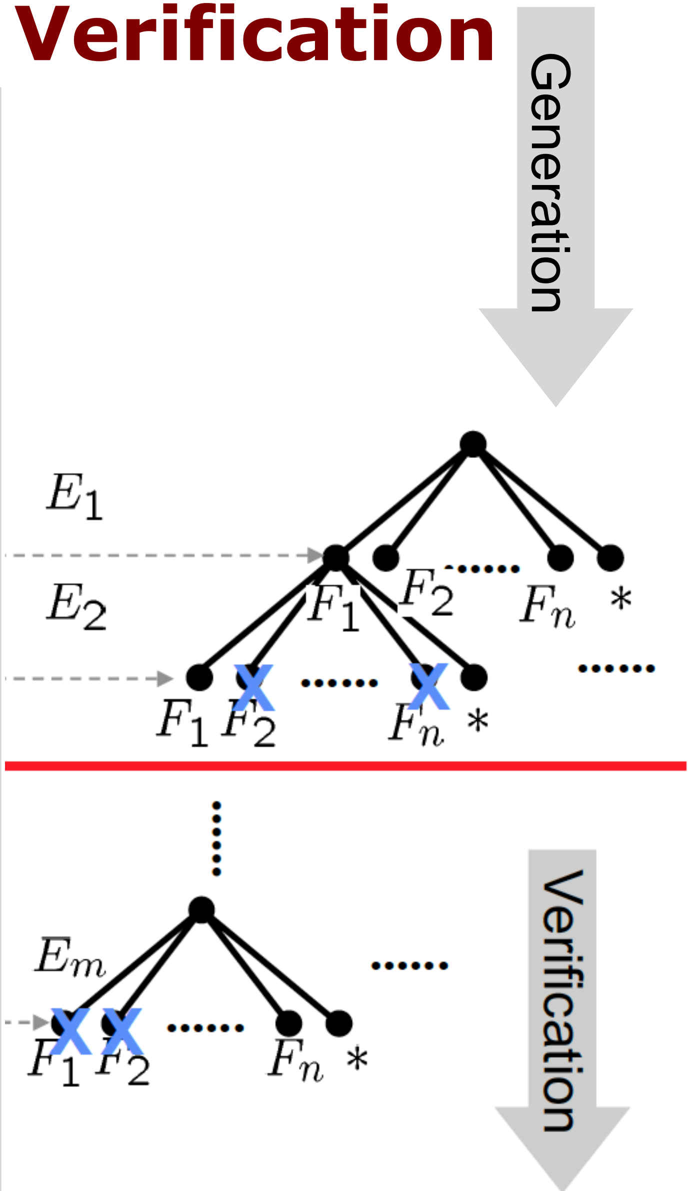




- Carrahan, Pardalos (1990)

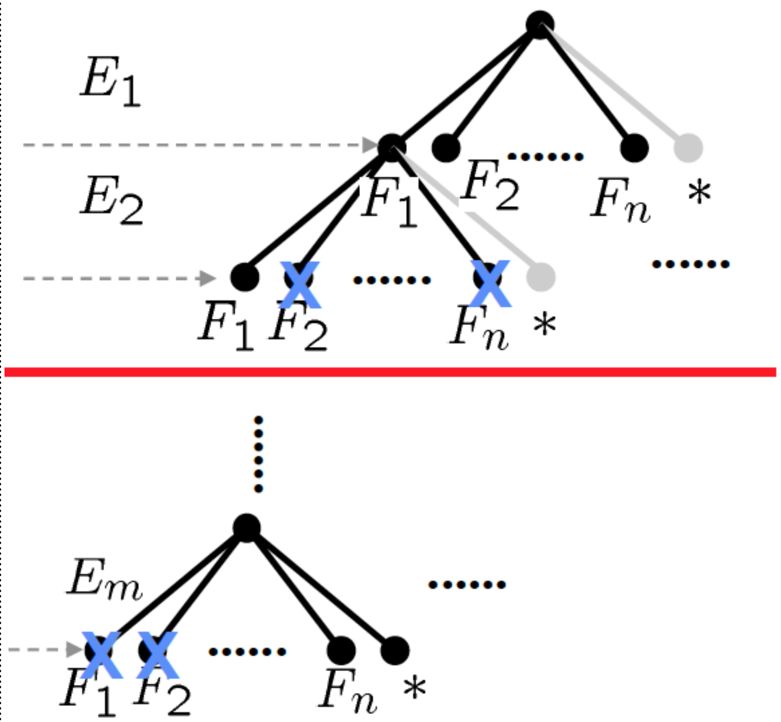
Algorithm 3: Generation Verification

```
procedure GV (H, i):
if i > m
   if pairings(H) > pairings(Best)
      Best = H
   fi
elseif pairings(H) == 3
   estimate_location_(H)
   if joint_compatibility(H)
      JCBB(H, i) -- hypohtesis verification
   fi
else
   for j in {1...n}
      if unary(i, j) ∧ binary(i, j, H)
         GV([H j], i + 1)
      fi
   rof
   if pairings(H) + m - i > pairings(Best)
      GV([H 0], i + 1)
   fi
fi
```



