The state of the art in the EKF solution to SLAM

José Neira
Universidad de Zaragoza

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Outline

1. Basic EKF SLAM
   - **Introduction: the need for SLAM**
   - The basic EKF SLAM algorithm
   - Feature Extraction
   - Continuous Data Association
   - The Loop Closing Problem

2. Advanced EKF SLAM
   - Computational complexity of EKF SLAM
   - Consistency of the EKF SLAM
   - SLAM using local maps
     - Sequential Map Joining
     - Divide and Conquer SLAM
     - Hierarchical SLAM

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Simultaneous Localization and Mapping

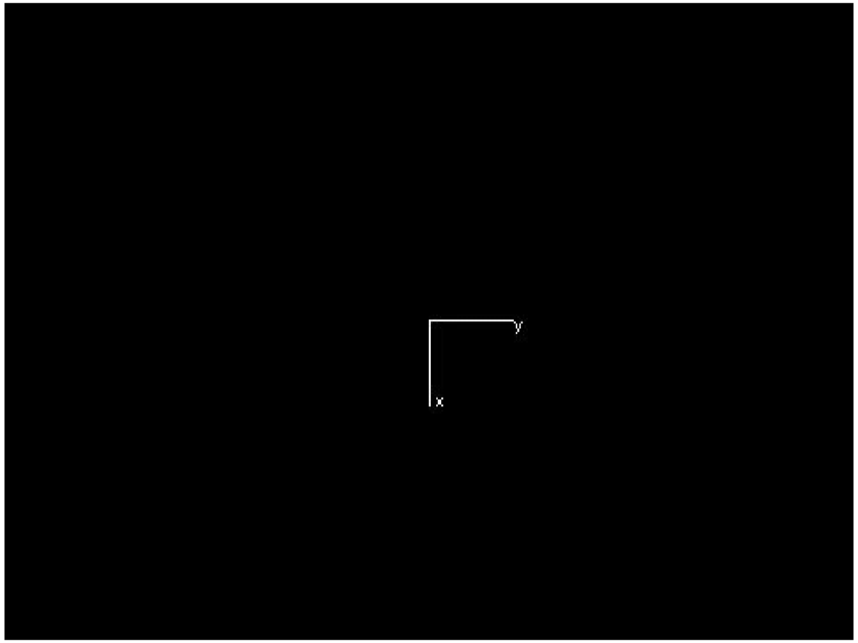
Is it possible to use a vehicle, starting at an

- **unknown initial location**, in an
- **unknown environment**, to
- **incrementally**

build a map of the environment,

- and **at the same time**

use the map to determine the vehicle location?



(image: Paul Newman)

Chicken and egg problem?

Simultaneous Localization and Mapping

The solution to the SLAM problem is, in many respects, a ‘Holy Grail’ of the autonomous vehicle research community, as the ability to build a map and navigate simultaneously would indeed make a robot ‘autonomous’.

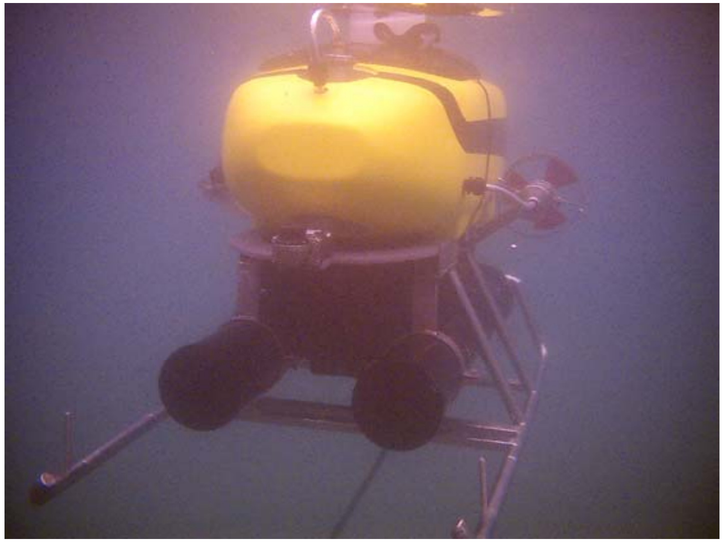
(Newman 1999, Leonard 2000, Thrun 2001)

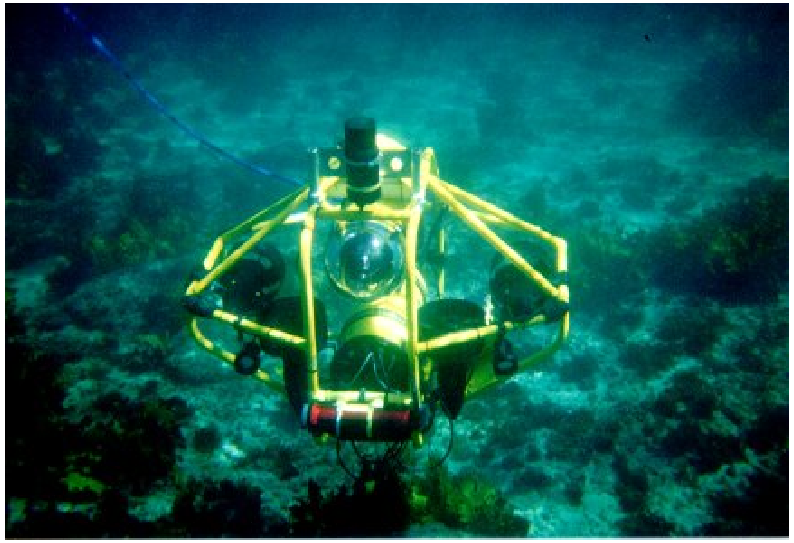
- There is a large amount of potential **applications**
- It gives the vehicle real **autonomy**
- A solution is indeed **possible**

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Mobile Robots

Mobile Robots





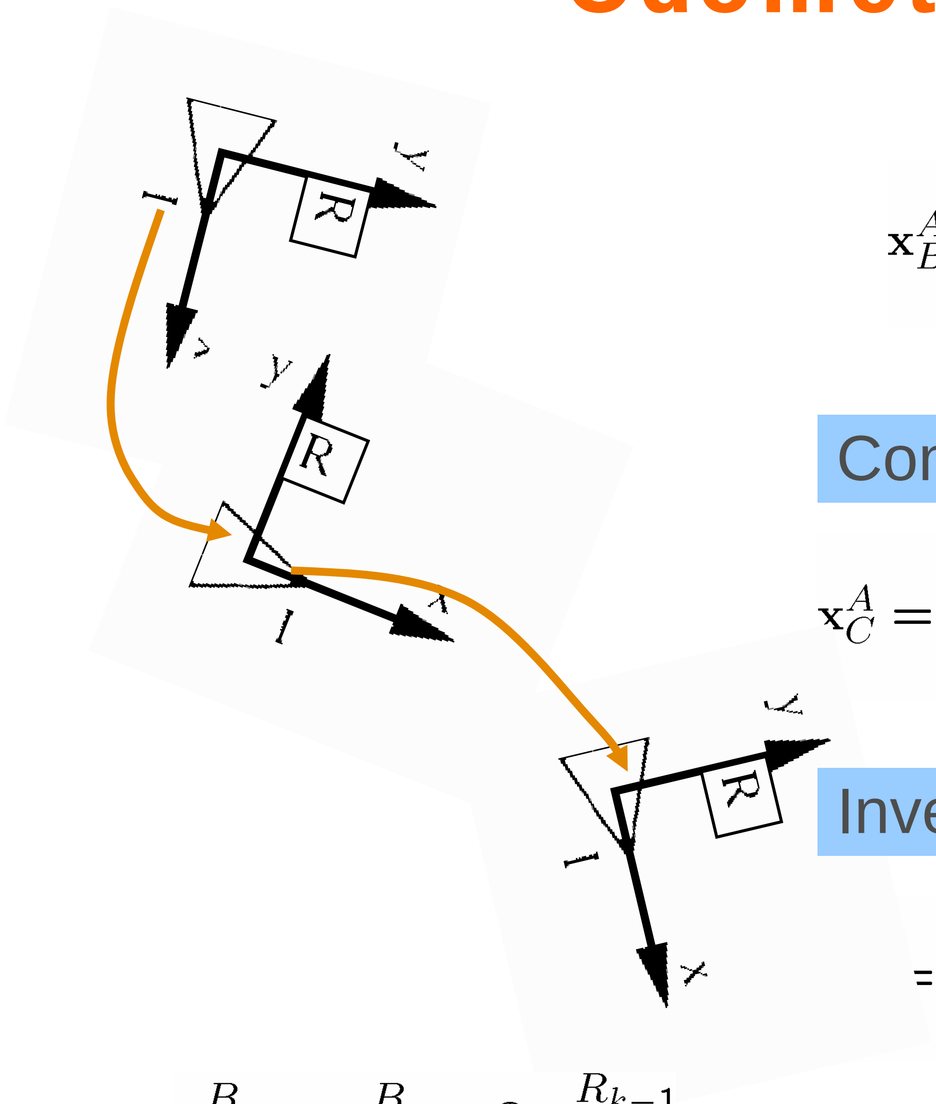
Mobile Robots

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Mobile Sensors

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Odometry 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:

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

Inversion:

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

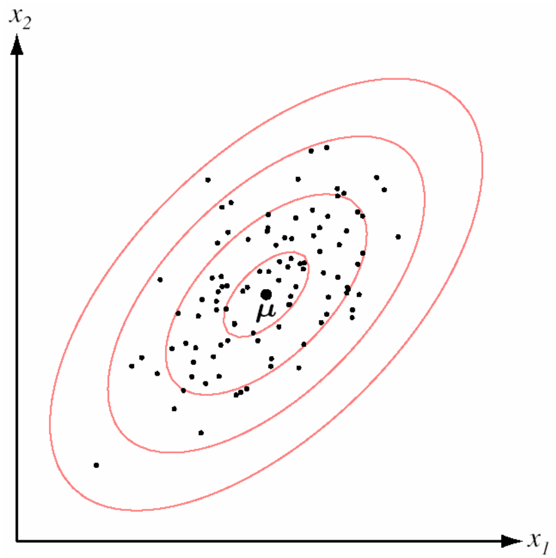
$$\mathbf{x}_{R_k}^B = \mathbf{x}_{R_{k-1}}^B \oplus \mathbf{x}_{R_k}^{R_{k-1}}$$

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Odometry in 2D

Odometry model:

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



Composition:

$$
\begin{aligned}
\hat{\mathbf{x}}_{R_k}^{B} &= \hat{\mathbf{x}}_{R_{k-1}}^{B} \oplus \hat{\mathbf{x}}_{R_k}^{R_{k-1}} \\
\mathbf{P}_{R_k} &\simeq J_1 \, \mathbf{P}_{R_{k-1}} J_1^T + J_2 \, \mathbf{Q}_k J_2^T
\end{aligned}
$$

$$
J_1 = \left. \frac{\partial \left(\mathbf{x}_{R_{k-1}}^{B} \oplus \mathbf{x}_{R_k}^{R_{k-1}} \right)}{\partial \mathbf{x}_{R_{k-1}}^{B}} \right|_{(\hat{\mathbf{x}}_{R_{k-1}}^{B}, \hat{\mathbf{x}}_{R_k}^{R_{k-1}})}
$$

$$
J_2 = \left. \frac{\partial \left(\mathbf{x}_{R_{k-1}}^{B} \oplus \mathbf{x}_{R_k}^{R_{k-1}} \right)}{\partial \mathbf{x}_{R_k}^{R_{k-1}}} \right|_{(\hat{\mathbf{x}}_{R_{k-1}}^{B}, \hat{\mathbf{x}}_{R_k}^{R_{k-1}})}
$$

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Odometry in 2D

Jacobians:

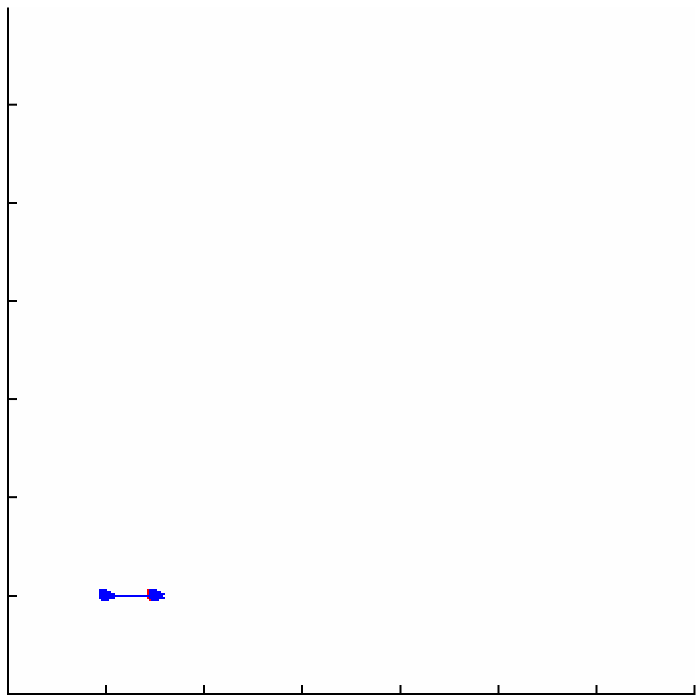
$$J_{1\oplus}\{\mathbf{x}_B^A, \mathbf{x}_C^B\} = \left.\frac{\partial\left(\mathbf{x}_B^A \oplus \mathbf{x}_C^B\right)}{\partial \mathbf{x}_B^A}\right|_{(\hat{\mathbf{x}}_B^A, \hat{\mathbf{x}}_C^B)} = \begin{bmatrix} 1 & 0 & -x_2 \sin\phi_1 - y_2 \cos\phi_1 \\ 0 & 1 & x_2 \cos\phi_1 - y_2 \sin\phi_1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

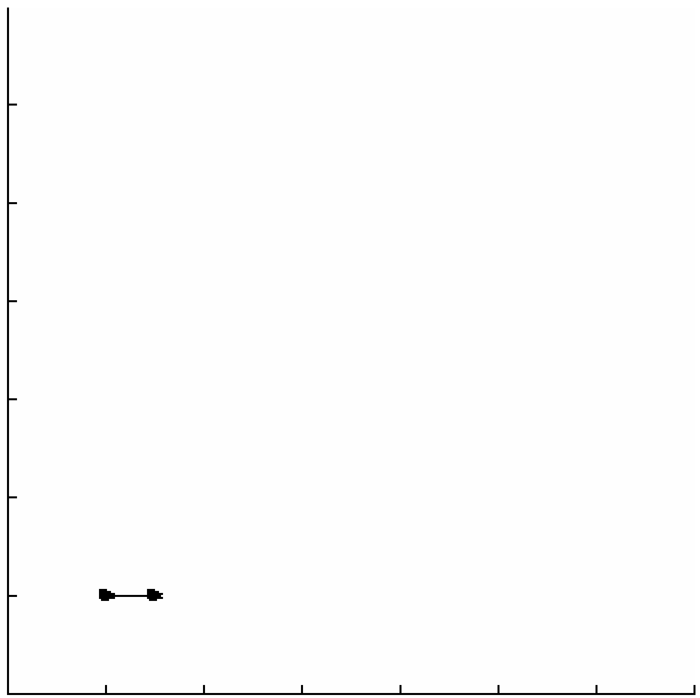
$$J_{2\oplus}\{\mathbf{x}_B^A, \mathbf{x}_C^B\} = \left.\frac{\partial\left(\mathbf{x}_B^A \oplus \mathbf{x}_C^B\right)}{\partial \mathbf{x}_C^B}\right|_{(\hat{\mathbf{x}}_B^A, \hat{\mathbf{x}}_C^B)} = \begin{bmatrix} \cos\phi_1 & -\sin\phi_1 & 0 \\ \sin\phi_1 & \cos\phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

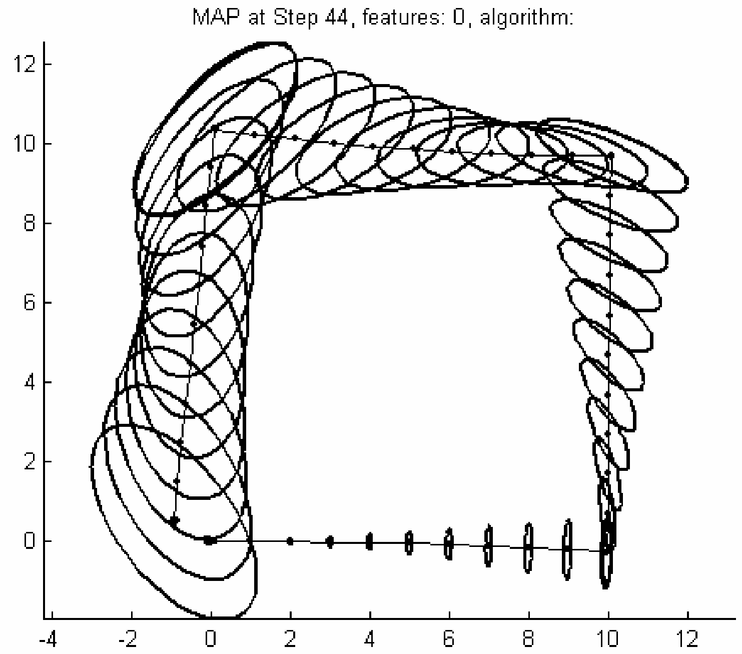
Odometry in 2D

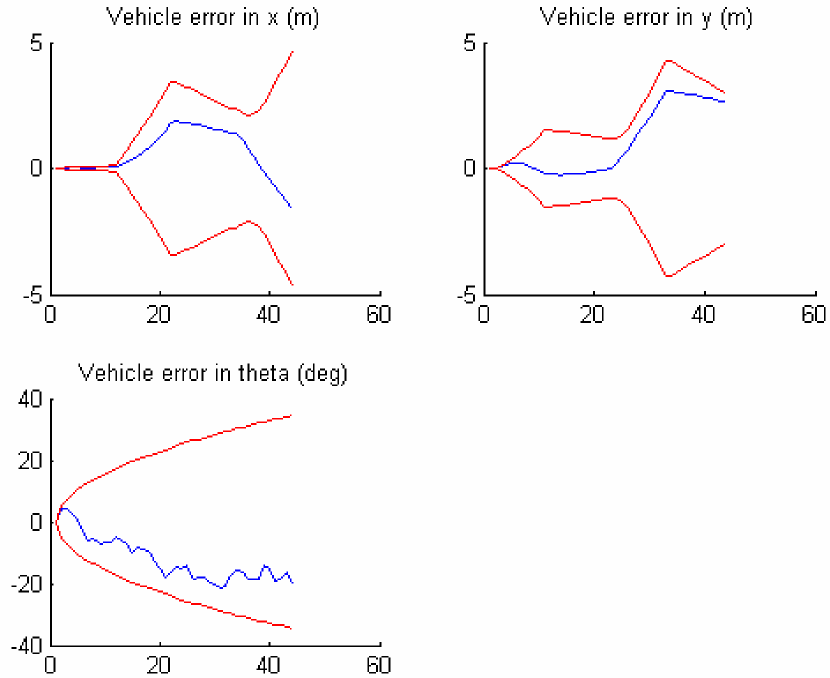




The need for SLAM

Error$\pm 2\sigma$ (prob. 0.95)

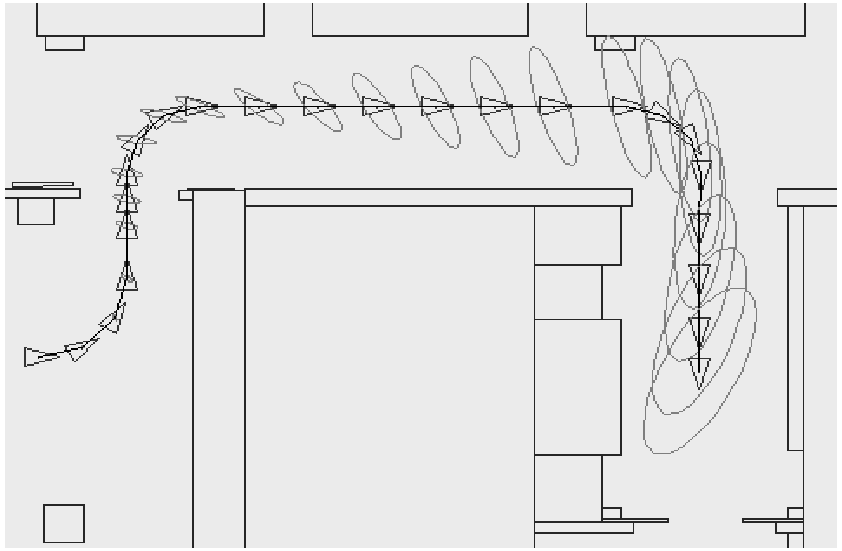


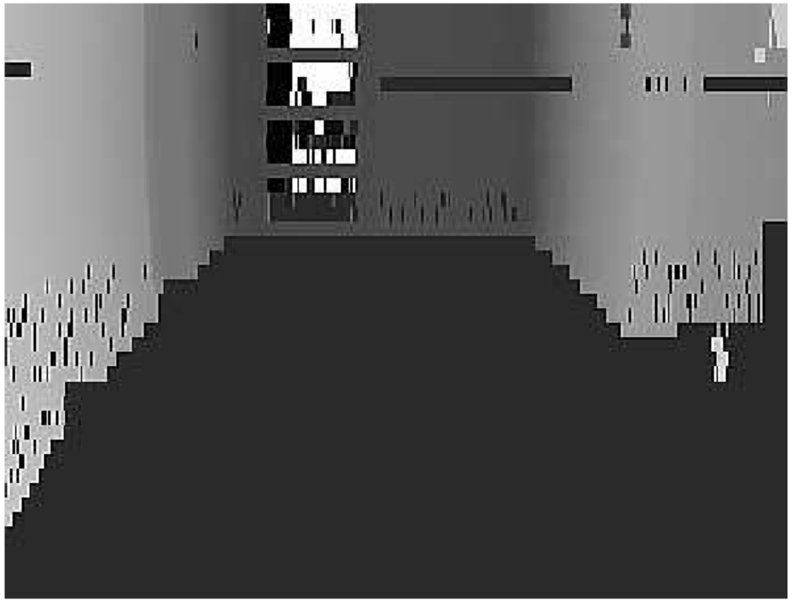


How can we avoid drift?

Map-based Localization

- Cumulative odometry errors
- A priori Map + Perception



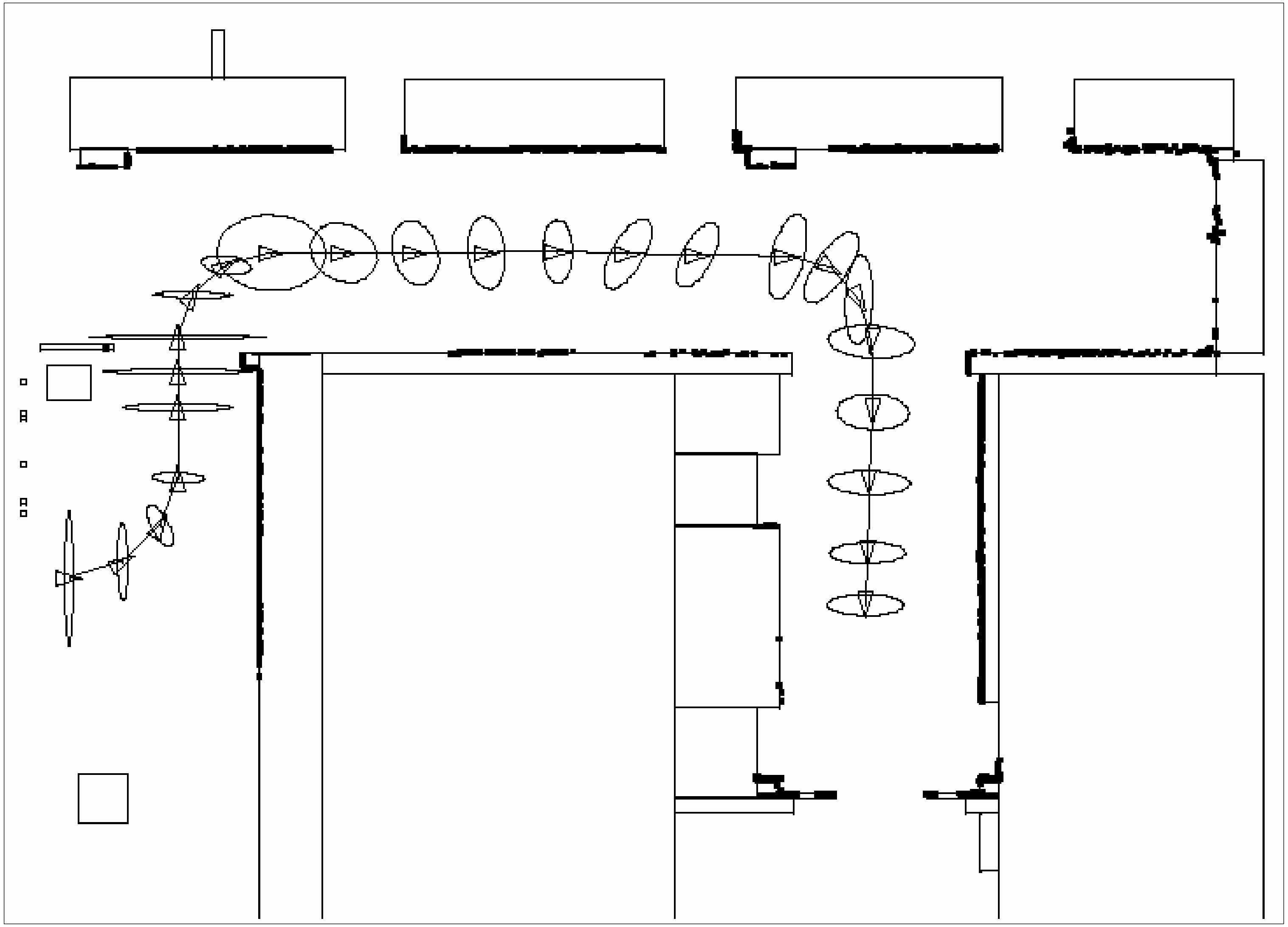


Laser

J. Neira, J.D. Tardós, J. Horn and G. Schmidt: **Fusing Range and Intensity Images for Mobile Robot Localization**, IEEE Trans. Robotics and Automation, Vol. 15, No. 1,Feb 1999, pp 76-84.

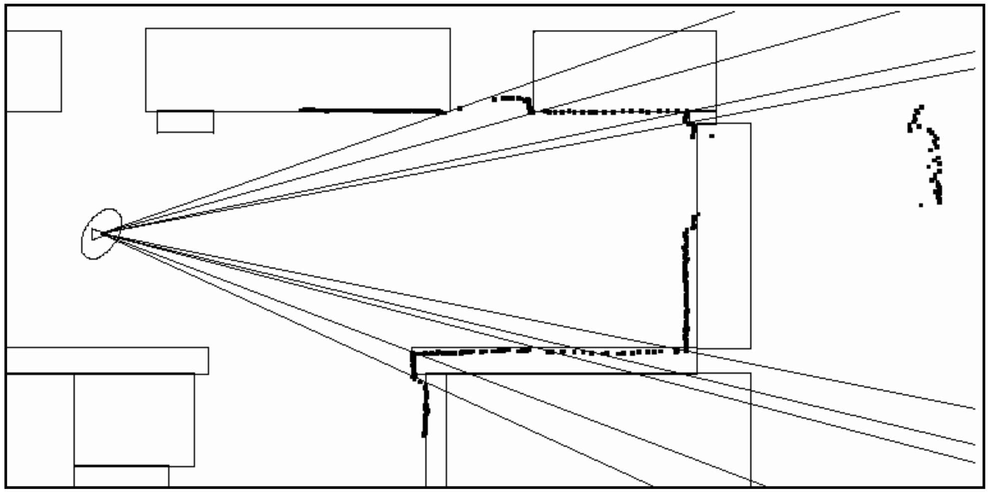
Multisensor Robot Localization

- Results (uncertainty x30)



The need for SLAM

- In many applications the environment is unknown
- A priori maps usually are:
  - Costly to obtain
  - Inaccurate
  - Incomplete
  - Out of date



➔ Map Building

Outline

1. Basic EKF SLAM
   – Introduction: the need for SLAM
   – **The basic EKF SLAM algorithm**
   – Feature Extraction
   – Continuous Data Association
   – The Loop Closing Problem

2. Advanced EKF SLAM
   – Computational complexity of EKF SLAM
   – Consistency of the EKF SLAM
   – SLAM using local maps
      » Sequential Map Joining
      » Divide and Conquer SLAM
      » Hierarchical SLAM

ROBOTICS UNIVERSIDAD DE ZARAGOZA

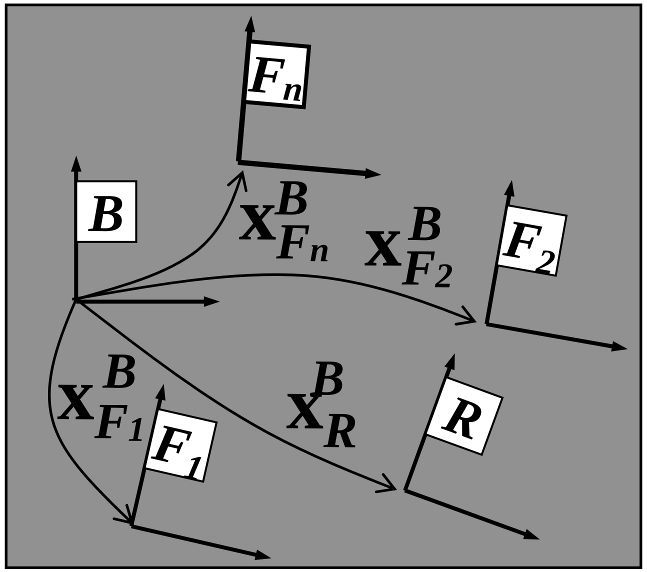
EKF-SLAM: approach

- Environment information related to a set of elements:

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

- represented by a **stochastic map**:

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



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

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

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:

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

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

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

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM

Algorithm 1 SLAM:

$\mathbf{x}_0^B = 0$; $\mathbf{P}_0^B = 0$ {*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$(\mathbf{x}_{k|k-1}^B, \mathbf{P}_{k|k-1}^B, \mathbf{z}_k, \mathbf{R}_k)$

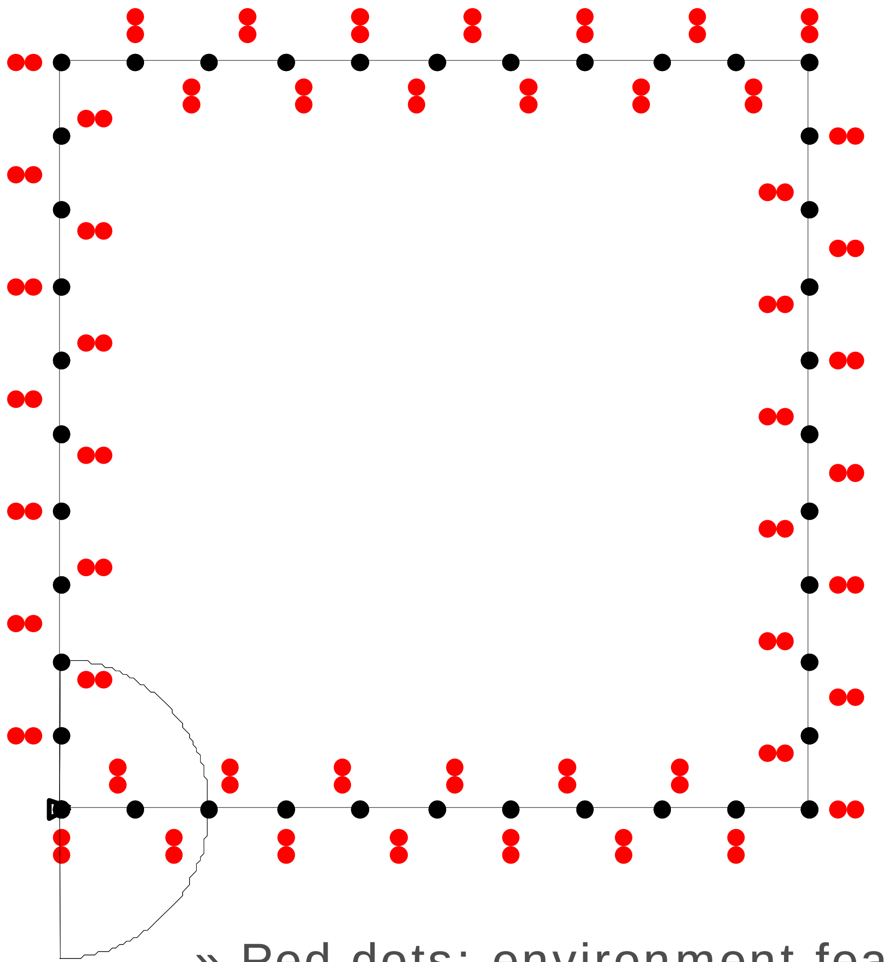
$[\mathbf{x}_k^B, \mathbf{P}_k^B]$ = update_map$(\mathbf{x}_{k|k-1}^B, \mathbf{P}_{k|k-1}^B, \mathbf{z}_k, \mathbf{R}_k, \mathcal{H}_k)$ {*EKF update*}

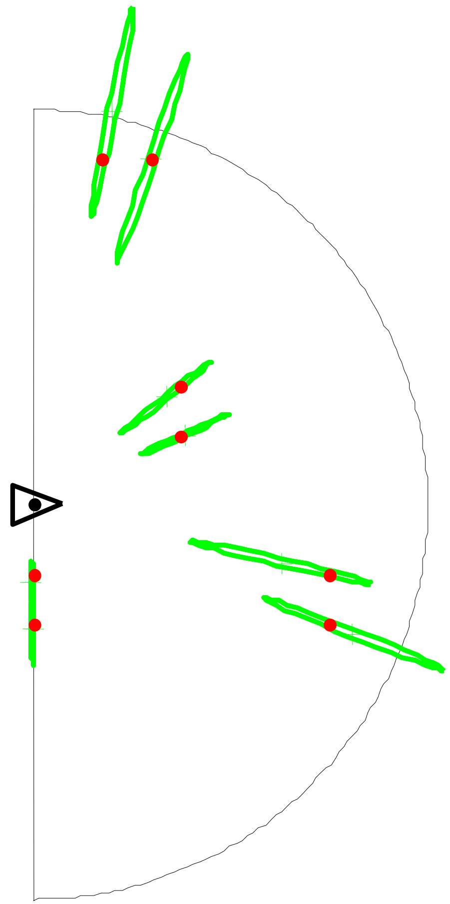
$[\mathbf{x}_k^B, \mathbf{P}_k^B]$ = add_new_features$(\mathbf{x}_k^B, \mathbf{P}_k^B, \mathbf{z}_k, \mathbf{R}_k, \mathcal{H}_k)$

end for

ROBOTICS UNIVERSIDAD DE ZARAGOZA

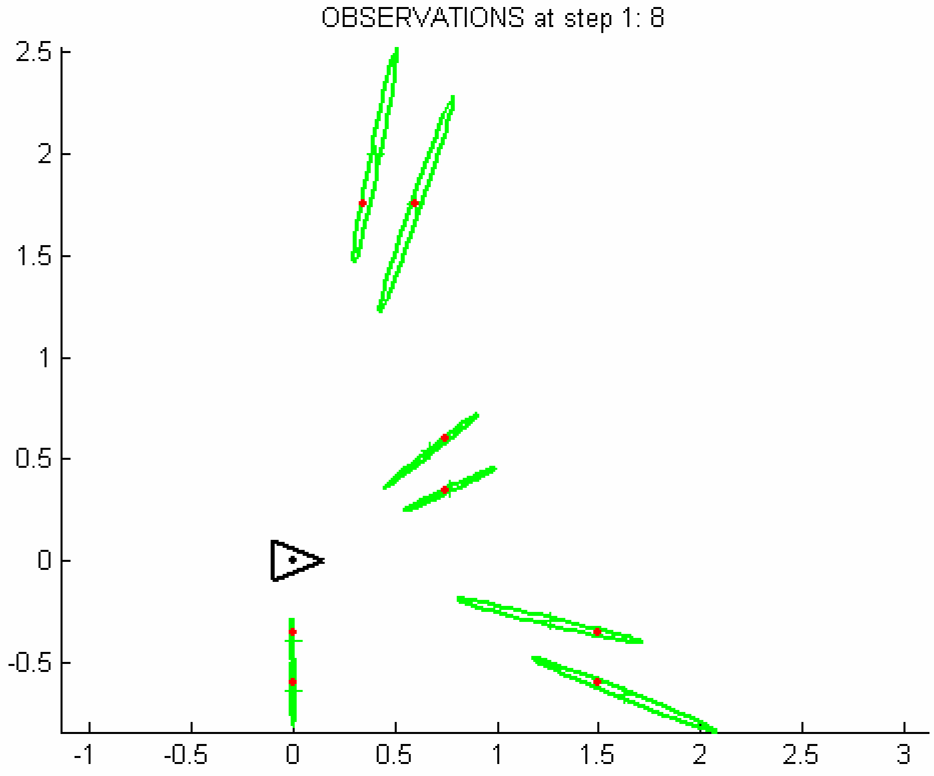
Example: SLAM in a cloister





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

The basic EKF SLAM Algorithm



Sensor measurements

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:

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

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

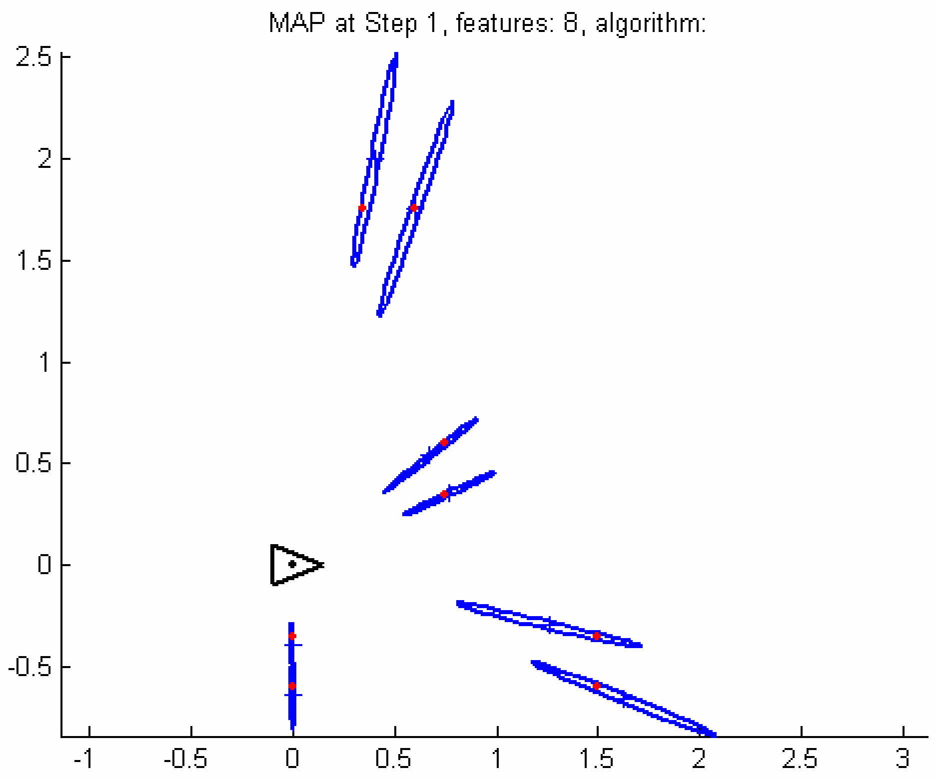
EKF-SLAM: add new features

$$\mathbf{P}_k^B = \begin{pmatrix} \mathbf{P}_R & \mathbf{P}_{RF_1} & \cdots & \mathbf{P}_{RF_n} \\ \mathbf{P}_{RF_1}^T & \mathbf{P}_{F_1} & \cdots & \mathbf{P}_{F_1F_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{RF_n}^T & \mathbf{P}_{F_1F_n}^T & \cdots & \mathbf{P}_{F_n} \end{pmatrix}$$

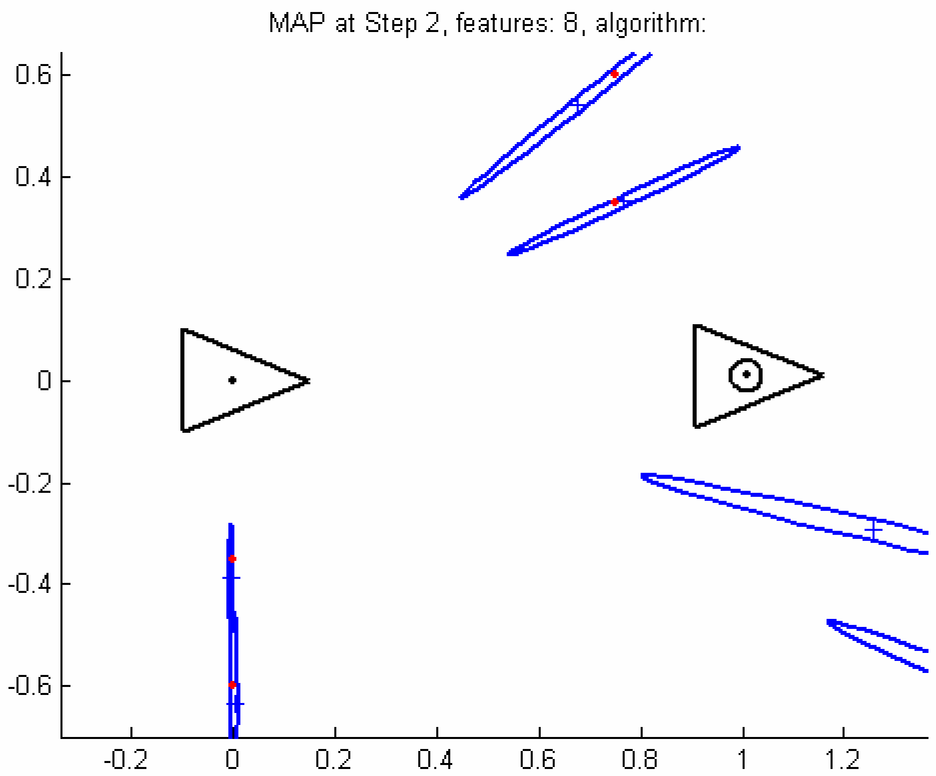
$$\mathbf{P}_{k+}^B = \begin{pmatrix} \mathbf{P}_R & \mathbf{P}_{RF_1} & \cdots & \mathbf{P}_{RF_n} & \mathbf{P}_R\mathbf{J}_{1\oplus}^T \\ \mathbf{P}_{RF_1}^T & \mathbf{P}_{F_1} & \cdots & \mathbf{P}_{F_1F_n} & \mathbf{P}_{RF_1}^T\mathbf{J}_{1\oplus}^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{P}_{RF_n}^T & \mathbf{P}_{F_1F_n}^T & \cdots & \mathbf{P}_{F_n} & \mathbf{P}_{RF_n}^T\mathbf{J}_{1\oplus}^T \\ \mathbf{J}_{1\oplus}\mathbf{P}_R & \mathbf{J}_{1\oplus}\mathbf{P}_{RF_1} & \cdots & \mathbf{J}_{1\oplus}\mathbf{P}_{RF_n} & \mathbf{J}_{1\oplus}\mathbf{P}_R\mathbf{J}_{1\oplus}^T + \mathbf{J}_{2\oplus}\mathbf{R}_k\mathbf{J}_{2\oplus}^T \end{pmatrix}$$

ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: add new features



EKF-SLAM: compute robot motion



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:

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

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

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

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

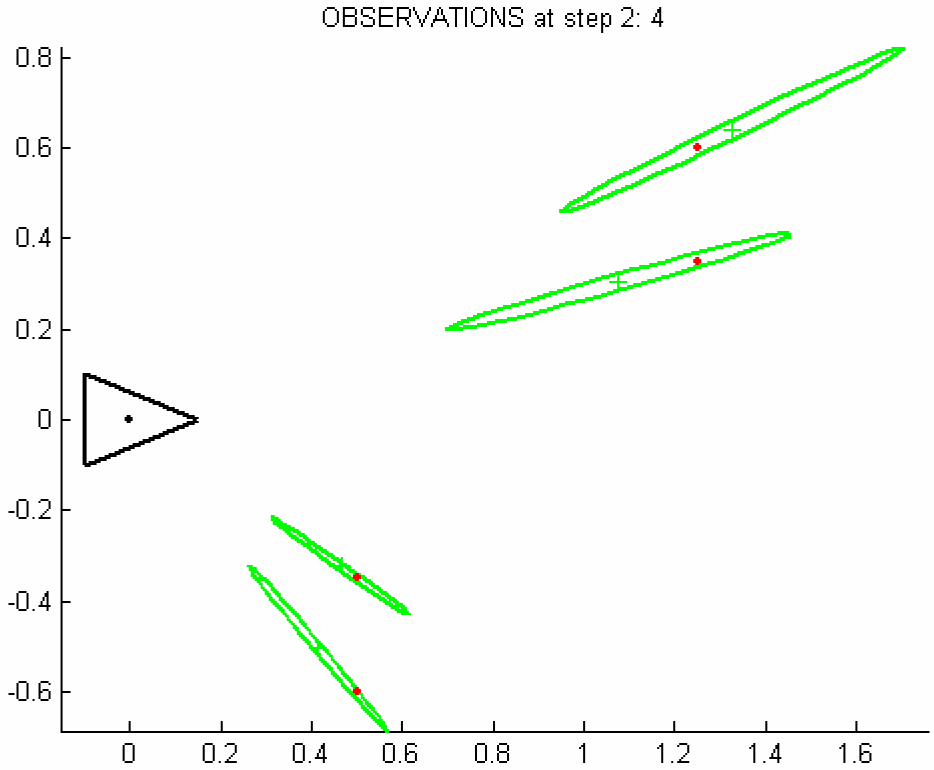
EKF-SLAM: compute robot motion

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

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

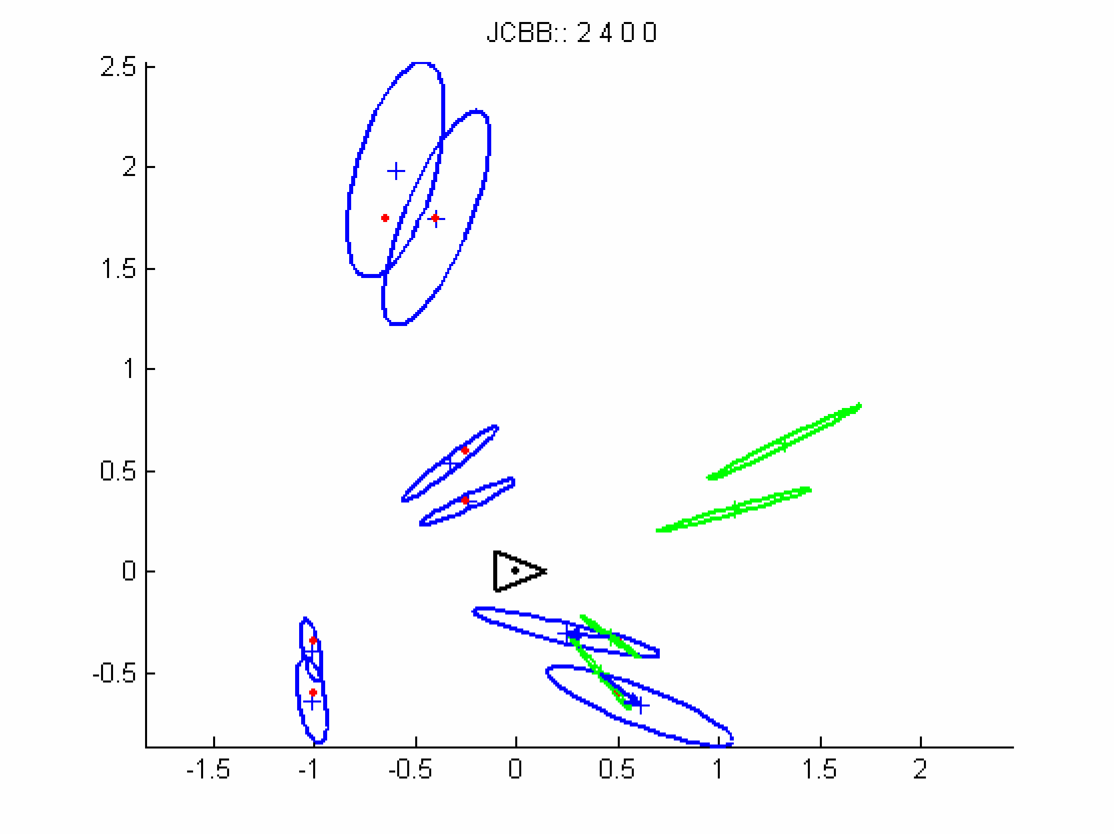
ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: Observations



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: Data association



Predicted map .vs. measurements

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) :

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

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: Observations

Measurement equation:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k^B) + \mathbf{w}_k$$

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

Linearization:

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

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k^B}\right|_{(\hat{\mathbf{x}}_{k|k-1}^B)} = \begin{pmatrix} \mathbf{H}_R & \mathbf{0} & \cdots & \mathbf{H}_F \cdots & \mathbf{0} \end{pmatrix}$$

$$\mathbf{H}_R = \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_{R_k}^B}\right|_{(\hat{\mathbf{x}}_{k|k-1}^B)} \; ; \; \mathbf{H}_F = \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_{F_k}^B}\right|_{(\hat{\mathbf{x}}_{k|k-1}^B)}$$

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:

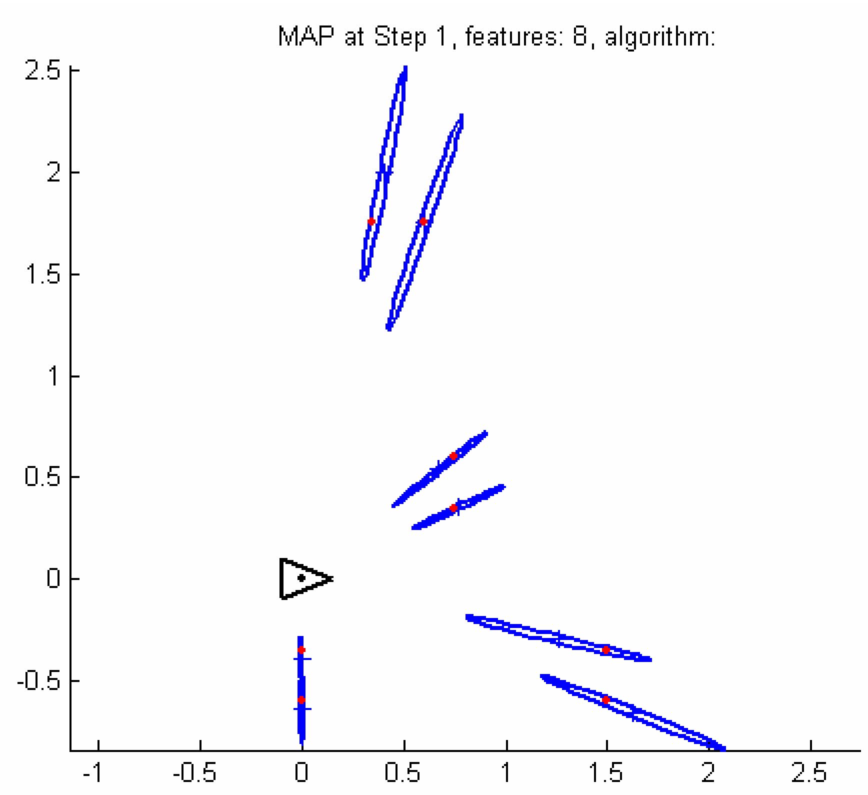
$$\mathbf{P}_k^B = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}^B$$

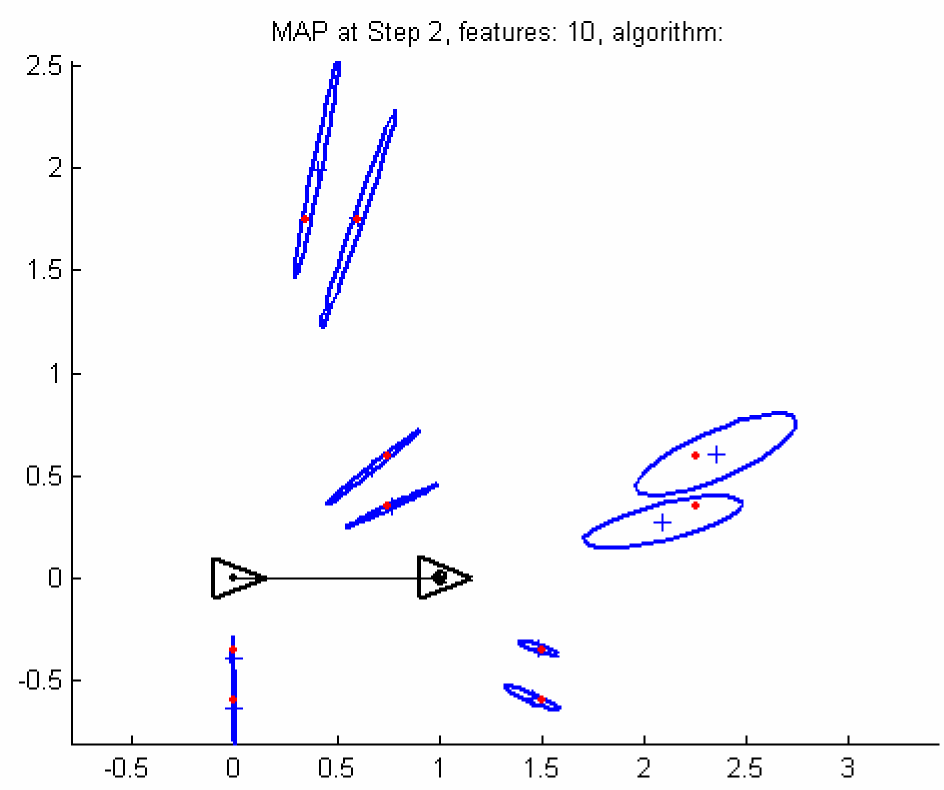
Filter gain:

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

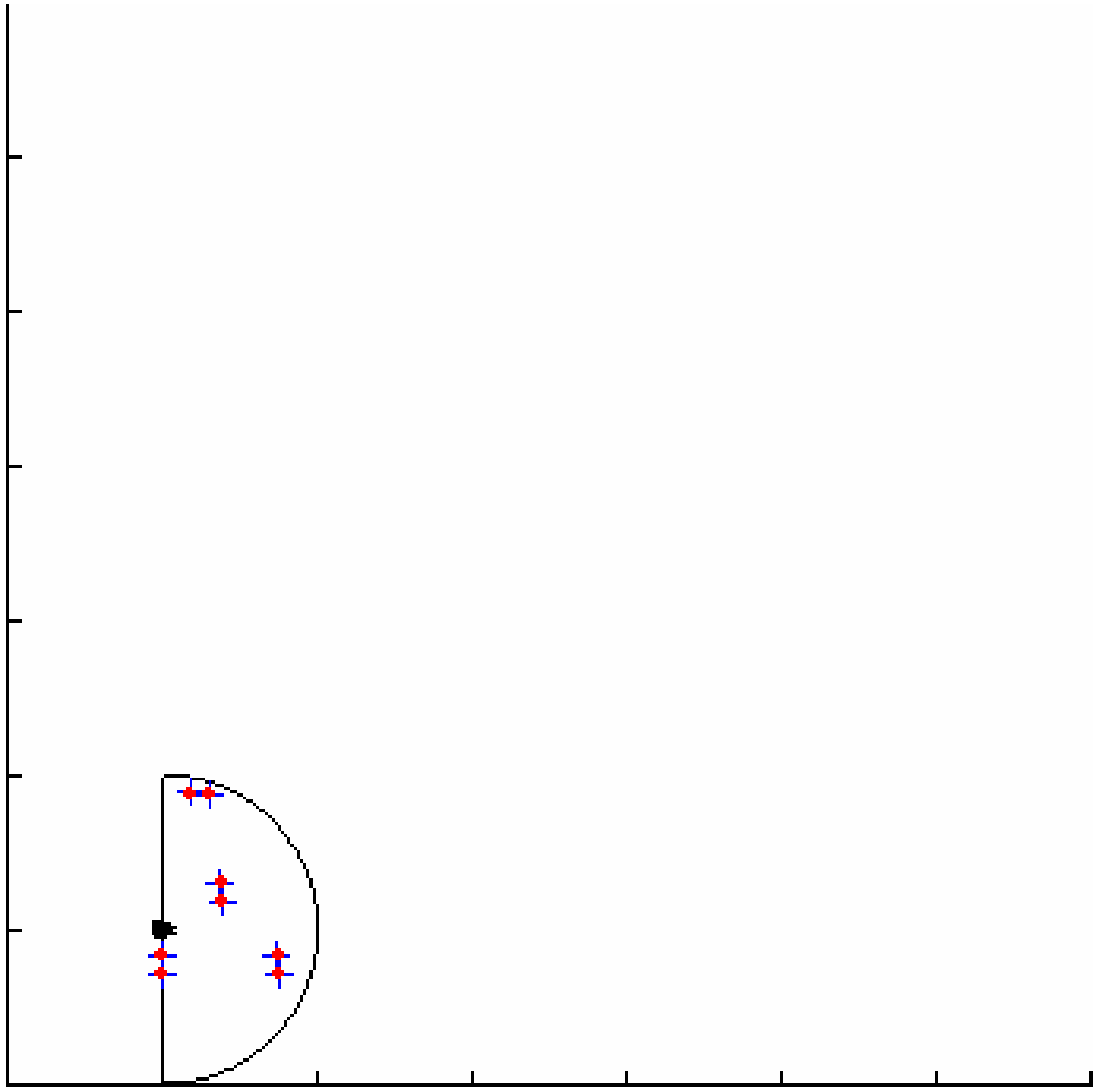
ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: map update



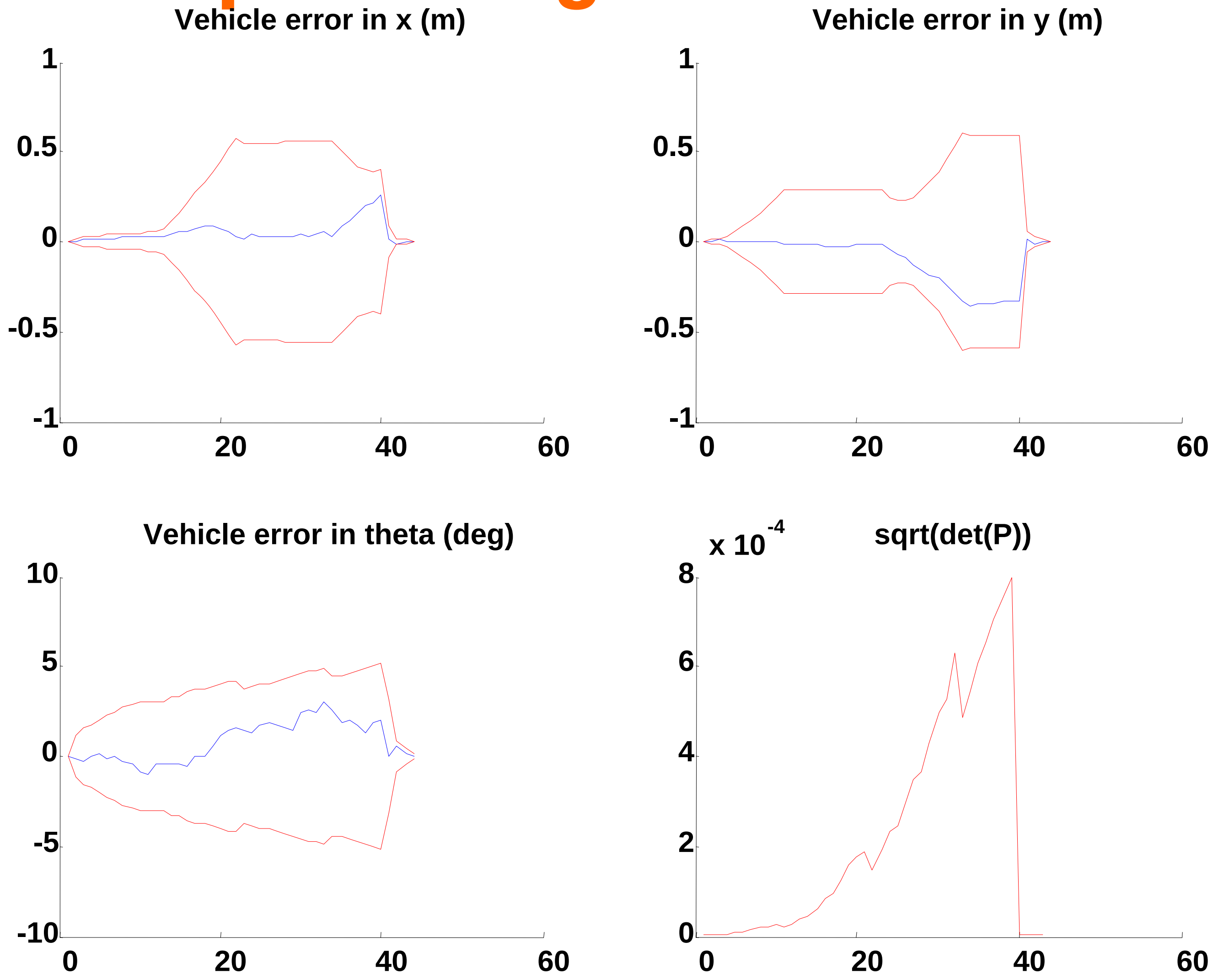


Why we do SLAM



Uncertainty still grows!

Loop closing in EKF-SLAM



Loop closing reduces uncertainty!

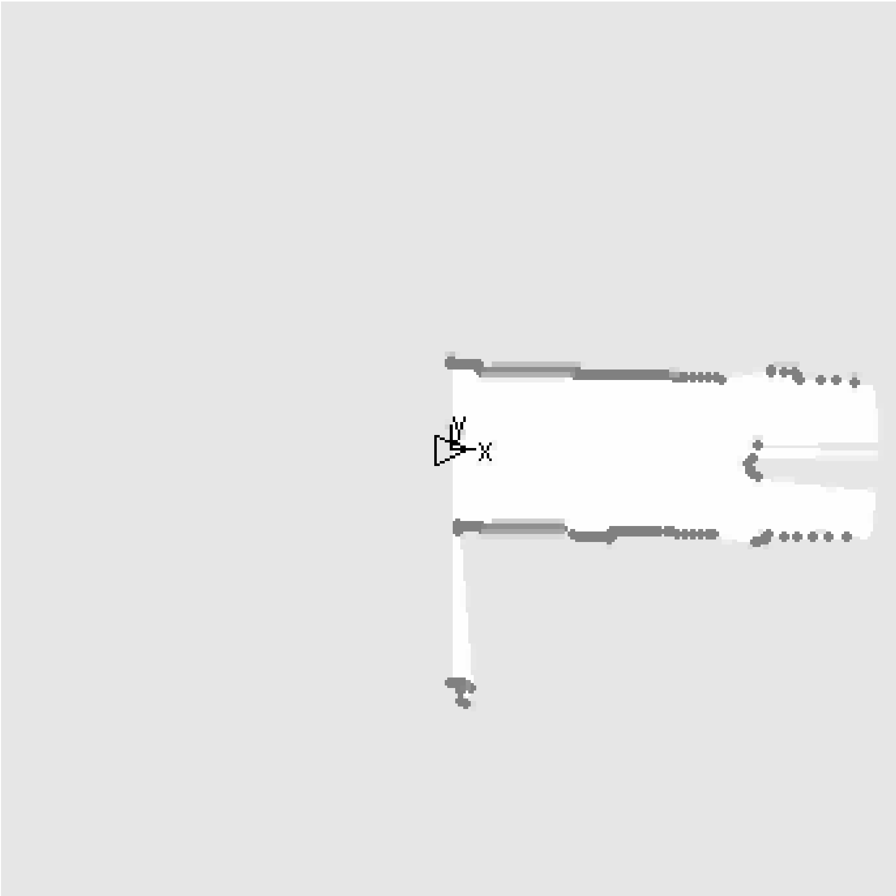
Outline

1. Basic EKF SLAM
   – Introduction: the need for SLAM
   – The basic EKF SLAM algorithm
   – **Feature Extraction**
   – Continuous Data Association
   – The Loop Closing Problem

2. Advanced EKF SLAM
   – Computational complexity of EKF SLAM
   – Consistency of the EKF SLAM
   – SLAM using local maps
      » Sequential Map Joining
      » Divide and Conquer SLAM
      » Hierarchical SLAM

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Feature extraction: Laser



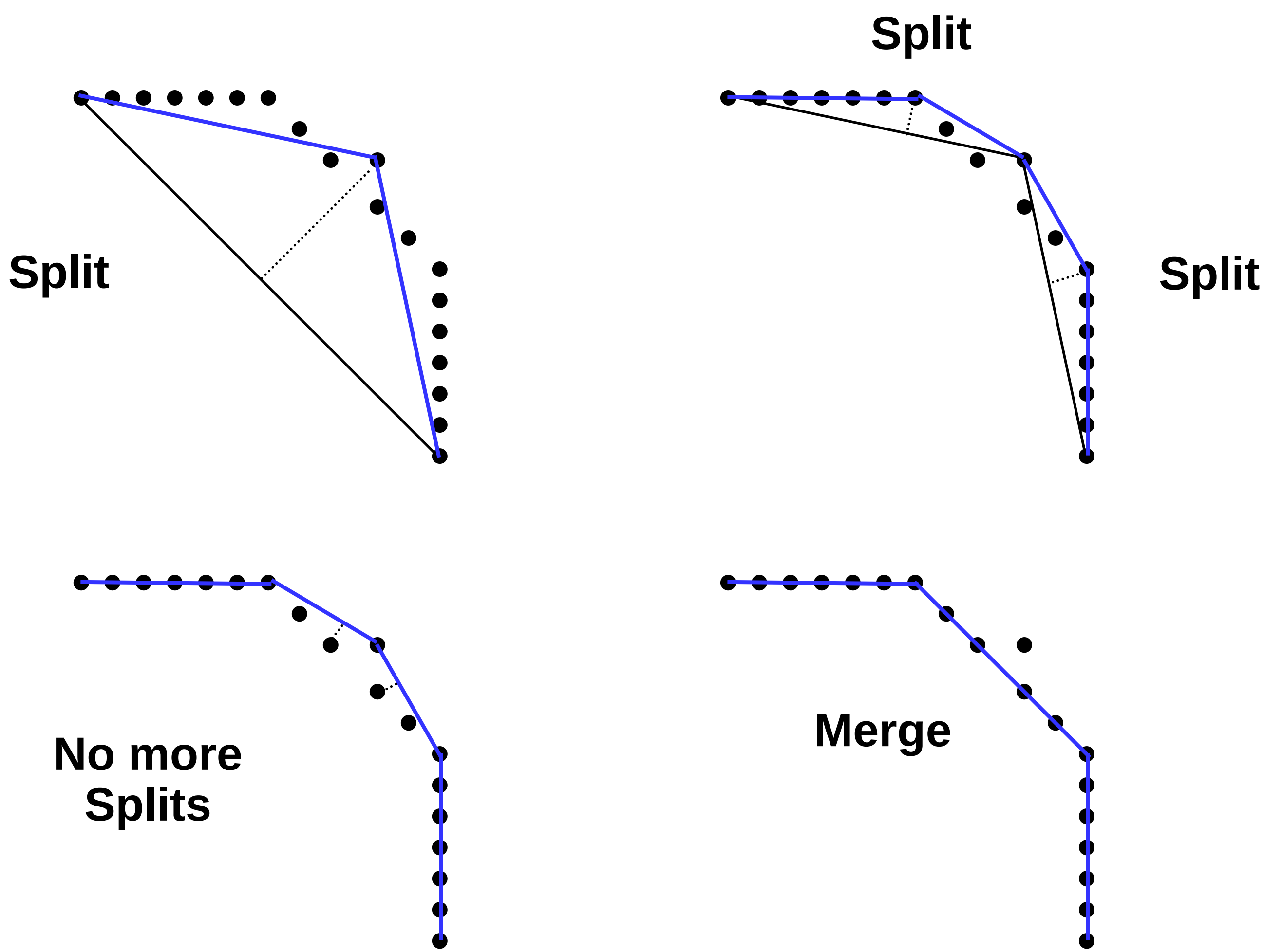
- Obtain line segments from a laser scan:
  - Segmentation
  - Line estimation

Split and merge:

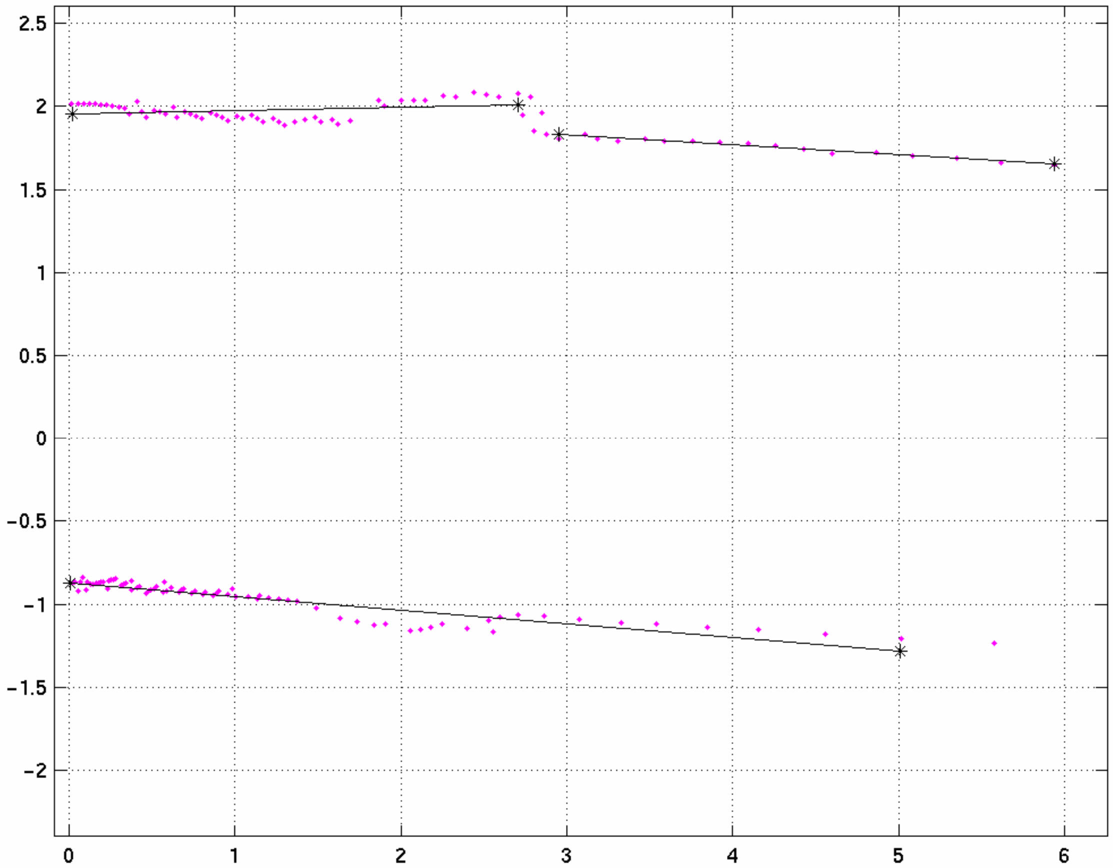
1. **Recursive Split:**
   1. Obtain the line passing by the two extreme points
   2. Obtain the point more distant to the line
   3. If distance > error_max, split and repeat with the left and right sub-scan
2. **Merge:**
   1. If two consecutive segments are close enough, obtain the common line and the more distant point
   2. If distance <= error_max, merge both segments
3. **Prune short** segments
4. **Estimate line equation**

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Split and Merge



Split and Merge

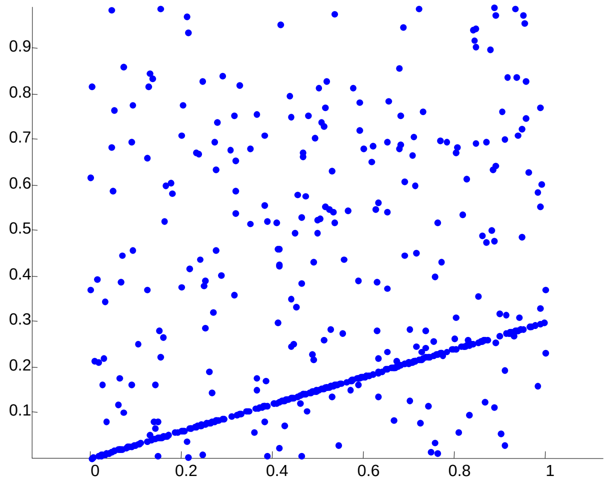


Not robust to complex and/or spurious data

RANSAC

- Given a model that requires n data points to compute a solution and a set of data points P, with #(P) > n :
  - Randomly select a subset S1 of n data points and compute the model M1
  - Determine the **consensus** set S1* of points is P compatible with M1 (within some error tolerance)
  - If #(S1*) > th, use S1* to compute (maybe using least squares) a new model M1*
  - If #(S1*) < th, randomly select another subset S2 and repeat
  - If, after t trials there is no consensus set with th points, return with failure

RANSAC



p no. of points

n points to build model

w probability that a point is good

$O(p^n)$_possible_models

z acceptable probability of failure

t tries ¿?

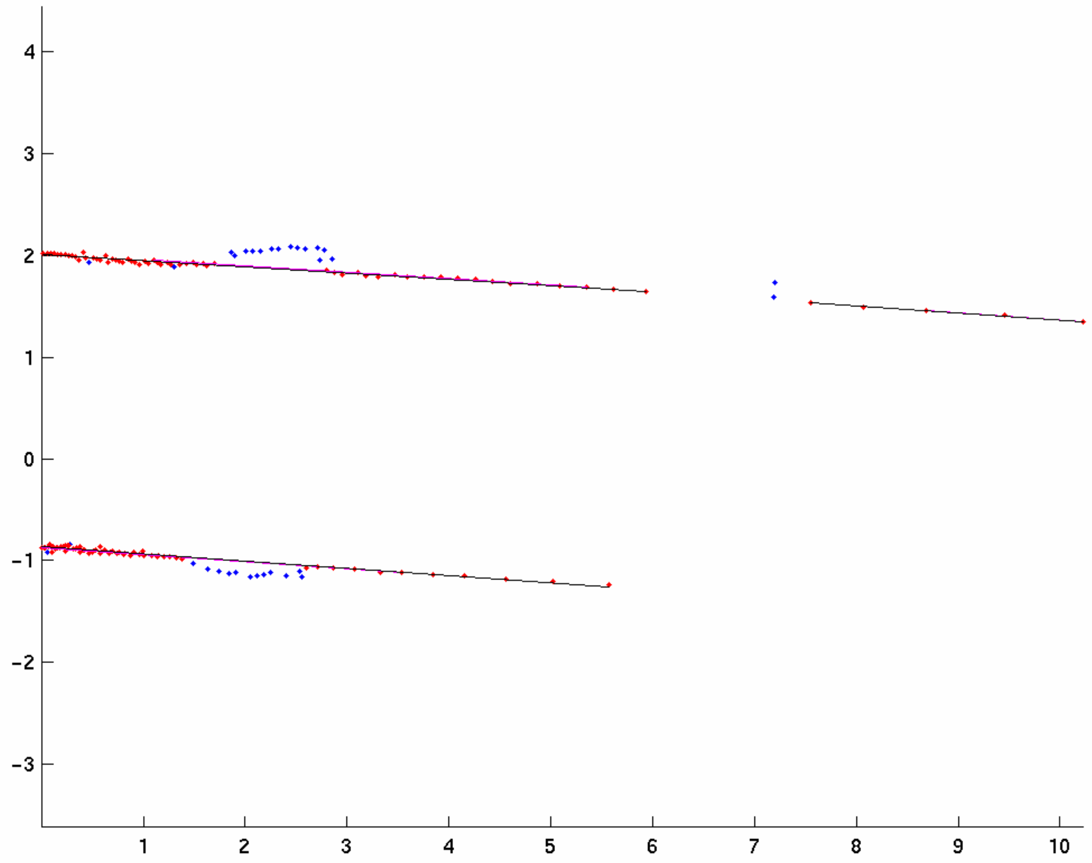
$$(1 - w^n)^t = z$$

$$t = \left\lceil \frac{\log z}{\log(1 - w^n)} \right\rceil$$

w	0,5
n	2
z	0,05
t	11

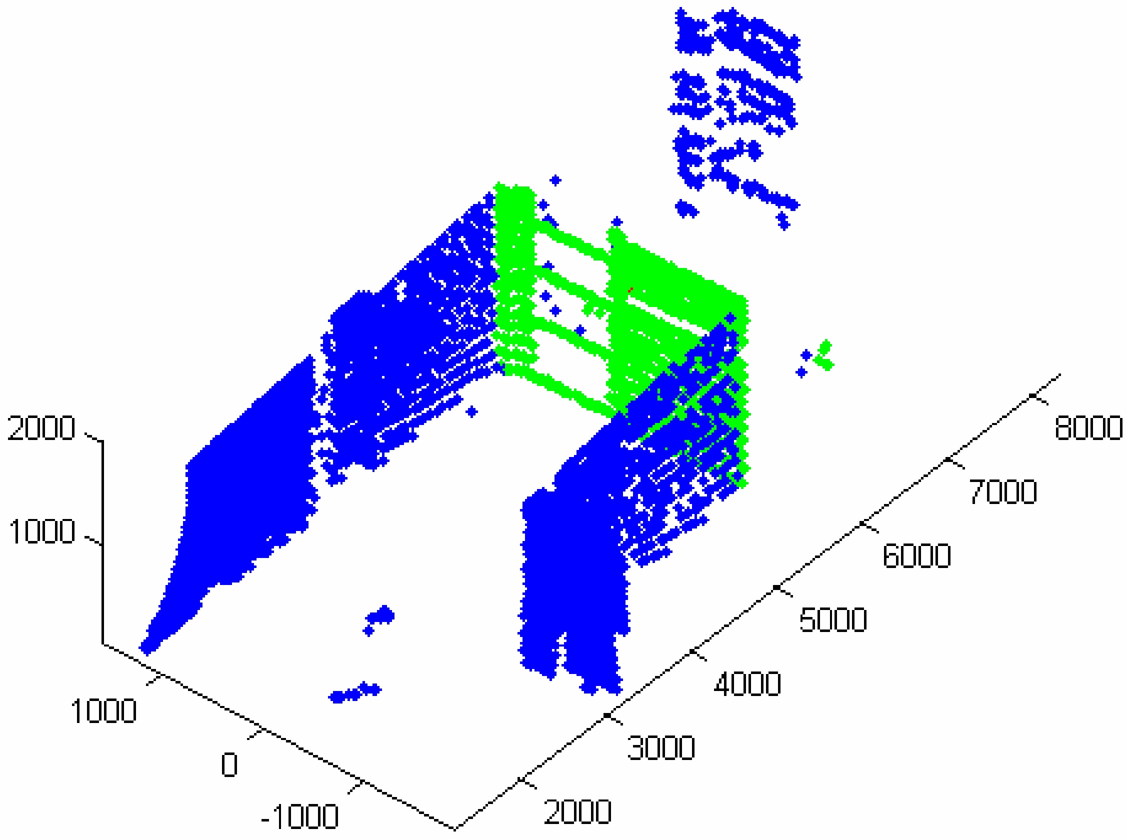
ROBOTICS UNIVERSIDAD DE ZARAGOZA

RANSAC



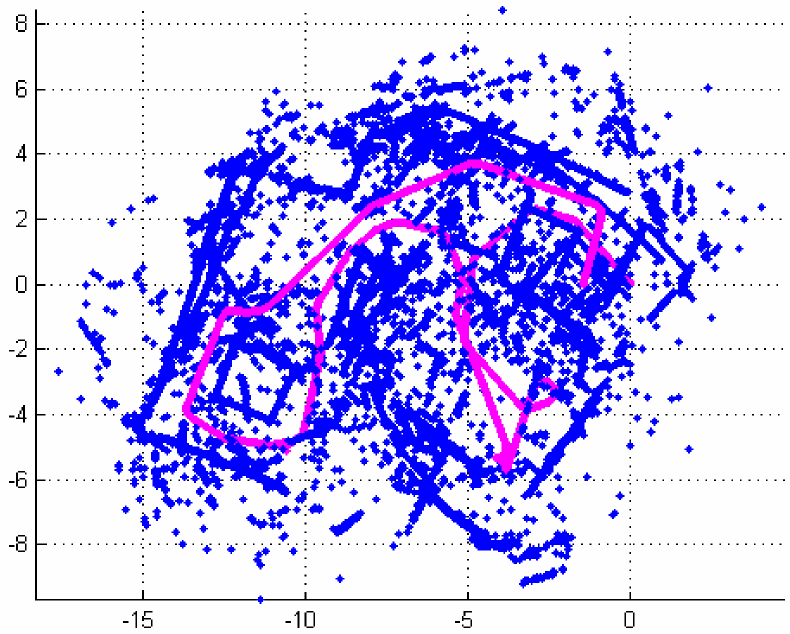
Robust statistics deal with spuriousness

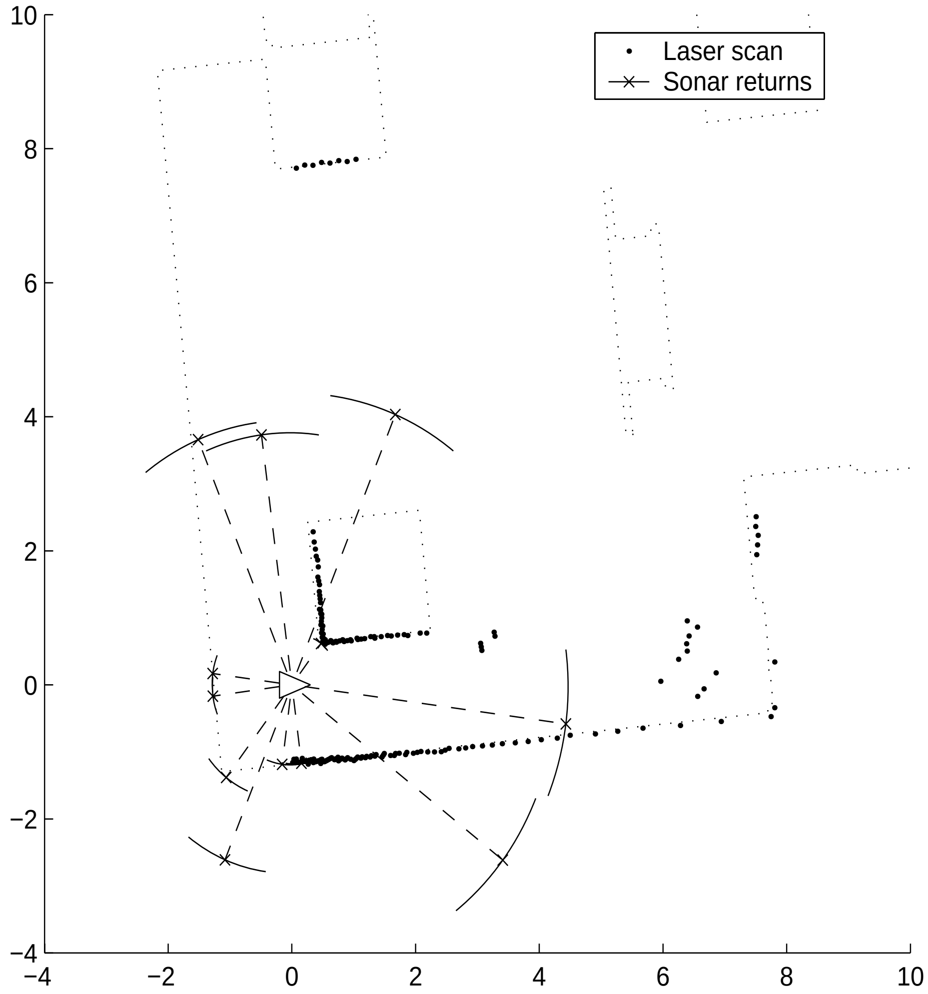
RANSAC for 3D planes



P.M. Newman, J.J. Leonard, J. Neira and J.D. Tardós: **Explore and Return: Experimental Validation of Real Time Concurrent Mapping and Localization**. IEEE Int. Conf. Robotics and Automation, May, 2002

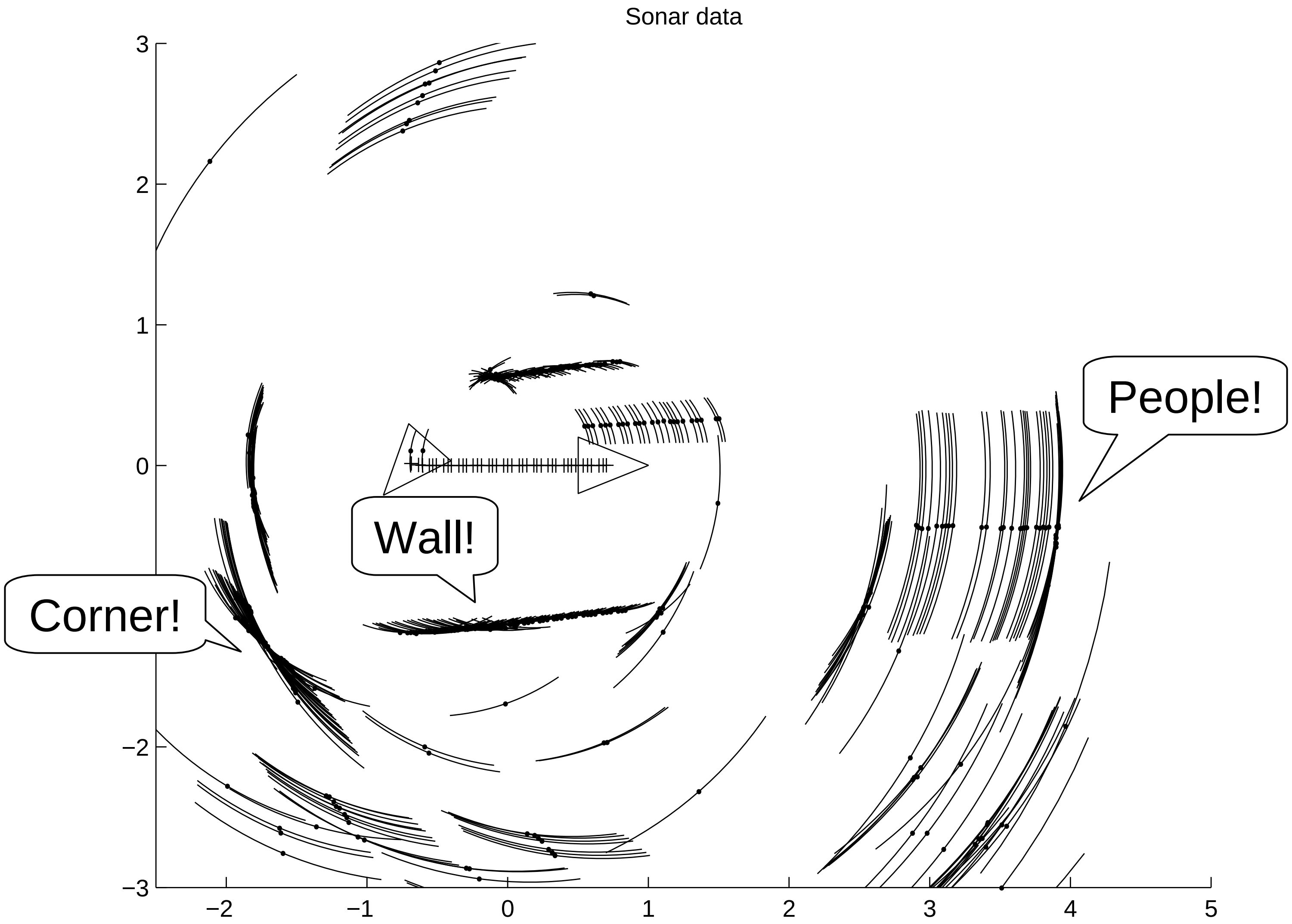
Sonar





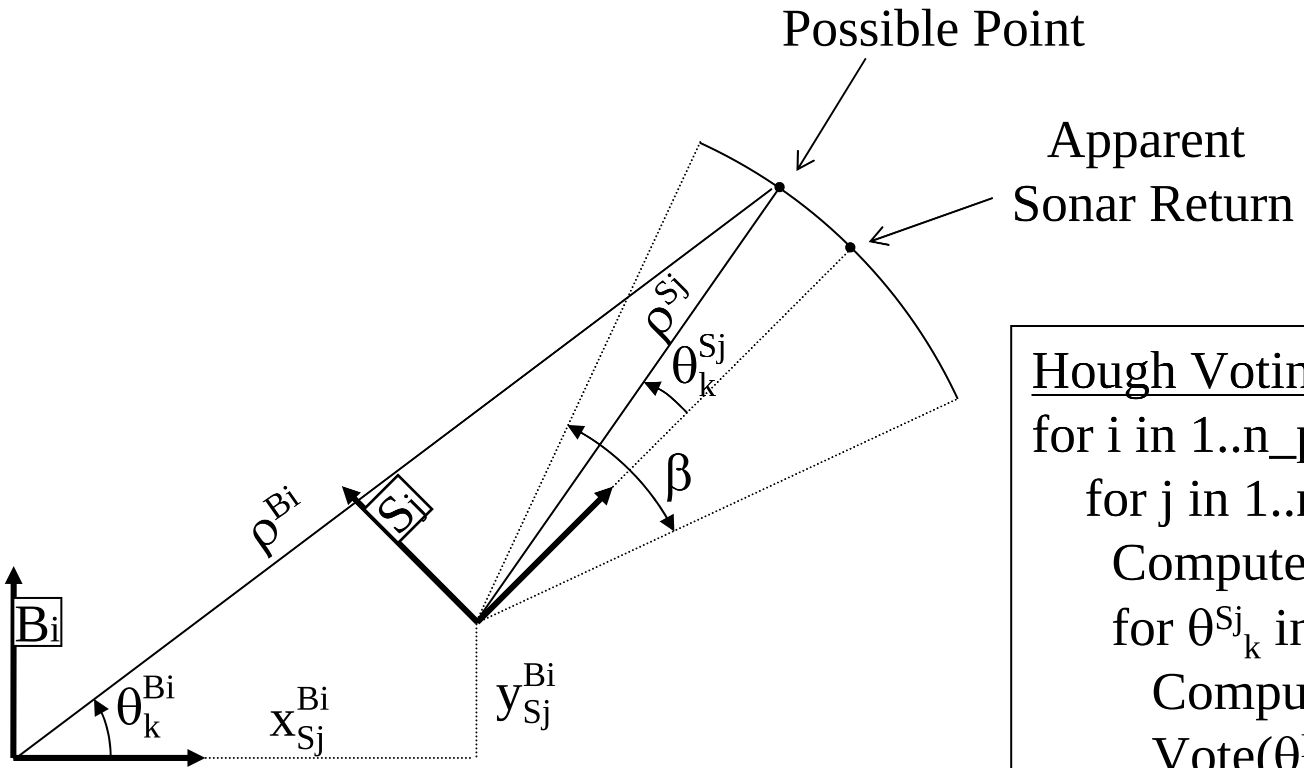
Very sparse and noisy data

Move and build a local map



Exploit redundancy

Sonar Model for Points

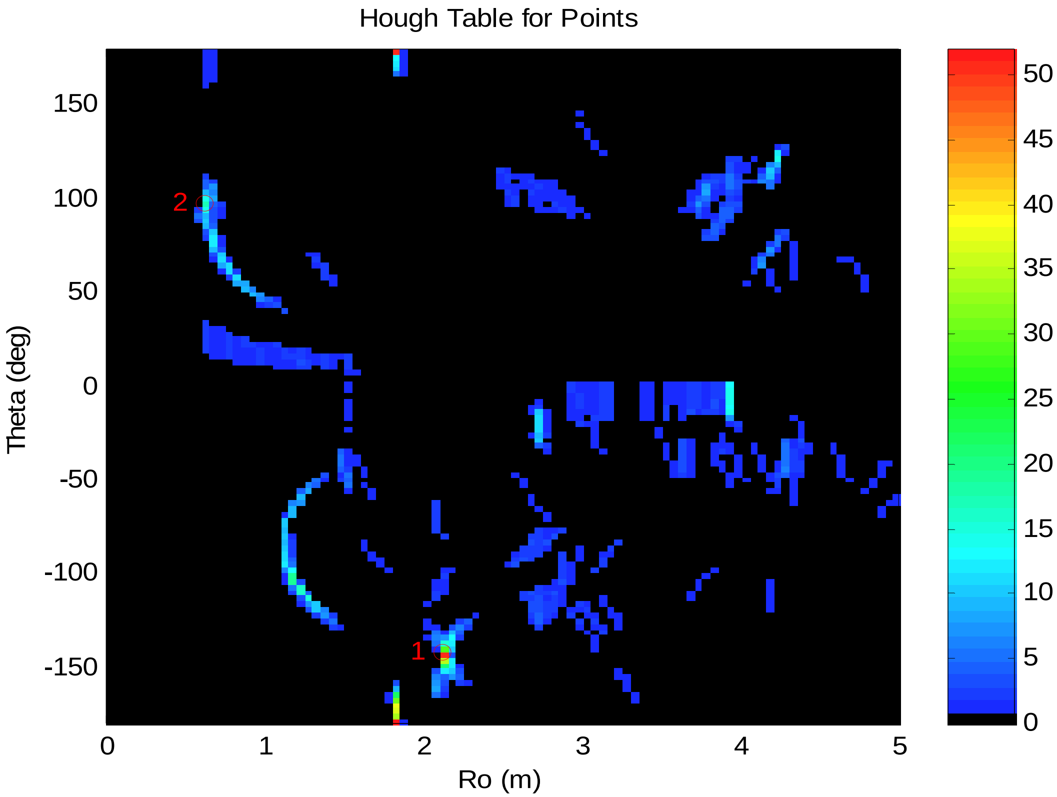


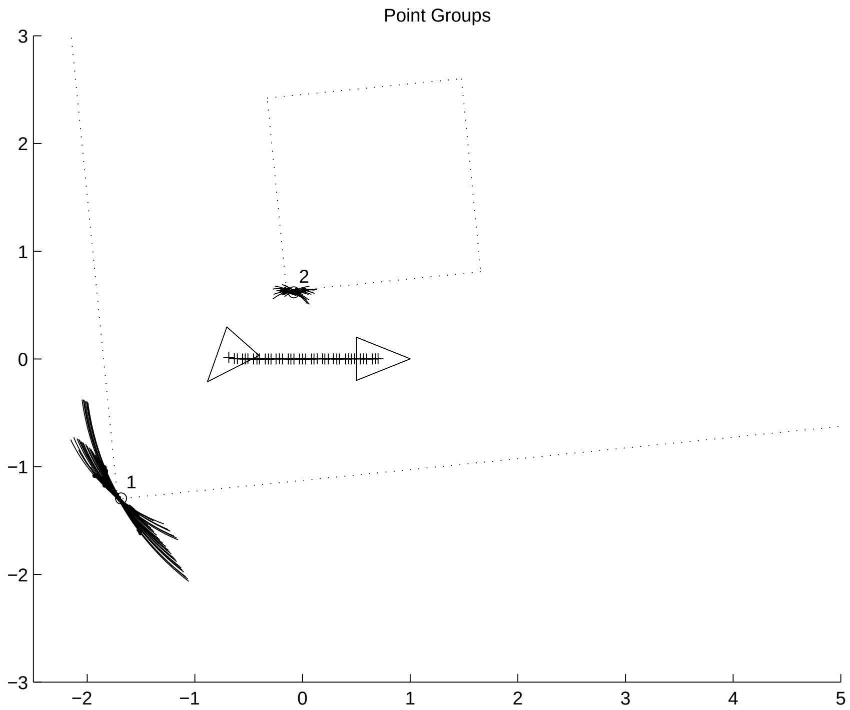
<u>Hough Voting</u>

for i in 1..n_positions
    for j in 1..n_sensors
        Compute $\mathbf{x}^{Bi}{}_{Sj}$
        for $\theta^{Sj}{}_{k}$ in $-\beta/2..\beta/2$ step δ
            Compute $\theta^{Bj}{}_{k}$ $\rho^{Bj}{}_{k}$
            Vote($\theta^{Bj}{}_{k}$, $\rho^{Bj}{}_{k}$)
        end
    end
end

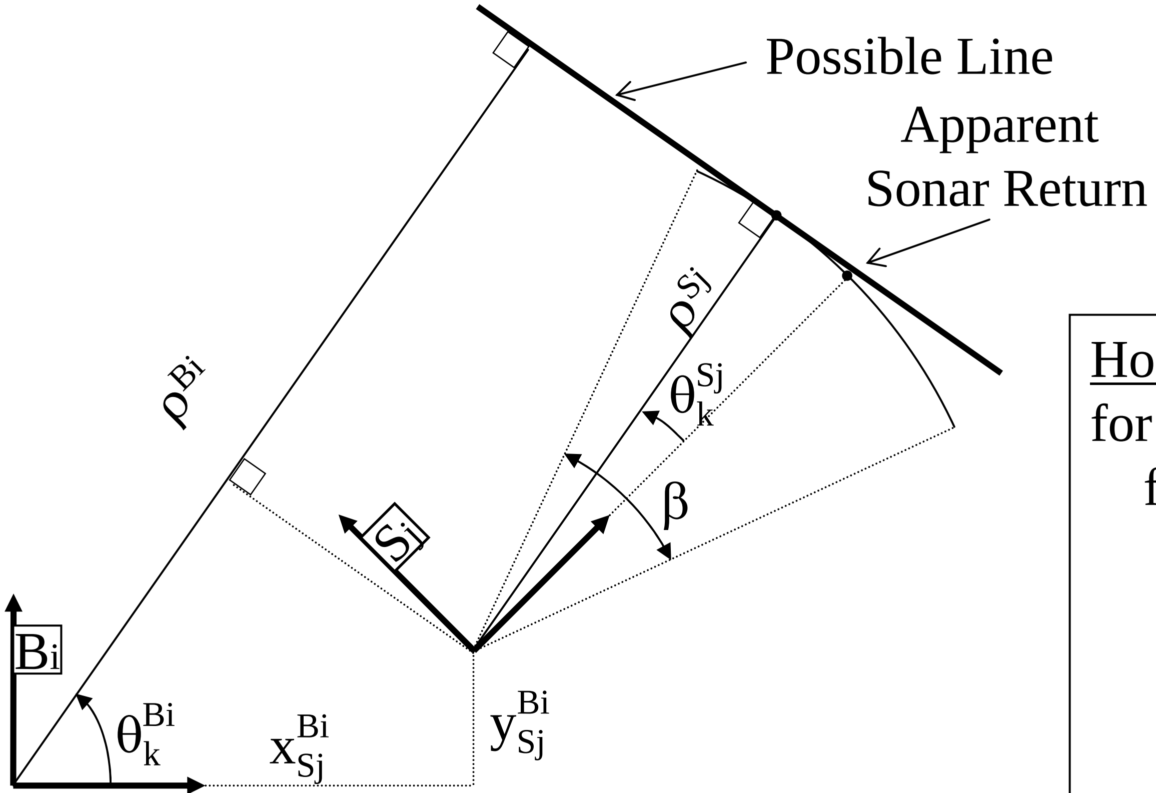
Hough Transform: Corners

- Sonar returns **vote** for points
- Look for local maxima





Sonar Model for Lines

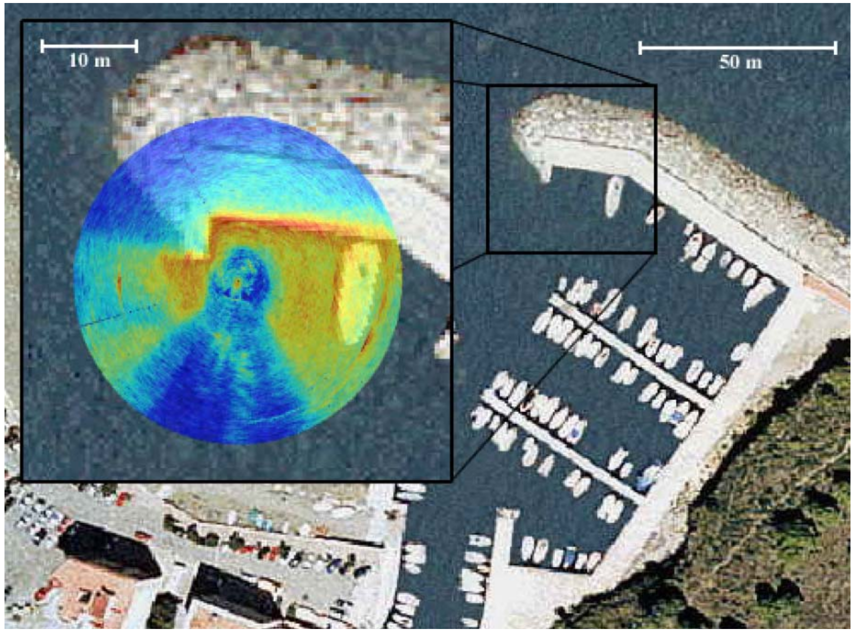


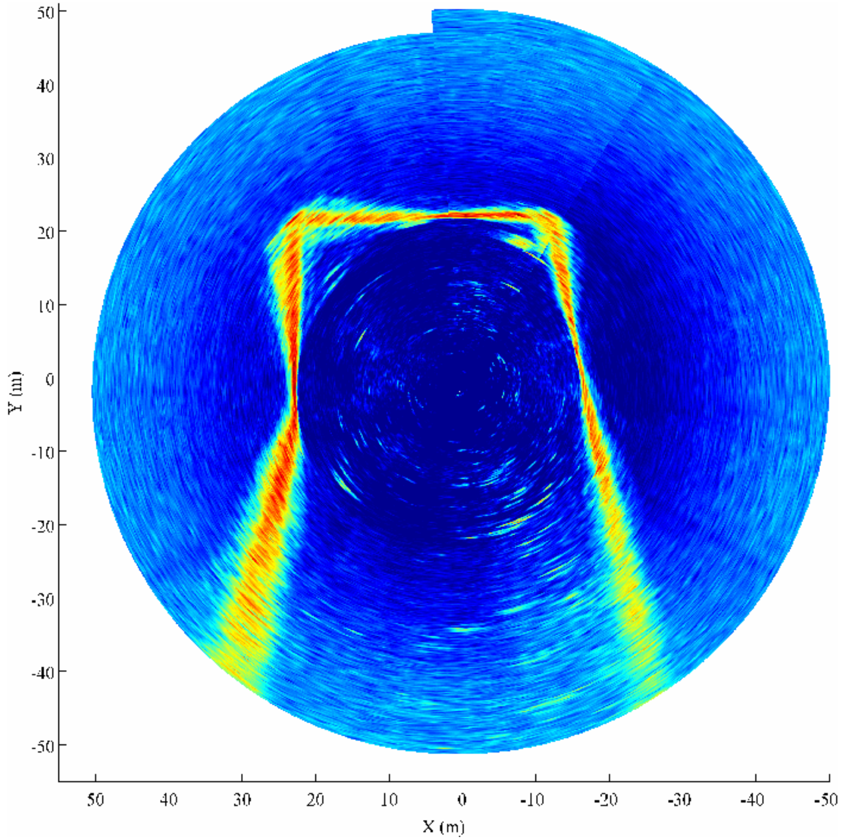
<u>Hough Voting</u>

for i in 1..n_positions
  for j in 1..n_sensors
    Compute $\mathbf{x}^{Bi}{}_{Sj}$
    for $\theta^{Sj}{}_k$ in $-\beta/2..\beta/2$ step δ
      Compute $\theta^{Bj}{}_k$ $\rho^{Bj}{}_k$
      Vote($\theta^{Bj}{}_k$, $\rho^{Bj}{}_k$)
    end
  end
end