Algorithm 4: RANSAC

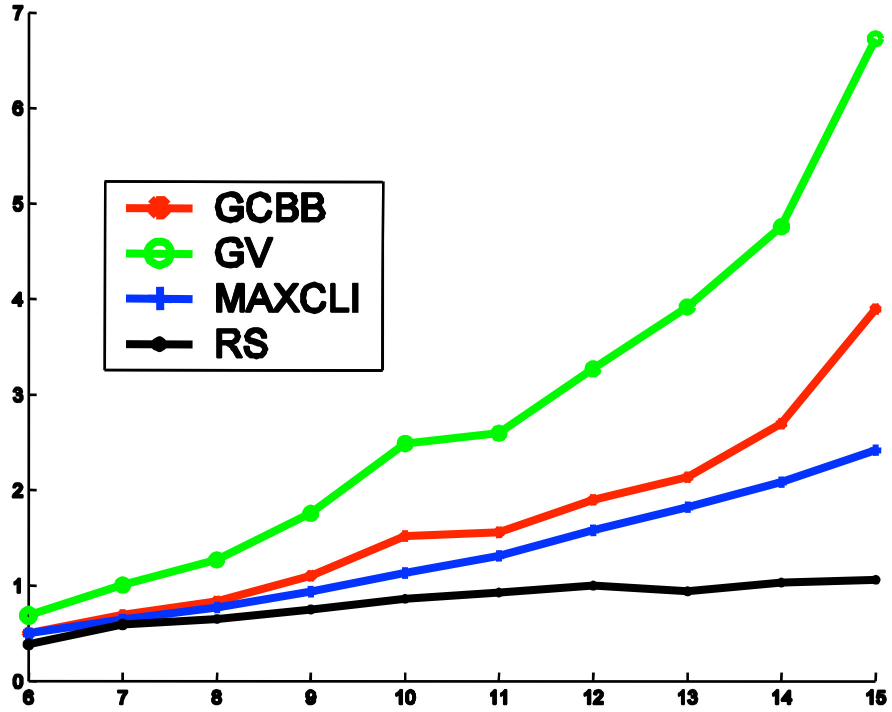
```
procedure RS (H, i):
if i > m
   if pairings(H) > pairings(Best)
      Best = H
   fi
elseif pairings(H) == 3
   estimate_location_(H)
   if joint_compatibility(H)
      JCBB(H, i) -- hypohtesis verification
   fi
else -- branch and bound without star node
   for j in {1...n}
      if unary(i, j) ∧ binary(i, j, H)
         RS([H j], i + 1)
      fi
   rof
fi
```



Experiments

1. No significant difference in **effectiveness** of the considered algorithms
2. All algorithms are made **linear** with the size of the map
3. **Efficiency** when the vehicle is in the map