Hough Transform

- Imaging sonar



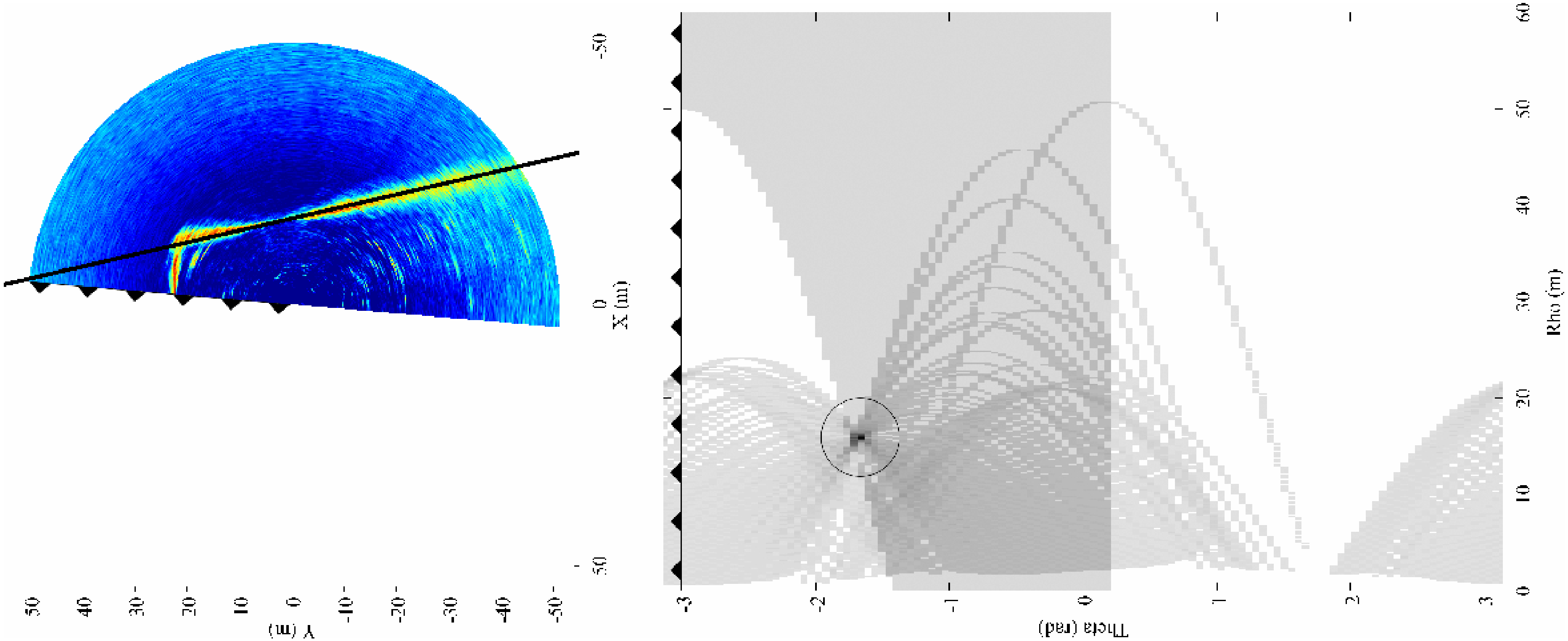


D. Ribas, P. Ridao, J. Neira, J.D. Tardós, **SLAM in Partially Structured Underwater Environments**, To appear in Journal of Field Robotics, 2008

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Hough Transform

- Imaging sonar



D. Ribas, P. Ridao, J. Neira, J.D. Tardós, **SLAM in Partially Structured Underwater Environments**, To appear in Journal of Field Robotics, 2008

Outline

1. Basic EKF SLAM
   - Introduction: the need for SLAM
   - The basic EKF SLAM algorithm
   - Feature Extraction
   - **Continuous Data Association**
   - The Loop Closing Problem

2. Advanced EKF SLAM
   - Computational complexity of EKF SLAM
   - Consistency of the EKF SLAM
   - SLAM using local maps
     - » Sequential Map Joining
     - » Divide and Conquer SLAM
     - » Hierarchical SLAM

ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM

Algorithm 1 SLAM:

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

The Data Association Problem

- ***n*** map features:

$$\mathcal{F} = \{F_1 \ldots F_n\}$$

- ***m*** sensor measurements:

$$\mathcal{E} = \{E_1 \ldots E_m\}$$

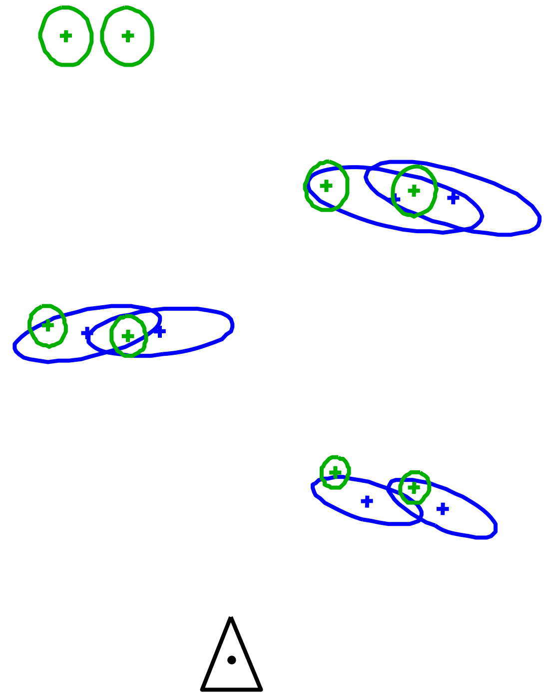
- Data association should return a hypothesis that associates each observation E_i with a feature F_{j_i}

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

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

- Non matched observations:

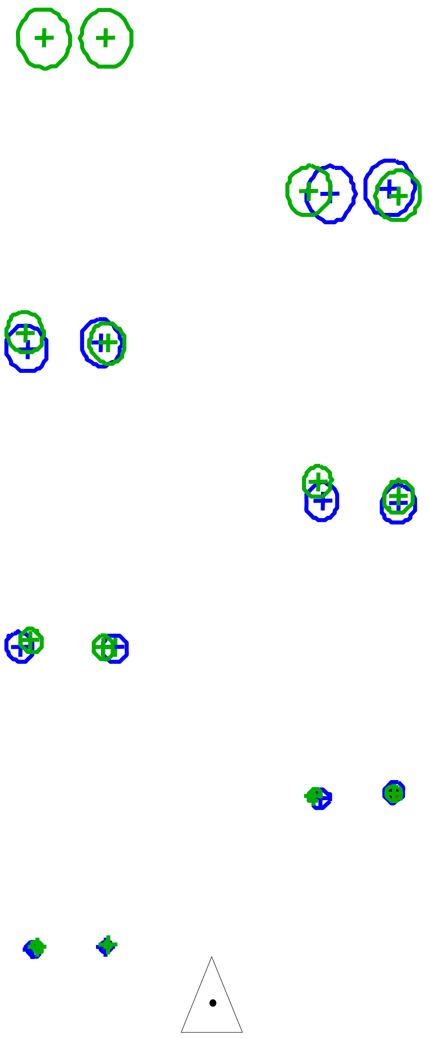
$$j_i = 0$$



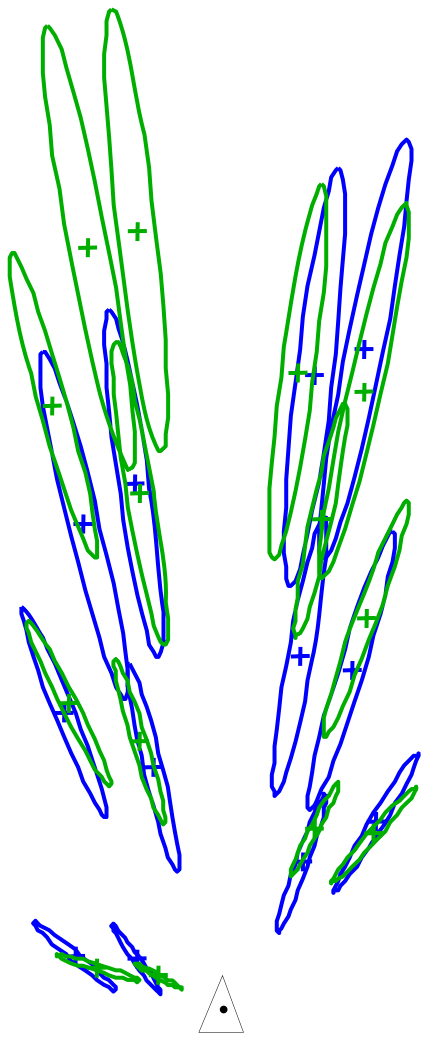
Green points: measurements
Blue Points: predicted features

When data association is difficult

- Low sensor error

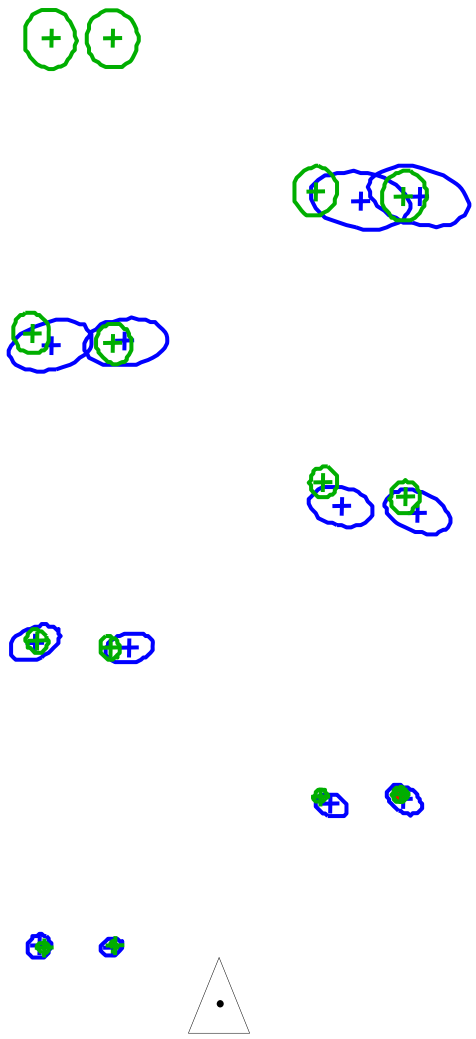


- High sensor error

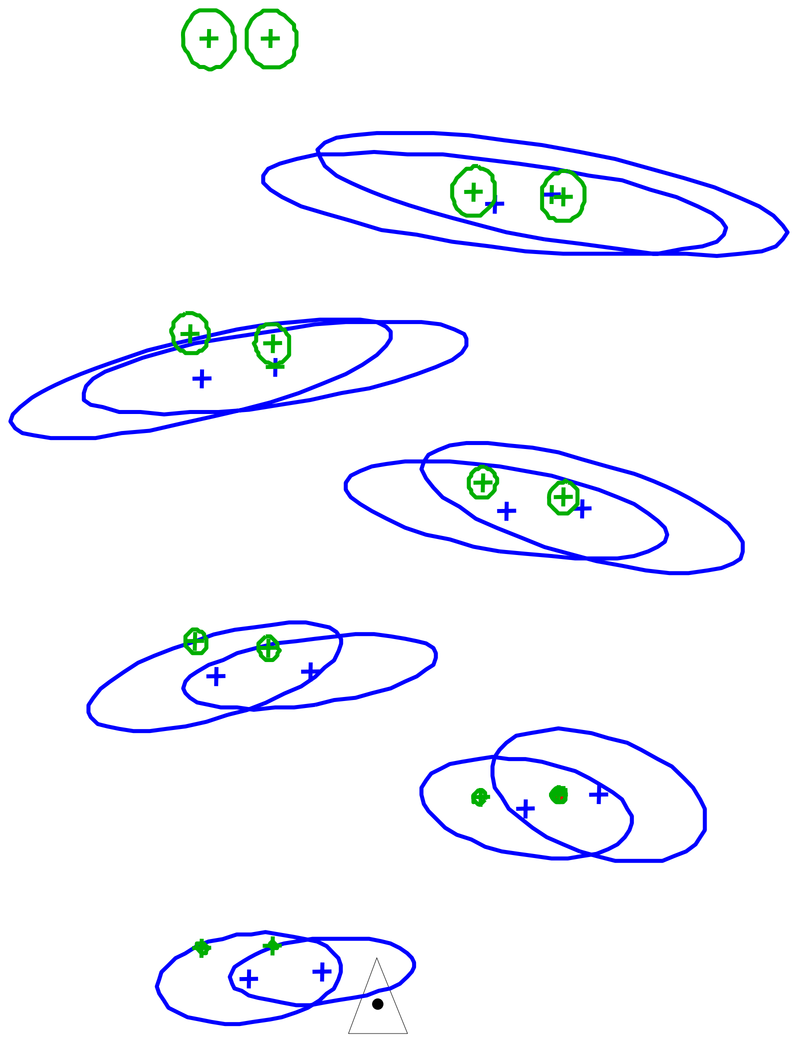


When data association is difficult

- Low odometry error

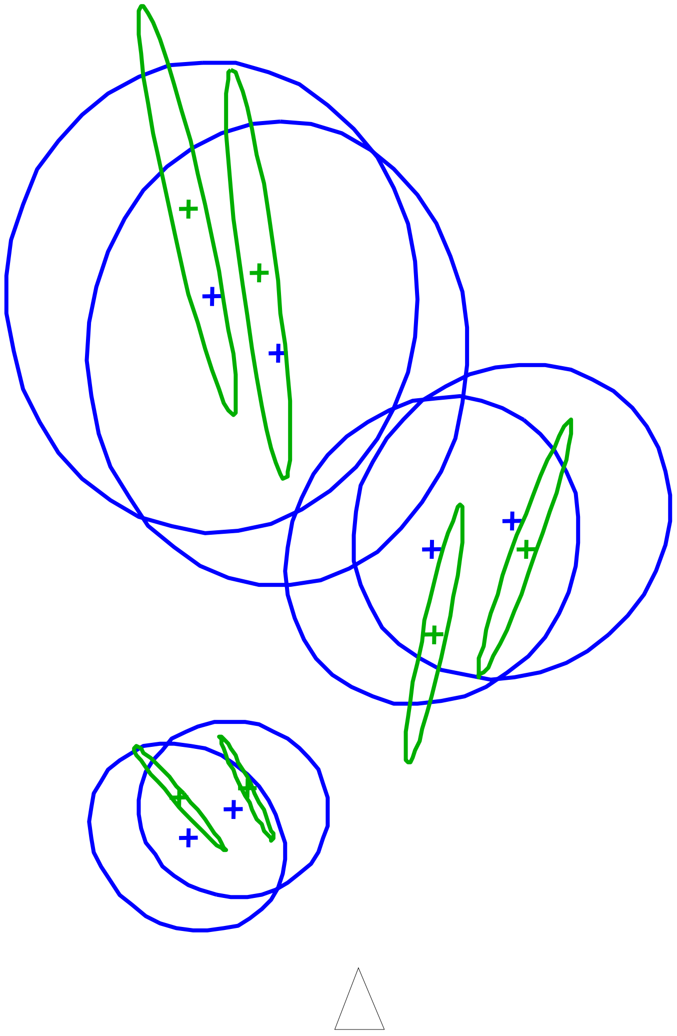


- High odometry error

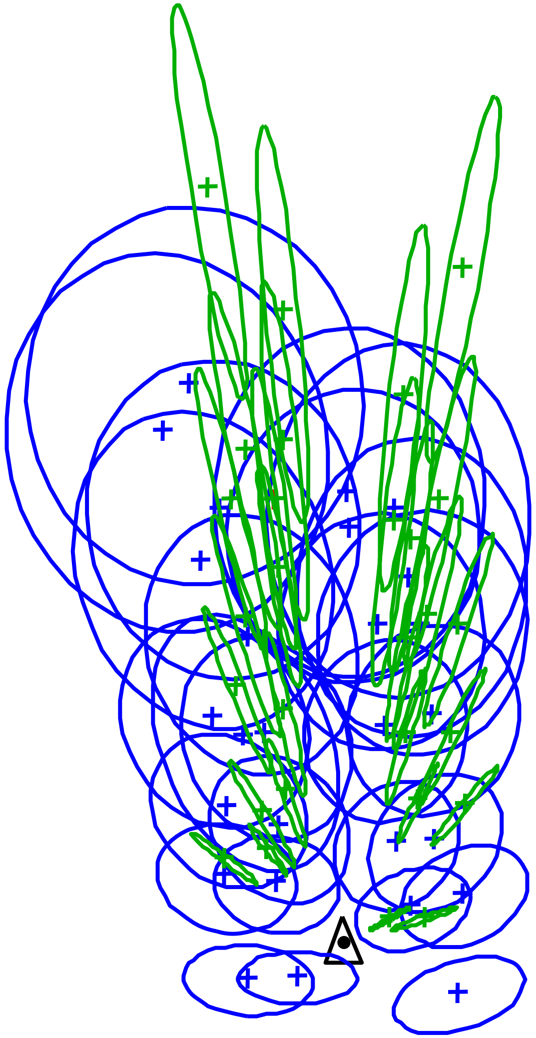


When data association is difficult

- Low feature density

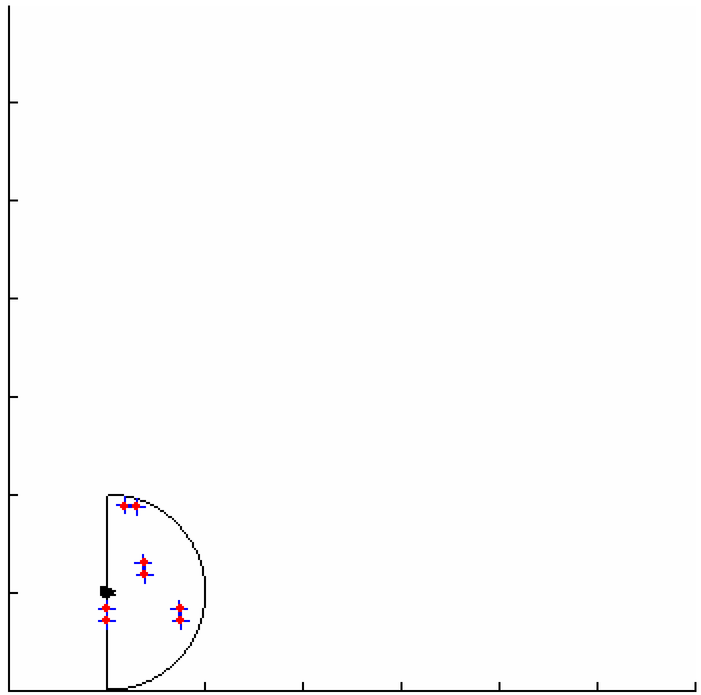


- High feature density

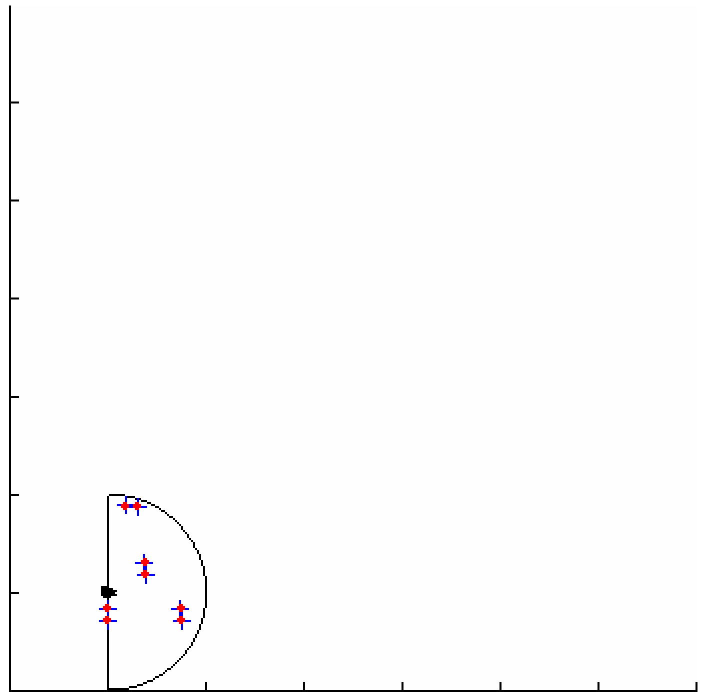


ROBOTICS
UNIVERSIDAD DE ZARAGOZA

How important is data association?

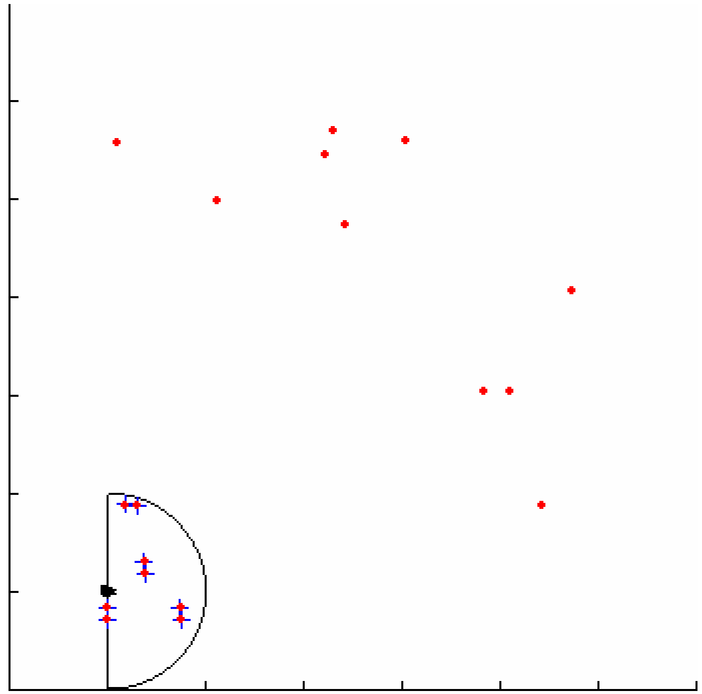


A good algorithm

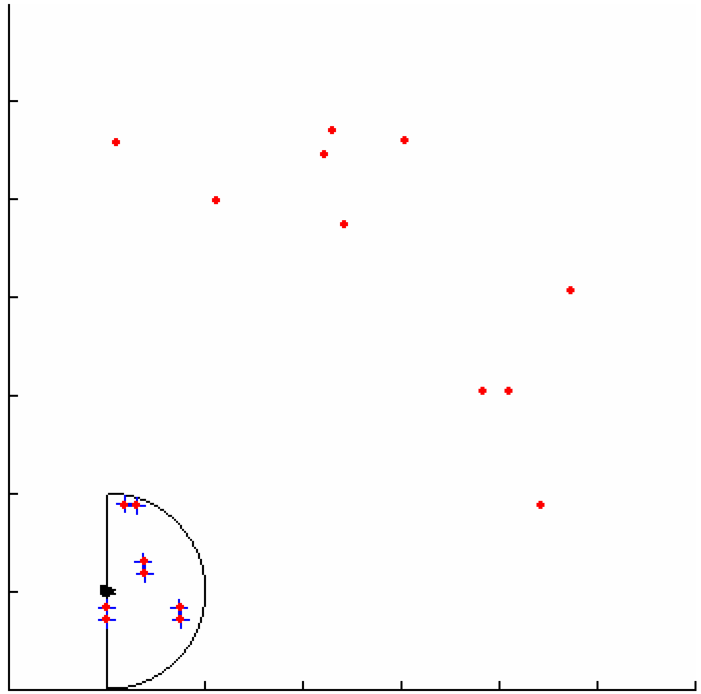


A **bad** algorithm

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 $\boldsymbol{E_i}$ with feature $\boldsymbol{F_j}$ is.....

	correct:	spurious:
error: $\mathbf{x} - \hat{\mathbf{x}}$	↓	↑
covariance: P	↓	↓
	Consistency	Divergence!

Individual Compatibility

- Measurement equation for observation E_i and feature F_j

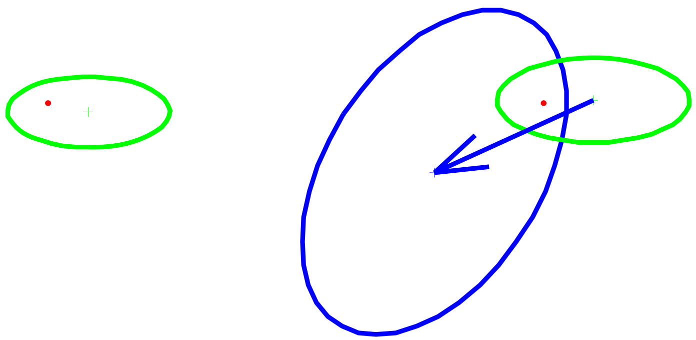
$$\mathbf{z}_i = \mathbf{h}_{ij}(\mathbf{x}^B) + \mathbf{w}_i$$
$$\mathbf{z}_i \simeq \mathbf{h}_{ij}(\hat{\mathbf{x}}^B) + \mathbf{H}_{ij}(\mathbf{x}^B - \hat{\mathbf{x}}^B)$$

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

- $\boldsymbol{E_i}$ and $\boldsymbol{F_j}$ are compatible if:

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

d = length(z_i)

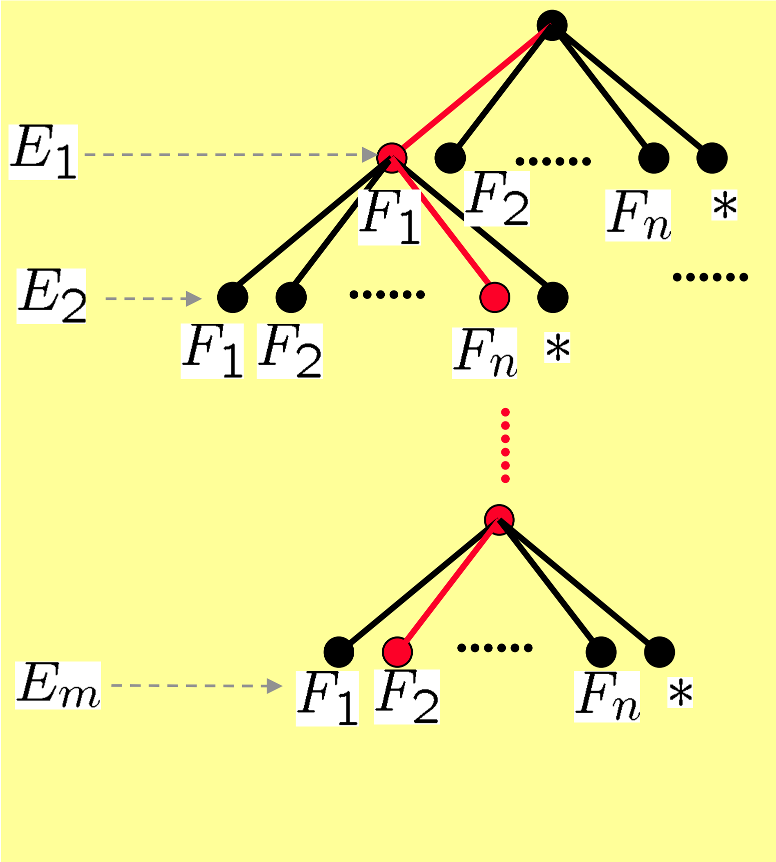


Nearest Neighbor

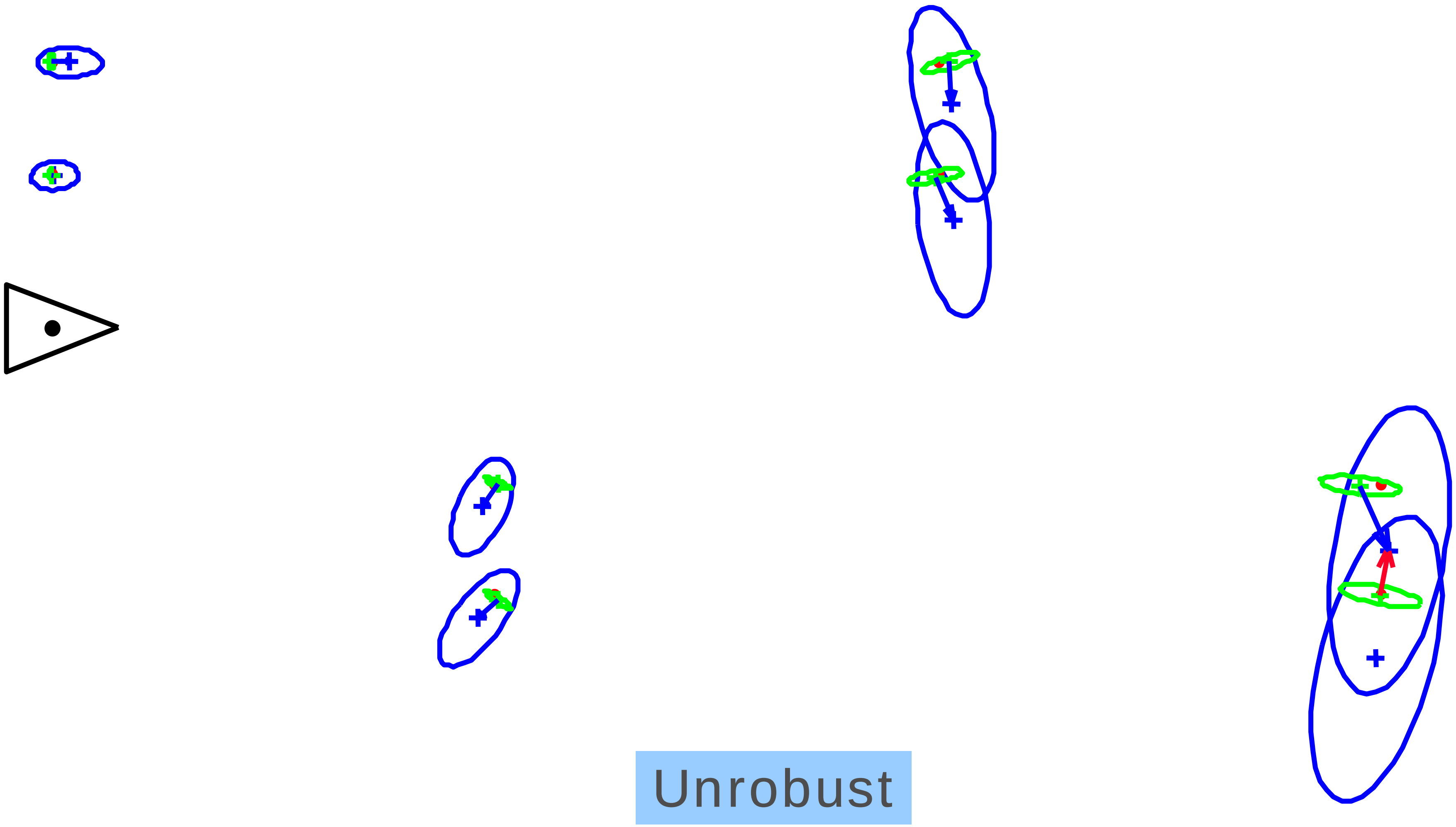
Algorithm 2 Individual Compatibility Nearest Neighbor ICNN $(E_{1\cdots m}, F_{1\cdots n})$

```
for i = 1 to m do {measurement E_i}
  D²_min ← mahalanobis2(E_i, F_1)
  nearest ← 1
  for j = 2 to n do {feature F_j}
    D²_ij ← mahalanobis2(E_i, F_j)
    if D²_ij < D²_min then
      nearest ← j
      D²_min ← D²_ij
    end if
  end for
  if D²_min ≤ χ²_{d_i,1−α} then
    H_i ← nearest
  else
    H_i ← 0
  end if
end for
return H
```

Greedy algorithm: *O(mn)*



The Fallacy of the Nearest Neighbor



Joint Compatibility

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

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

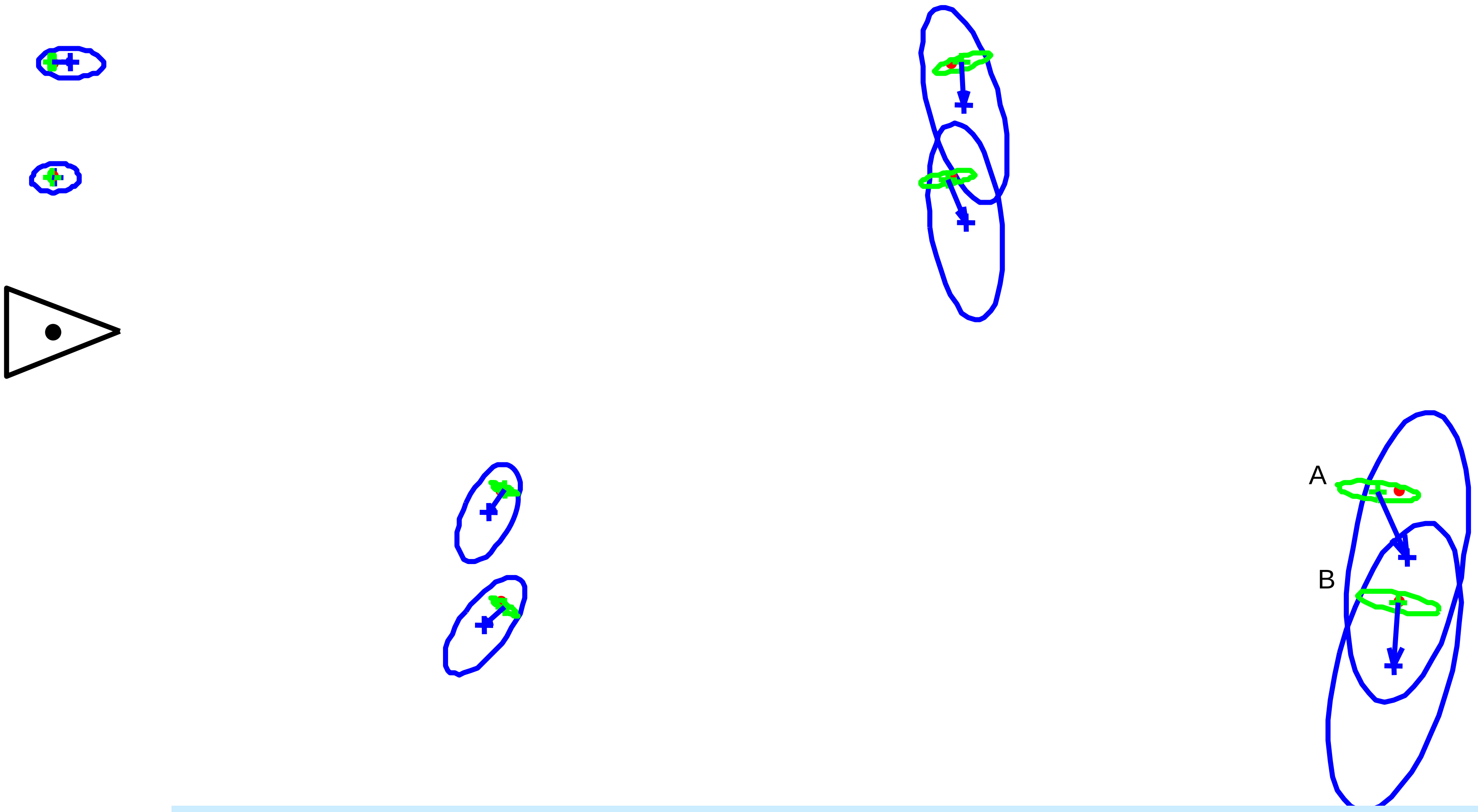
- The joint hypothesis is compatible if:

$$\begin{aligned} D^2_{\mathcal{H}} &= (\mathbf{z}_{\mathcal{H}} - \mathbf{h}_{\mathcal{H}}(\hat{\mathbf{x}}^B))^T C^{-1}_{\mathcal{H}} (\mathbf{z}_{\mathcal{H}} - \mathbf{h}_{\mathcal{H}}(\hat{\mathbf{x}}^B)) < \chi^2_{d,\alpha} \\ C_{\mathcal{H}} &= \mathbf{H}_{\mathcal{H}} \mathbf{P}^B \mathbf{H}^T_{\mathcal{H}} + \mathbf{R}_{\mathcal{H}} \end{aligned}$$

d = length(z)

ROBOTICS UNIVERSIDAD DE ZARAGOZA

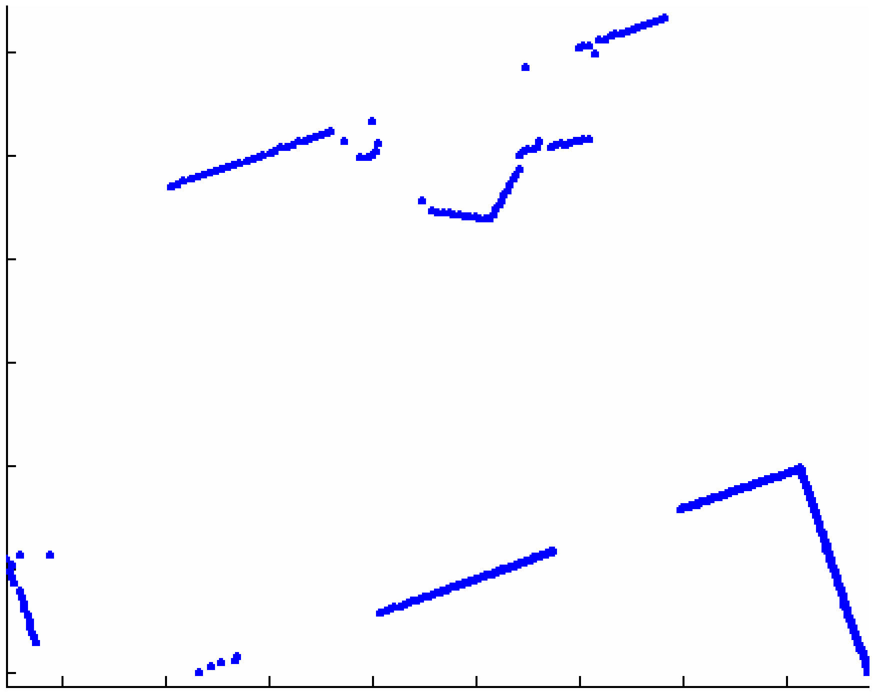
Jointly Compatible Pairings

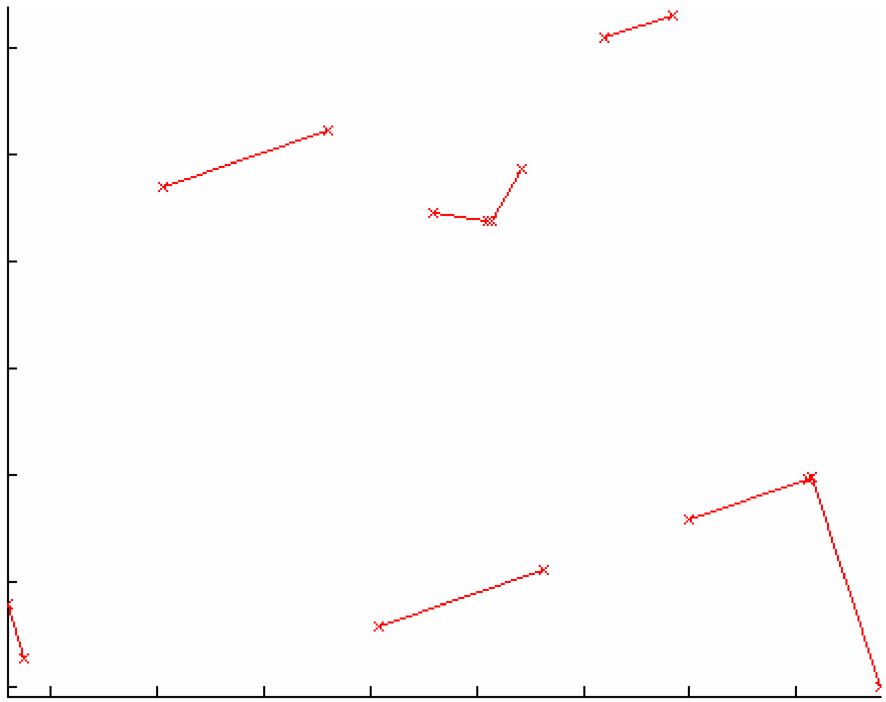


J. Neira, J.D. Tardós. **Data Association in Stochastic Mapping using the Joint Compatibility Test** IEEE Trans. Robotics and Automation, Vol. 17, No. 6, Dec 2001, pp 890 –897

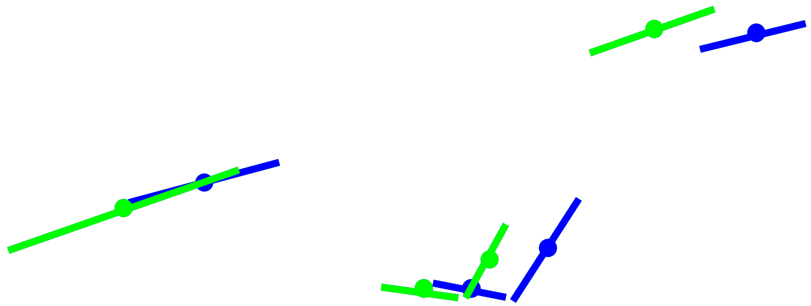
SLAM without odometry

- No estimation of the vehicle motion
- Segments in the environment

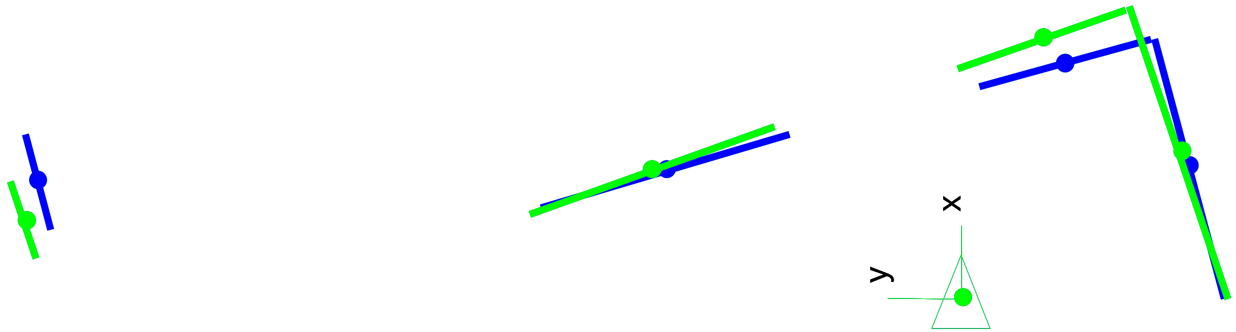




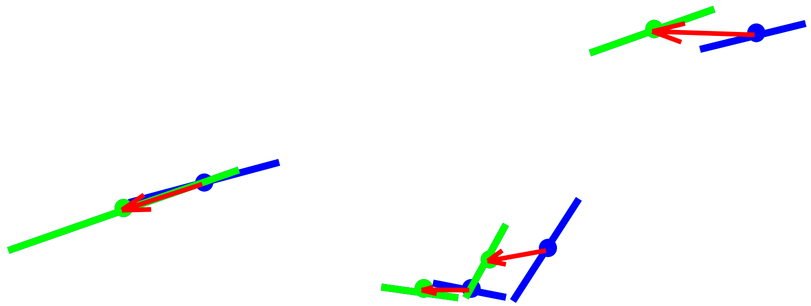
SLAM without odometry



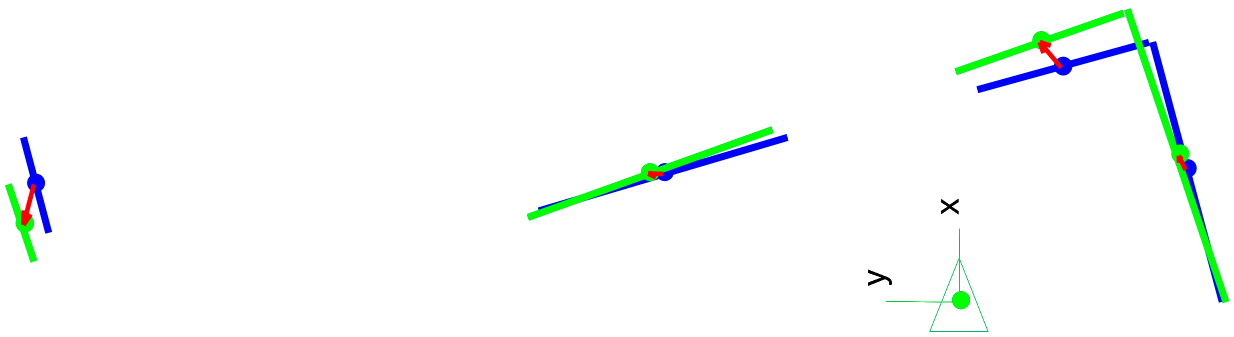
Assuming small motions



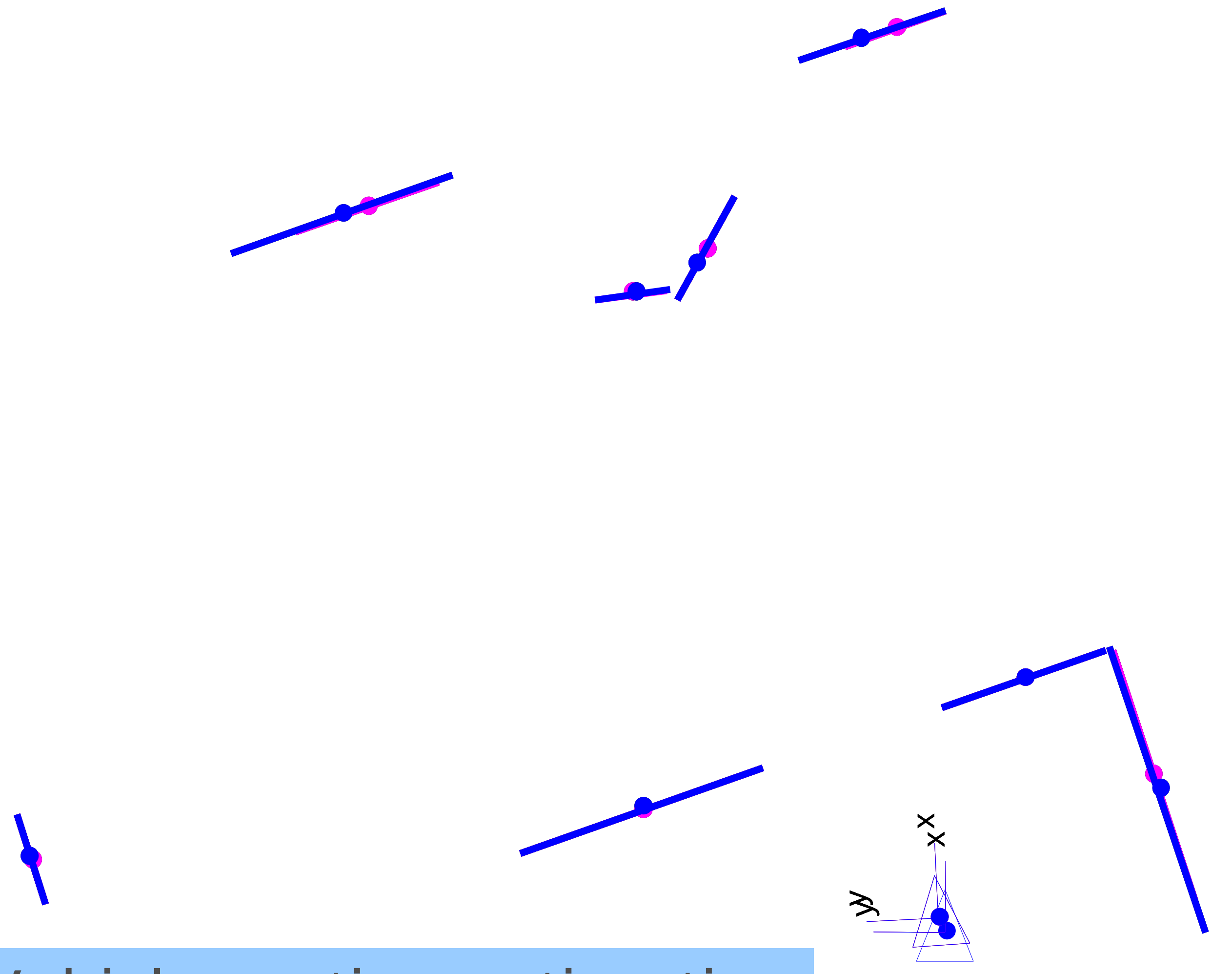
SLAM without odometry



Data association using Joint Compatibility

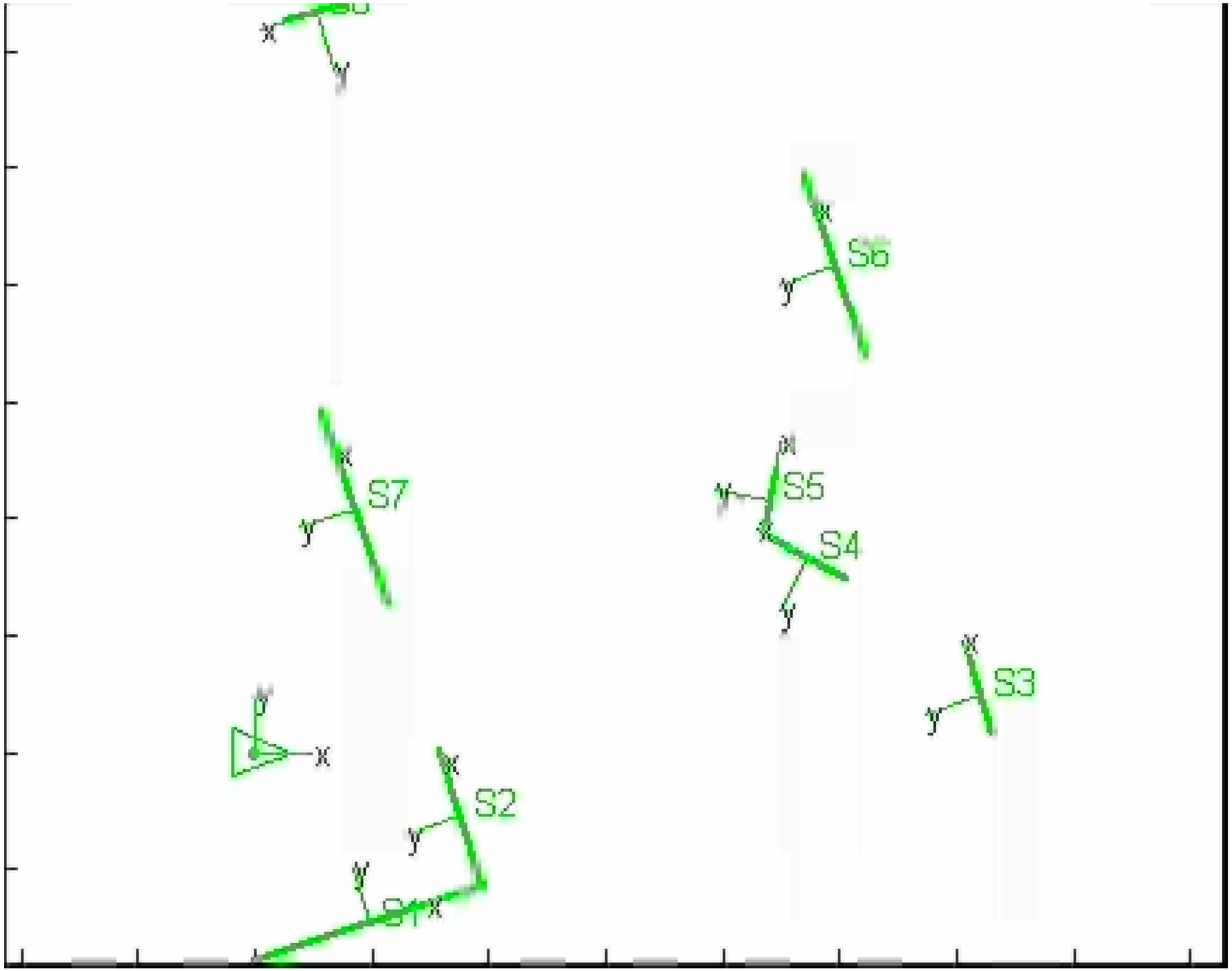


SLAM without odometry



Vehicle motion estimation

SLAM without odometry



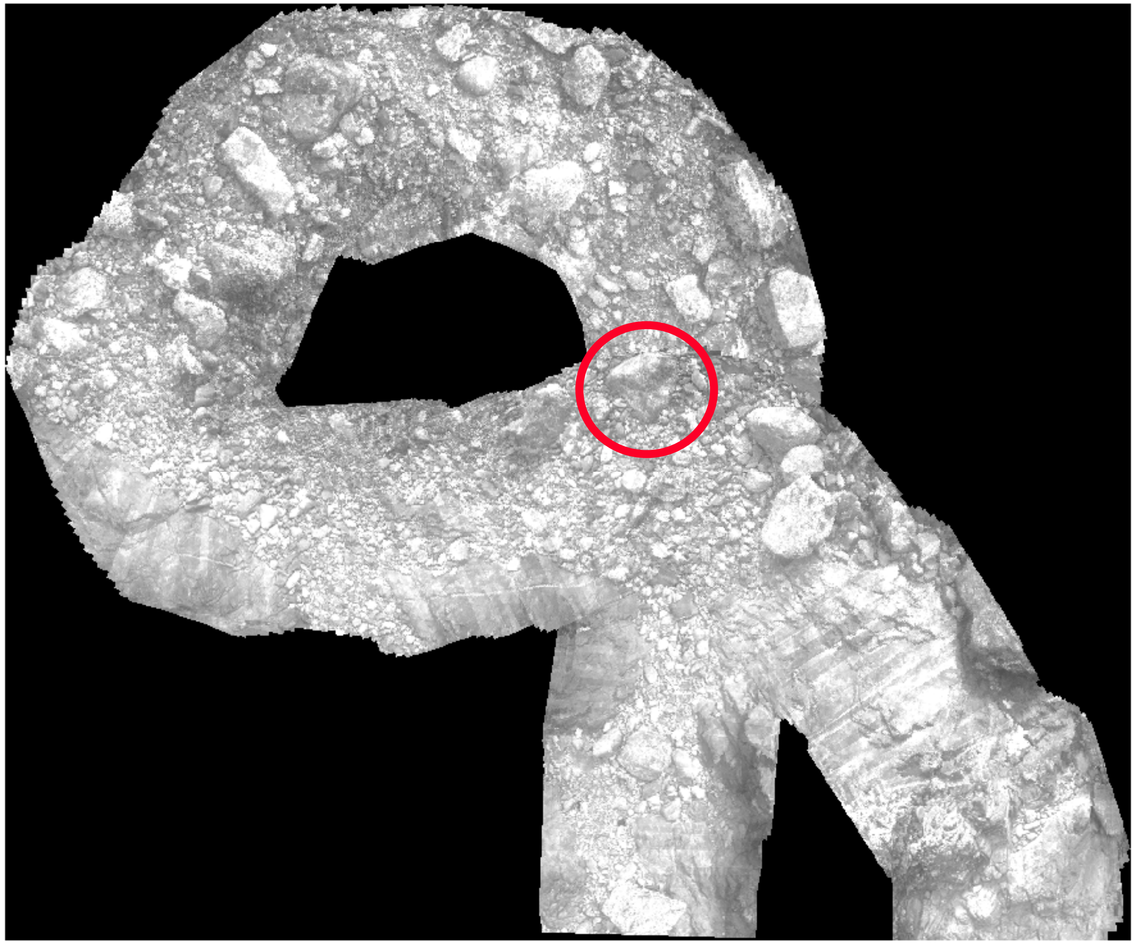
Outline

1. Basic EKF SLAM
   - Introduction: the need for SLAM
   - The basic EKF SLAM algorithm
   - Feature Extraction
   - Continuous Data Association
   - **The Loop Closing Problem**

2. Advanced EKF SLAM
   - Computational complexity of EKF SLAM
   - Consistency of the EKF SLAM
   - SLAM using local maps
     - Sequential Map Joining
     - Divide and Conquer SLAM
     - Hierarchical SLAM

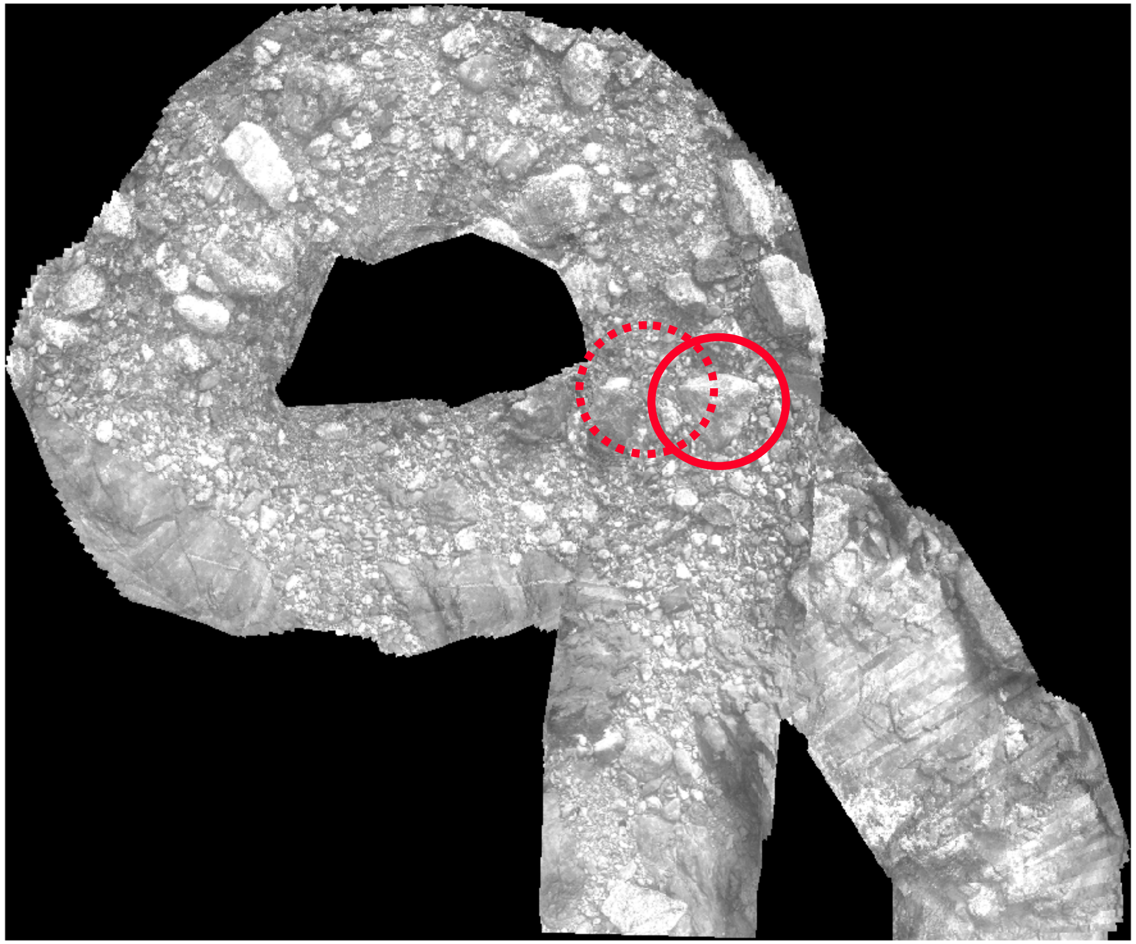
ROBOTICS UNIVERSIDAD DE ZARAGOZA

Loop closing in mosaicing
use first



Joint work with R. García, University of Girona

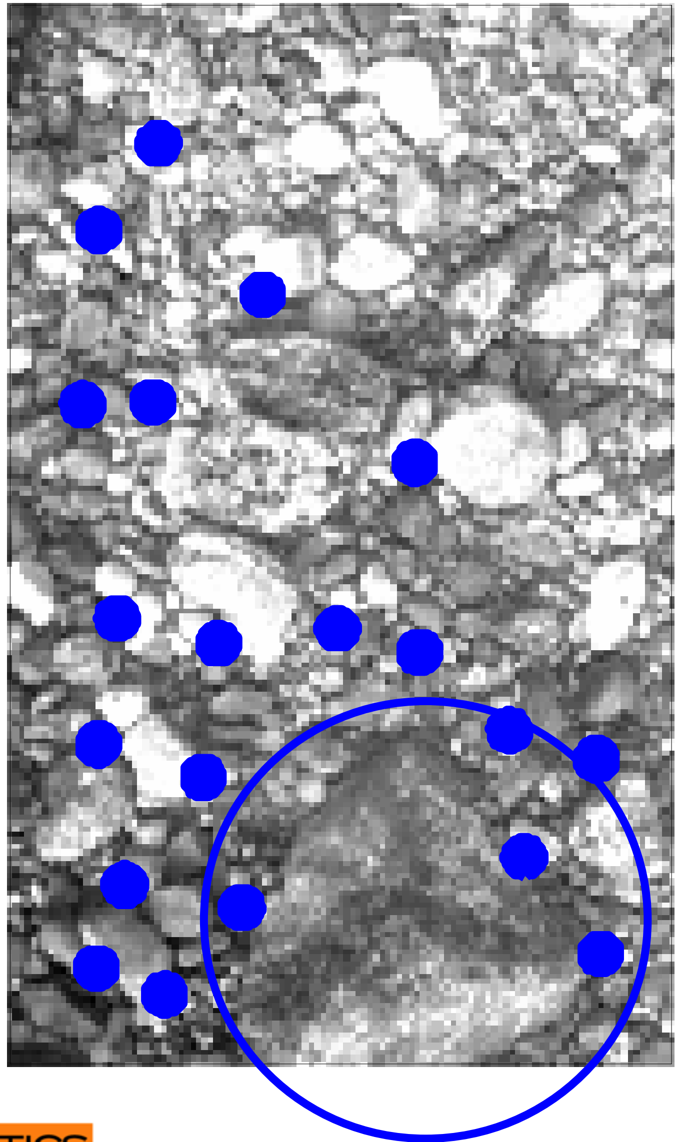
Loop closing in mosaicing: use Last



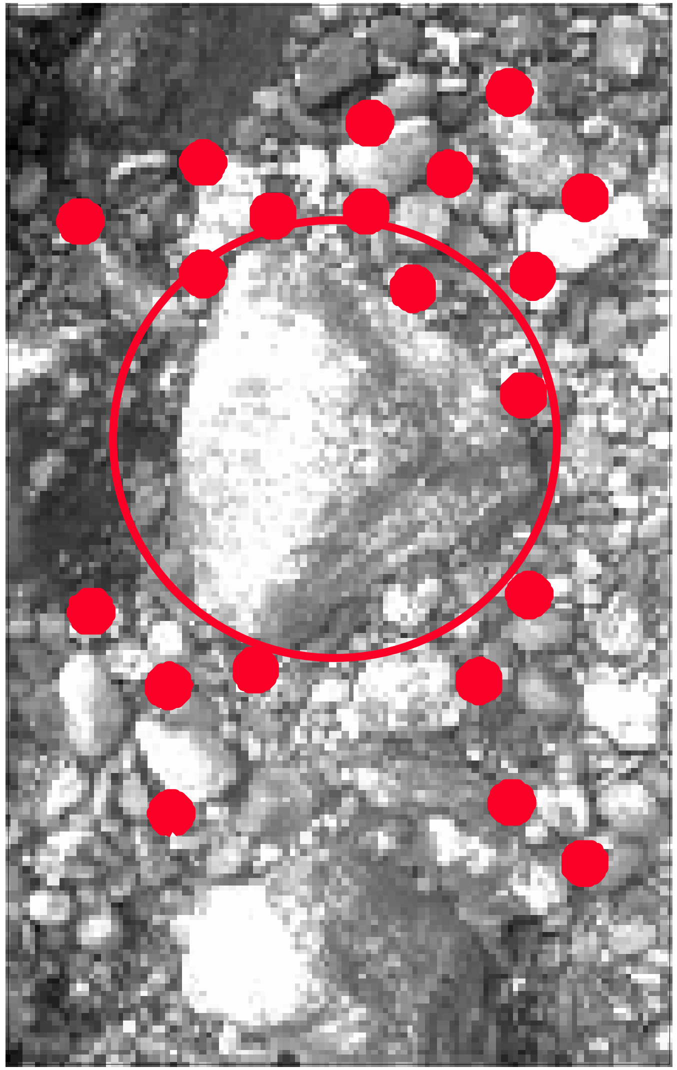
Sequential mosaicing is a form of odometry

The loop closing problem

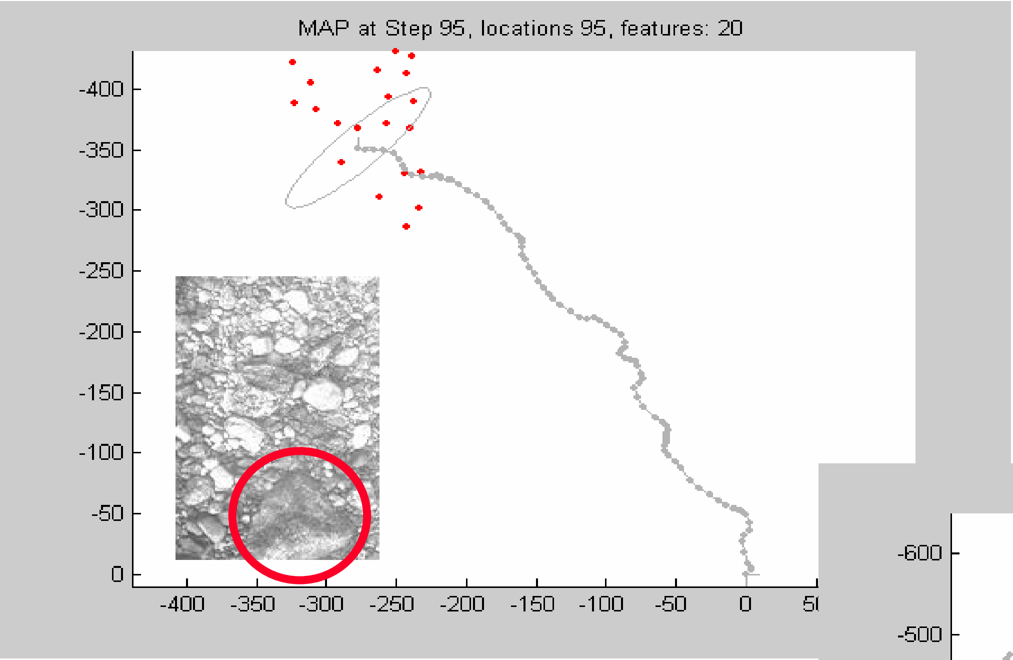
- Loop beginning

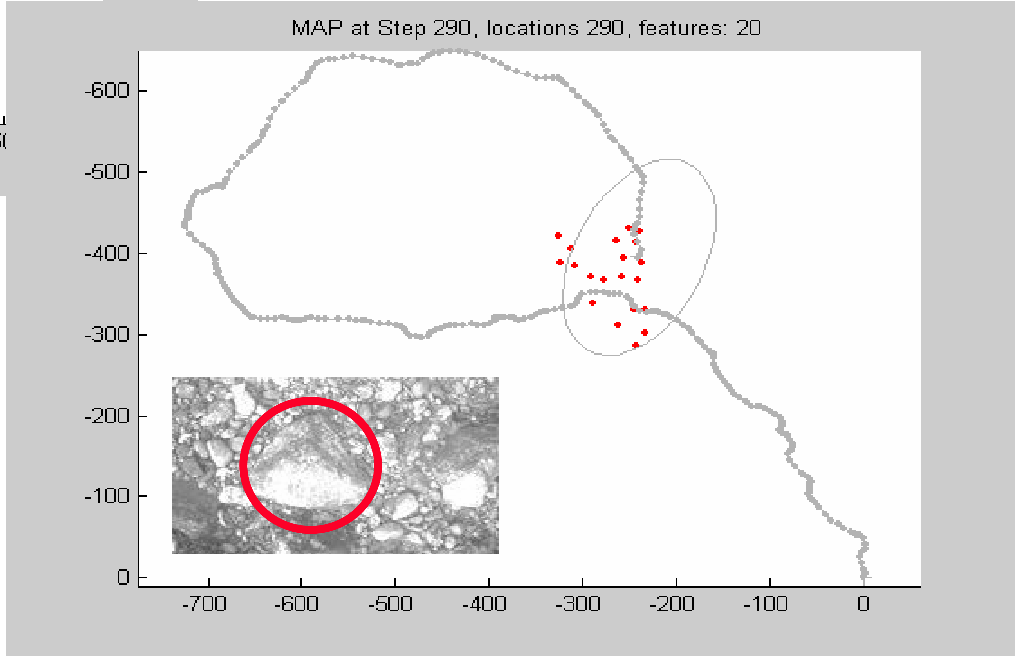


- Loop end

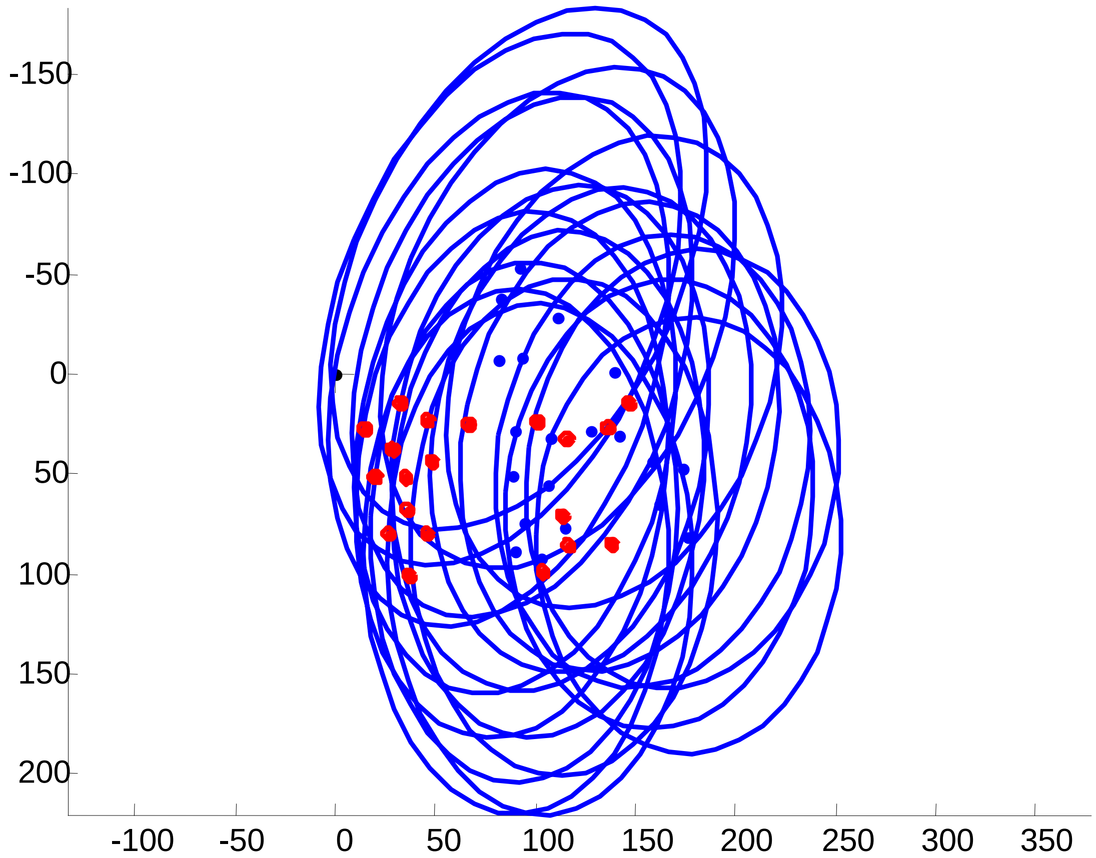


The loop closing problem





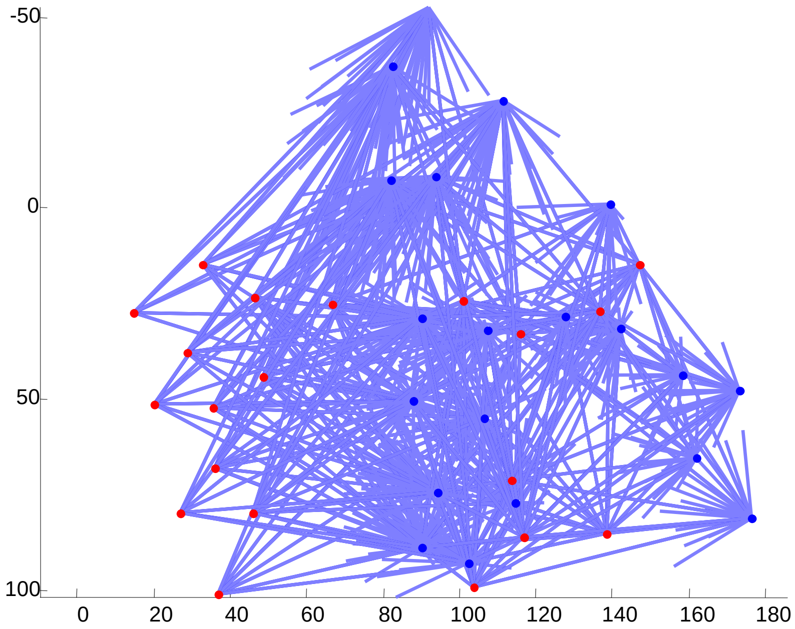
The loop closing problem



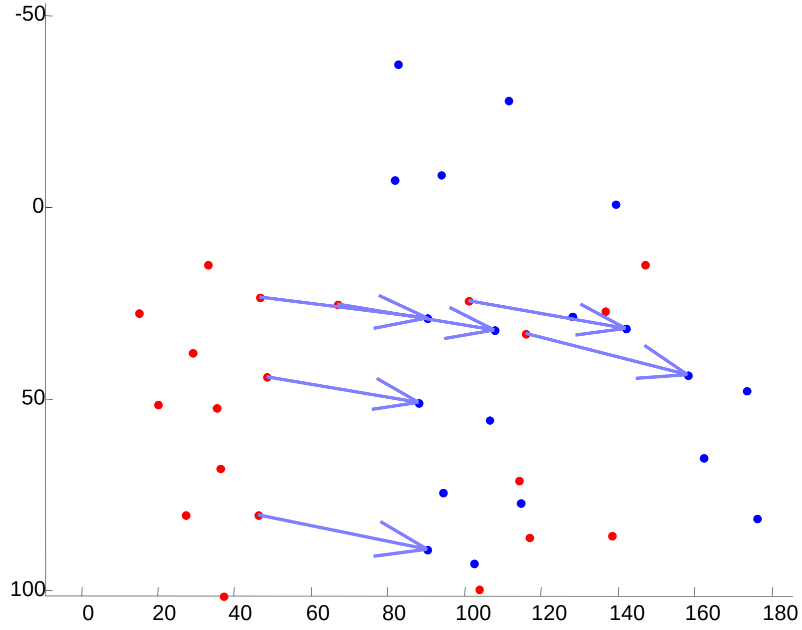
Measurements (red) and predicted features (blue)

The loop closing problem

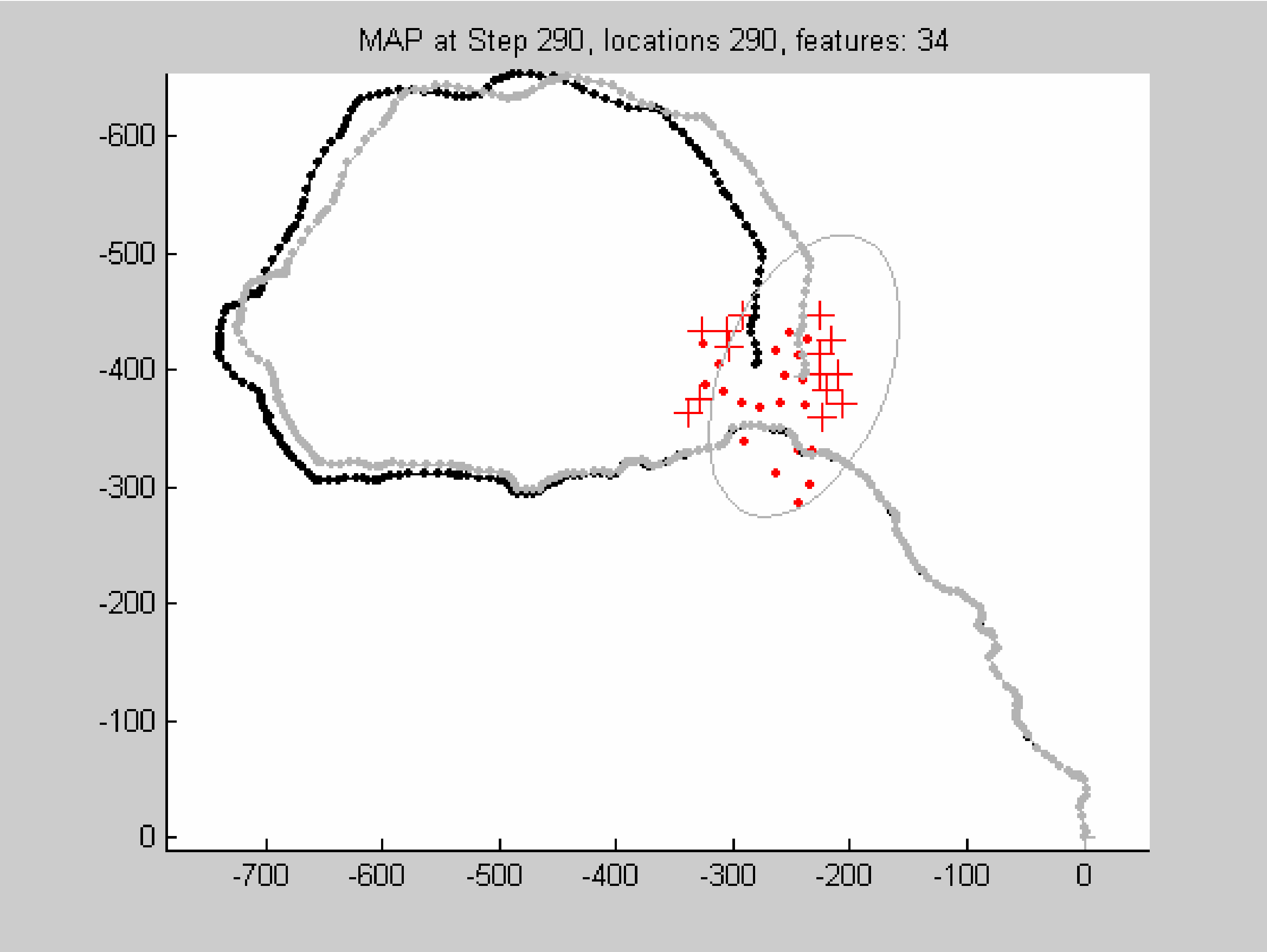
- Individual compatibility



- Joint Compatibility



The loop closing problem



Outline

1. Basic EKF SLAM
   - Introduction: the need for SLAM
   - The basic EKF SLAM algorithm
   - Feature Extraction
   - Continuous Data Association
   - The Loop Closing Problem

2. Advanced EKF SLAM
   - **Computational complexity of EKF SLAM**
   - Consistency of the EKF SLAM
   - SLAM using local maps
     - Sequential Map Joining
     - Divide and Conquer SLAM
     - Hierarchical SLAM

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

The EKF SLAM algorithm

Algorithm 1 SLAM:

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

$[\mathbf{z}_0, \mathbf{R}_0] = \text{get\_measurements}$

$[\mathbf{x}_0^B, \mathbf{P}_0^B] = \text{add\_new\_features}(\mathbf{x}_0^B, \mathbf{P}_0^B, \mathbf{z}_0, \mathbf{R}_0)$

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

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

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

$[\mathbf{z}_k, \mathbf{R}_k] = \text{get\_measurements}$

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

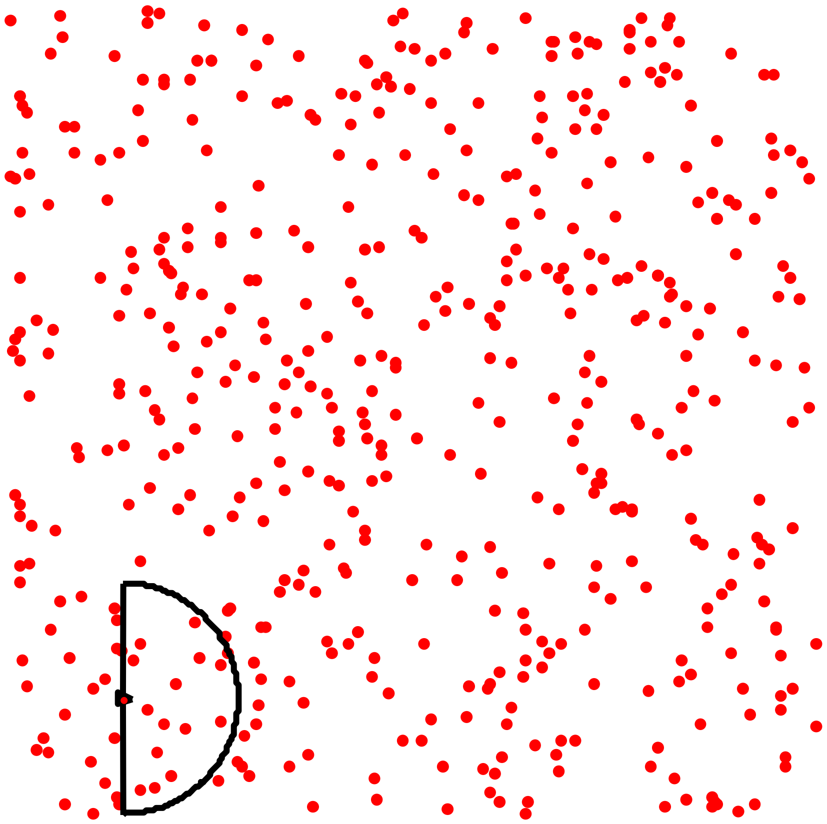
$[\mathbf{x}_k^B, \mathbf{P}_k^B] = \text{EKF\_update}(\mathbf{x}_{k|k-1}^B, \mathbf{P}_{k|k-1}^B, \mathbf{z}_k, \mathbf{R}_k, \mathcal{H}_k)$

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

end for

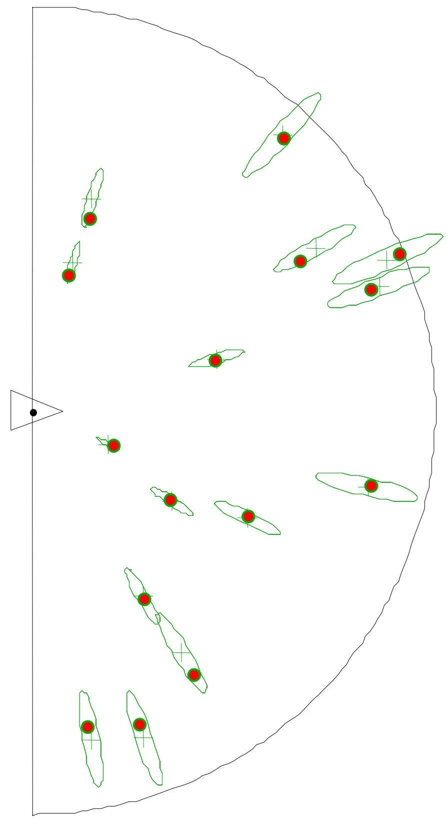
Without loss of generality...

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



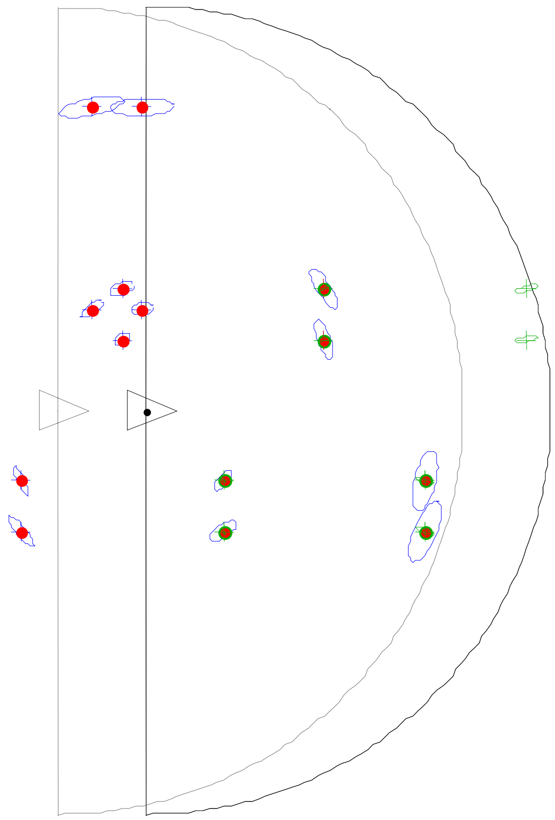
Without loss of generality...

- 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 prediction step

EKF SLAM prediction

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

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

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF SLAM: prediction

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

O(n)

O(1)

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

EKF prediction is *O(n)*

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Adding new features

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:

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

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF SLAM: add new features

$$\mathbf{P}_k = \begin{pmatrix} \mathbf{P}_R & \mathbf{P}_{RF_1} & \cdots & \mathbf{P}_{RF_n} \\ \mathbf{P}_{RF_1}^T & \mathbf{P}_{F_1} & \cdots & \mathbf{P}_{F_1F_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{RF_n}^T & \mathbf{P}_{F_1F_n}^T & \cdots & \mathbf{P}_{F_n} \end{pmatrix}$$

O(n)

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

O(s)

Adding new features is *O(n)*

The update step

EKF SLAM: map update

m **observations:**

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}^B_{\mathcal{F}_k}) + \mathbf{w}_k$$

$$\mathbf{z}_k \simeq \mathbf{h}_k(\hat{\mathbf{x}}^B_{\mathcal{F}_{k|k-1}}) + \mathbf{H}_k(\mathbf{x}^B_{\mathcal{F}_k} - \hat{\mathbf{x}}^B_{\mathcal{F}_{k|k-1}})$$

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}^B_{\mathcal{F}_k}}\right|_{(\hat{\mathbf{x}}^B_{\mathcal{F}_{k|k-1}})}$$

Innovation Matrix: $\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k$

Filter gain: $\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{S}_k)^{-1}$

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

Covariance update: $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_{k|k-1}$

The innovation matrix

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

rxr — *rxn* — *nxn* — *nxr* — *rxr*

rxn

rxr

rxr

$rn^2 + r^2n + r^2$

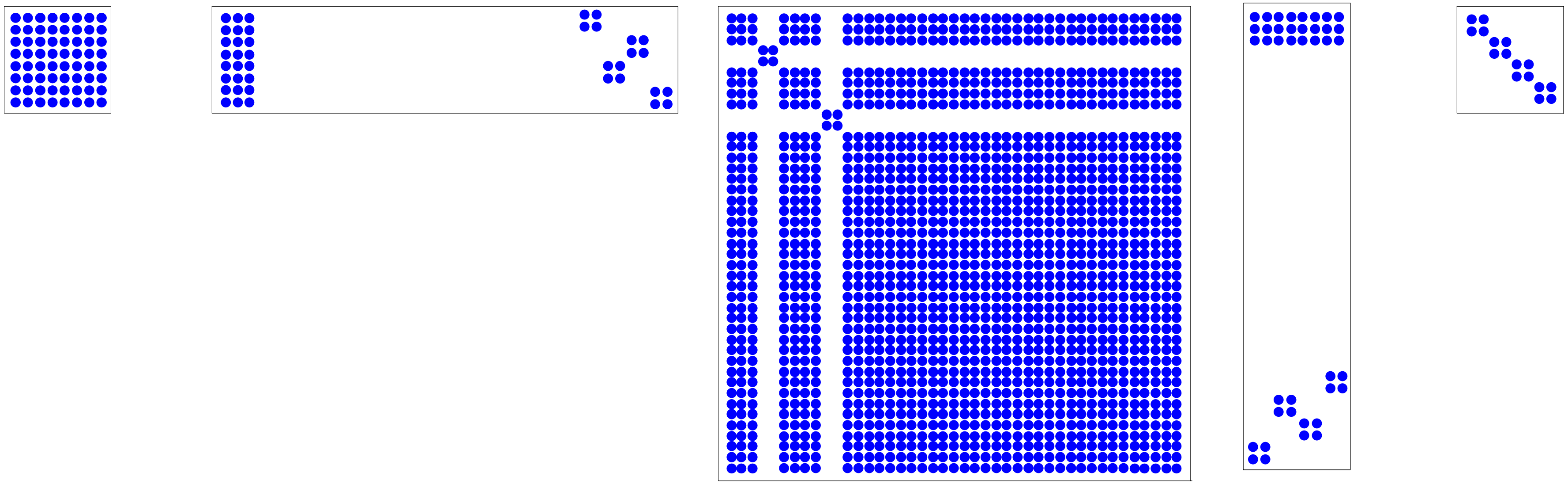
$O(n^2)$ **?**

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

The innovation matrix

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

rxr *rxc* *nxn* *cxr* *rxr*

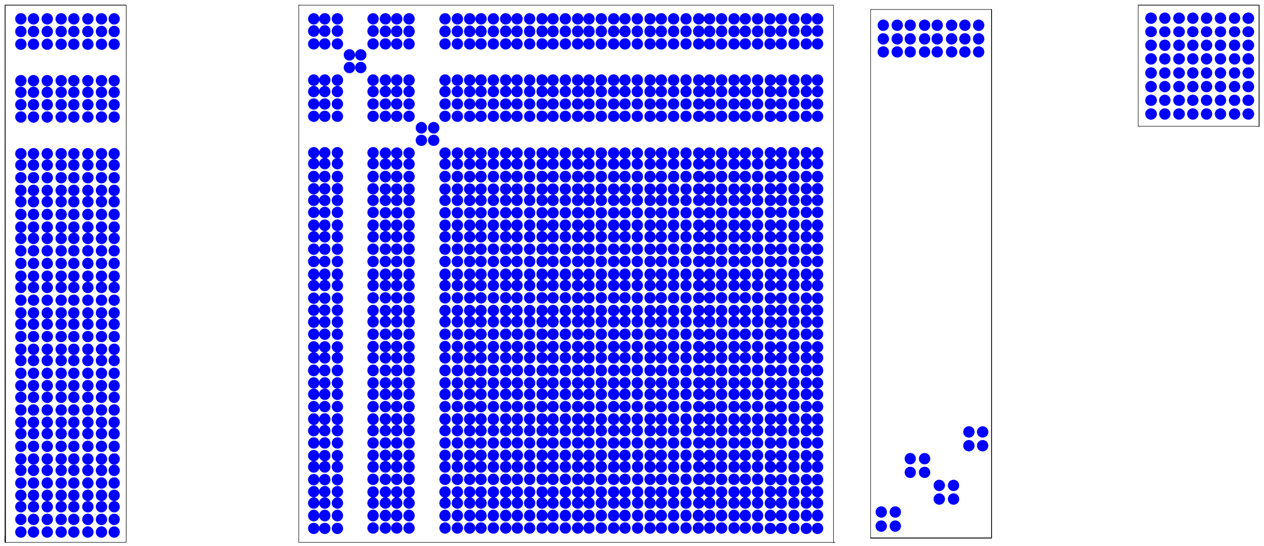


O(rn)

The Kalman gain matrix

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

nxr *nxn* *nxc* *rxr*



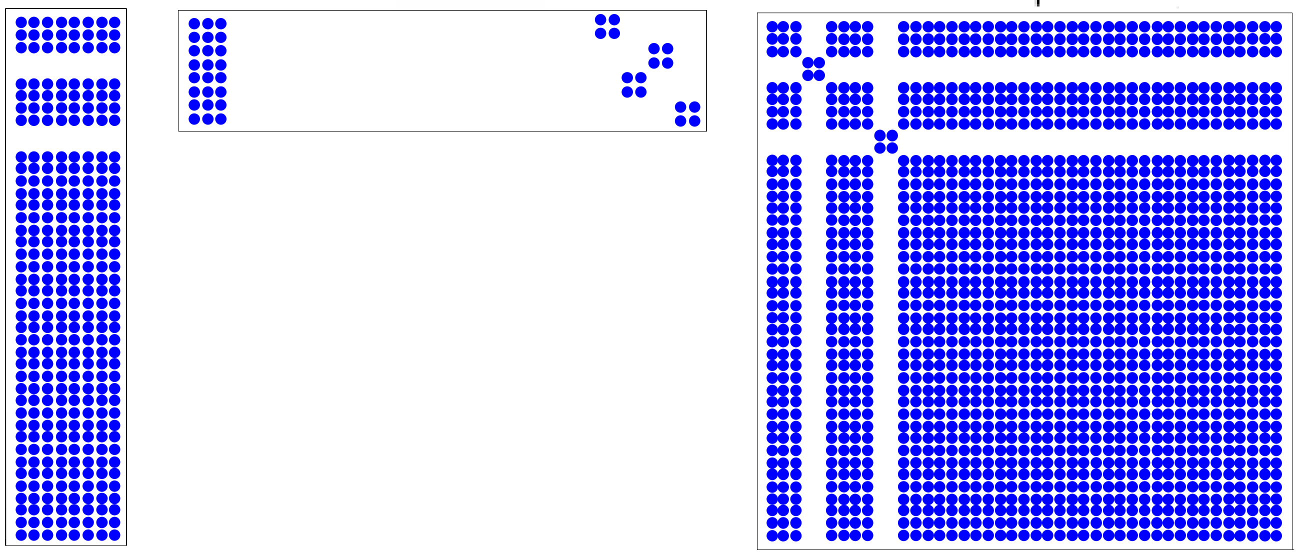
$O(r^2 n)$

The covariance matrix

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

..... = ... $\mathbf{K}_k$ $\mathbf{H}_k$ $\mathbf{P}_{k|k-1}$

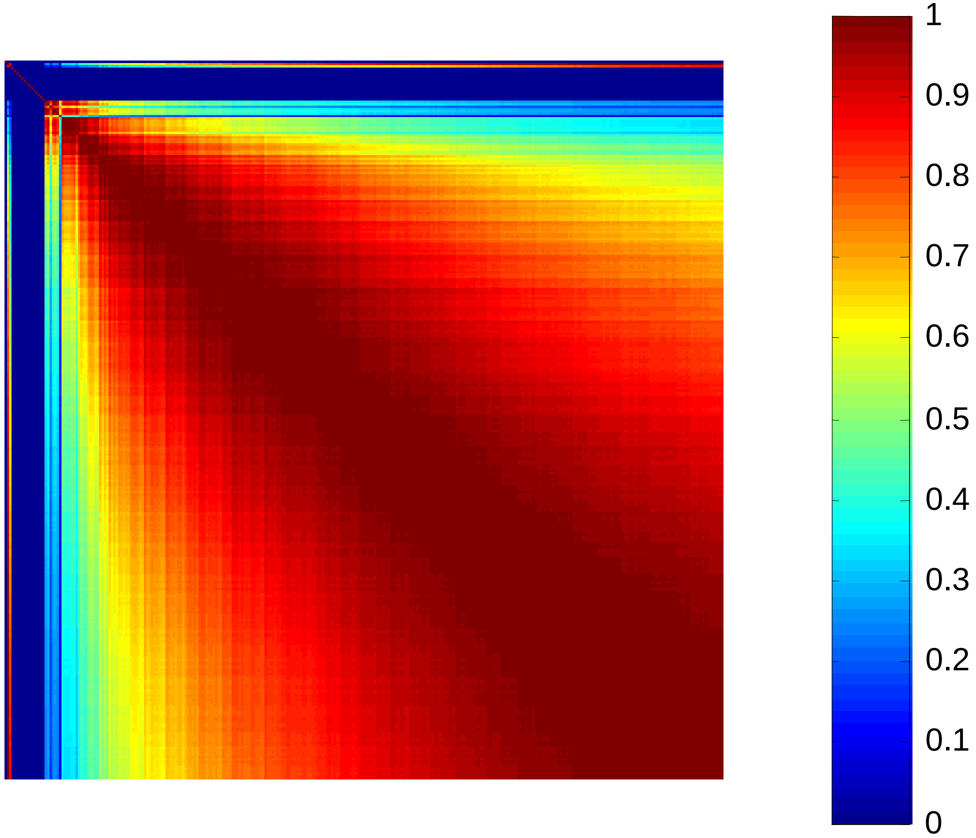
....



nxr *cxn* *nxn*

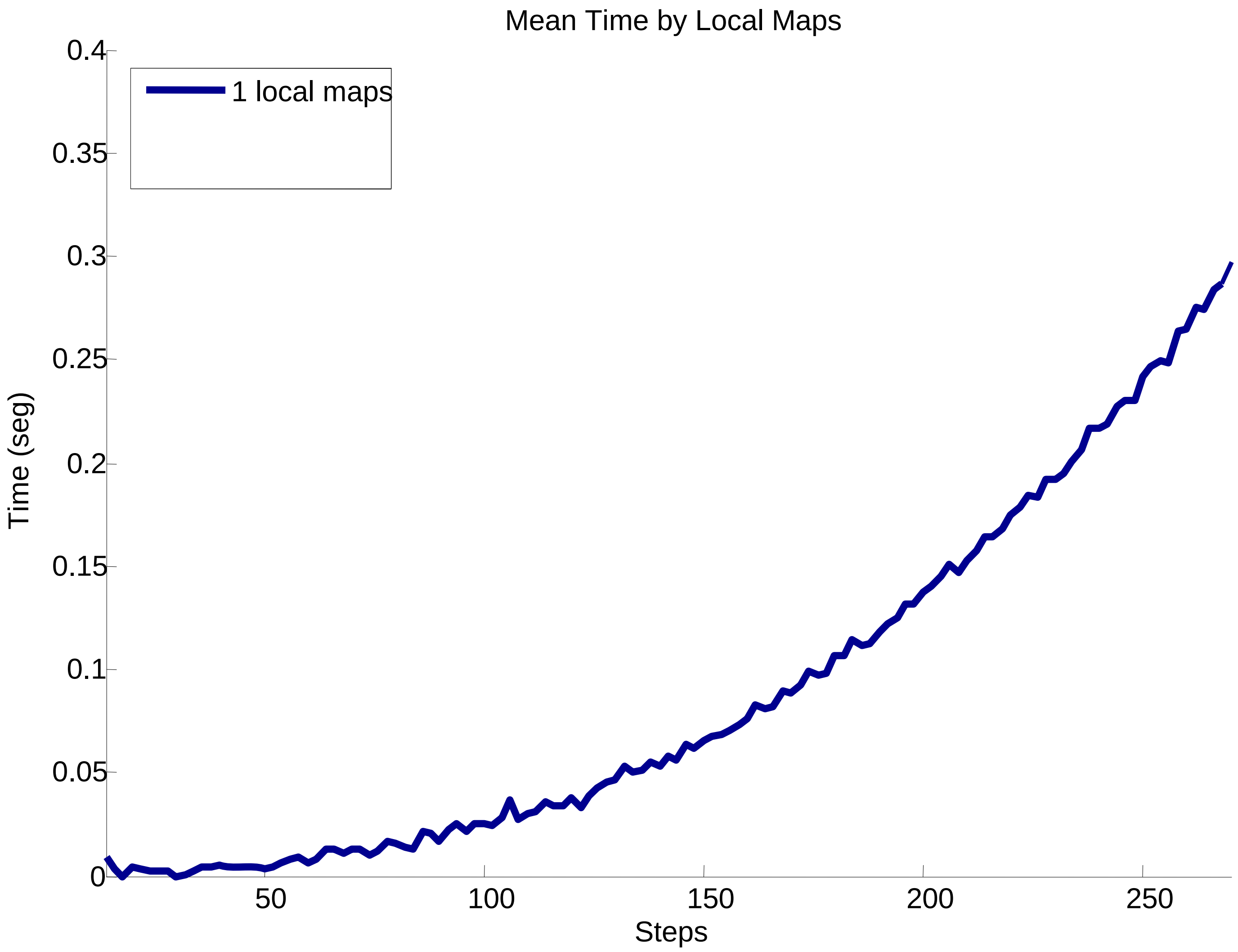
EKF update step is *O(n²)*

The mixed blessing of Covariance



- Covariance provides data association
- But the covariance matrix is full

EKF-SLAM updates are $O(n^2)$

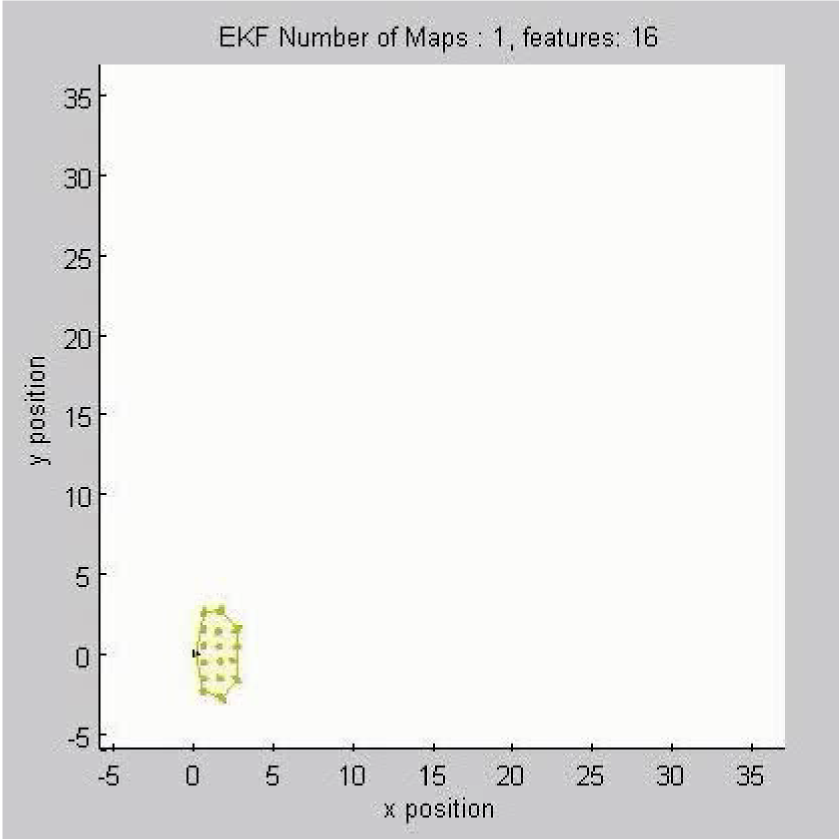


Computational Cost of EKF SLAM

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

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

$O(n^2)$



ROBOTICS UNIVERSIDAD DE ZARAGOZA

Outline

1. Basic EKF SLAM
   - Introduction: the need for SLAM
   - The basic EKF SLAM algorithm
   - Feature Extraction
   - Continuous Data Association
   - The Loop Closing Problem

2. Advanced EKF SLAM
   - Computational complexity of EKF SLAM
   - **Consistency of the EKF SLAM**
   - SLAM using local maps
     - Sequential Map Joining
     - Divide and Conquer SLAM
     - Hierarchical SLAM

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Consistency of EKF-SLAM

- Nice **"convergence"** properties of $\mathbf{P}_k^W$ (Dissanayake et al. 2001):
  - Landmark covariance decreases monotonically
  - In the limit, landmarks become fully correlated
  - In the limit, landmark covariance reaches a lower bound related to the initial vehicle covariance

- But SLAM is a non-linear problem
  - The inherent approximations due to linearizations can lead to **divergence (inconsistency)** of the EKF
    - » see for example (Jazwinski, 1970)

ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: 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:

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

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

Linearization

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

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: Map Update

Feature observations:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k^B) + \mathbf{w}_k$$

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

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k^B}\right|_{(\hat{\mathbf{x}}_{k|k-1}^B)}$$

Linearization

EKF map update:

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

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

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Consistency Testing

1. Normalized Estimation Error Squared NEES

$$D^2 = \left(\mathbf{x}_k^W - \hat{\mathbf{x}}_k^W\right)^T \left(\mathbf{P}_k^W\right)^{-1} \left(\mathbf{x}_k^W - \hat{\mathbf{x}}_k^W\right)$$

$$D^2 \leq \chi^2_{r,1-\alpha}$$

True map required → Simulations

2. Innovation test (observation *i* → map feature *j*)

$$D^2_{ij} = \left(\mathbf{z}_i - \mathbf{h}_j(\hat{\mathbf{x}}_k^W)\right)^T \left(\mathbf{H}_j \mathbf{P}_k^W \mathbf{H}_j^T + \mathbf{R}_i\right)^{-1} \left(\mathbf{z}_i - \mathbf{h}_j(\hat{\mathbf{x}}_k^W)\right)$$

$$D^2_{ij} \leq \chi^2_{d,1-\alpha}$$

Critical when closing big loops

ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: Simulation

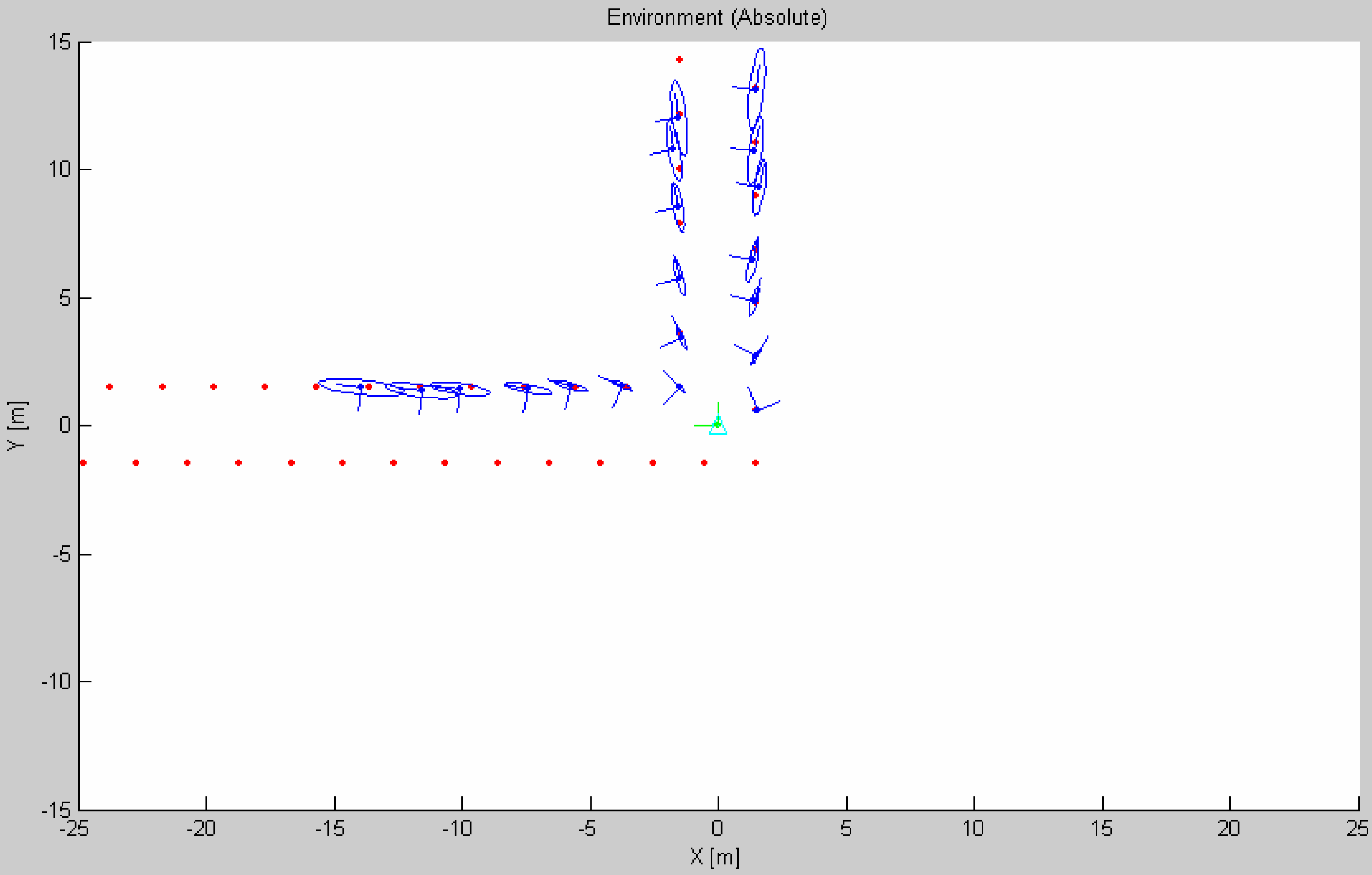
- Simulation conditions
  - Perfect data association
  - Ideal odometry and measurement noise
    - » white, Gaussian, known covariance

- Advantages of simulation:
  - Consistency can be tested against the true map
  - A simulation with noise=0 gives the theoretical map covariance (without linearization errors)

ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: Simulation

Known data association and noise model

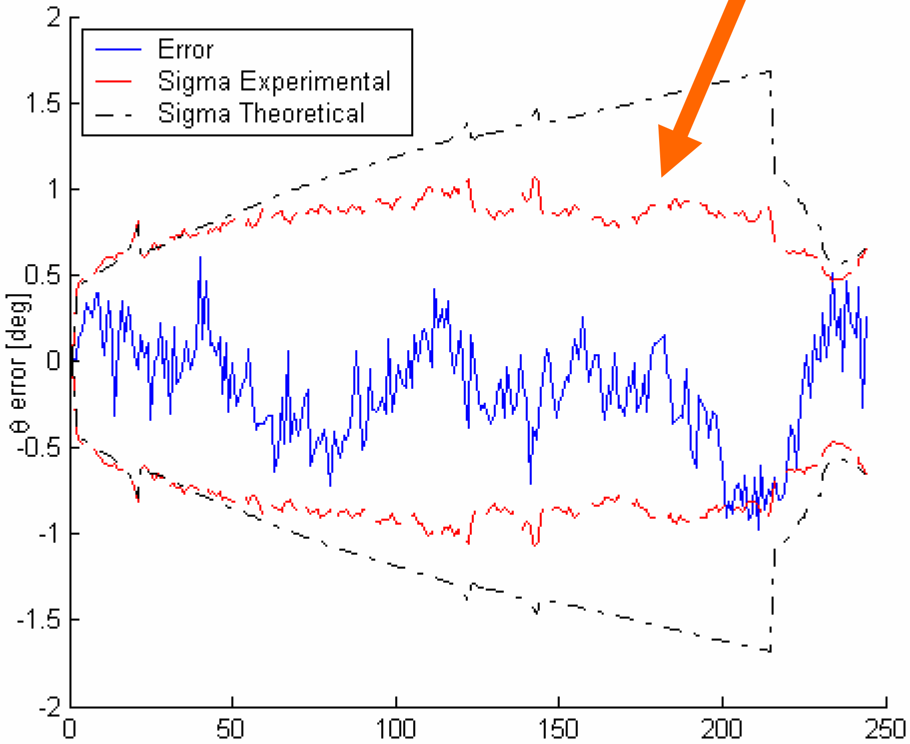


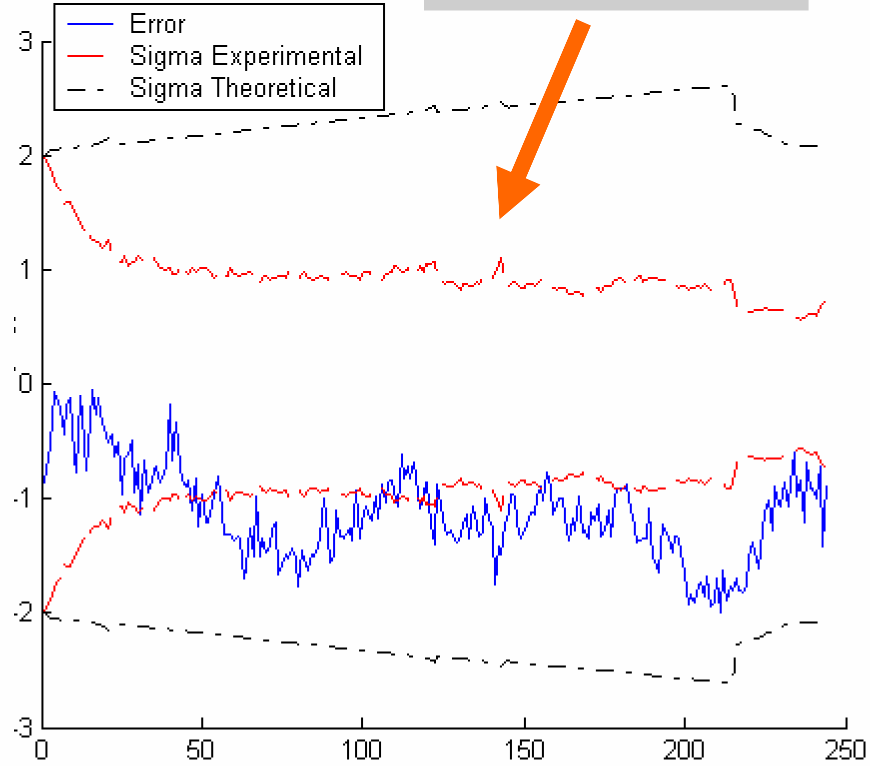
ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: Covariance

Optimistic

Violates lower bound





Initial uncertainty = 0

Initial uncertainty > 0

J.A. Castellanos, J. Neira, J.D. Tardós, **Limits to the Consistency of EKF-based SLAM**, 5th IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon, July 2004

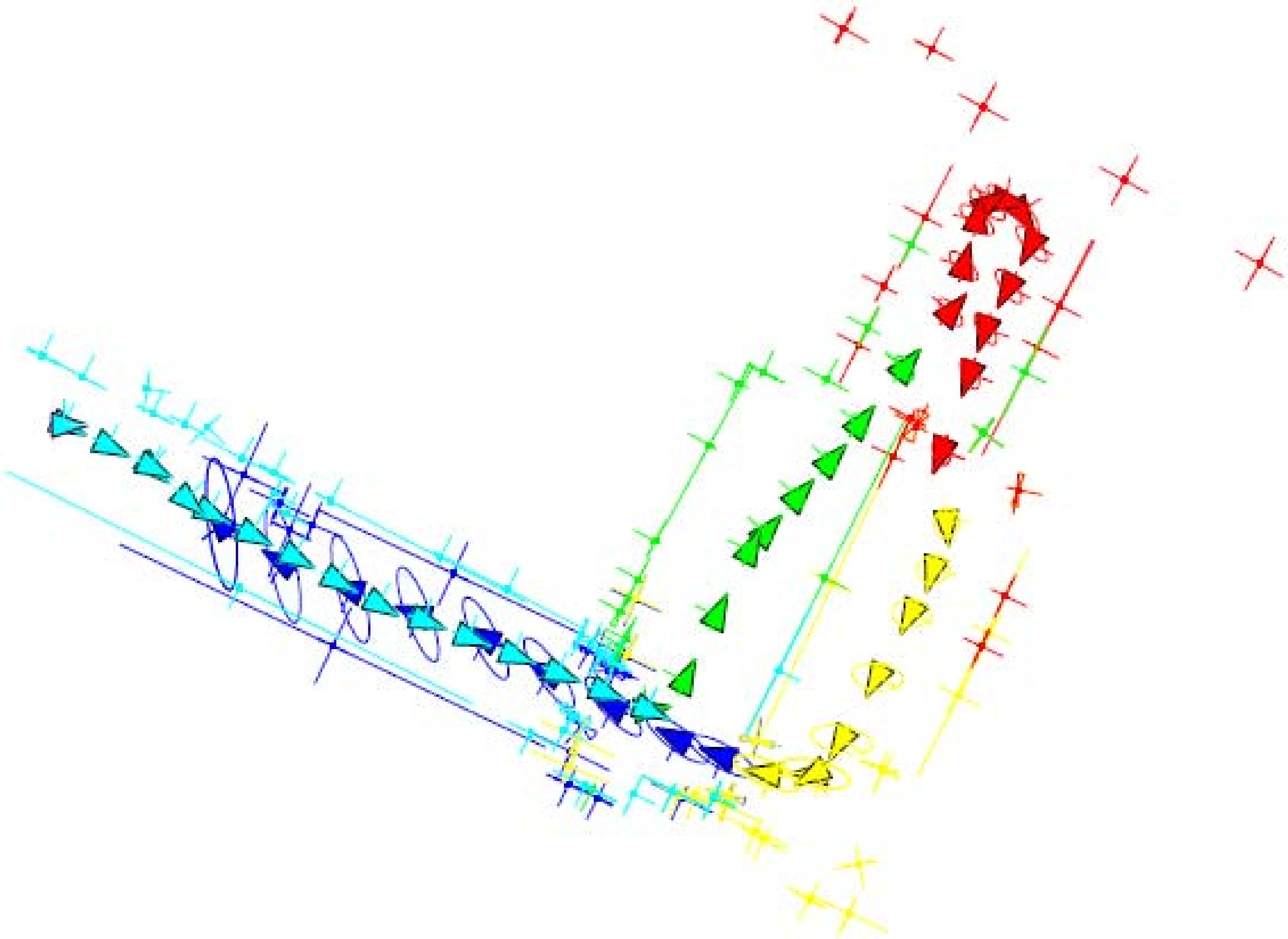
Outline

1. Basic EKF SLAM
   - Introduction: the need for SLAM
   - The basic EKF SLAM algorithm
   - Feature Extraction
   - Continuous Data Association
   - The Loop Closing Problem

2. Advanced EKF SLAM
   - Computational complexity of EKF SLAM
   - Consistency of the EKF SLAM
   - SLAM using local maps
     - **Sequential Map Joining**
     - Divide and Conquer SLAM
     - Hierarchical SLAM

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Local submaps



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Local map building

- Periodically, the robot starts a new map, relative to its current location:

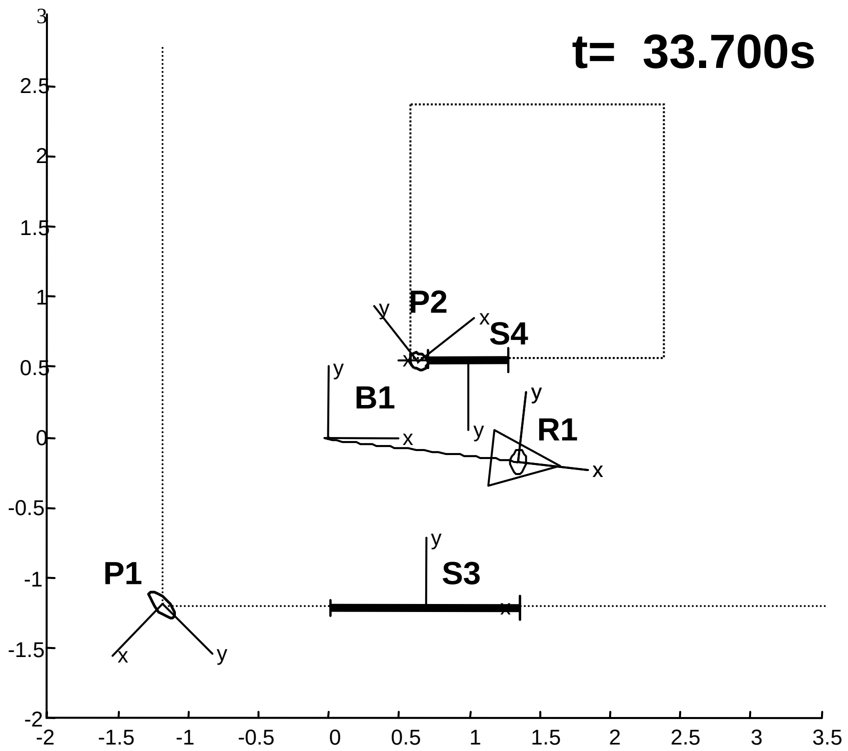
$$\hat{\mathbf{x}}^{B}_{R_0} = 0$$
$$\mathbf{P}^{B}_{R_0} = 0$$

- Given measurements:

$$D^{1\ldots k_1} = \left\{\mathbf{u}_1\,\mathbf{z}_1 \ldots \mathbf{u}_{k_1}\,\mathbf{z}_{k_1}\right\}$$
$$\mathbf{u}_k = \hat{\mathbf{x}}^{R_{k-1}}_{R_k}$$

- EKF approximates the conditional mean:

$$\hat{\mathbf{x}}^{B_1}_{\mathcal{F}_1} \simeq E\left[\mathbf{x}^{B_1}_{\mathcal{F}_1} \mid D^{1\ldots k_1},\ \mathcal{H}^{1\ldots k_1}\right]$$



Local map building

- Second map: $D^{k_1+1 \ldots k_2} = \left\{ \mathbf{u}_{k_1+1} \, \mathbf{z}_{k_1+1} \ldots \mathbf{u}_{k_2} \, \mathbf{z}_{k_2} \right\}$

$$\hat{\mathbf{x}}^{B_2}_{\mathcal{F}_2} \simeq E\left[\mathbf{x}^{B_2}_{\mathcal{F}_2} \mid D^{k_1+1 \ldots k_2}, \, \mathcal{H}^{k_1+1 \ldots k_2} \right]$$

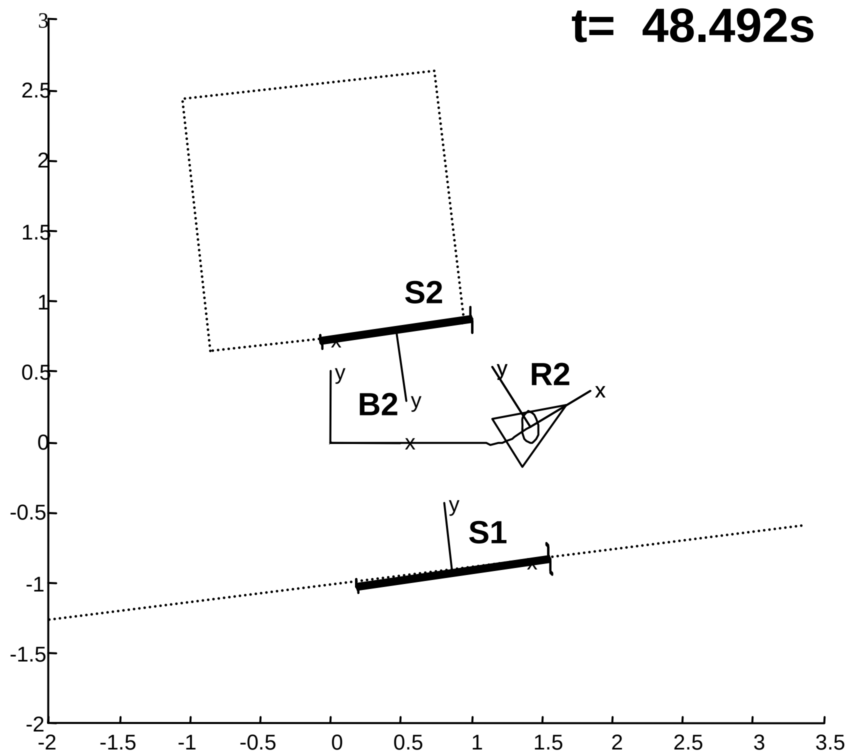
- No information is shared:

$$D^{1 \ldots k_1} \cap D^{k_1+1 \ldots k_2} = \emptyset$$

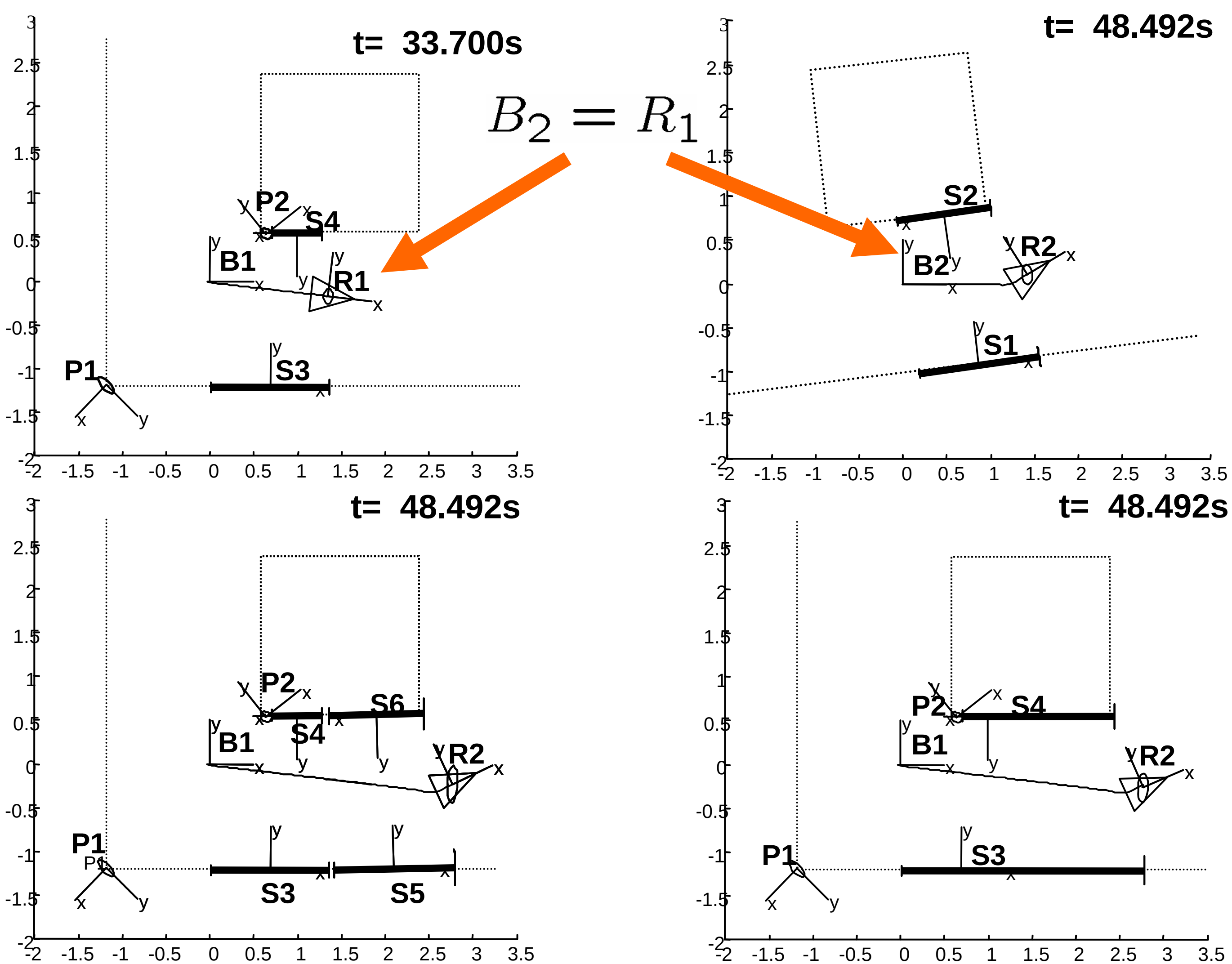
Maps are uncorrelated

- Common reference:

$$B_2 = R_1$$



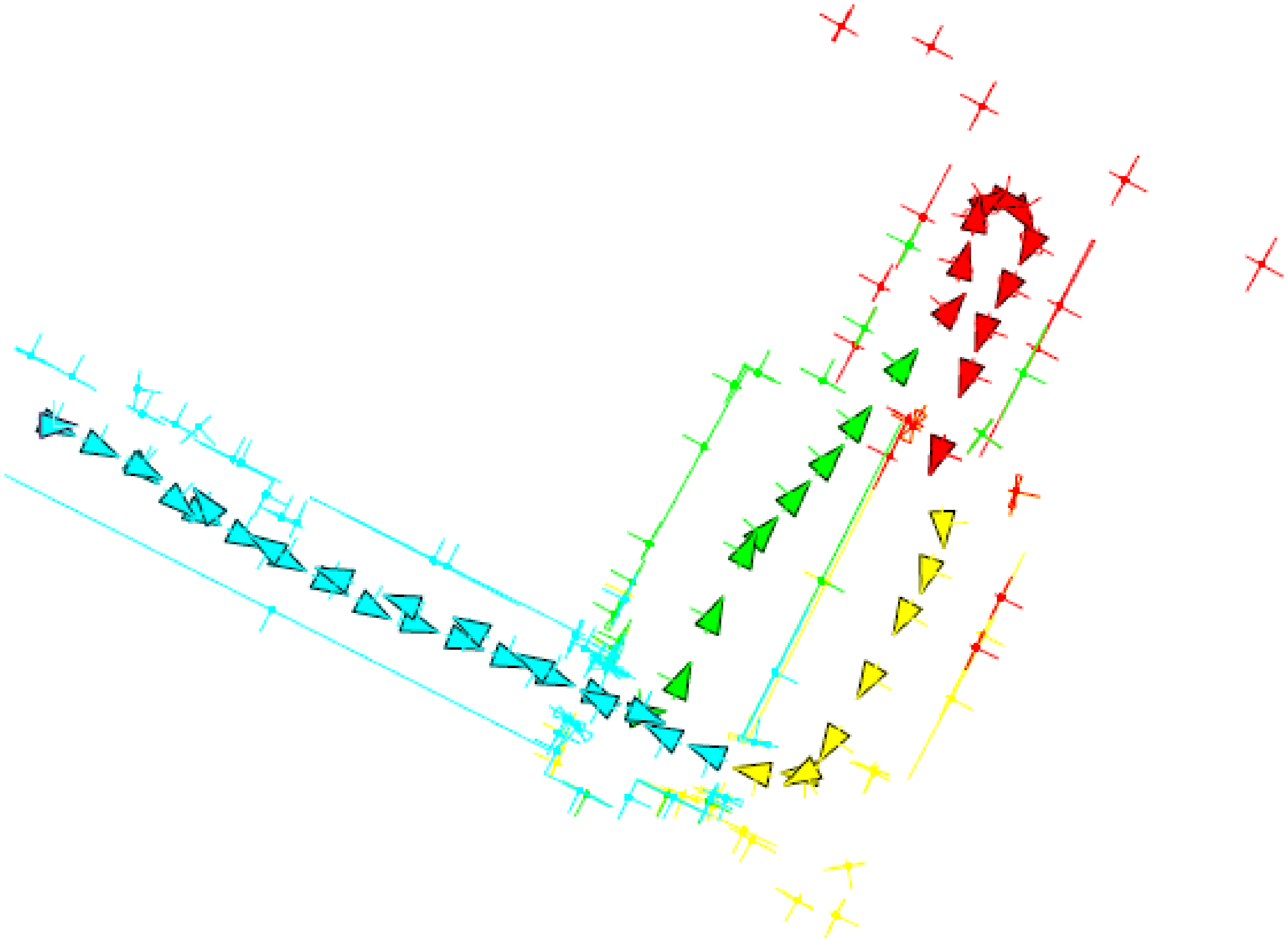
Map Joining: Example



Joined map **After matching and fusion**

J.D. Tardós, J. Neira, P.M. Newman and J.J. Leonard, Robust Mapping and Localization in Indoor Environments using Sonar Data, The Int. Journal of Robotics Research, Vol. 21, No. 4, April, 2002, pp 311 –330

Map Joining



Map Joining Step

- New state vector:

$$\hat{\mathbf{x}}_{\mathcal{A}+\mathcal{B}} = \begin{bmatrix} \hat{\mathbf{x}}_{\mathcal{A}} \\ \hat{\mathbf{x}}_{\mathcal{B}} \end{bmatrix}$$

- New covariance matrix:

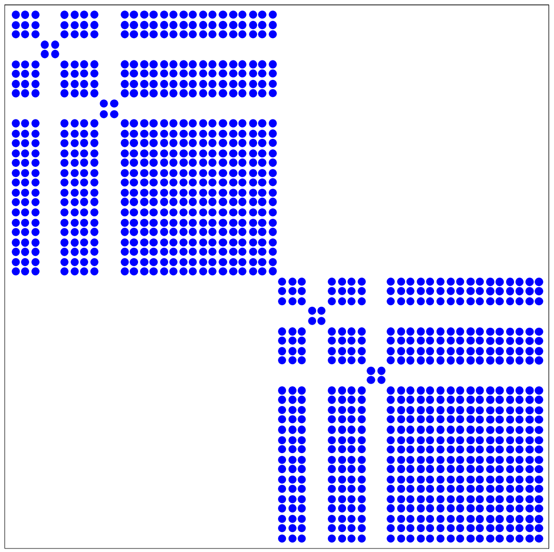
$$\mathbf{P}^{-}_{\mathcal{A}+\mathcal{B}} = \begin{bmatrix} \mathbf{P}_{\mathcal{A}} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{P}_{\mathcal{B}} \end{bmatrix}$$

$$\mathbf{P}^{-}_{\mathcal{A}+\mathcal{B}} = \quad \mathbf{P}_{\mathcal{A}}$$

nxn

n1xn1

n = n1+n2



$\mathbf{P}_{\mathcal{B}}$

n2xn2

Matching and Fusion Step

- Matching function:

$$\mathbf{f}_{ij_i}(\mathbf{x}) = \mathbf{0}$$

- Joint matching function for the hypothesis:

$$\mathbf{f}_{\mathcal{H}}(\mathbf{x}) = \begin{bmatrix} \mathbf{f}_{1j_1}(\mathbf{x}) \\ \vdots \\ \mathbf{f}_{mj_m}(\mathbf{x}) \end{bmatrix} \simeq \mathbf{h}_{\mathcal{H}} + \mathbf{H}_{\mathcal{H}}(\mathbf{x} - \hat{\mathbf{x}}) = \mathbf{0}$$

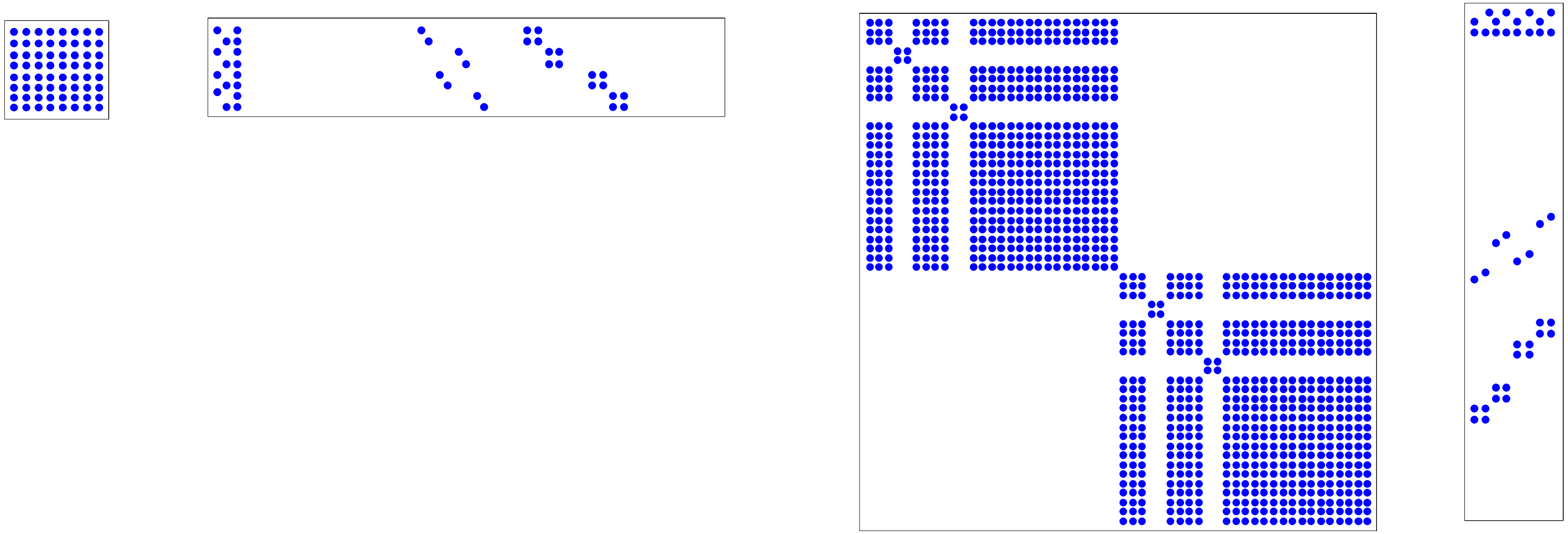
- Map update using EKF:

$$\begin{aligned} \mathbf{S}_{\mathcal{H}} &= \mathbf{H}_{\mathcal{H}} \mathbf{P}^{-}_{\mathcal{A}+\mathcal{B}} \mathbf{H}^{T}_{\mathcal{H}} \\ \mathbf{K}_{\mathcal{H}} &= \mathbf{P}^{-}_{\mathcal{A}+\mathcal{B}} \mathbf{H}^{T}_{\mathcal{H}} \left(\mathbf{S}_{\mathcal{H}}\right)^{-1} \\ \mathbf{P}_{\mathcal{A}+\mathcal{B}} &= \left(\mathbf{I} - \mathbf{K}_{\mathcal{H}} \mathbf{H}_{\mathcal{H}}\right) \mathbf{P}^{-}_{\mathcal{A}+\mathcal{B}} \end{aligned}$$

The innovation matrix

$$\mathbf{S}_{\mathcal{H}} = \mathbf{H}_{\mathcal{H}} \mathbf{P}^{-}_{\mathcal{A}+\mathcal{B}} \mathbf{H}^{T}_{\mathcal{H}}$$

rxr rxc nxn cxr

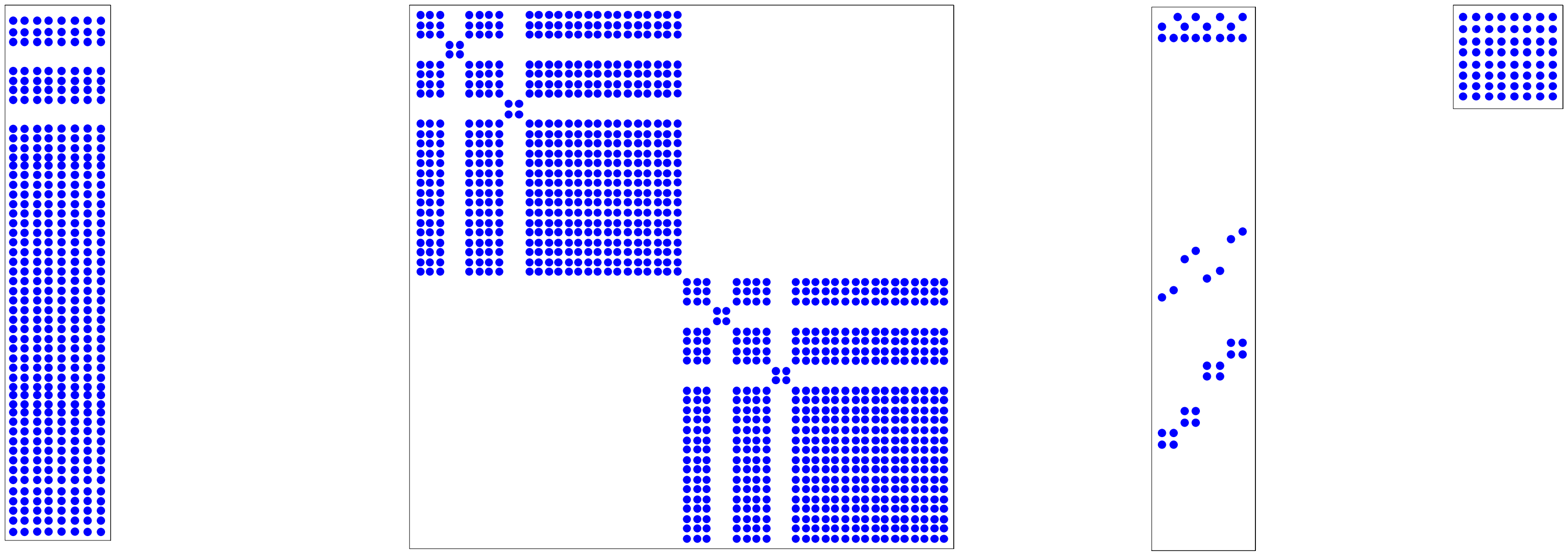


O(rn)

The gain matrix

$$\mathbf{K}_{\mathcal{H}} = \mathbf{P}^{-}_{\mathcal{A}+\mathcal{B}} \; \mathbf{H}^{T}_{\mathcal{H}} \; (\mathbf{S}_{\mathcal{H}})^{-1}$$

nxr *nxn* *cxr* *rxr*



$O(r^2 n)$

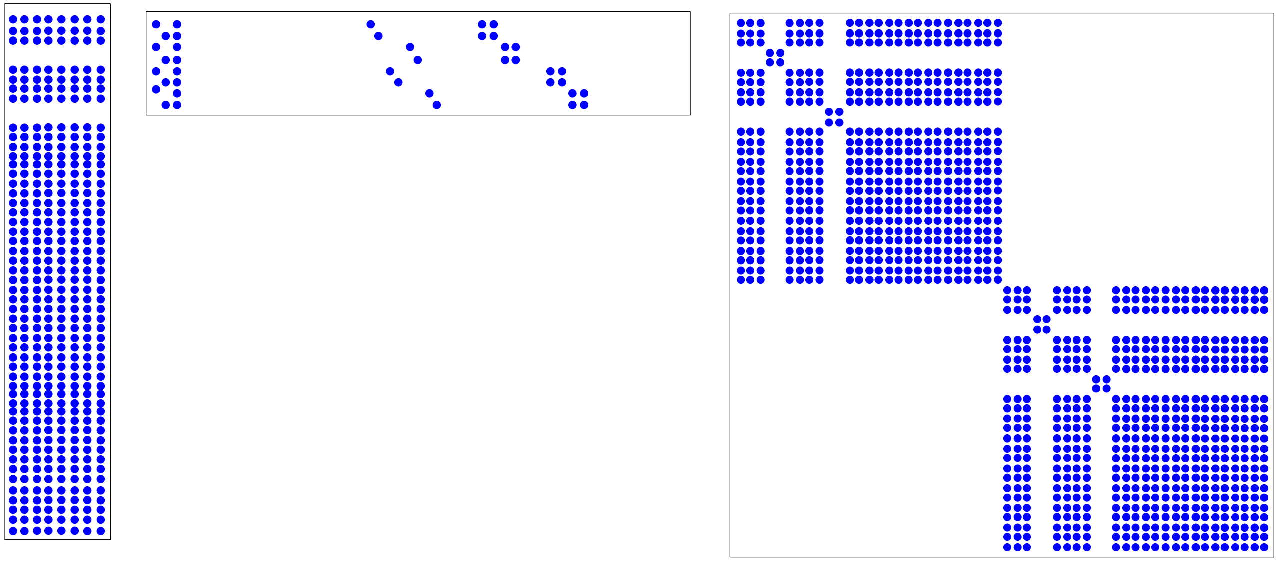
The update

$$\mathbf{P}_{\mathcal{A}+\mathcal{B}} = (\mathbf{I} - \mathbf{K}_{\mathcal{H}}\mathbf{H}_{\mathcal{H}})\,\mathbf{P}^{-}_{\mathcal{A}+\mathcal{B}}$$

$$\ldots\ldots = \ldots\ldots\ \mathbf{K}_{\mathcal{H}} \quad \mathbf{H}_{\mathcal{H}} \quad \mathbf{P}^{-}_{\mathcal{A}+\mathcal{B}}$$

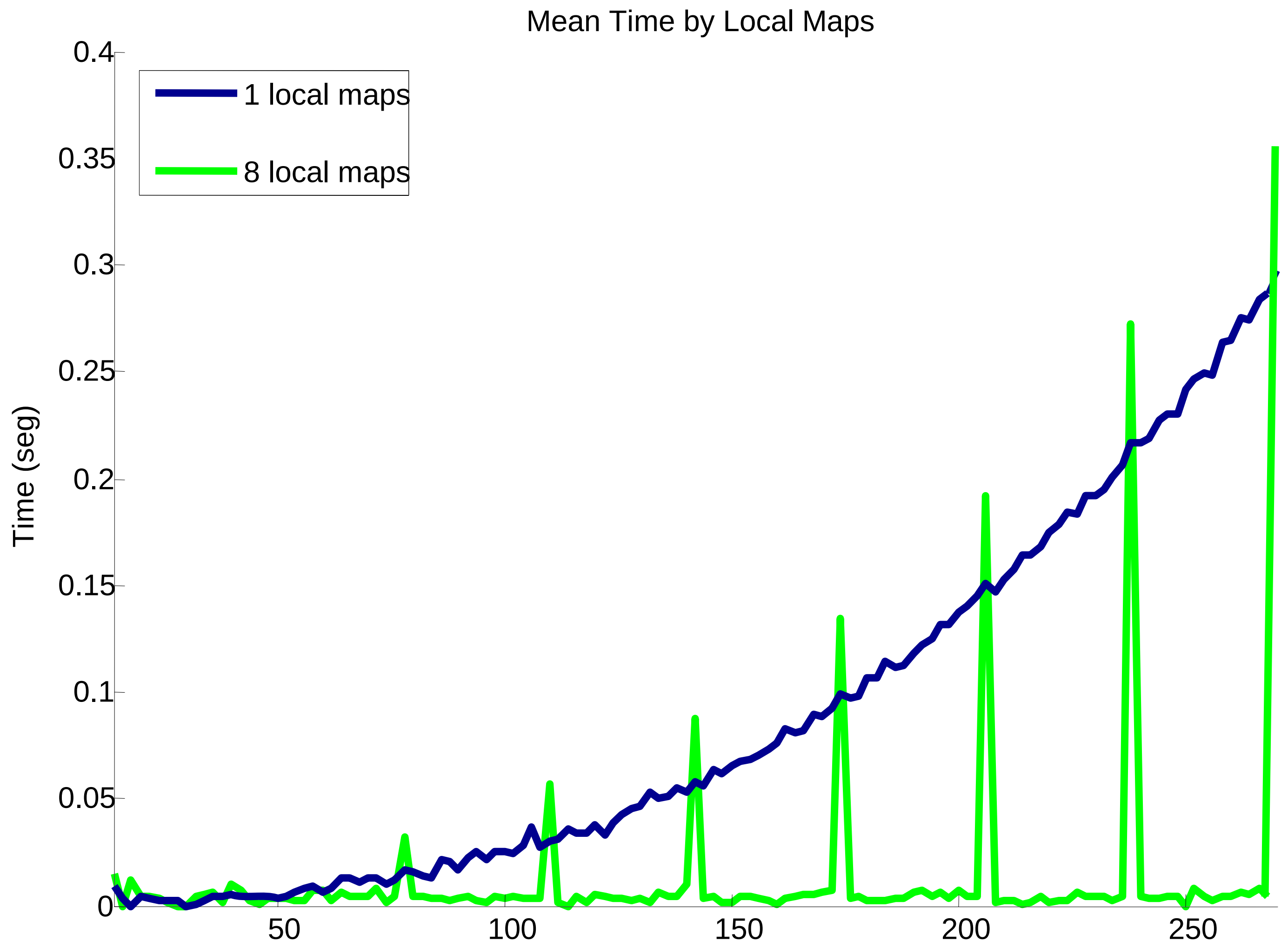
nxr *rxc* *nxn*

....