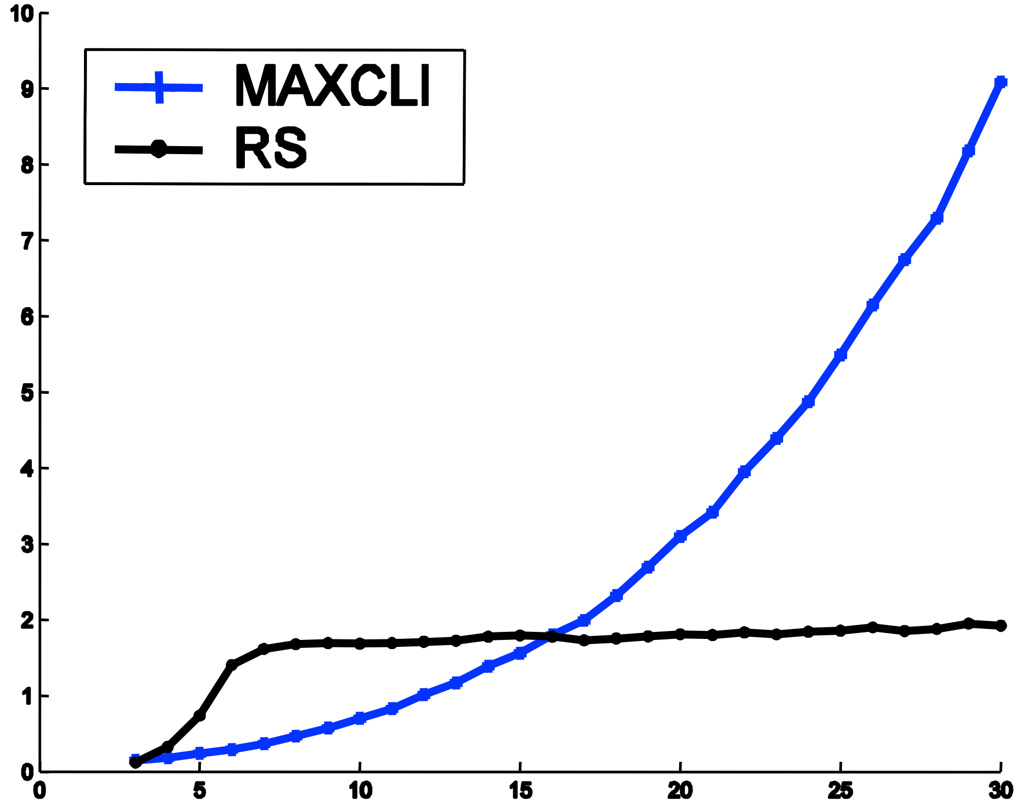
mean time .vs. m



Experiments

4. When the vehicle is **NOT** in the map:
   random measurements, 100 for each m=3..30

mean time .vs. m



J. Neira, J.D. Tardós, J.A. Castellanos, **Linear time vehicle relocation in SLAM.** IEEE Int. Conf. Robotics and Automation, Taipei, Taiwan, May, 2003