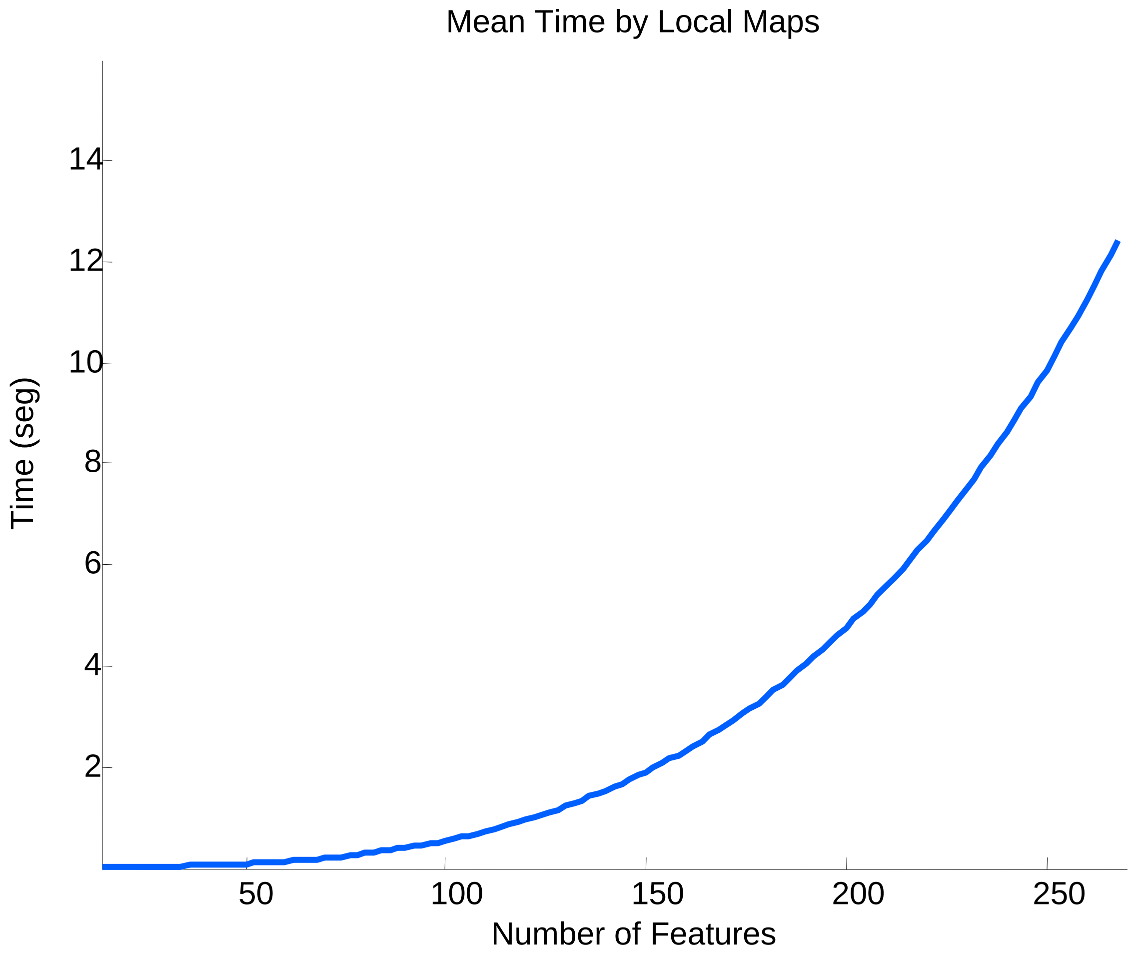


Matching and fusion is $O(n^2)$

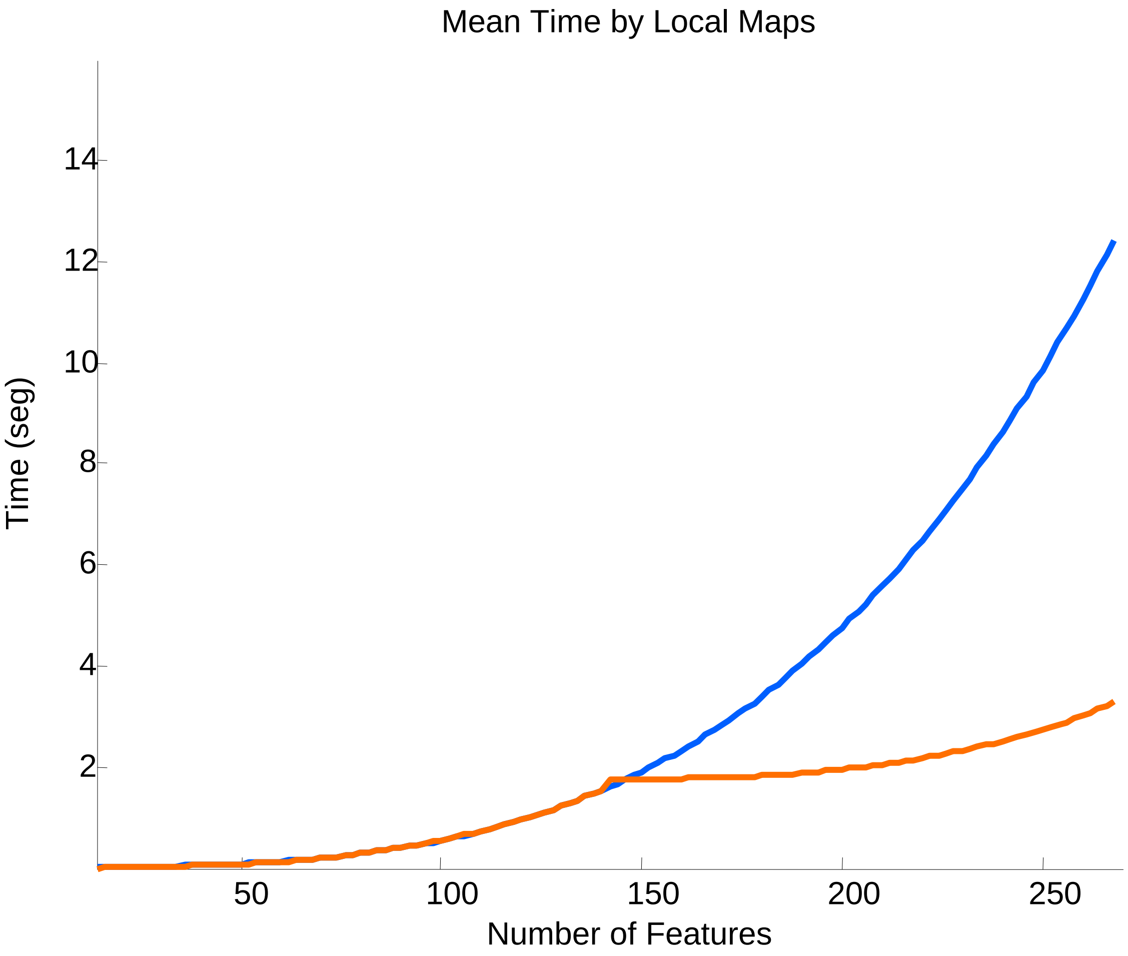
Cost of Map Joining per step



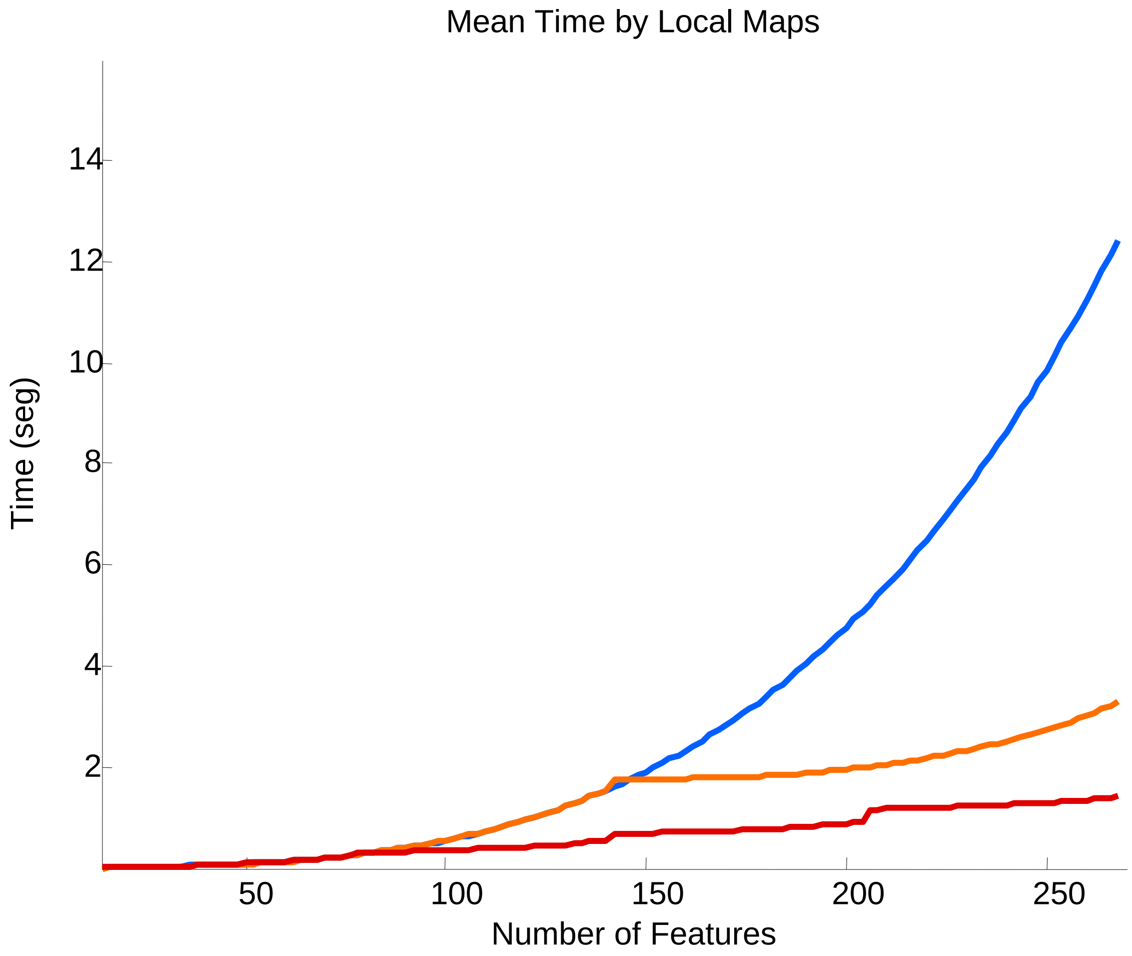
Total SLAM with Map Joining



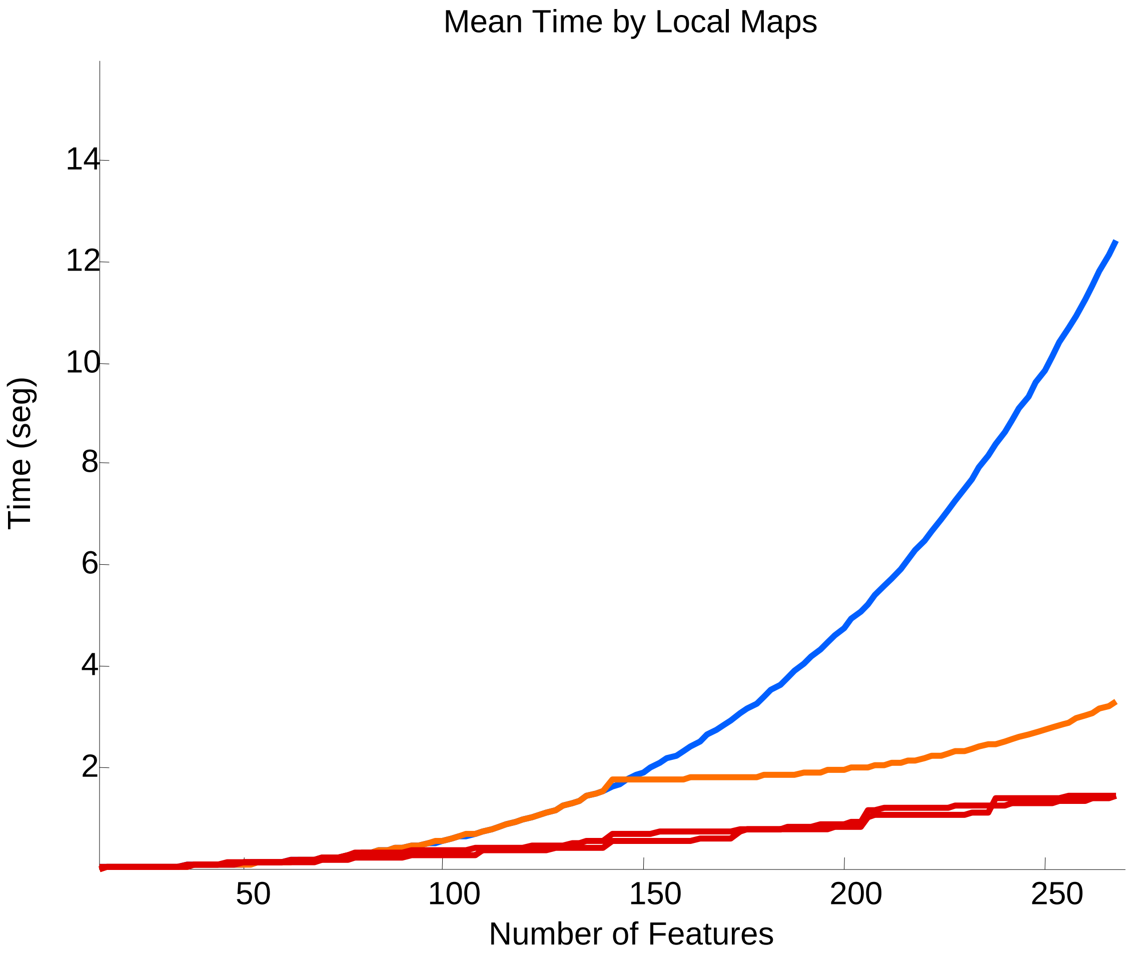
Total SLAM with Map Joining



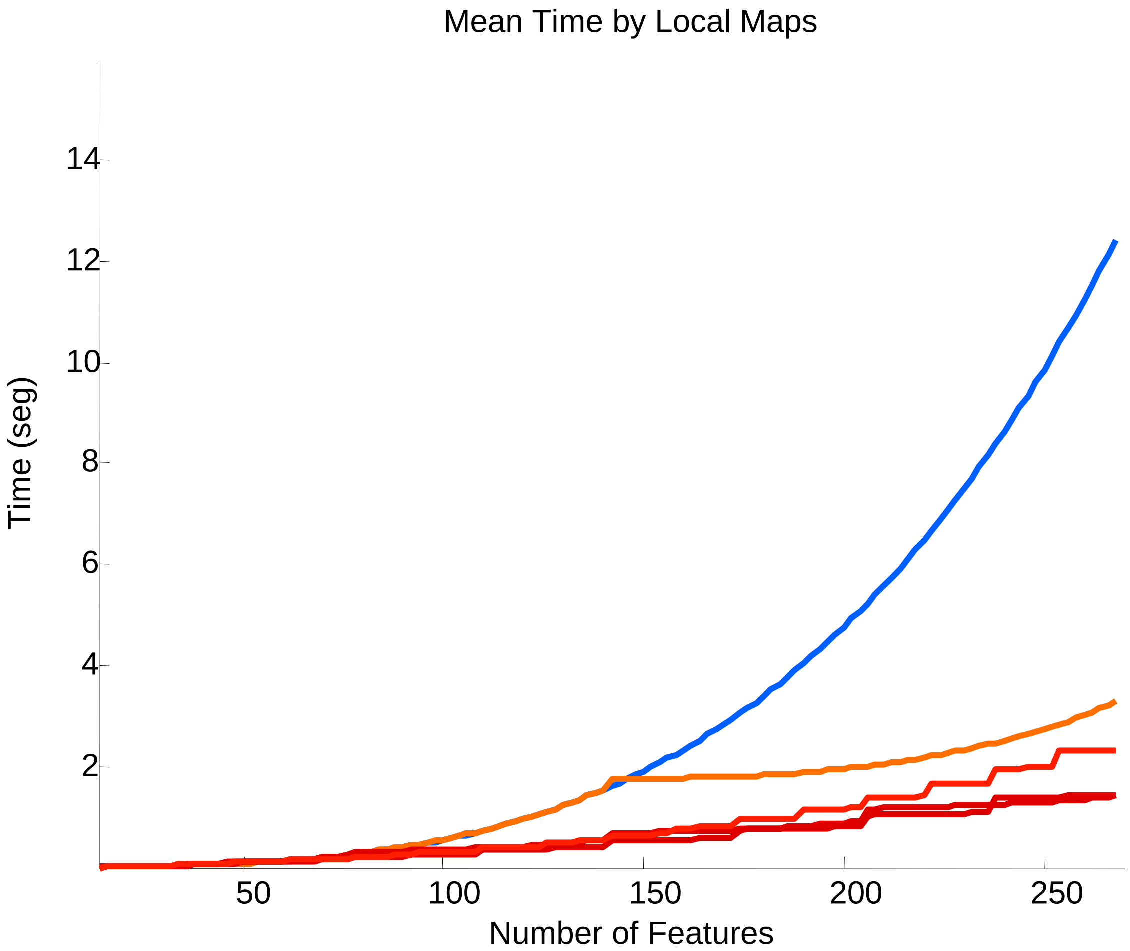
Total SLAM with Map Joining



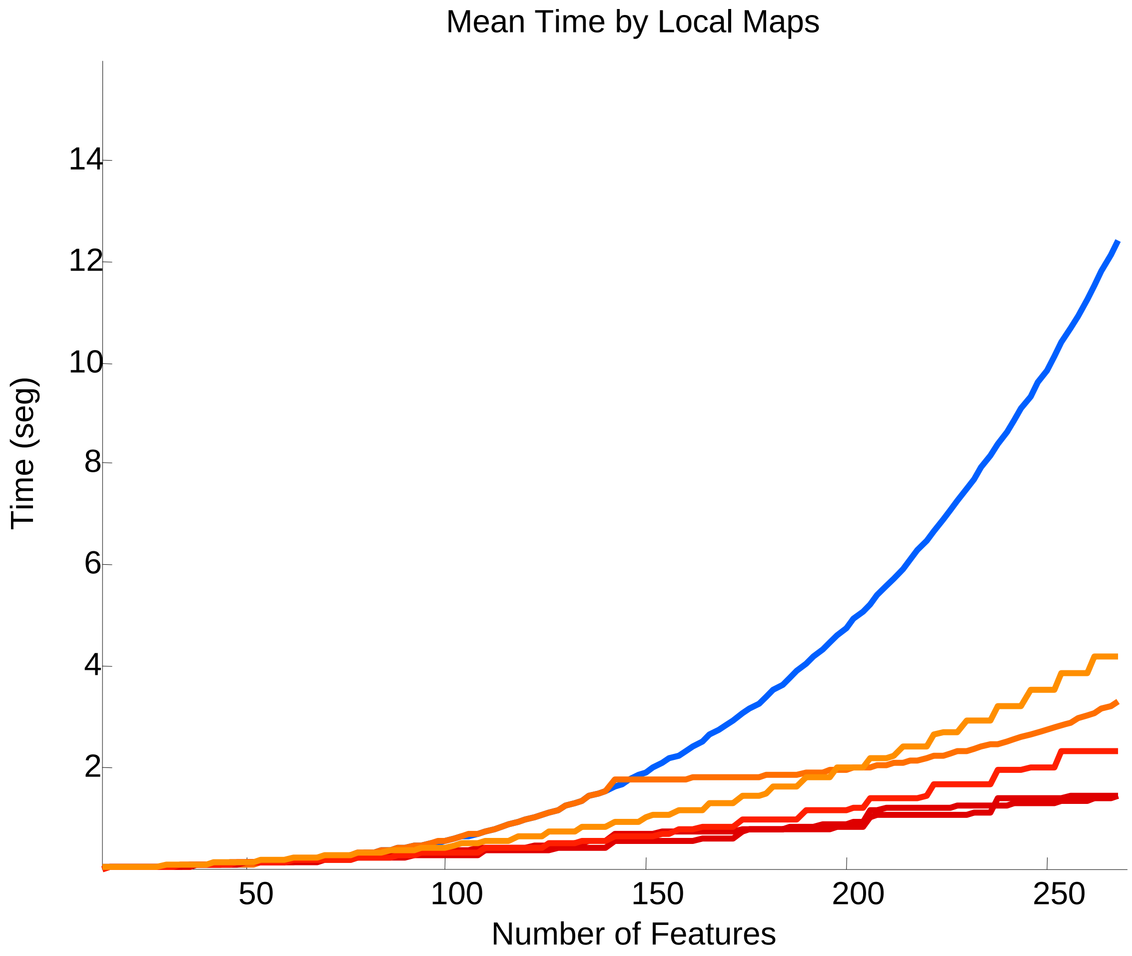
Total SLAM with Map Joining



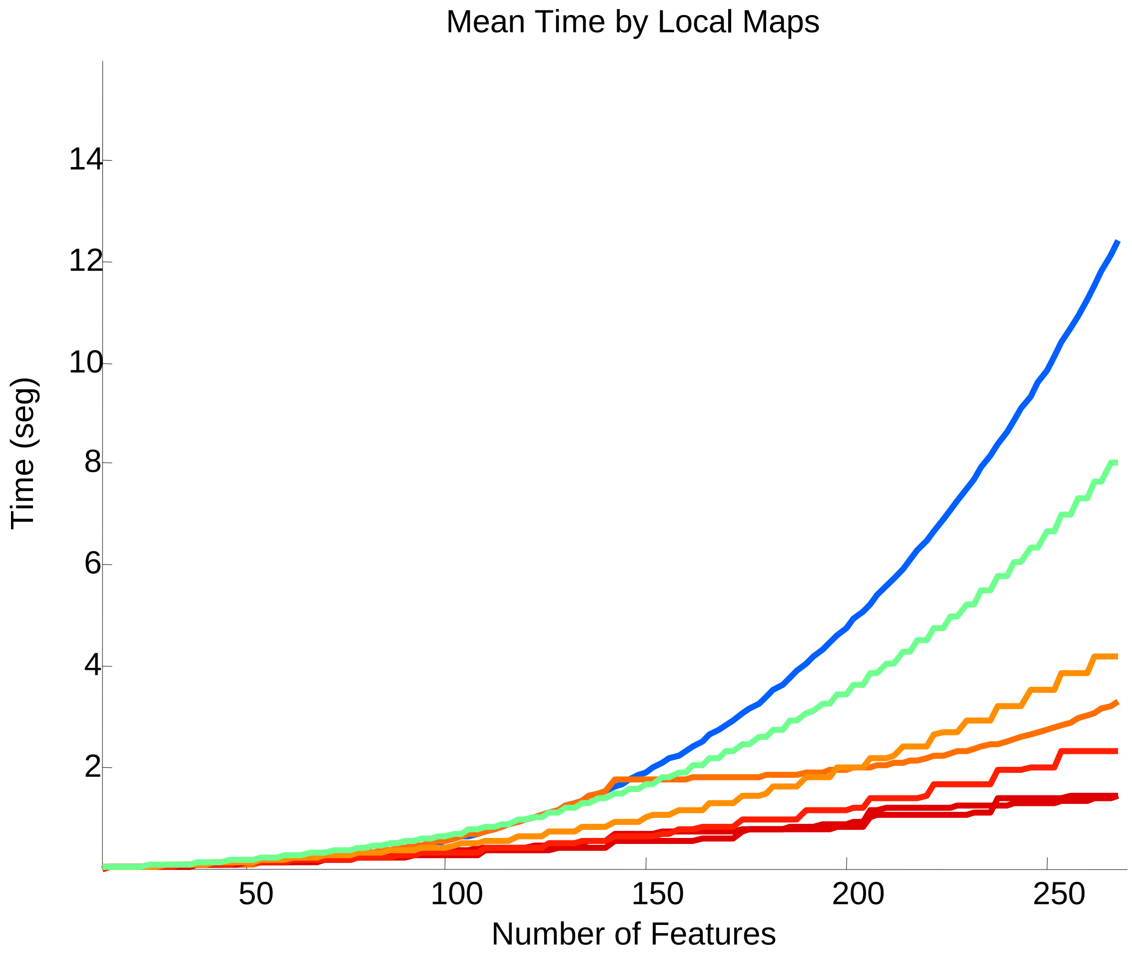
Total SLAM with Map Joining



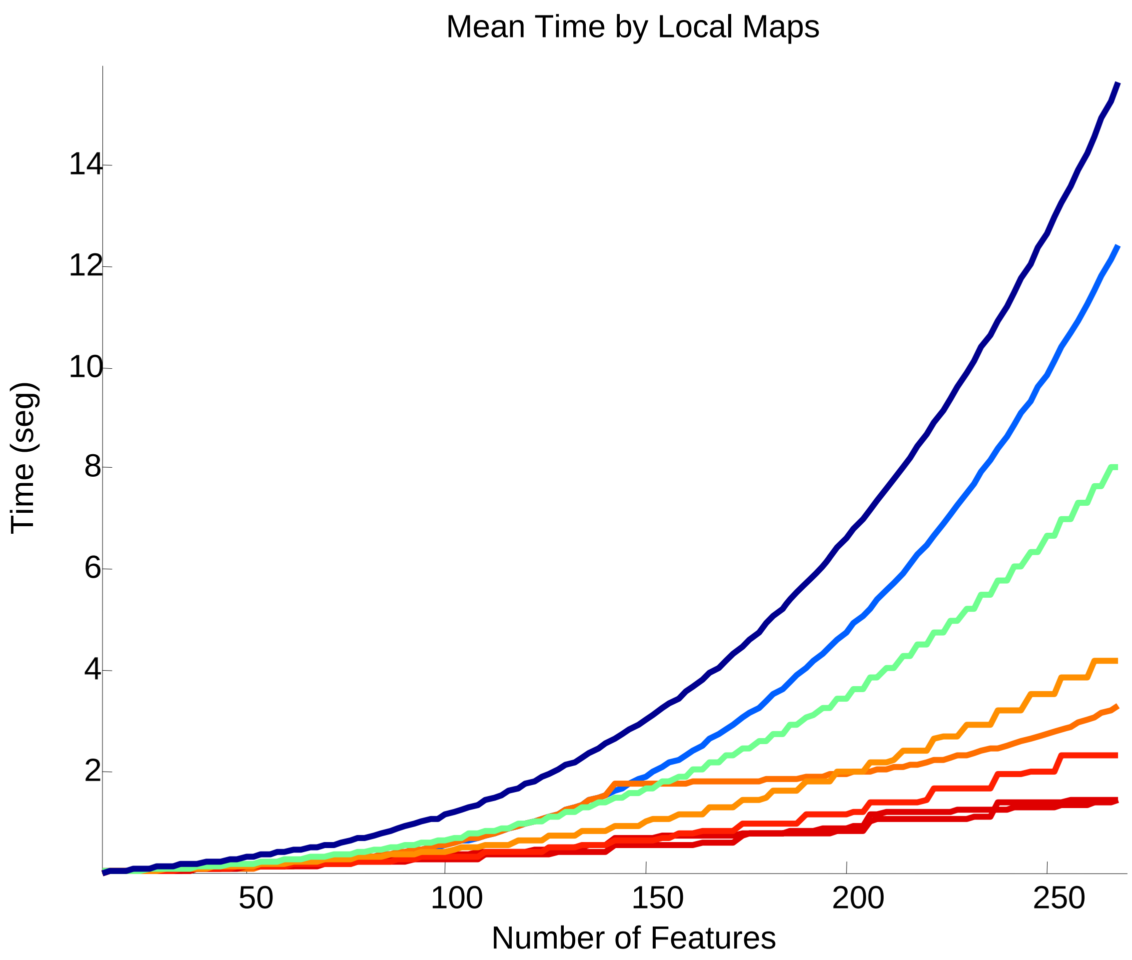
Total SLAM with Map Joining



Total SLAM with Map Joining

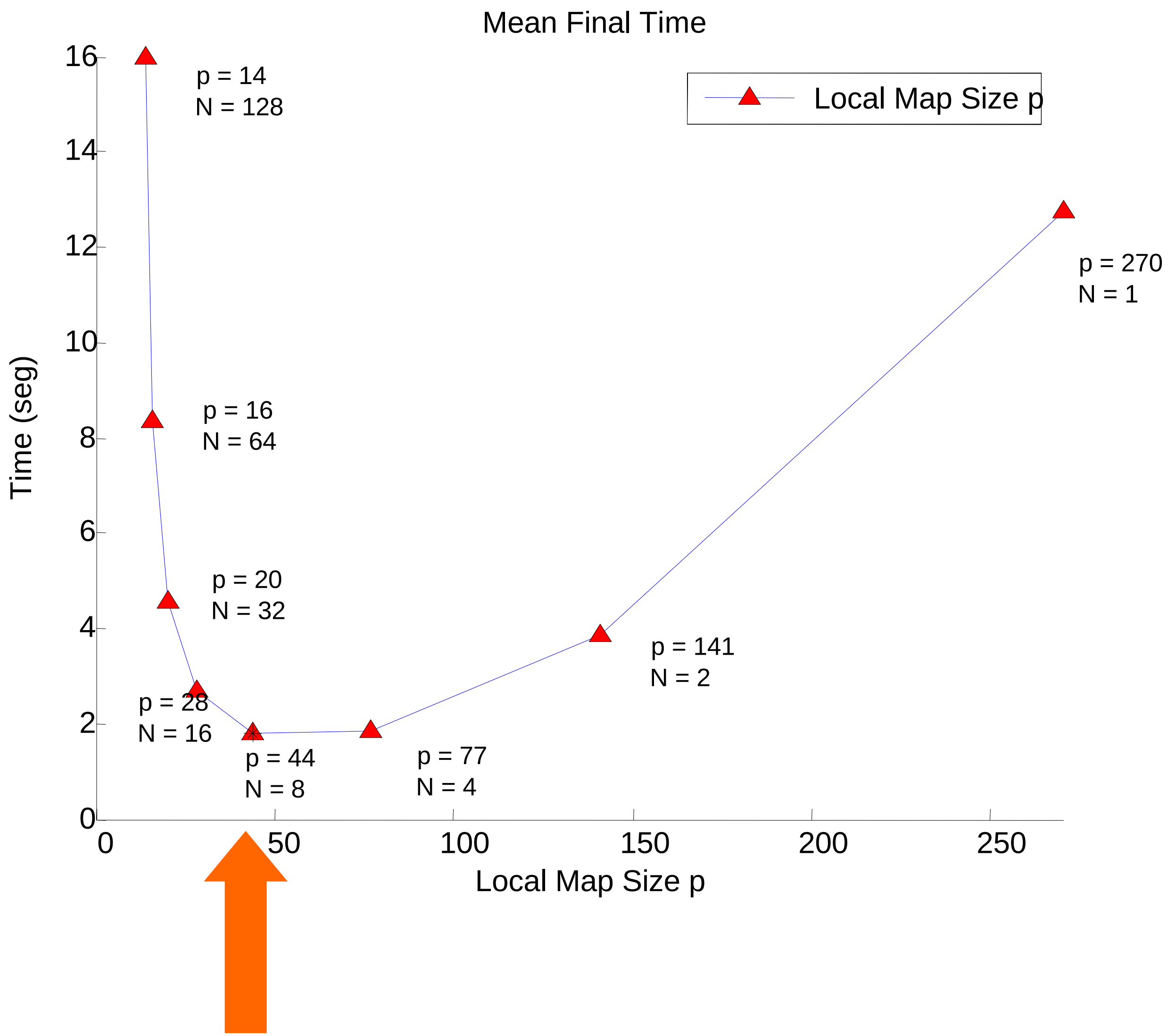


Total SLAM with Map Joining



There is an optimal submap size

Optimal local map size



Consistency Testing

1. Define the consistency index CI as:

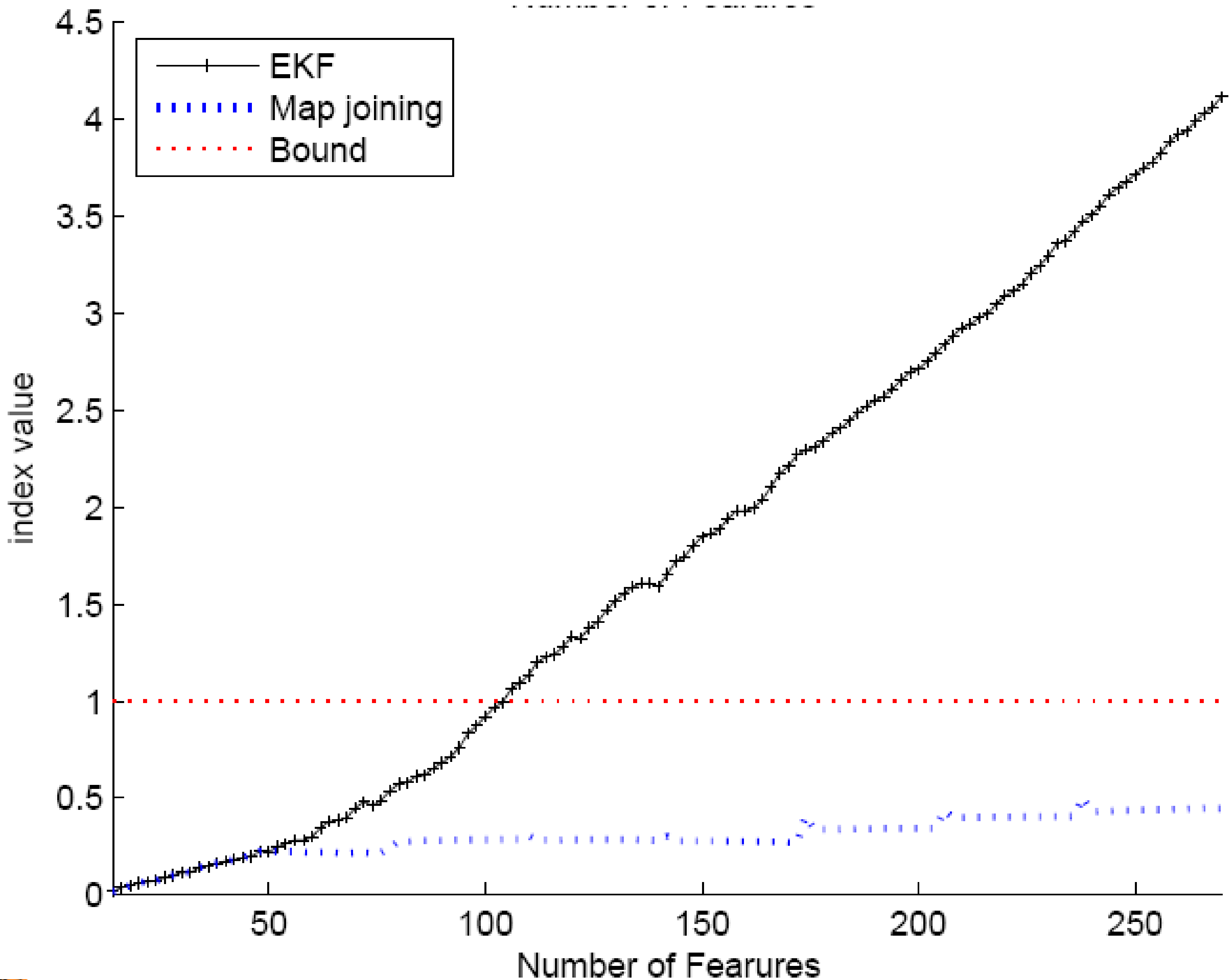
$$CI = \frac{D^2}{\chi^2_{r,1-\alpha}},$$

2. When CI < 1, the estimation is consistent with ground truth

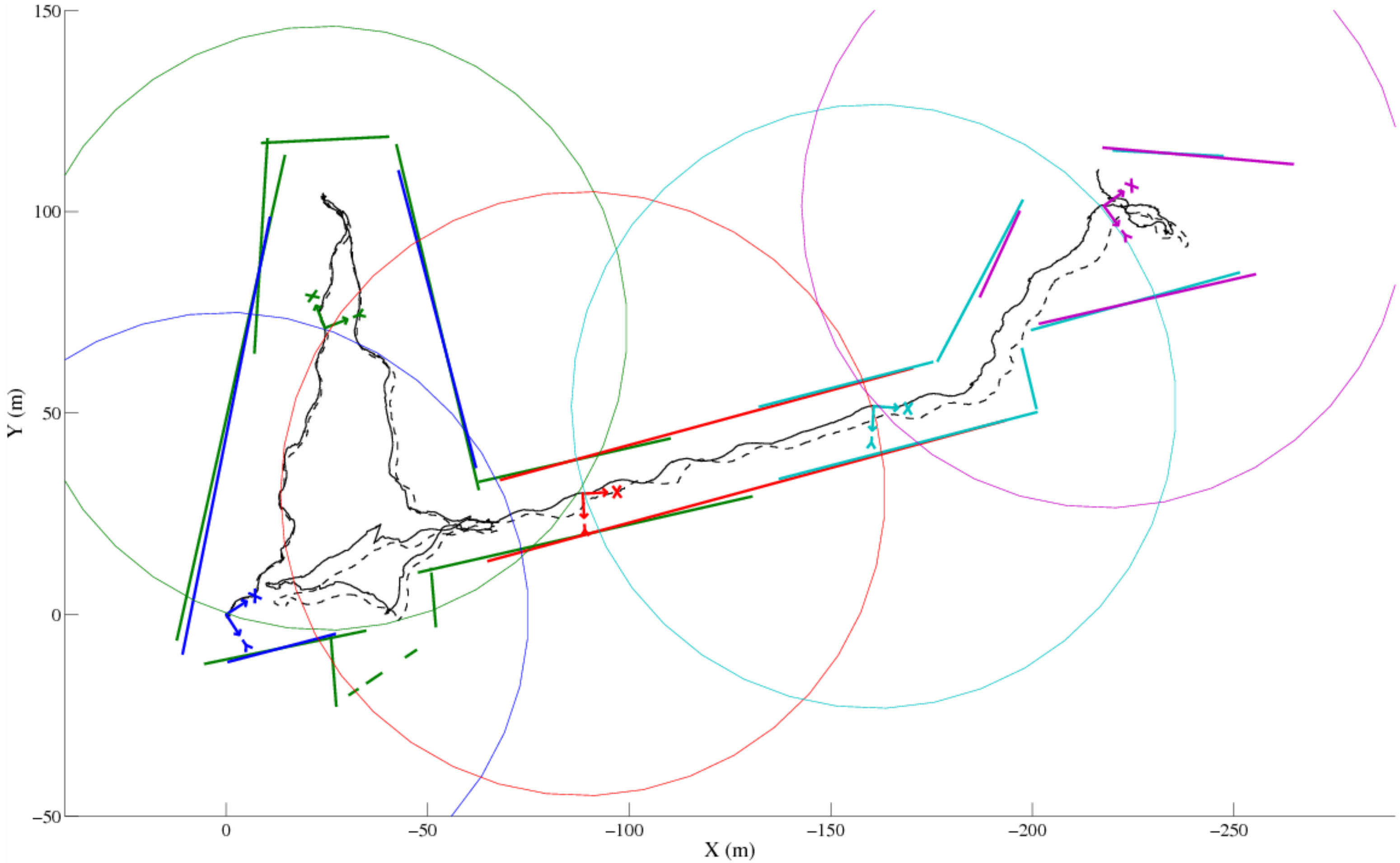
3. When CI > 1, the estimation is inconsistent

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Mean feature location



Underwater SLAM



D. Ribas, P. Ridao, J. Neira, J.D. Tardós, **SLAM in Partially Structured Underwater Environments**, To appear in Journal of Field Robotics, 2008

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Underwater SLAM

D. Ribas, P. Ridao, J. Neira, J.D. Tardós, **SLAM in Partially Structured Underwater Environments**, To appear in Journal of Field Robotics, 2008

Underwater SLAM

D. Ribas, P. Ridao, J. Neira, J.D. Tardós, **SLAM in Partially Structured Underwater Environments**, To appear in Journal of Field Robotics, 2008

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Map Joining is a 'more for less' algorithm:

- It improves the consistency of the resulting map
- It reduces the computational cost

Limitations:

- Map joining is still $O(n^2)$
- Environment size must be known

Overcoming limitations:

- Can we have EKF-SLAM updates in $O(n)$?

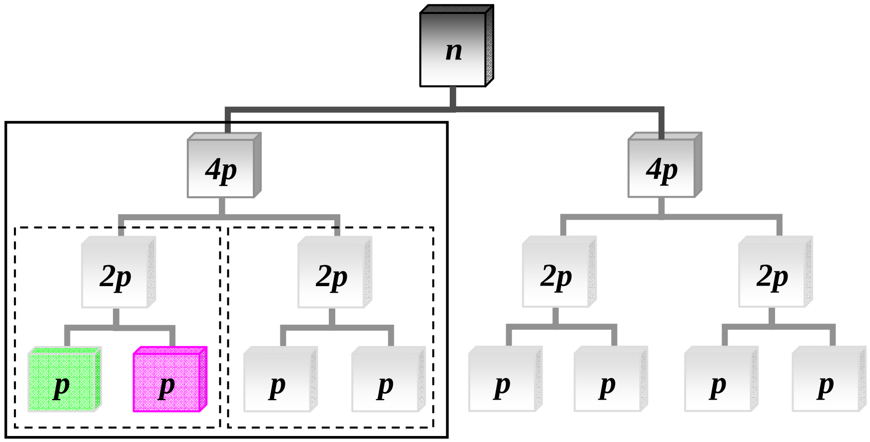
Outline

1. Basic EKF SLAM
    - Introduction: the need for SLAM
    - The basic EKF SLAM algorithm
    - Feature Extraction
    - Continuous Data Association
    - The Loop Closing Problem

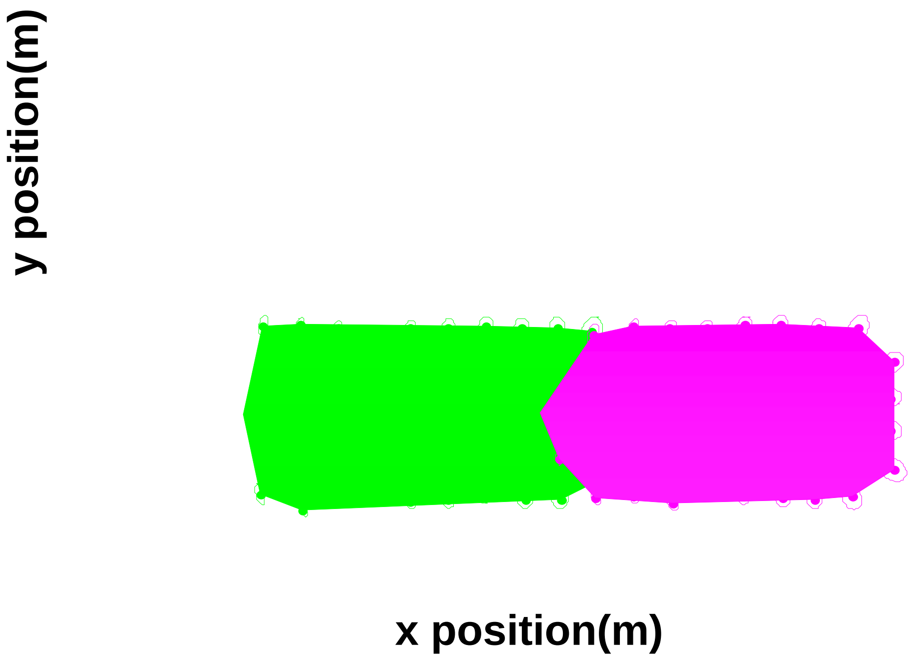
2. Advanced EKF SLAM
    - Computational complexity of EKF SLAM
    - Consistency of the EKF SLAM
    - SLAM using local maps
        - Sequential Map Joining
        - **Divide and Conquer SLAM**
        - Hierarchical SLAM

ROBOTICS UNIVERSIDAD DE ZARAGOZA

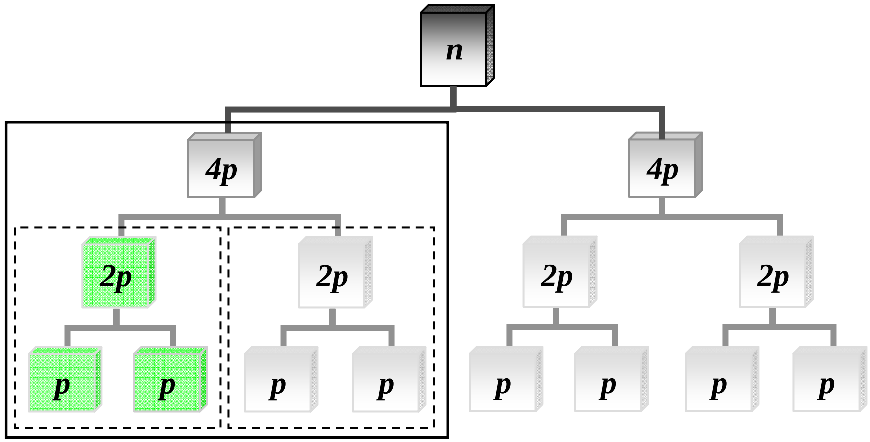
D&C SLAM: Map Hierarchy



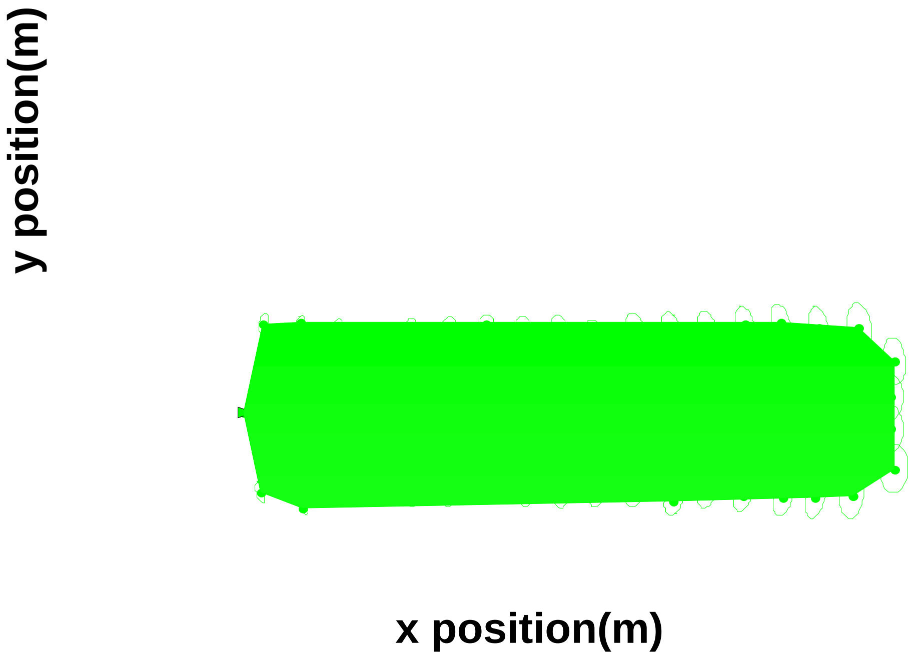
Number of Maps : 2



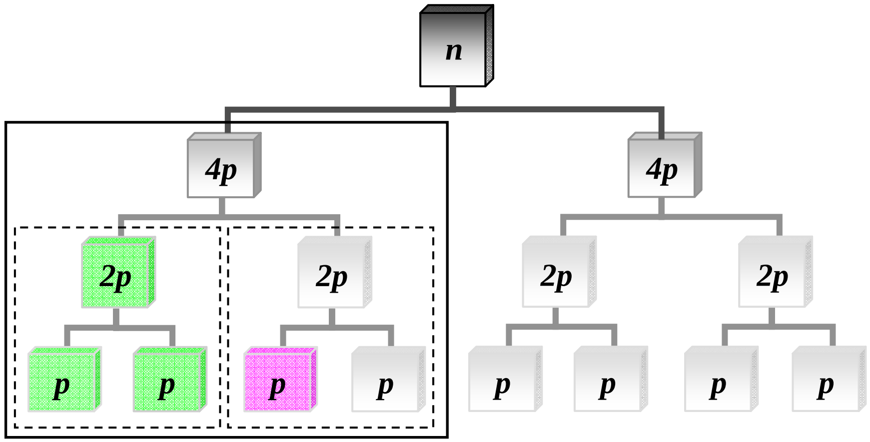
D&C SLAM



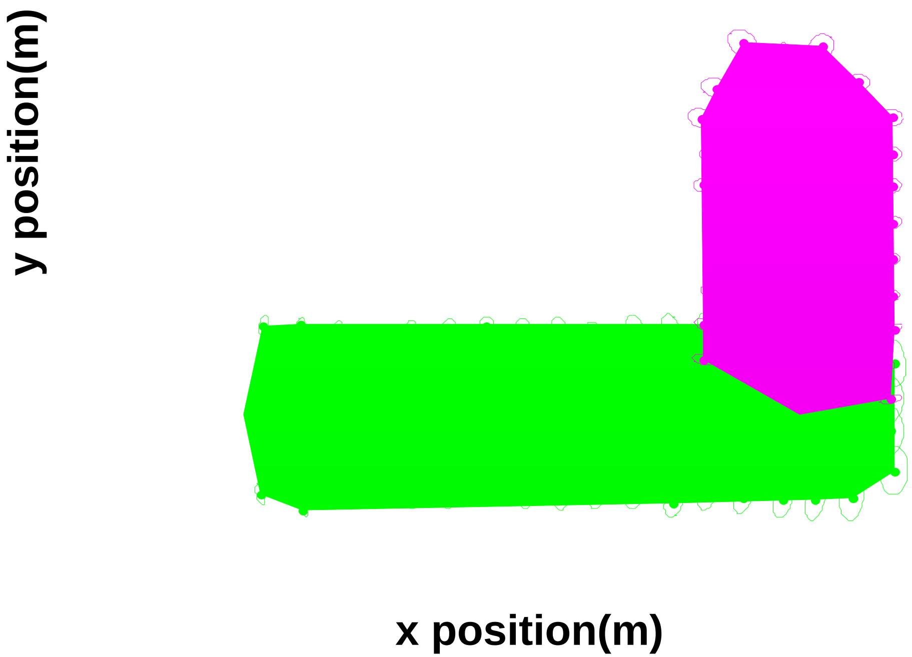
Number of Maps : 1



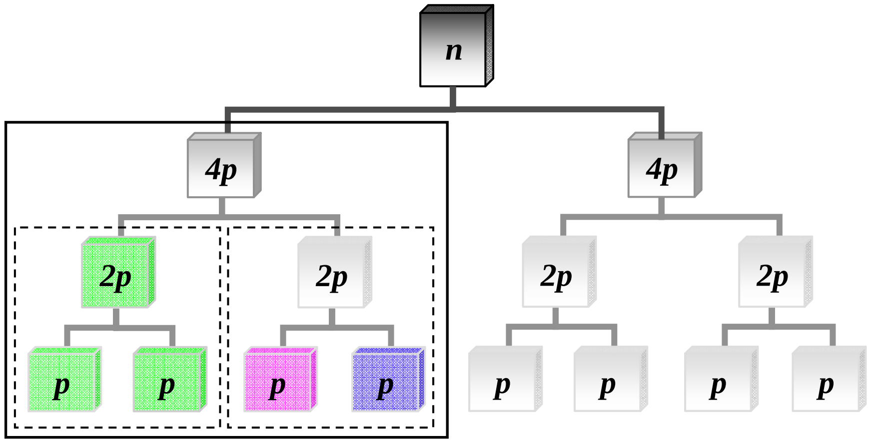
D&C SLAM



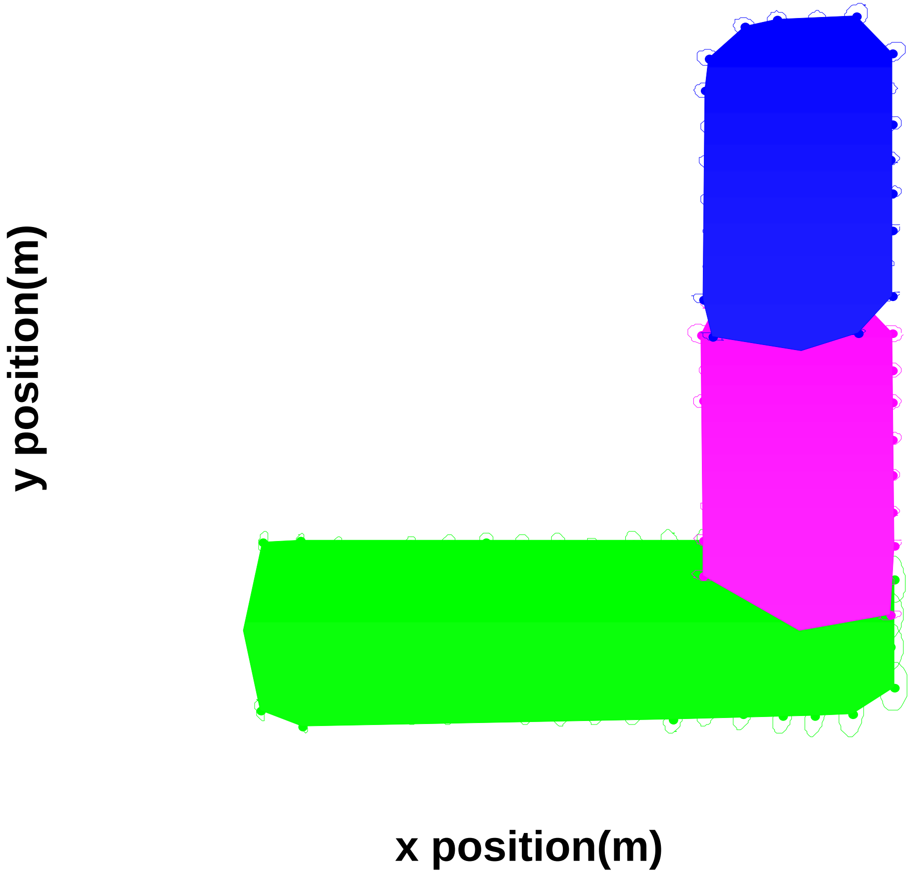
Number of Maps : 2



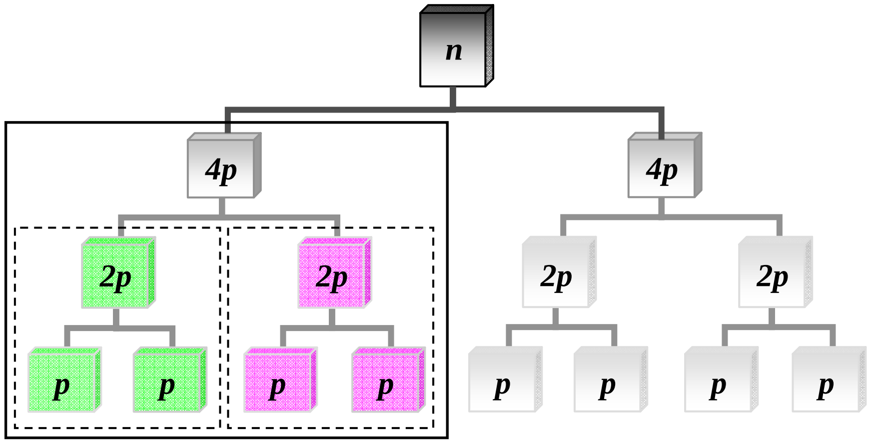
D&C SLAM



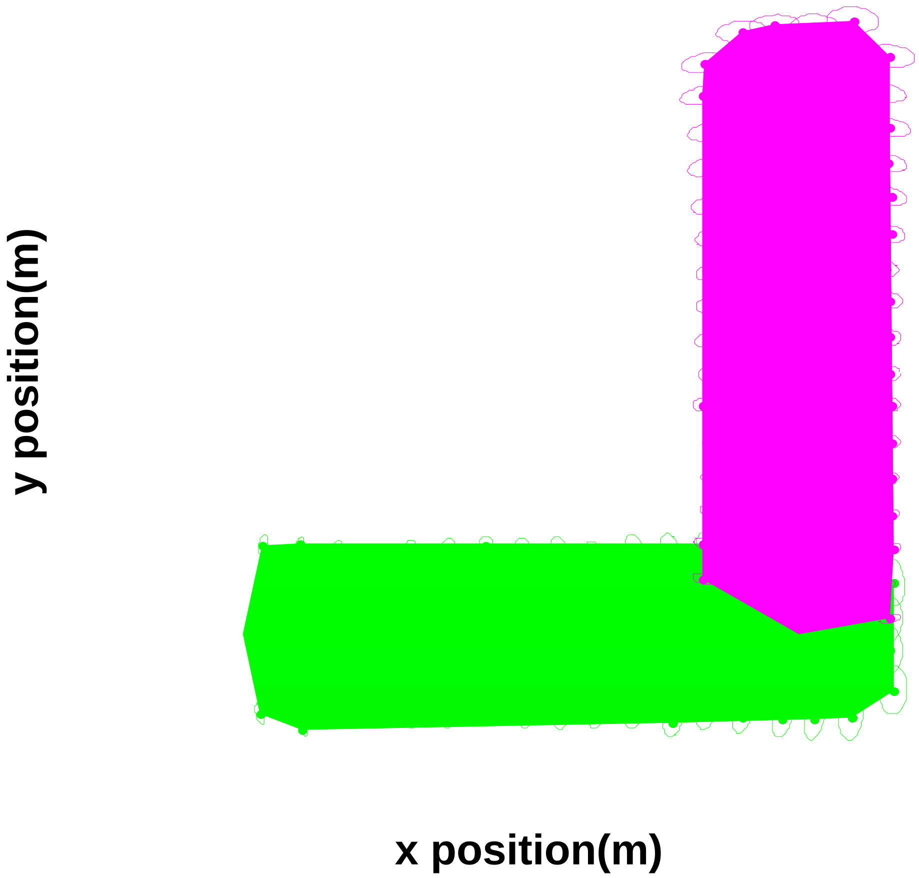
Number of Maps : 3



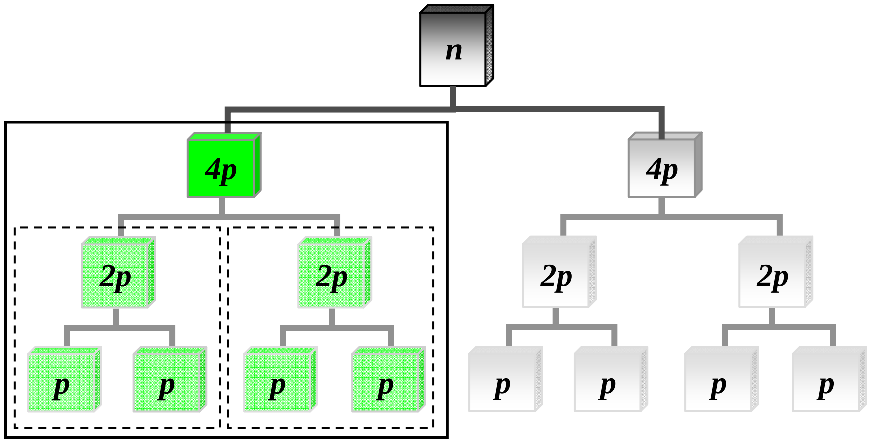
D&C SLAM



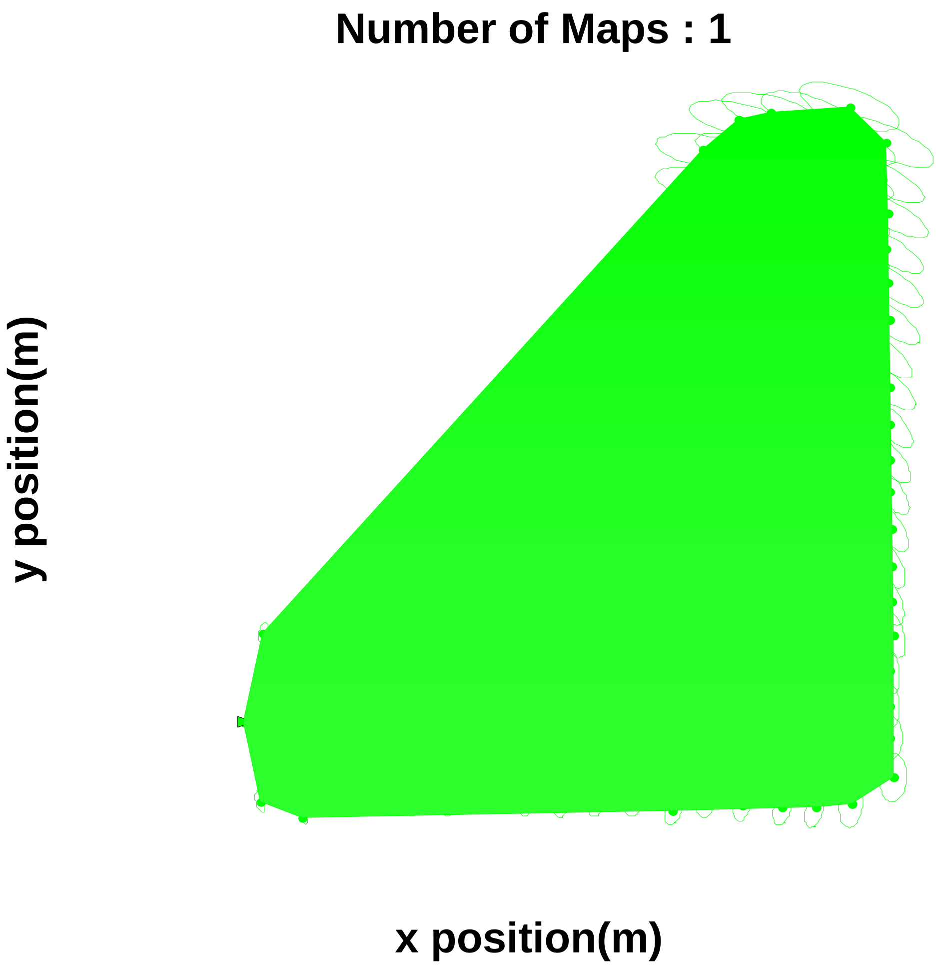
Number of Maps : 2



D&C SLAM

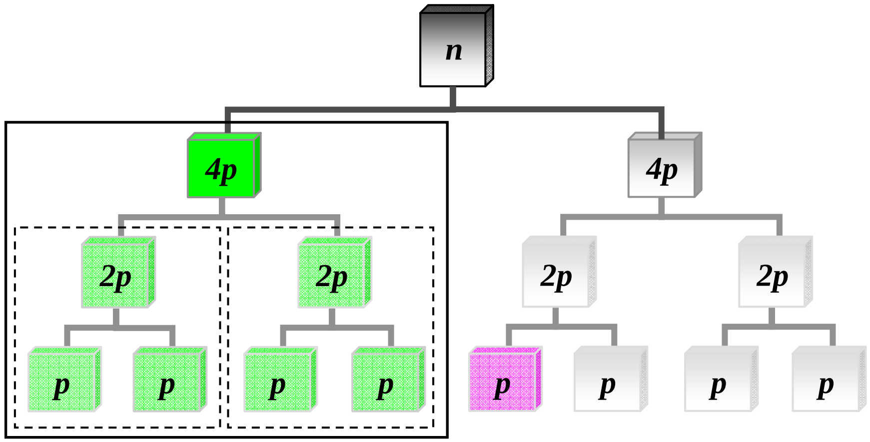


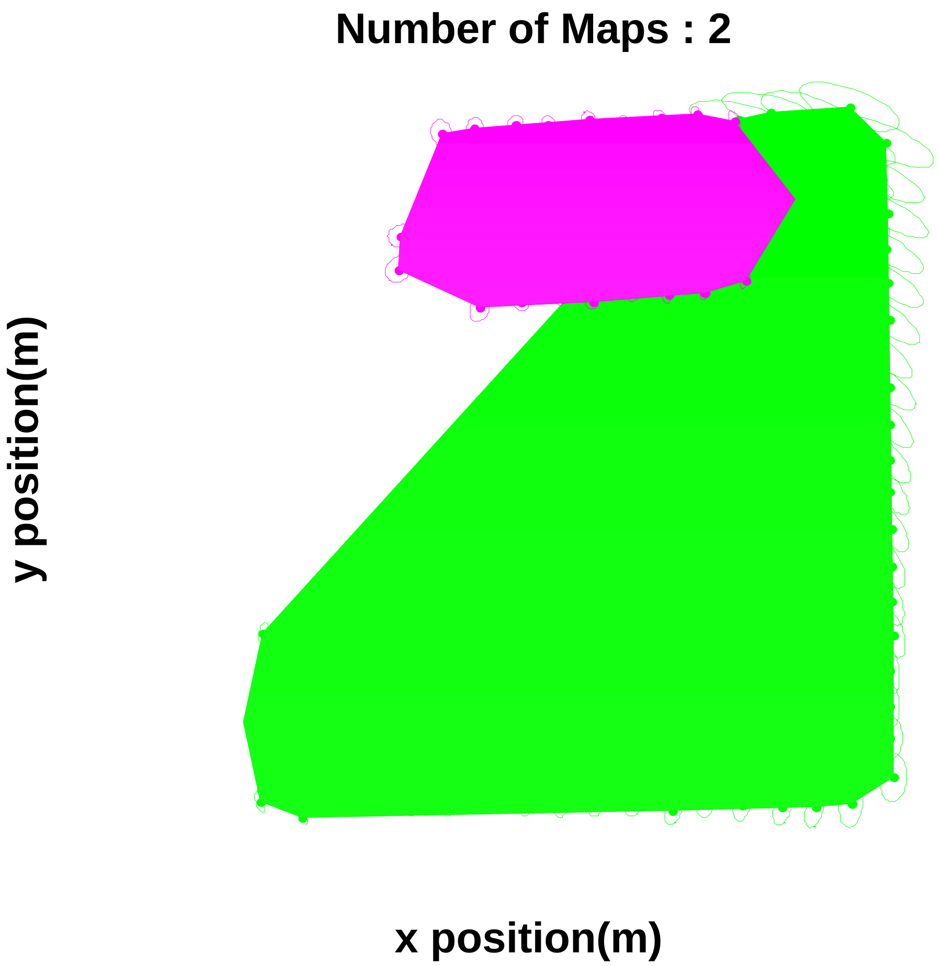
n
4p
4p
2p
2p
2p
2p
p
p
p
p
p
p
p
p



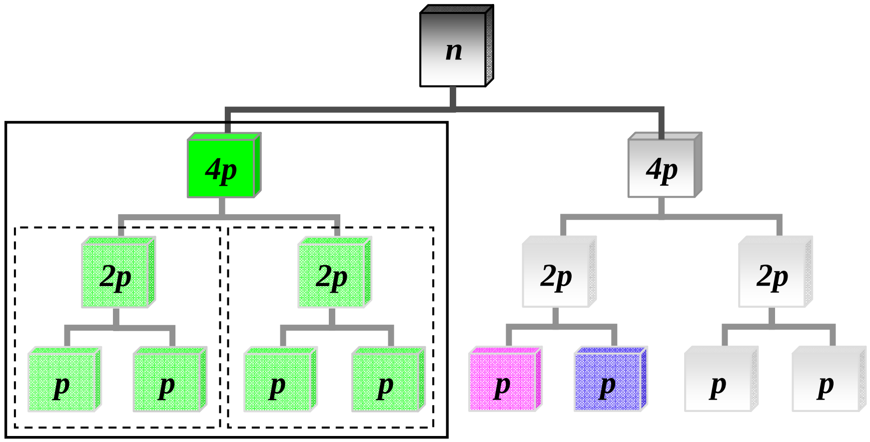
Number of Maps : 1
y position(m)
x position(m)

D&C SLAM

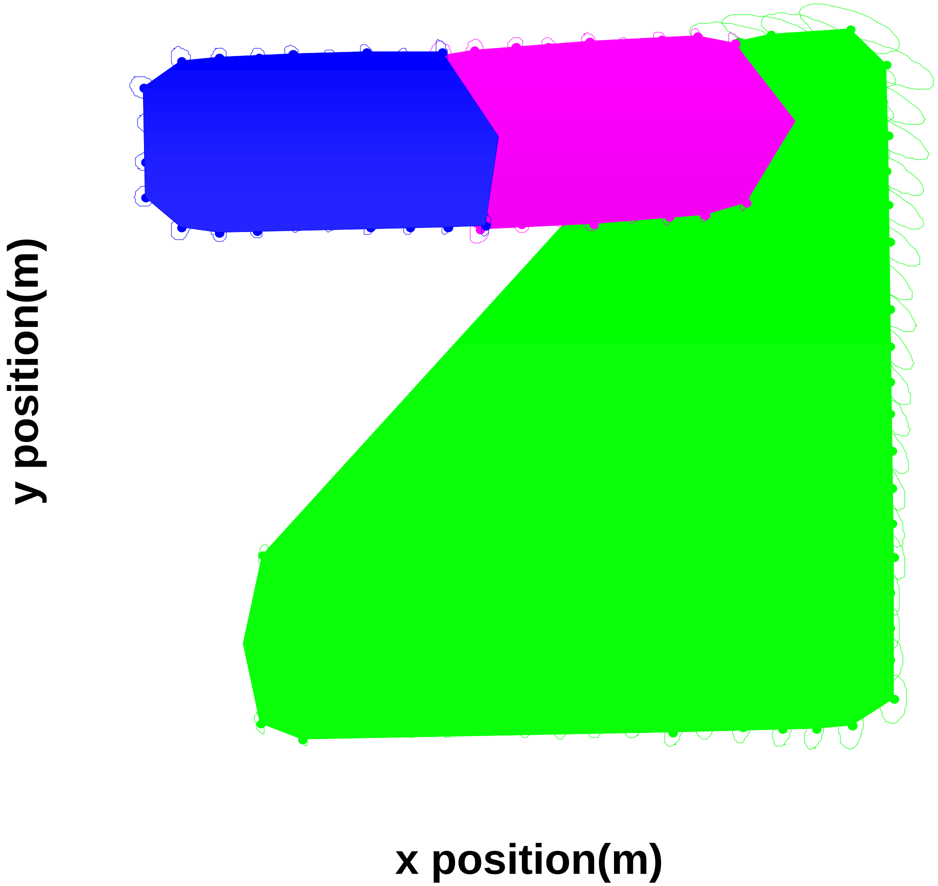

n
4p
4p
2p
2p
2p
2p
p
p
p
p
p
p
p
p


Number of Maps : 2
y position(m)
x position(m)

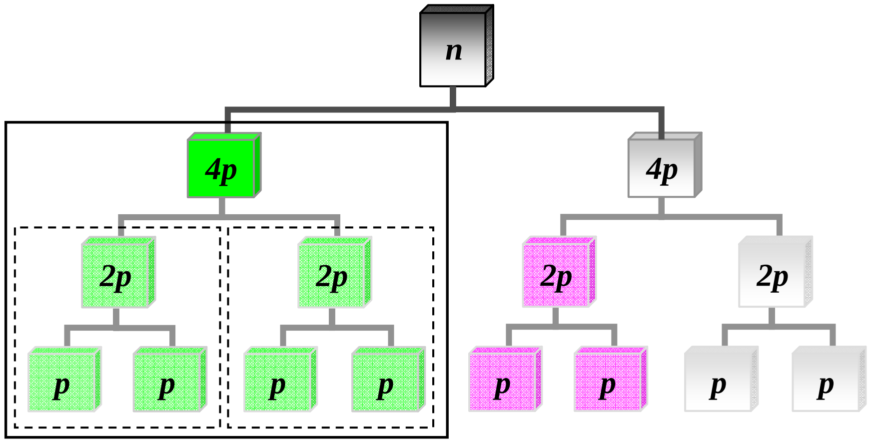
D&C SLAM

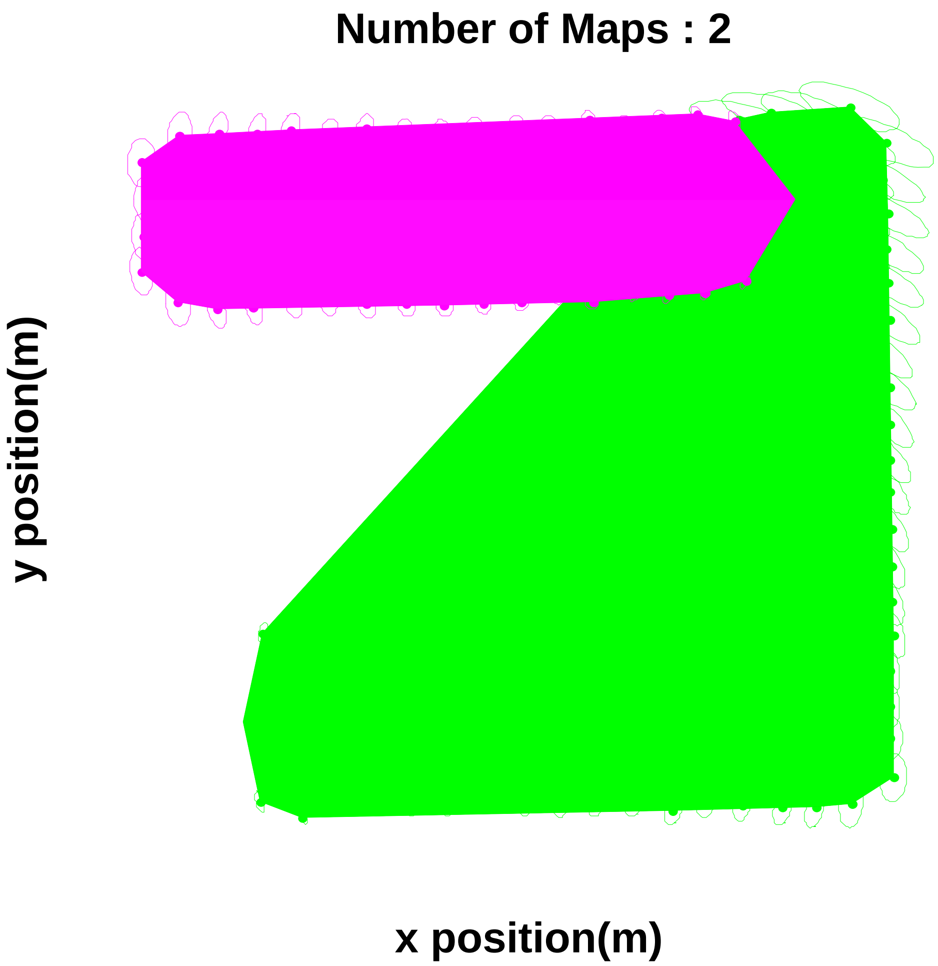


Number of Maps : 3

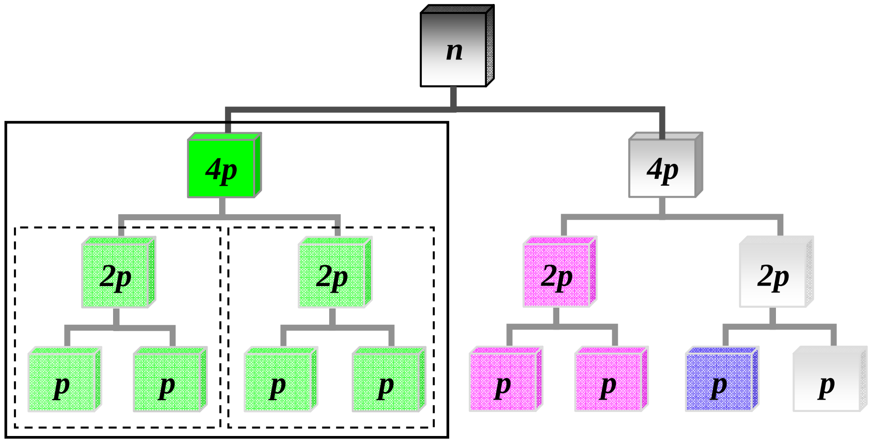


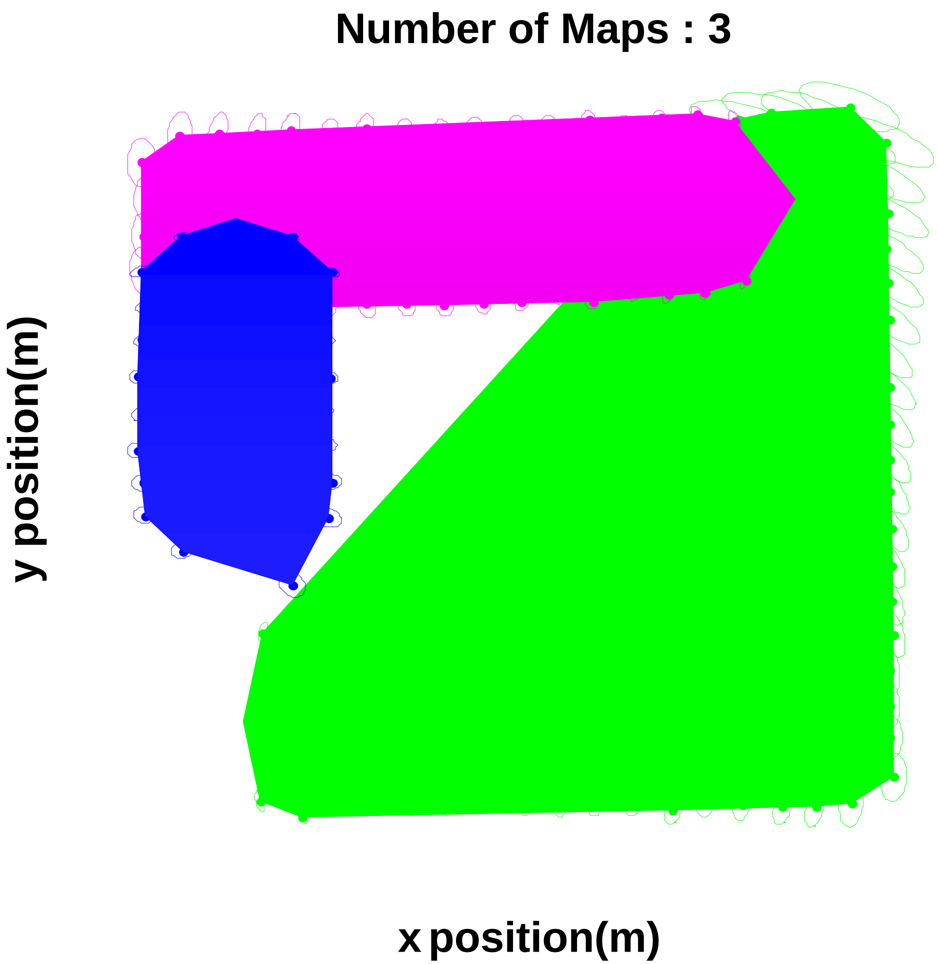
D&C SLAM


n
4p
4p
2p
2p
2p
2p
p
p
p
p
p
p
p
p

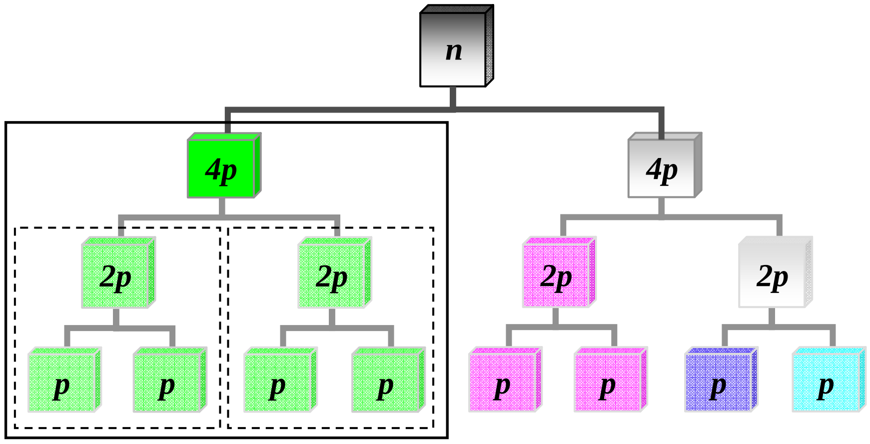

Number of Maps : 2
y position(m)
x position(m)

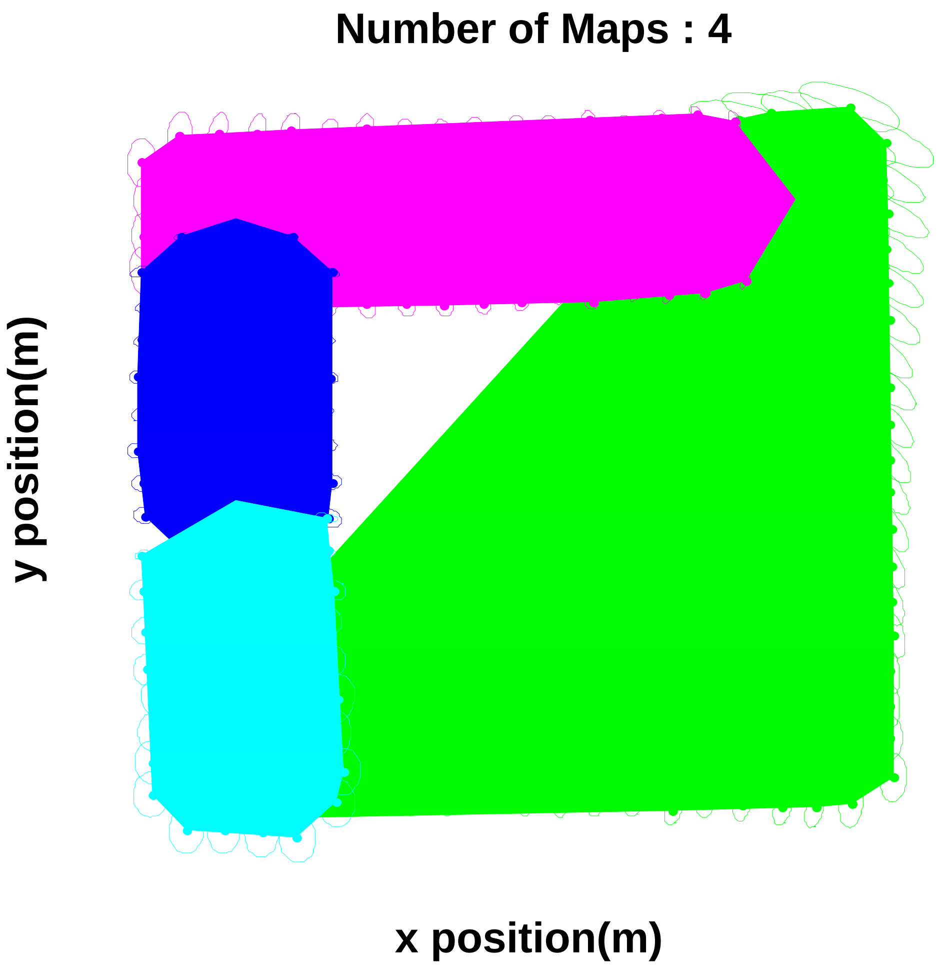
D&C SLAM


n
4p
4p
2p
2p
2p
2p
p
p
p
p
p
p
p
p

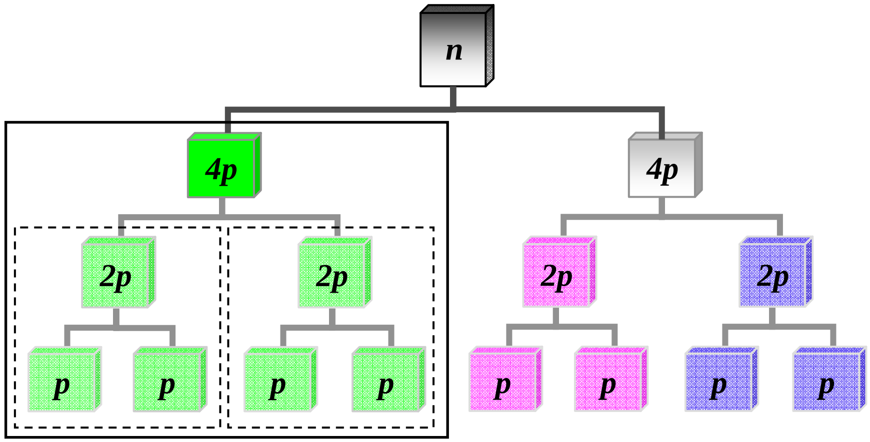

Number of Maps : 3
y position(m)
x position(m)

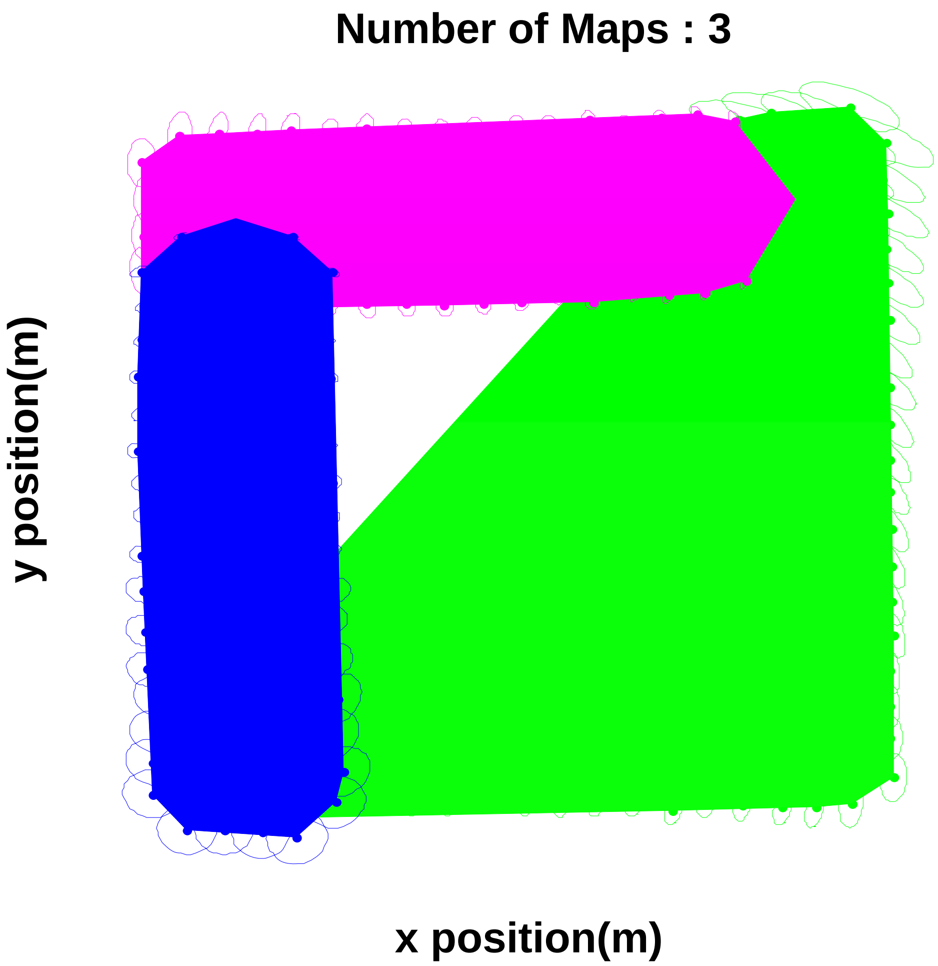
D&C SLAM


n
4p
4p
2p
2p
2p
2p
p
p
p
p
p
p
p
p


Number of Maps : 4
y position(m)
x position(m)

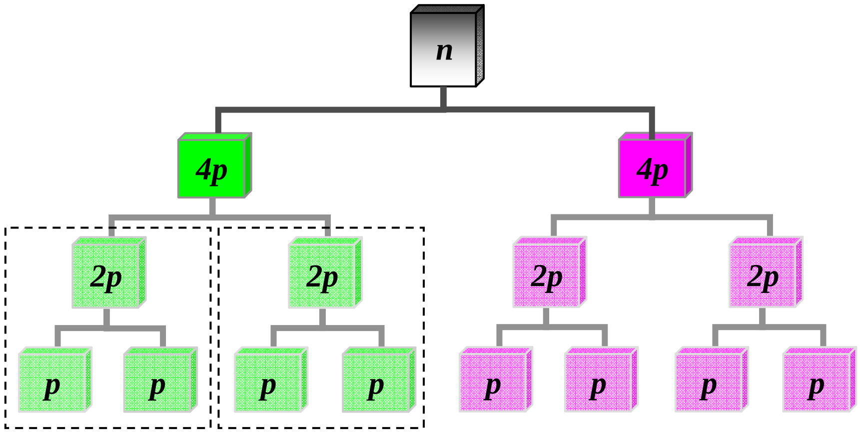
D&C SLAM

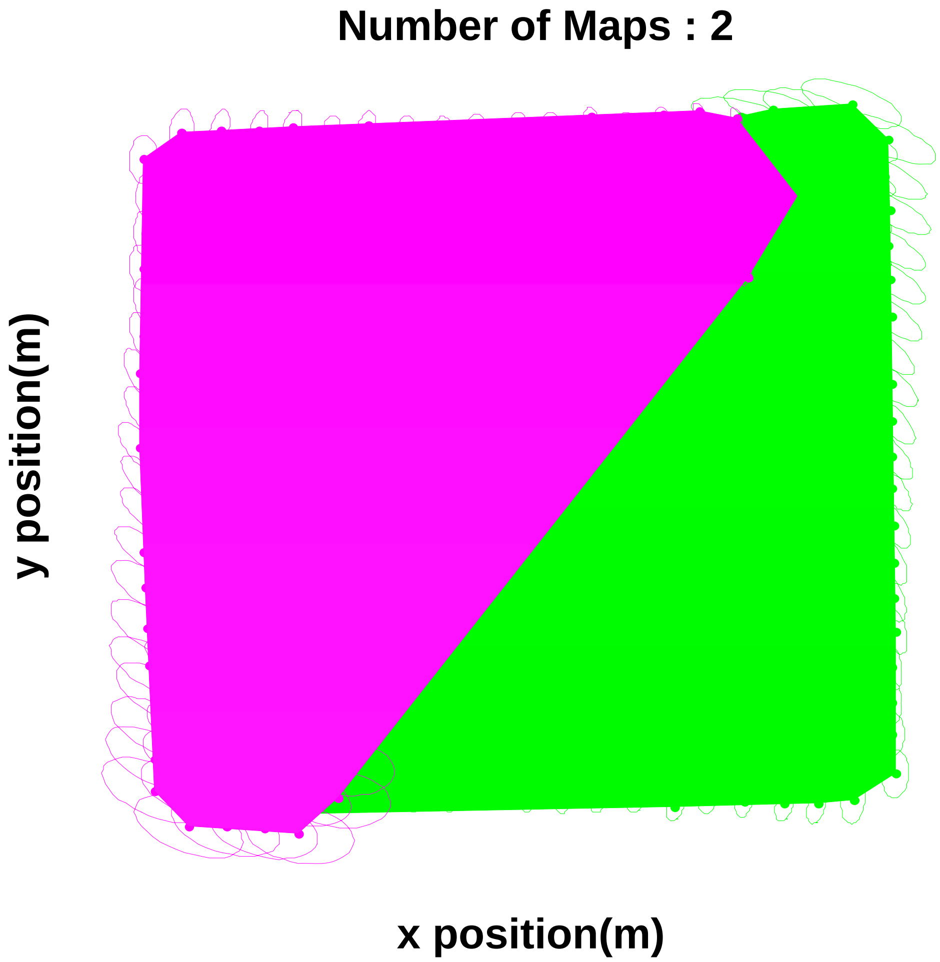




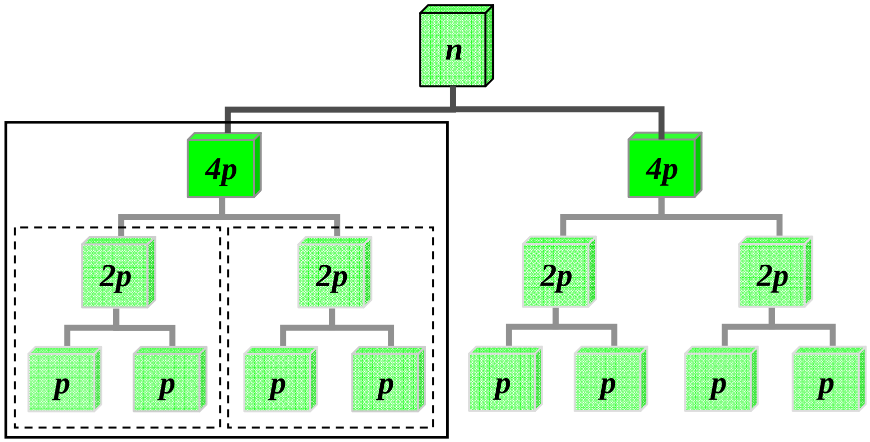
ROBOTICS UNIVERSIDAD DE ZARAGOZA

D&C SLAM

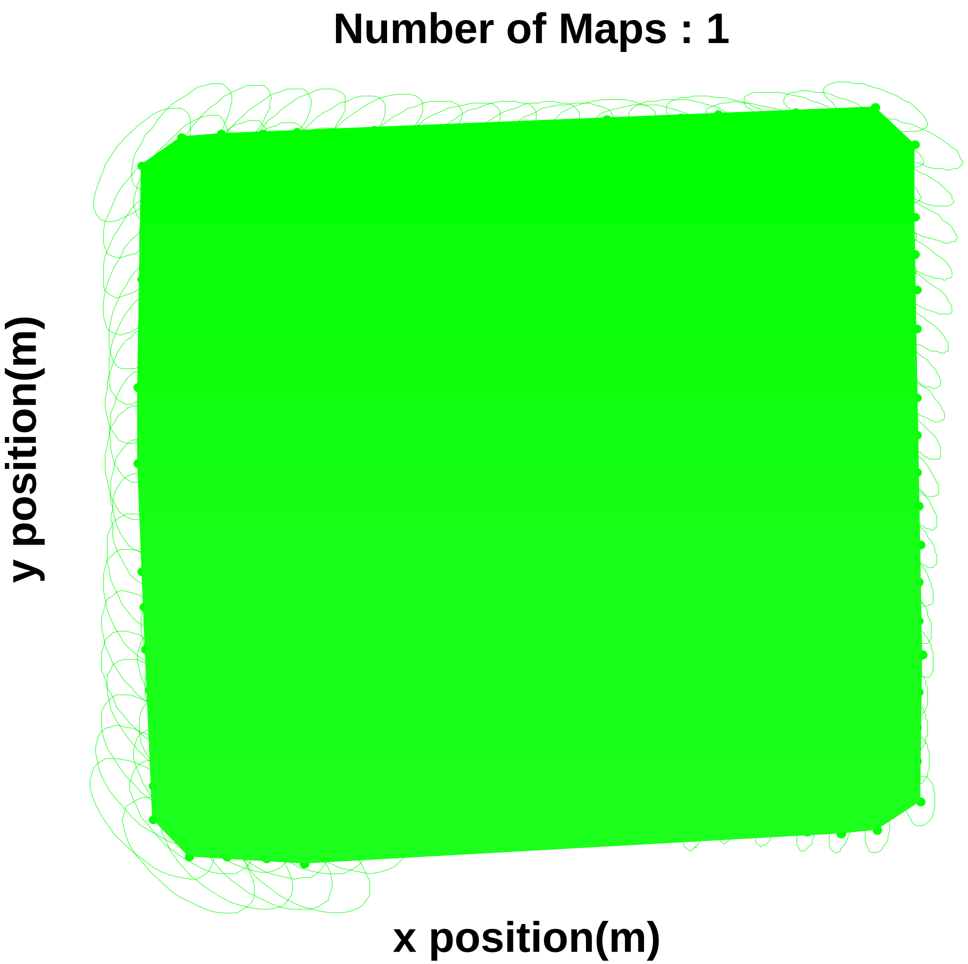

n
4p
4p
2p
2p
2p
2p
p
p
p
p
p
p
p
p


Number of Maps : 2
y position(m)
x position(m)

D&C SLAM

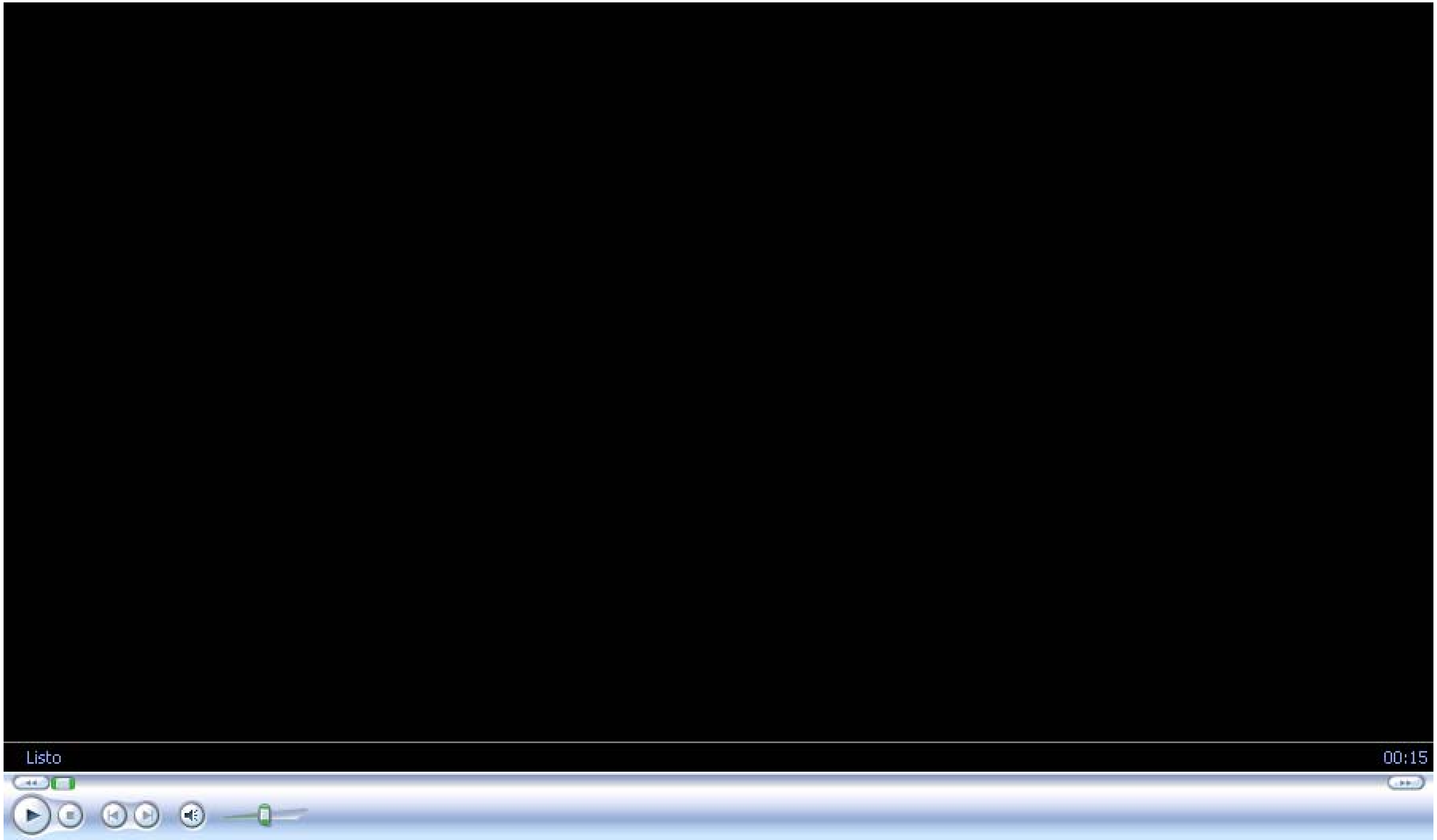


n
4p
4p
2p
2p
2p
2p
p
p
p
p
p
p
p
p



Number of Maps : 1
y position(m)
x position(m)

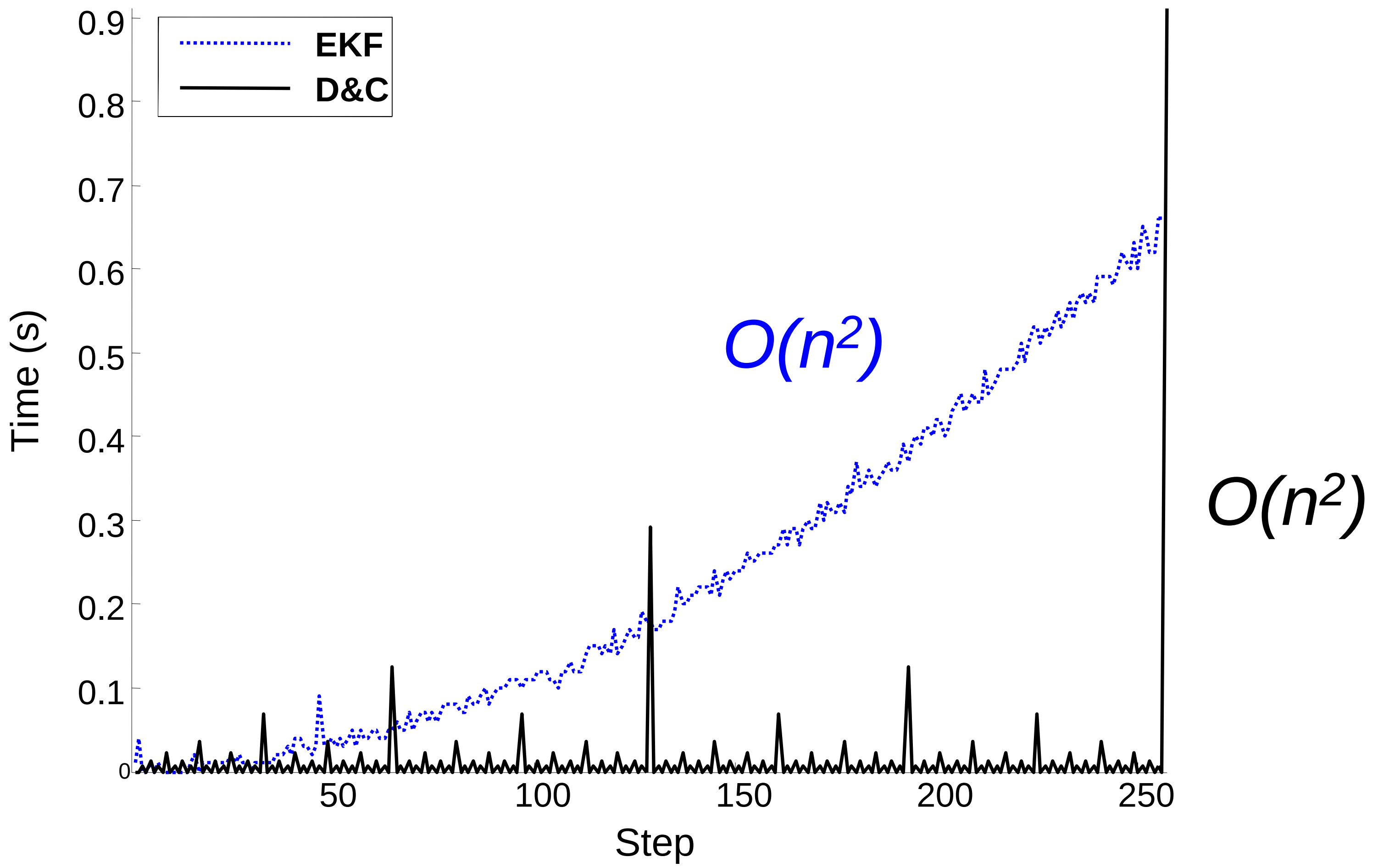
Loop Trajectory



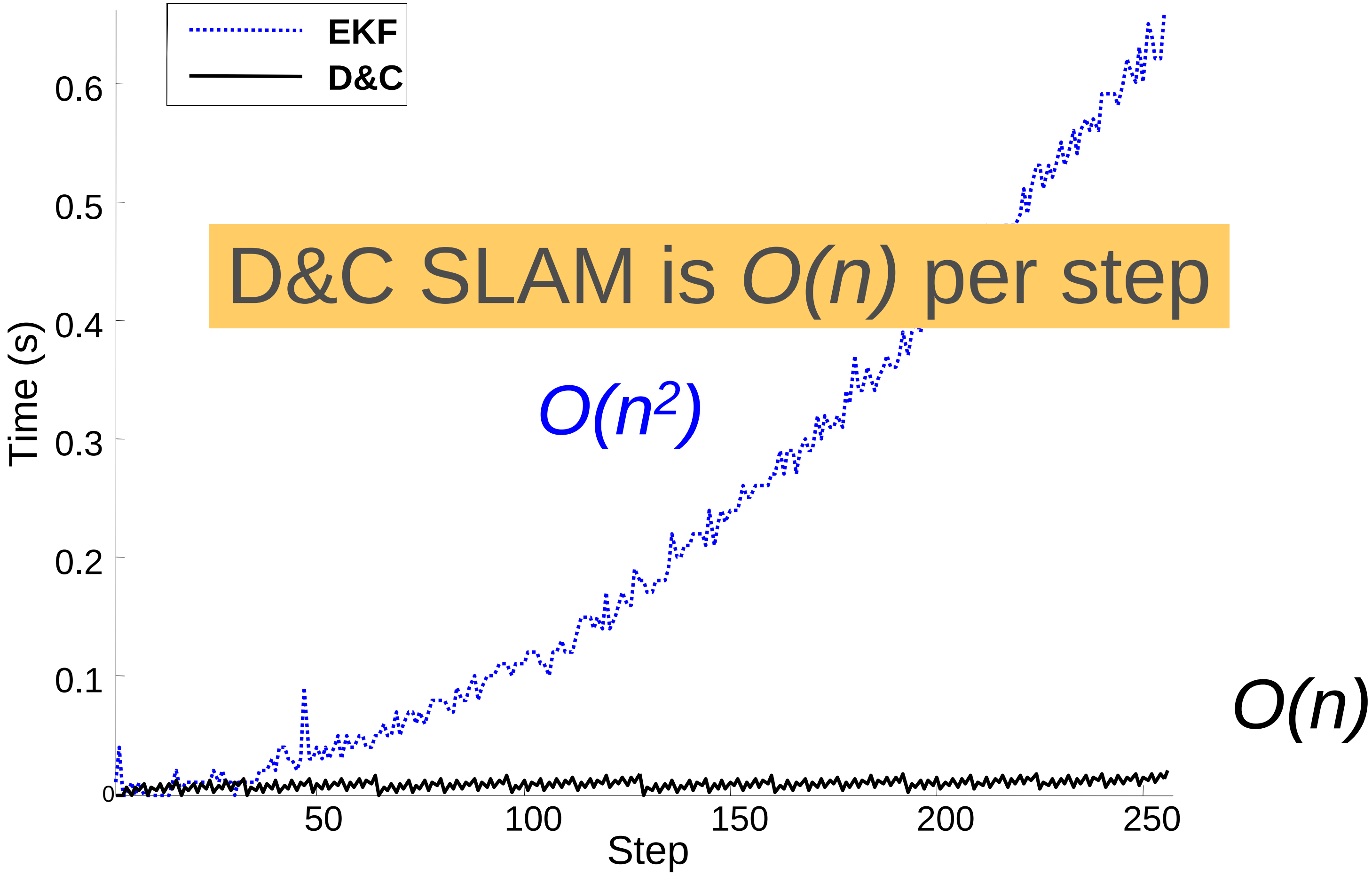
D&C is ~1/7 of EKF total time

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Computational cost per step

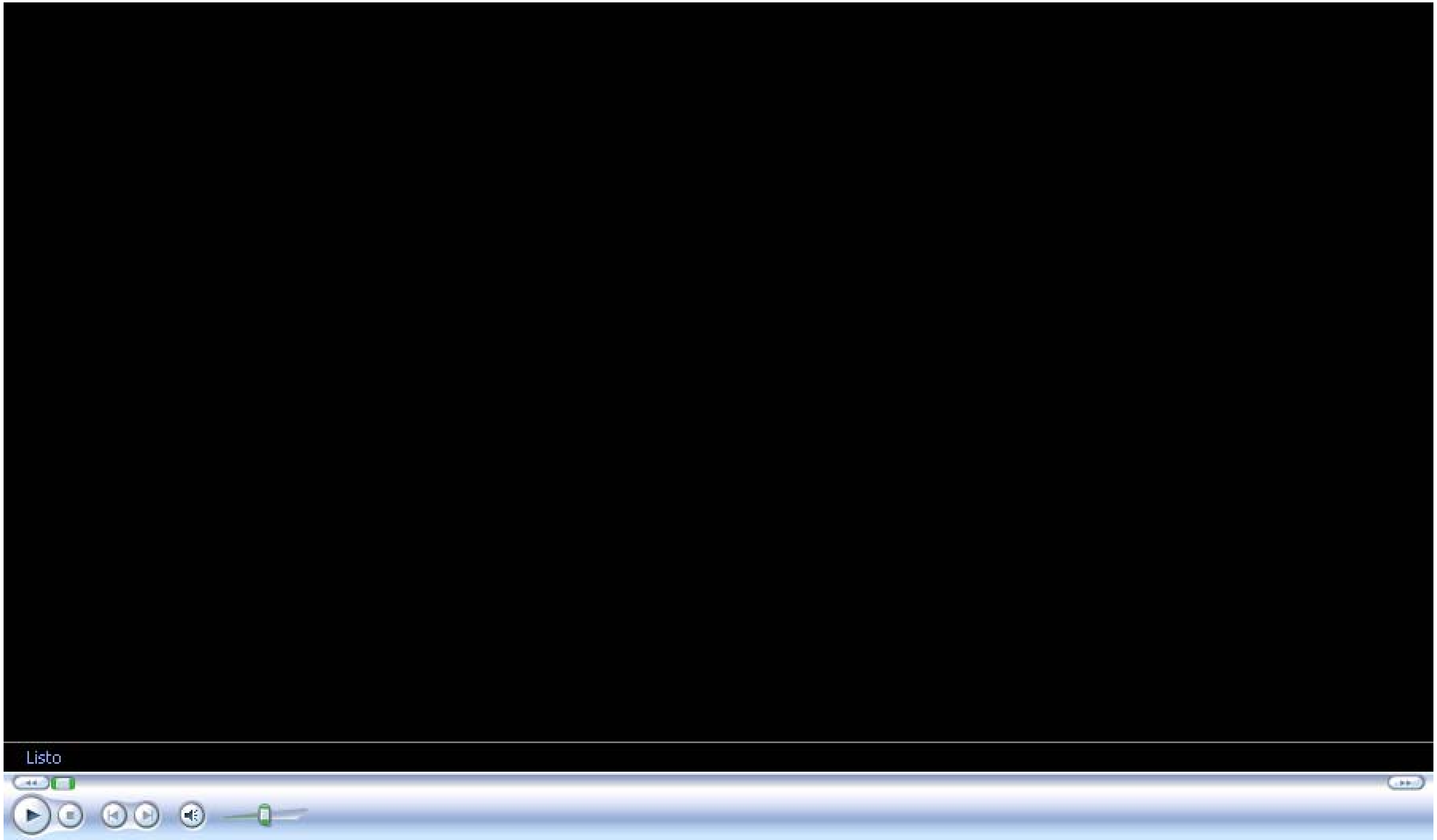


Amortized cost per step



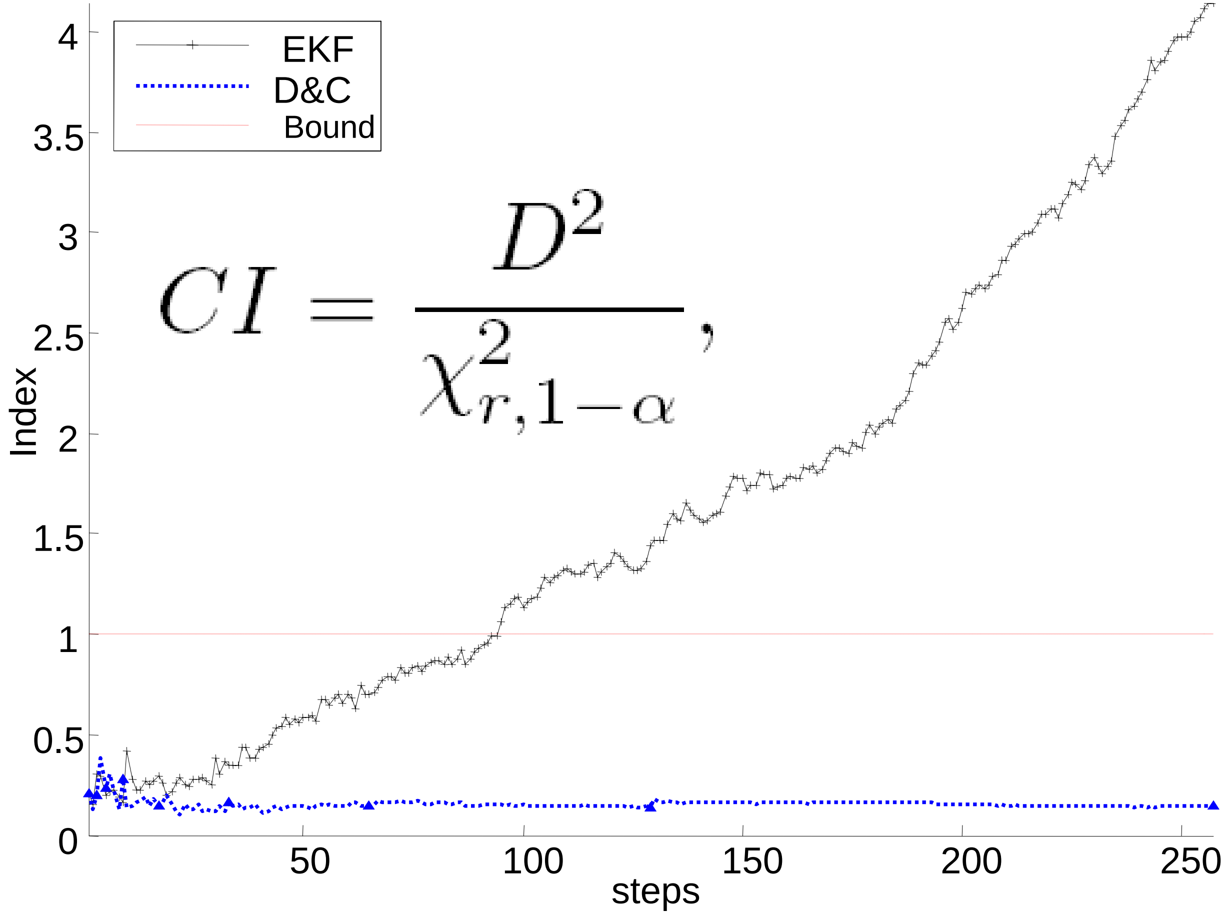
- The full map can be recovered at any time in a single $O(n^2)$ step
- But no part of the algorithm or process requires it

EKF SLAM consistency



D&C SLAM is more consistent!