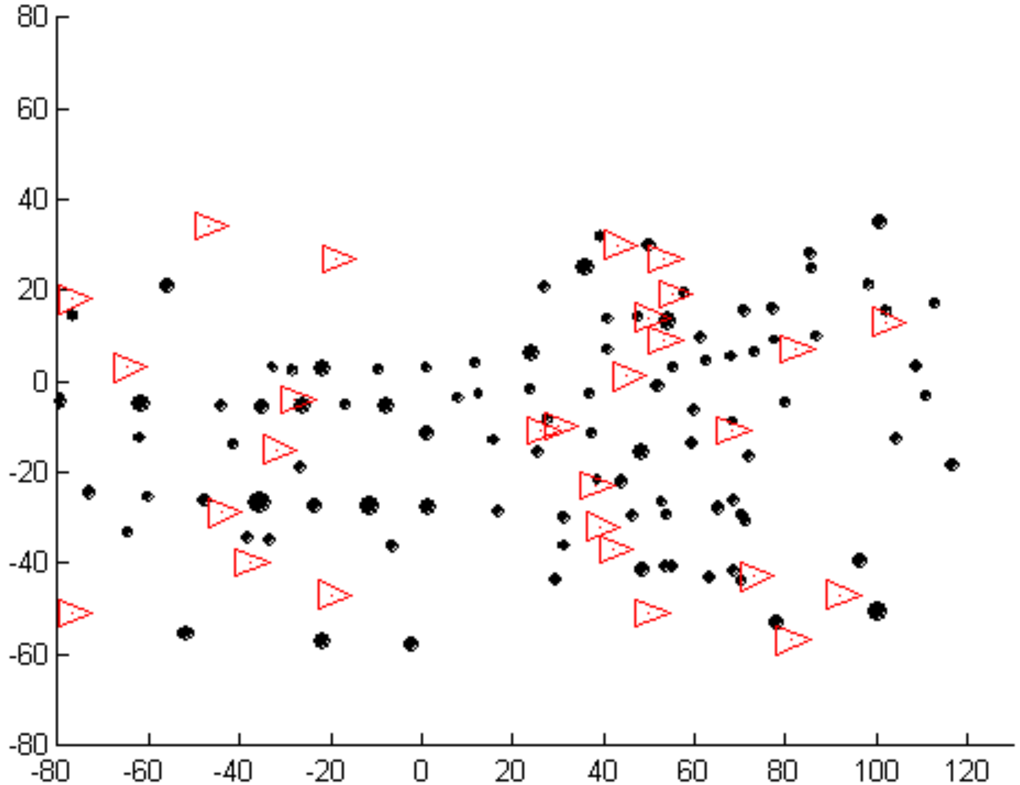
In configuration space

In Configuration Space: RANDOM sampling

- Consider *s* randomly chosen vehicle locations hypotheses:

$$\mathbf{x} = (x, y, \phi)^t$$

$$x \in [x_{min}, x_{max}]$$
$$y \in [y_{min}, y_{max}]$$
$$\phi \in [\phi_{min}, \phi_{max}]$$



- Monte Carlo Localization

Alternative 1: location-driven

- Consider each alternative **location hypothesis** in turn

Algorithm 1 Loc_driven:

$votes = 0$

for each hypothesis $\mathbf{x} \in \mathbf{X}$ **do**

  for each measurement $\mathbf{z}_i \in \mathbf{Z}$ **do**

    $\mathbf{F}_i = \texttt{predict\_features}(\mathbf{x}, \mathbf{z}_i)$

    if $\texttt{any\_compatible\_feature}(\mathbf{F}_i, \mathbf{F})$ **then**

      $votes(\mathbf{x}) = votes(\mathbf{x}) + 1$

    end if

  end for

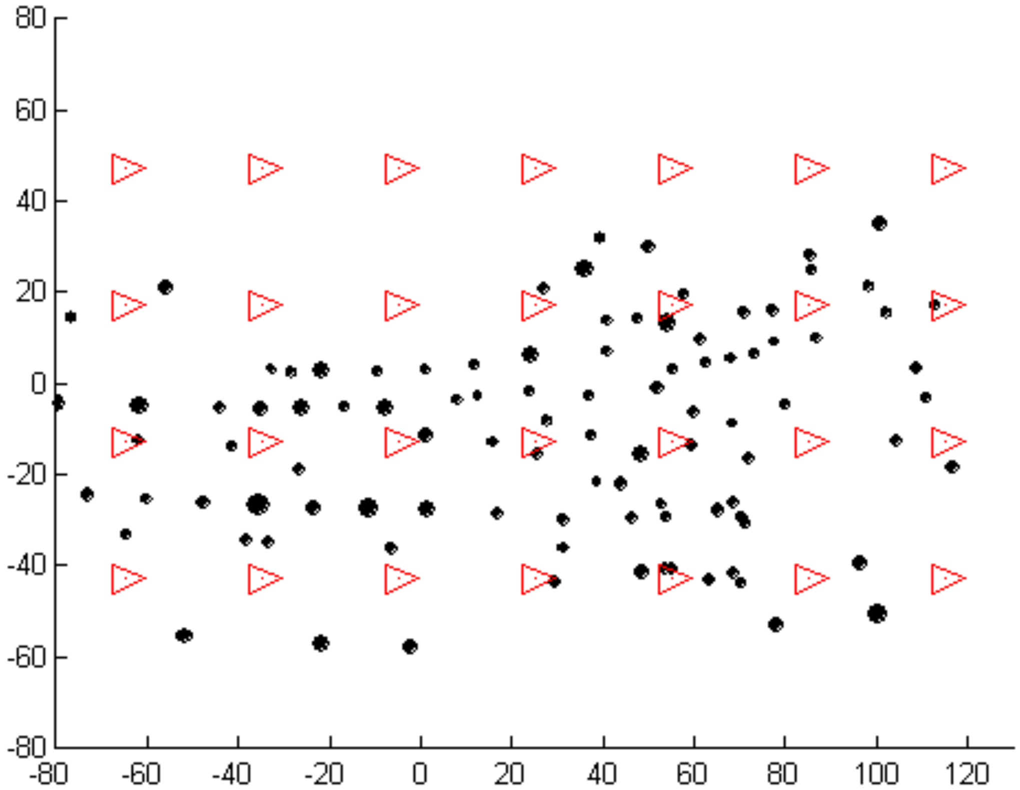
end for

In Configuration Space: GRID sampling

- Uniformly tessellate the space in $s = n_x \, n_y \, n_\phi$ grid cells:

$$\mathbf{x} = (x, y, \phi)^t$$

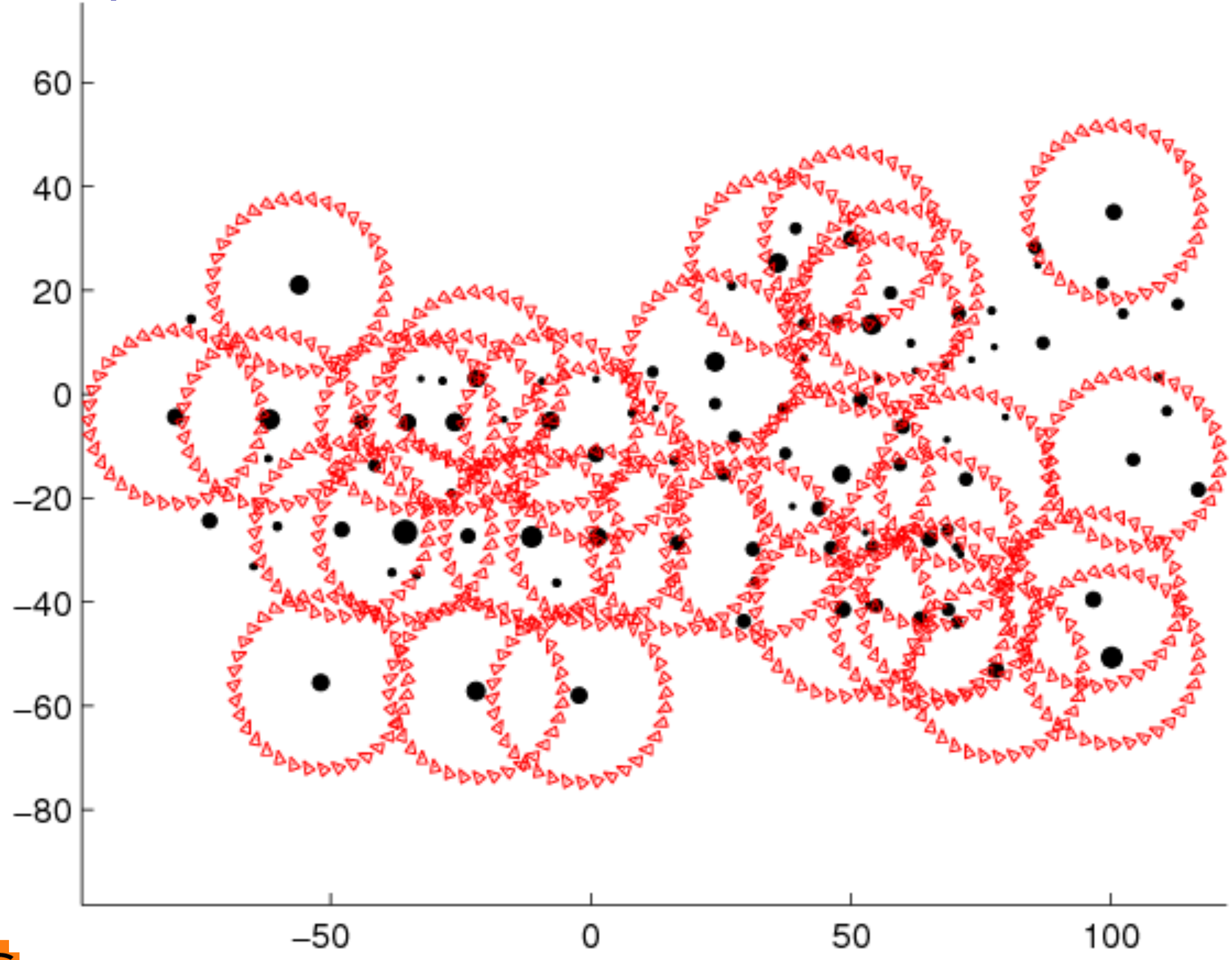
$$\begin{aligned} x &\in [x_{min}, x_{max}] \\ y &\in [y_{min}, y_{max}] \\ \phi &\in [\phi_{min}, \phi_{max}] \end{aligned}$$