Improved Consistency over EKF SLAM



$$CI = \frac{D^2}{\chi^2_{r,1-\alpha}},$$

D&C SLAM is always more consistent

6DOF SLAM

- Experimental setup

A bumblebee, a laptop and a firewire cable

Pure Stereo SLAM

180m indoor loop, CPS, Zaragoza

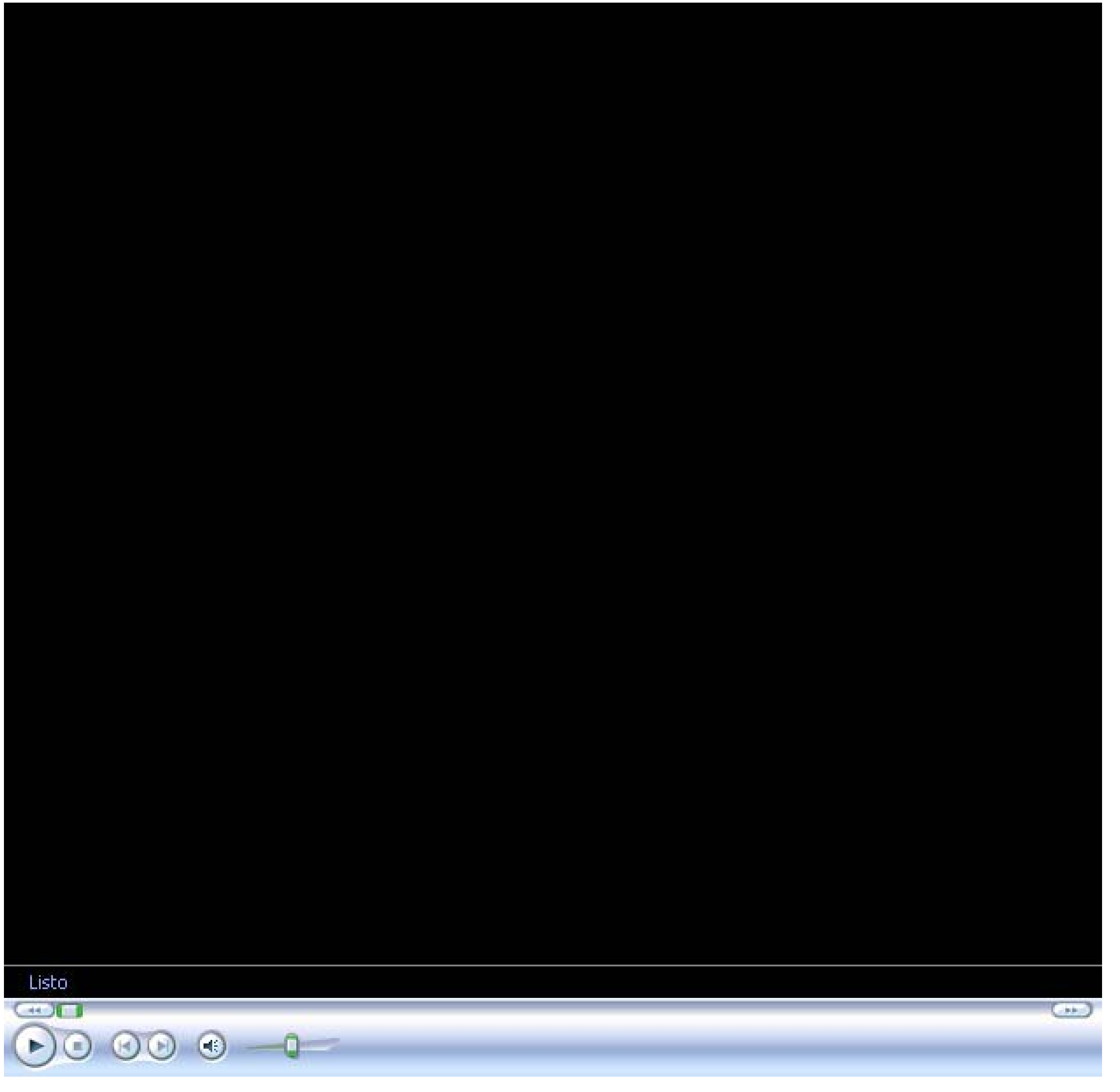
ROBOTICS UNIVERSIDAD DE ZARAGOZA

Pure Stereo SLAM

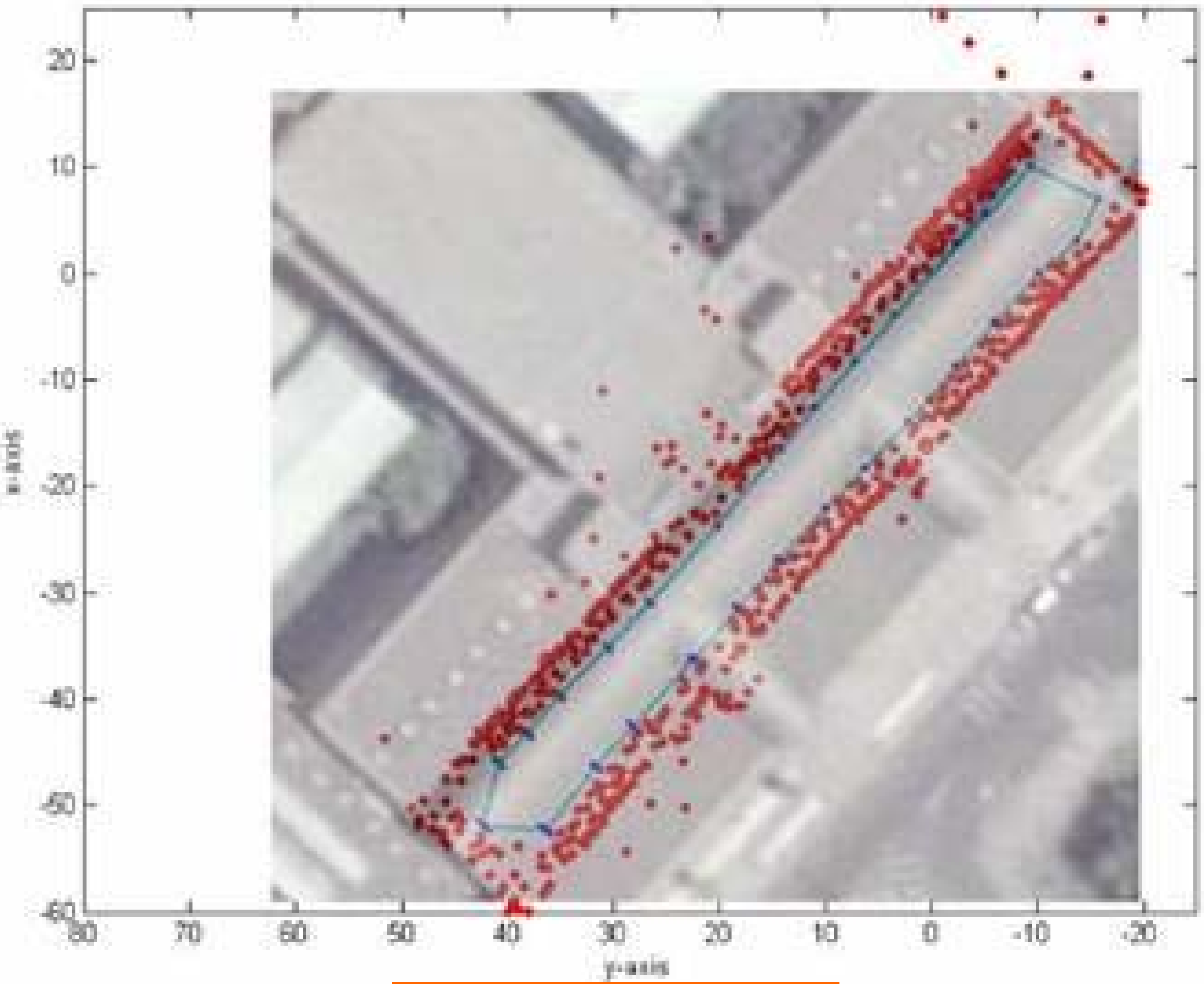
150m outdoor loop, public square, Zaragoza

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

6DOF SLAM with stereo



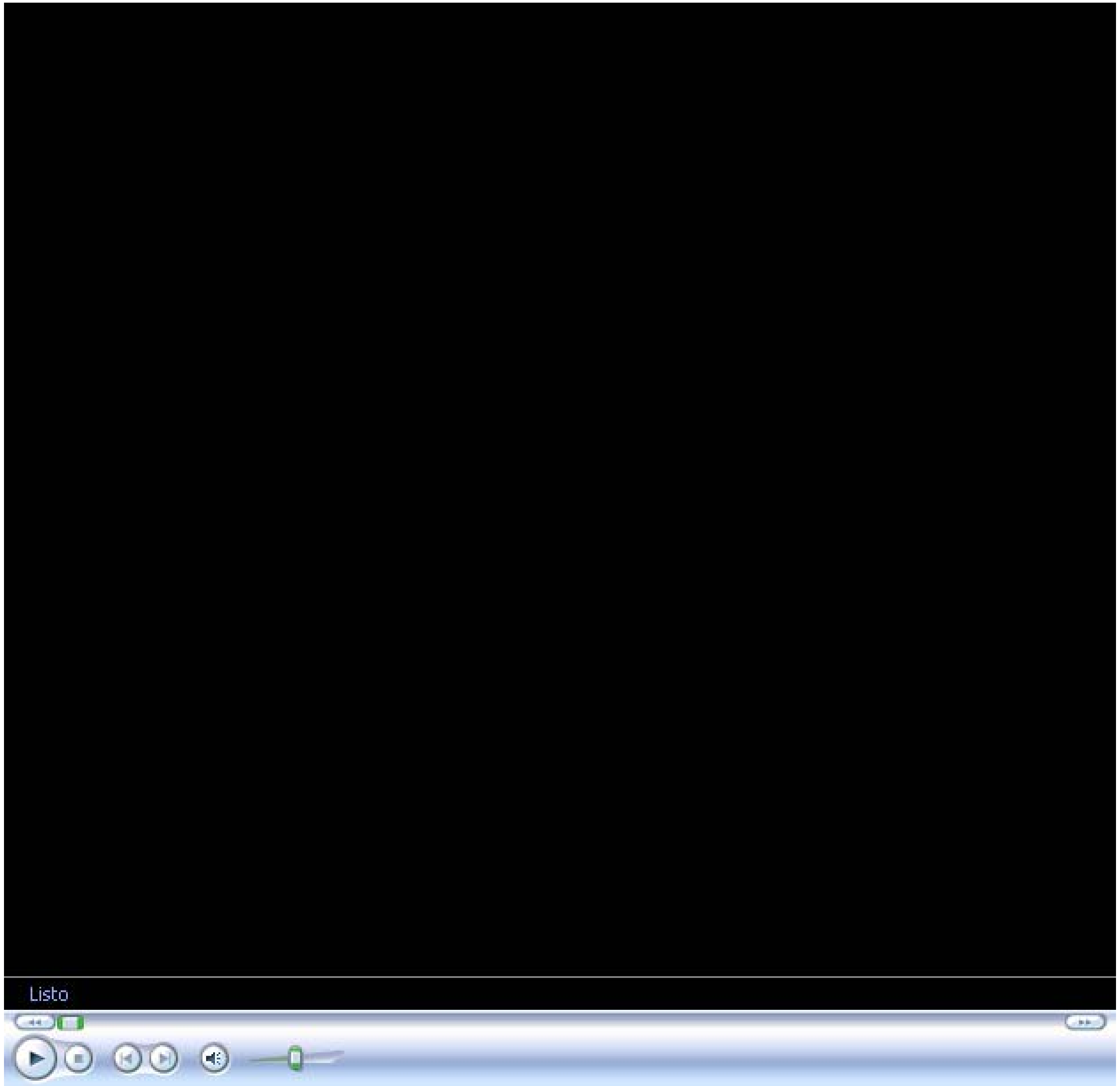
6Dof Stereo SLAM, indoors



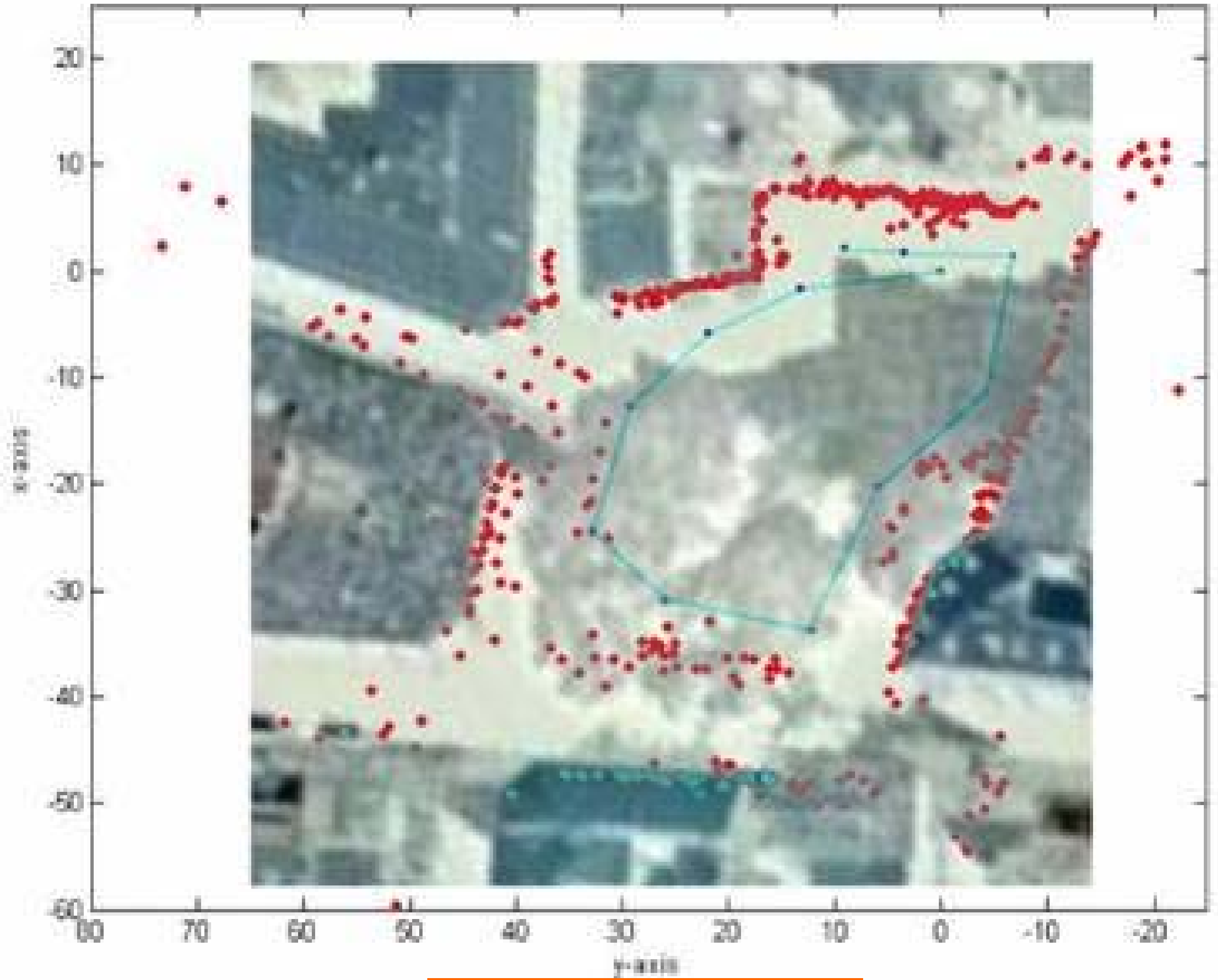
180 m loop

ROBOTICS UNIVERSIDAD DE ZARAGOZA

6DOF SLAM with stereo



6Dof Stereo SLAM, outdoors



150 m loop

Outline

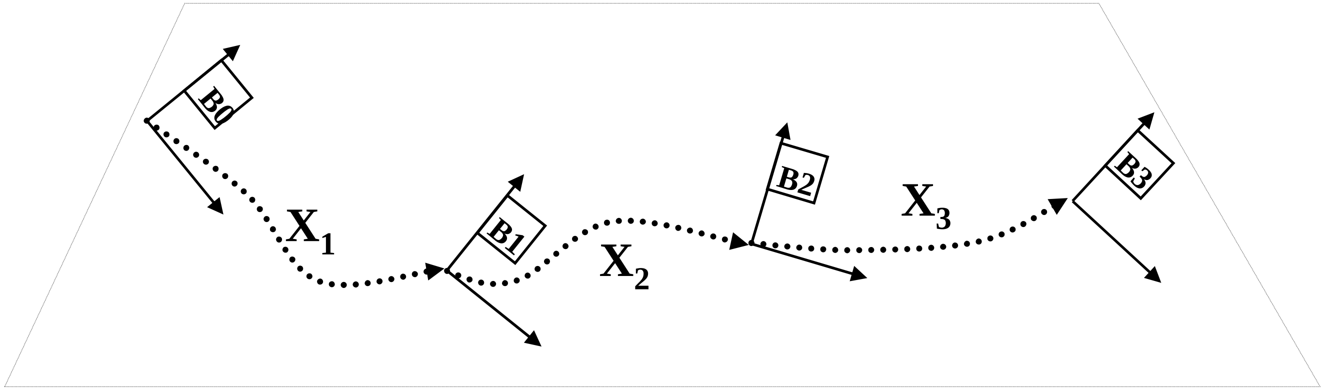
1. Basic EKF SLAM
   - Introduction: the need for SLAM
   - The basic EKF SLAM algorithm
   - Feature Extraction
   - Continuous Data Association
   - The Loop Closing Problem

2. Advanced EKF SLAM
   - Computational complexity of EKF SLAM
   - Consistency of the EKF SLAM
   - SLAM using local maps
     - Sequential Map Joining
     - Divide and Conquer SLAM
     - **Hierarchical SLAM**

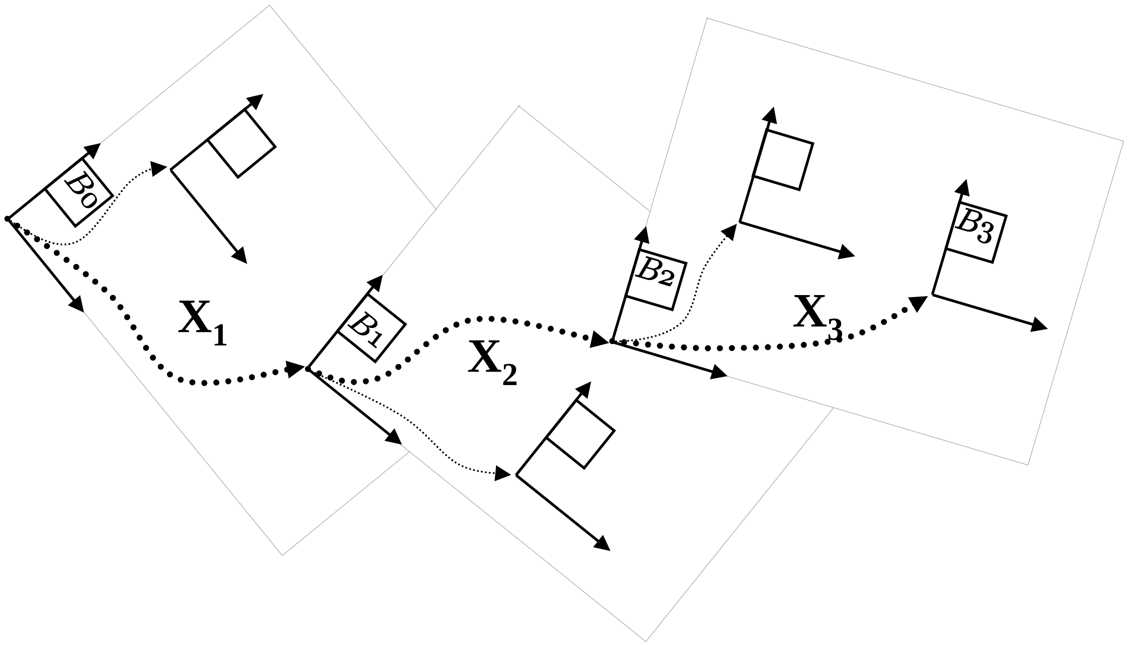
ROBOTICS UNIVERSIDAD DE ZARAGOZA

Hierarchical SLAM

- Global level: adjacency graph and relative stochastic map



- Local level: statistically independent local maps



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Hierarchical SLAM

- Local maps:

$$\hat{\mathbf{x}}^B_{\mathcal{F}} = \begin{bmatrix} \hat{\mathbf{x}}^B_R \\ \vdots \\ \hat{\mathbf{x}}^B_{F_n} \end{bmatrix}; \ \mathbf{P}_{\mathcal{F}} = \begin{bmatrix} \mathbf{P}^B_R & \cdots & \mathbf{P}^B_{RF_n} \\ \vdots & \ddots & \vdots \\ \mathbf{P}^B_{F_nR} & \cdots & \mathbf{P}^B_{F_nF_n} \end{bmatrix}$$

- Global relative map:

$$\hat{\mathbf{x}} = \begin{bmatrix} \vdots \\ \hat{\mathbf{x}}_i \\ \vdots \end{bmatrix}; \ \mathbf{P} = \begin{bmatrix} . & 0 & 0 \\ 0 & \mathbf{P}_i & 0 \\ 0 & 0 & . \end{bmatrix}$$

Block diagonal

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Hierarchical SLAM

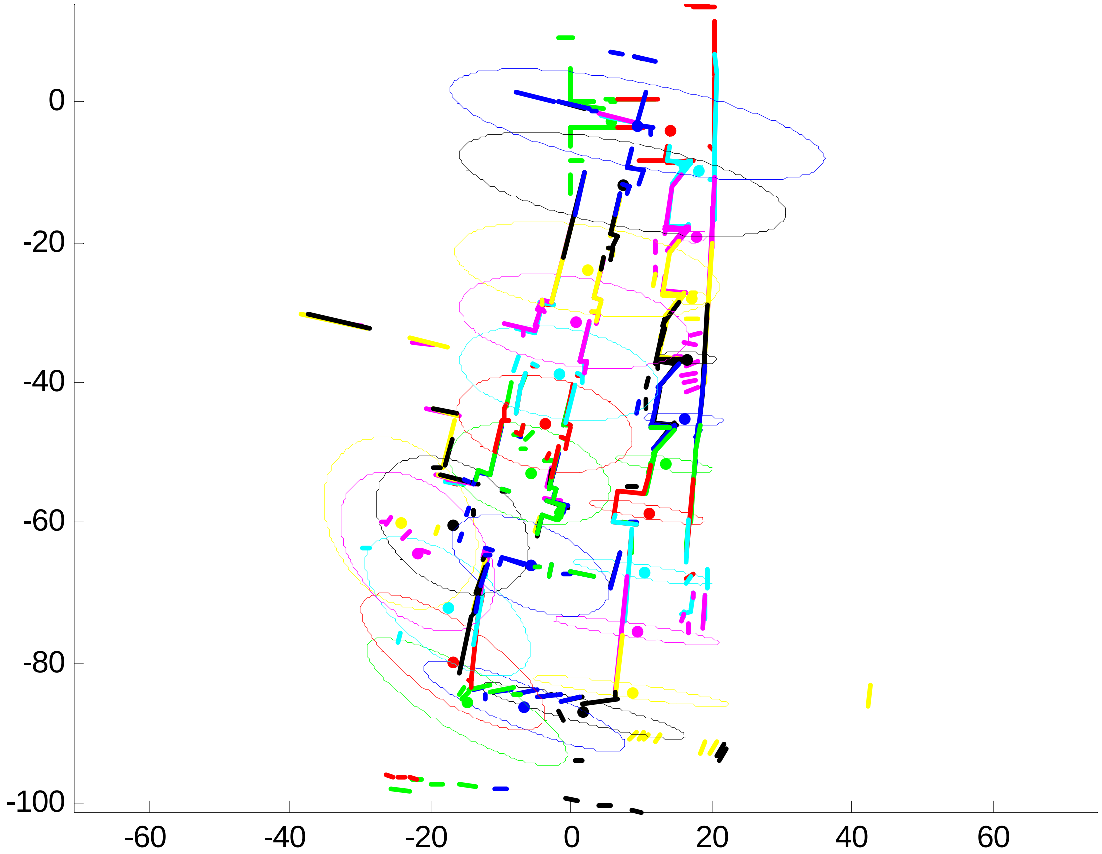
- Global relative map before loop closing:

$$\hat{\mathbf{x}} = \begin{bmatrix} \vdots \\ \hat{\mathbf{x}}_{i+1} \\ \vdots \\ \hat{\mathbf{x}}_j \end{bmatrix} ; \; \mathbf{P} = \begin{bmatrix} . & 0 & 0 & 0 \\ 0 & \mathbf{P}_{i+1} & 0 & 0 \\ 0 & 0 & . & 0 \\ 0 & 0 & 0 & \mathbf{P}_j \end{bmatrix}$$

- After loop closing:

$$\hat{\mathbf{x}} = \begin{bmatrix} \vdots \\ \hat{\mathbf{x}}_{i+1} \\ \hat{\mathbf{x}}_{ij} \\ \vdots \\ \hat{\mathbf{x}}_j \end{bmatrix} ; \; \mathbf{P} = \begin{bmatrix} . & 0 & 0 & 0 & 0 \\ 0 & \mathbf{P}_{i+1} & \mathbf{P}_{i+1,ij} & 0 & 0 \\ 0 & \mathbf{P}_{i+1,ij}^T & \mathbf{P}_{ij} & 0 & 0 \\ 0 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & 0 & \mathbf{P}_j \end{bmatrix}$$

Imposing loop constraints



$$\mathbf{h}(\mathbf{x}) \equiv \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_{n-1} \oplus \mathbf{x}_n$$

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Nonlinear constrained optimization

- Minimize corrections to the global map, subject to the loop constraint:

$$\min_{\mathbf{x}} \frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})^T \mathbf{P}^{-1}(\mathbf{x}-\hat{\mathbf{x}})$$
$$\mathbf{h}(\mathbf{x}) = \mathbf{0}$$

- Sequential Quadratic Programming (SQP) :

$$\mathbf{H}_i = \left[\left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_1}\right|_{\hat{\mathbf{x}}_i} \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_2}\right|_{\hat{\mathbf{x}}_i} \cdots \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_{n-1}}\right|_{\hat{\mathbf{x}}_i} \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_n}\right|_{\hat{\mathbf{x}}_i} \right]$$

$$\begin{aligned}\mathbf{P}_i &= \mathbf{P}_0 - \mathbf{P}_0\mathbf{H}_i^T\left(\mathbf{H}_i\mathbf{P}_0\mathbf{H}_i^T\right)^{-1}\mathbf{H}_i\mathbf{P}_0 \\ \hat{\mathbf{x}}_{i+1} &= \hat{\mathbf{x}}_i - \mathbf{P}_i\mathbf{P}_0^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_0) - \mathbf{P}_0\mathbf{H}_i^T\left(\mathbf{H}_i\mathbf{P}_0\mathbf{H}_i^T\right)^{-1}\hat{\mathbf{h}}_i\end{aligned}$$

  - » Iterate until convergence

Nonlinear constrained optimization

- A more efficient version:

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_0 + \mathbf{P}_0 \mathbf{H}_i^T \left(\mathbf{H}_i \mathbf{P}_0 \mathbf{H}_i^T \right)^{-1} \left(\mathbf{H}_i \left(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_0 \right) - \hat{\mathbf{h}}_i \right)$$

  » Iterate until convergence

- Complexity:
  - P_0 is block diagonal
  - H_i is sparse with nonzeros only for the maps in the loop
  - The iteration is linear with the number of maps in the loop
- Convergence:
  - Converges in 2 or 3 iterations (for loops around 300m)
  - For bigger errors, may it converge to a local minimum ??

Nonlinear constrained optimization

- Generalization to closing several loops simultaneously:

$$\mathbf{h}_j(\mathbf{x}) = \mathbf{x}_{j_1} \oplus \mathbf{x}_{j_2} \oplus \cdots \oplus \mathbf{x}_{j_{n_j-1}} \oplus \mathbf{x}_{j_{n_j}} = 0$$

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_l \end{bmatrix}$$

Iterated Extended Kalman Filter

- Jacobian of the measurement function:

$$\mathbf{H}_i = \left[\left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_1}\right|_{\hat{\mathbf{x}}_i} \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_2}\right|_{\hat{\mathbf{x}}_i} \cdots \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_{n-1}}\right|_{\hat{\mathbf{x}}_i} \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_n}\right|_{\hat{\mathbf{x}}_i} \right]$$

- Iterated EKF equations:

$$\begin{aligned} \mathbf{P}_i &= \mathbf{P}_0 - \mathbf{P}_0\mathbf{H}_i^T \left[\mathbf{H}_i\mathbf{P}_0\mathbf{H}_i^T + \mathbf{P}_z\right]^{-1} \mathbf{H}_i\mathbf{P}_0 \\ \hat{\mathbf{x}}_{i+1} &= \hat{\mathbf{x}}_i - \mathbf{P}_i\mathbf{P}_0^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_0) + \mathbf{P}_0\mathbf{H}_i^T \left(\mathbf{H}_i\mathbf{P}_0\mathbf{H}_i^T + \mathbf{P}_z\right)^{-1} \left(\mathbf{z} - \hat{\mathbf{h}}_i\right) \end{aligned}$$

- With exact loop constraint, z = 0 and P_z = 0, IEKF is equivalent to nonlinear optimization with SQP

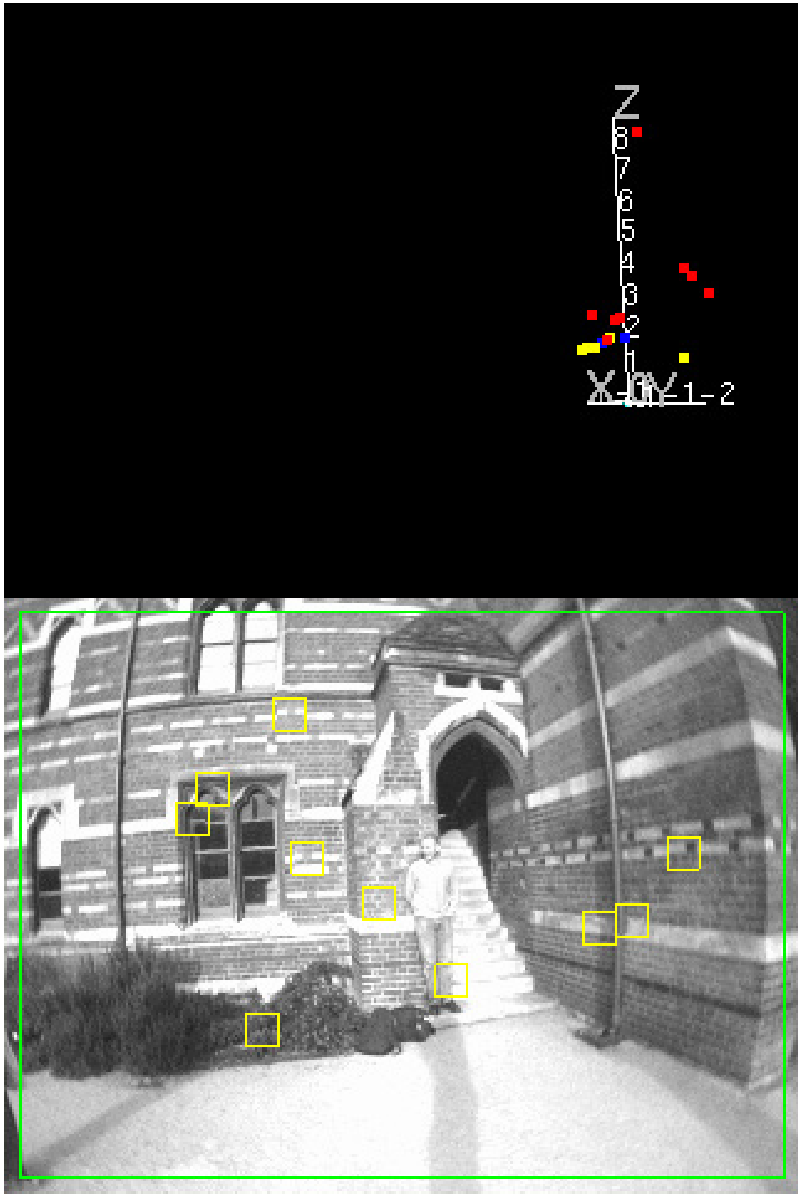
C. Estrada, J. Neira, J.D. Tardós, *Hierarchical SLAM: real-time accurate map-ping of large environments*. IEEE Transactions on Robotics, 2005

Experiment

- Double blind review experimental set

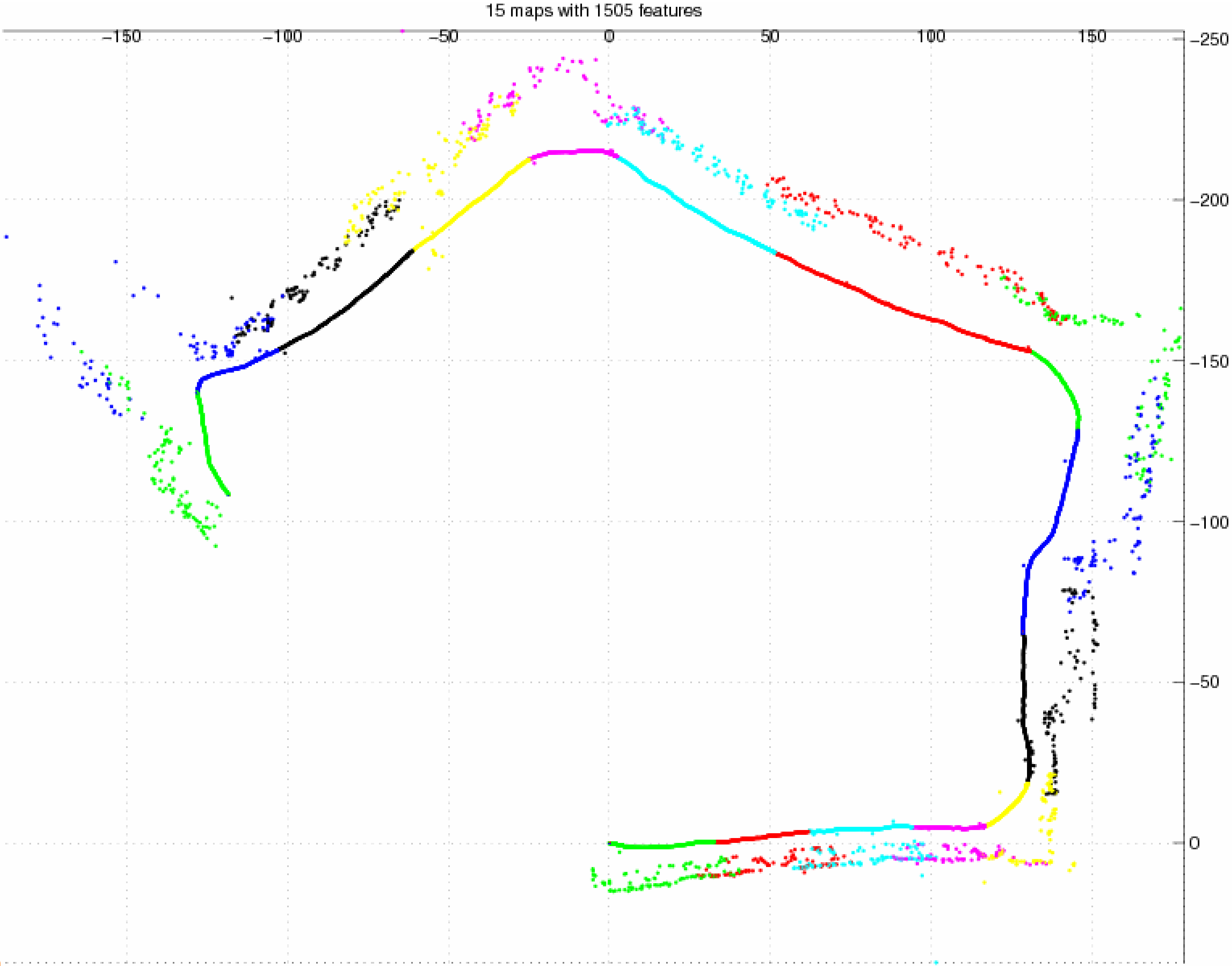
L. Clemente, A. Davison, I. Reid, J. Neira and J.D. Tardós
Mapping Large Loops with a Single Hand-Held Camera
2007 Robotics: Science and Systems, June 27-30, Atlanta, USA

Keble College, Oxford

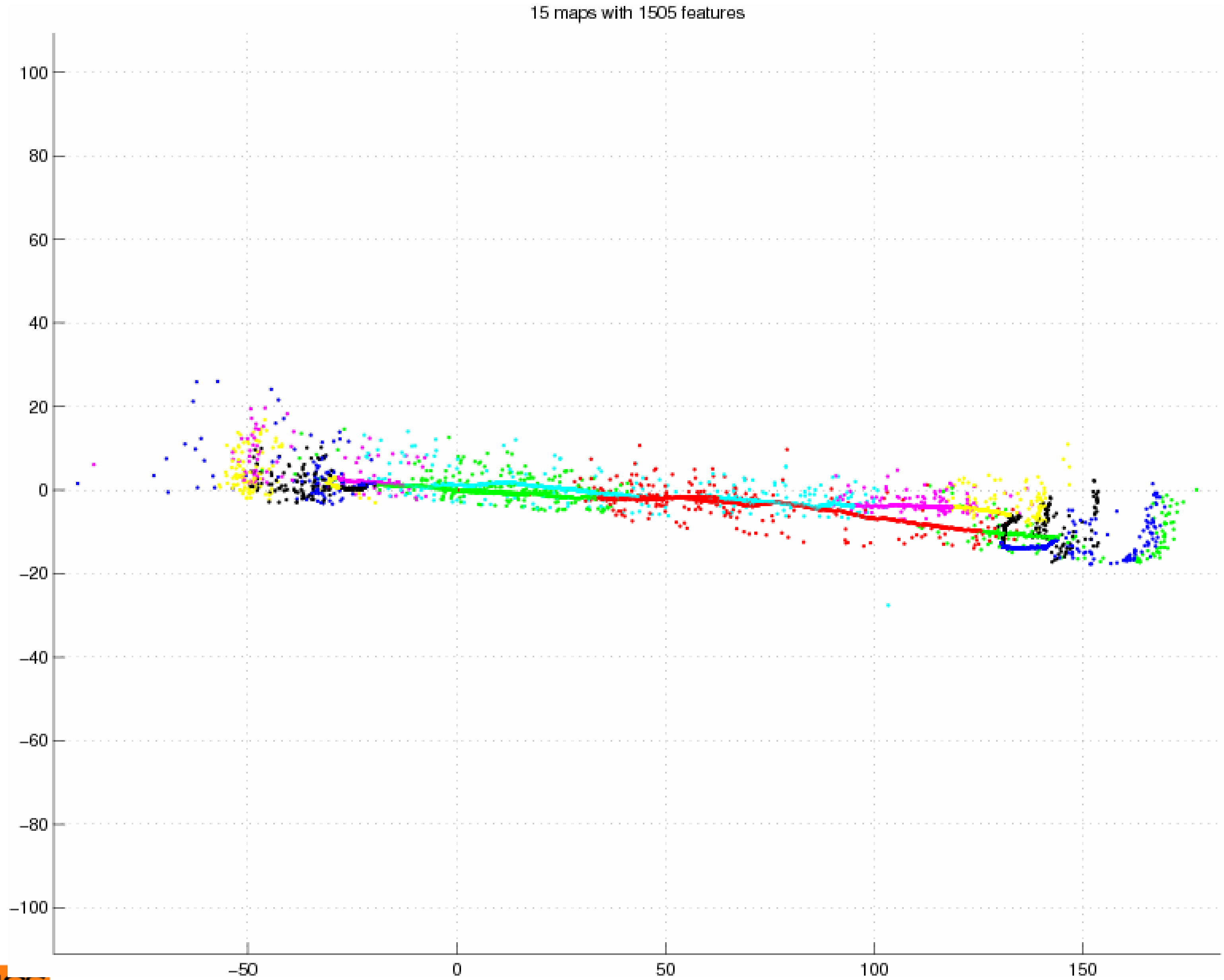


ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Sequence of local maps

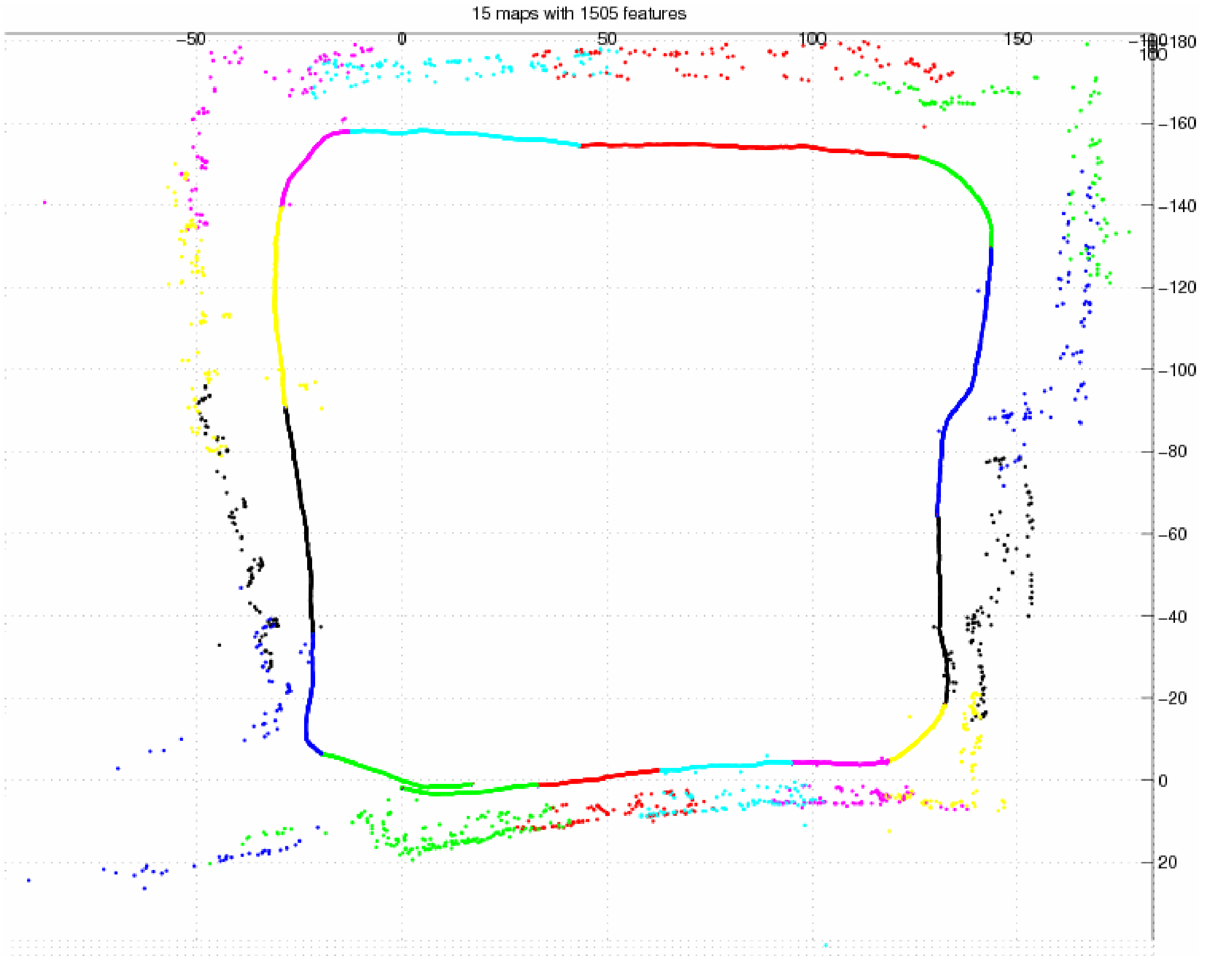


Sequence of local maps

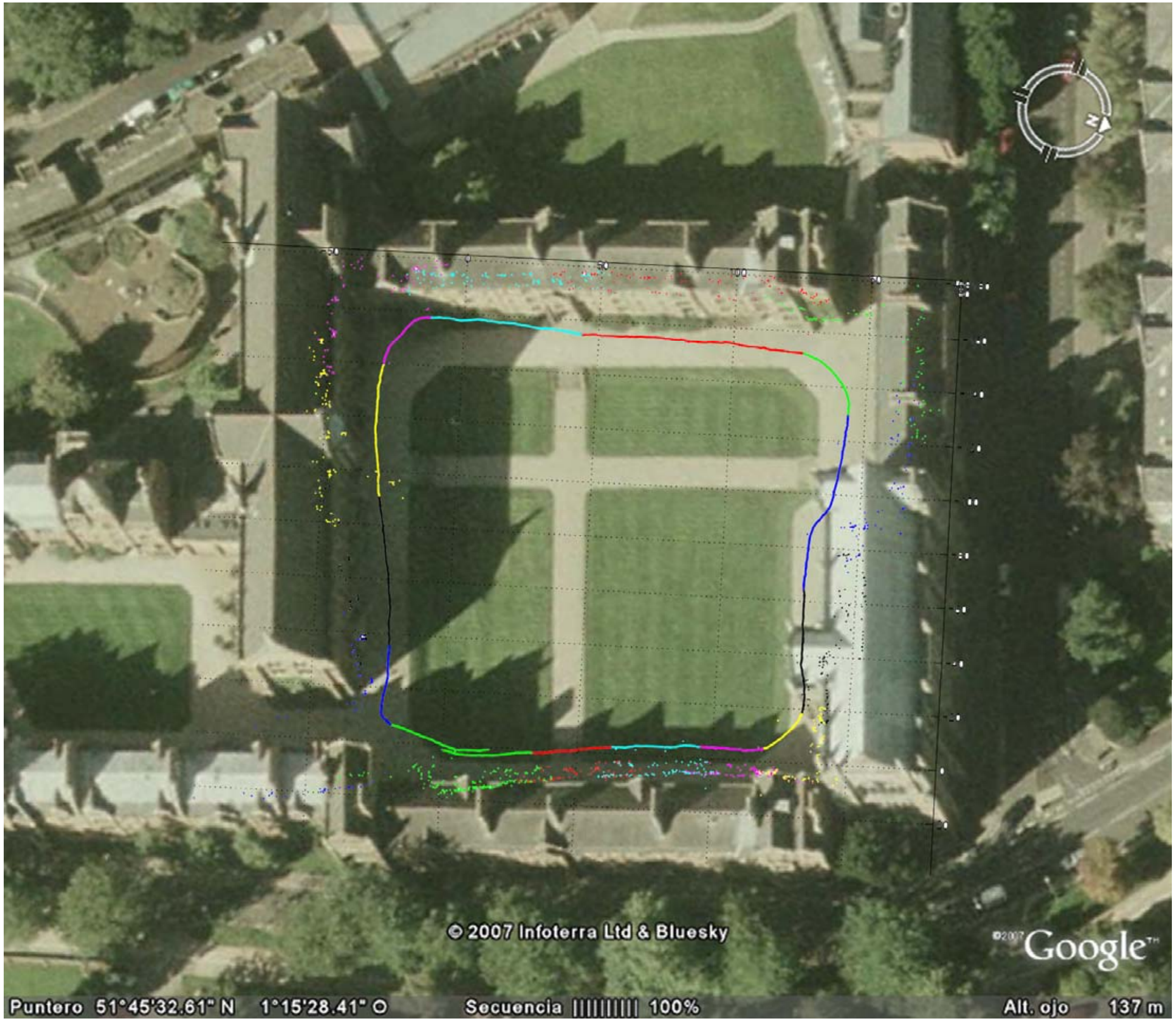


ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Loop closing



Keble College, Oxford (290m)



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Results

- Local Map building in real-time @30Hz
  - 60 features per map using inverse depth
  - Bigger maps if converted to (x,y,z)

- Loop optimization takes 800ms (6 iterations)

- The scale drifts along the map

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Conclusions

	Loop 30m	Loop 300m	Longer loop
EKF-SLAM Nearest Neighbor	weak	--- inconsistent	---
EKF-SLAM Joint Compatib.	very good	--- inconsistent	---
Map Joining Joint Compatib.	**excellent**	weak	---
Hierarchical SLAM Relocation	overkill	**excellent**	future work

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Recommended Readings

- J.D. Tardós, J. Neira, P. Newman, and J. Leonard. **Robust Mapping and Localization in Indoor Environments using Sonar Data**, Int. J. Robotics Research, Vol. 21, No. 4, April 2002, pp 311 –330
- J.A. Castellanos, J. Neira, J.D. Tardós, **Limits to the Consistency of EKF-based SLAM**, 5th IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon, July 2004.
- C. Estrada, J. Neira, J.D. Tardós, **Hierarchical SLAM: real-time accurate mapping of large environments**. To appear in the IEEE Transactions on Robotics
- Grimson, W. E. L., **Object Recognition by Computer: The Role of Geometric Constraints,** The MIT Press, Cambridge, Mass., 1990
- José A. Castellanos and Juan D. Tardós, **Mobile Robot Localization and Map Building: A Multisensor Fusion Approach**, Kluwer Ac. Pub., Boston, 1999
- J. Neira and J.D. Tardós, **Data Association in Stochastic Mapping Using the Joint Compatibility Test**, IEEE Trans. Robotics and Automation, vol. 17, no. 6, pp. 890-897, Dec 2001.
- P.M. Newman, J.J. Leonard, J. Neira and J.D. Tardós: **Explore and Return: Experimental Validation of Real Time Concurrent Mapping and Localization**. IEEE Int. Conf. Robotics and Automation, May, 2002.
- J. Neira, J.D. Tardós, J.A. Castellanos, **Linear time vehicle relocation in SLAM**. 2003 IEEE Int. Conf. Robotics and Automation, Taipei, Taiwan, May, 2003.
- D. Ortín, J. Neira, J.M.M. Montiel, **Relocation using Laser and Vision.** 2004 IEEE Int. Conf. Robotics and Automation, New Orleans, USA, April, 2004.

INFORMATION

1st Summer School 2002, Stockholm

http://www.cas.kth.se/SLAM/

2nd Summer School 2004, Toulouse

http://www2.laas.fr/SLAM/

3rd Summer Schhol 2006, Oxford

http://www.robots.ox.ac.uk/~SSS06/

ROBOTICS UNIVERSIDAD DE ZARAGOZA

BIBLIOGRAPHY