- Markov

Voting in configuration space

- Each measurement-feature pairing constrains the set of possible vehicle locations:



Alternative 2: pairing-driven

- Consider each **measurement-feature pairing** in turn

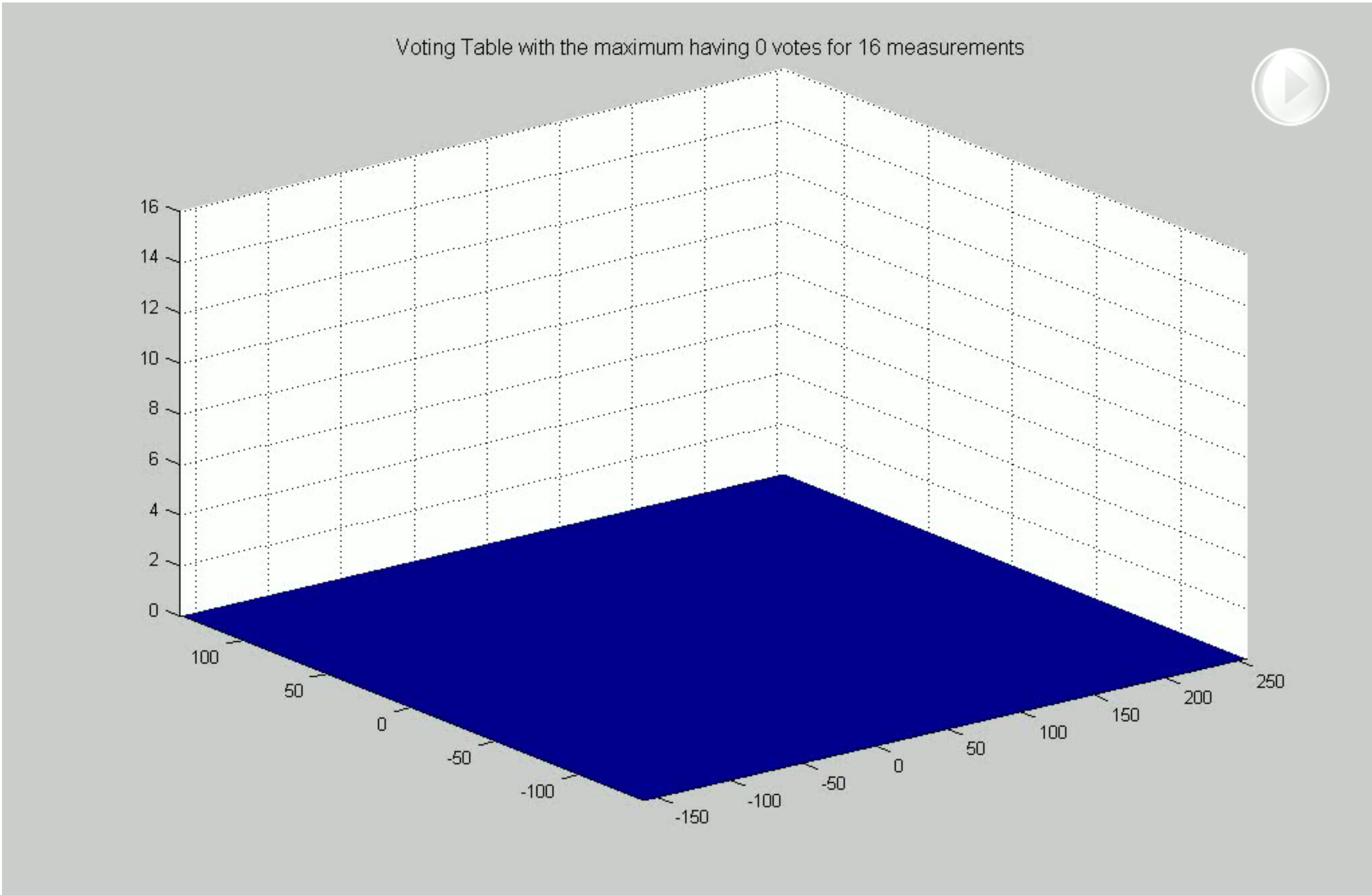
Algorithm 2 `Pair_driven`:

$votes = 0$
for each measurement $\mathbf{z}_i \in \mathbf{Z}$ **do**
  for each feature $\mathbf{f}_j \in \mathbf{F}$ **do**
    $\mathbf{X}_{ij} = \texttt{hypothesize\_locations}(\mathbf{f}_j, \mathbf{z}_i)$
    $\mathbf{X}_v = \texttt{compute\_compatible\_locations}(\mathbf{X}_{ij}, \mathbf{X})$
    $votes(\mathbf{X}_v) = votes(\mathbf{X}_v) + 1$
  end for
end for

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Results

– Resolution: 1.5m for x and y, and 1º for theta



Computational Complexity

Sensor	Loc_driven	Pair_driven
Range and bearing	$O(n_x \cdot n_y \cdot n_\phi \cdot m)$	$O(n \cdot m \cdot n_\phi)$
Range-only	$O(n_x \cdot n_y \cdot n_\phi \cdot m \cdot n_\theta)$	$O(n \cdot m \cdot n_\phi \cdot n_\theta)$
Bearing-only	$O(n_x \cdot n_y \cdot n_\phi \cdot m \cdot n_r)$	$O(n \cdot m \cdot n_\phi \cdot n_r)$

- How do they compare?

$$\frac{\texttt{Pair\_driven}}{\texttt{Loc\_driven}} = \frac{n}{n_x \cdot n_y} = \rho$$

- Pair_driven is better in proportion to the **density of features** in the environment.
- Victoria Park: 18321 sq. m.,sample every 1.5m, expect pair_driven to be **82** faster.

Conclusions

- Data association in SLAM: algorithms based on some form of **consensus** provide the best results
  - » Joint Compatibility
  - » RANSAC
  - » Hough Transform

ROBOTICS
UNIVERSIDAD DE ZARAGOZA