SLAM: Simultaneous Robot Localization and Mapping

Jose A. Castellanos
Jose Neira

{jacaste, jneira} at unizar.es
Computer Science and Systems Engineering Dept.
University of Zaragoza
Spain

UZ – Robotics Group

- Founded in 1982
- 13 Faculty Members
- 14 PhD Students, 20 Graduate Students
- Research lines
  - Multisensor Fusion and Integration
  - Localization and Map Building, SLAM
  - Planning and navigation
  - Multi-robot systems
  - Perception
  - Computer vision
  - 3D scene reconstruction
  - Embedded systems

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

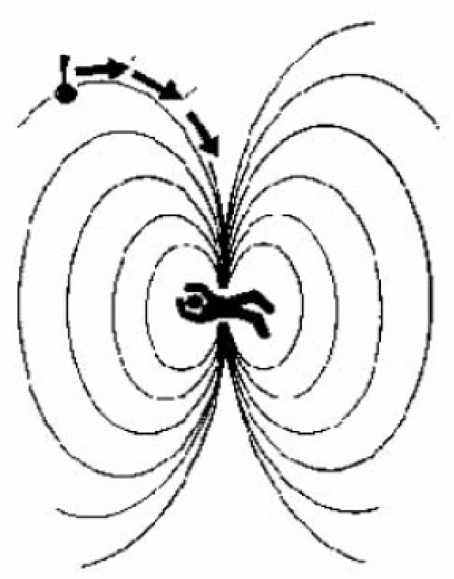
Recent Projects

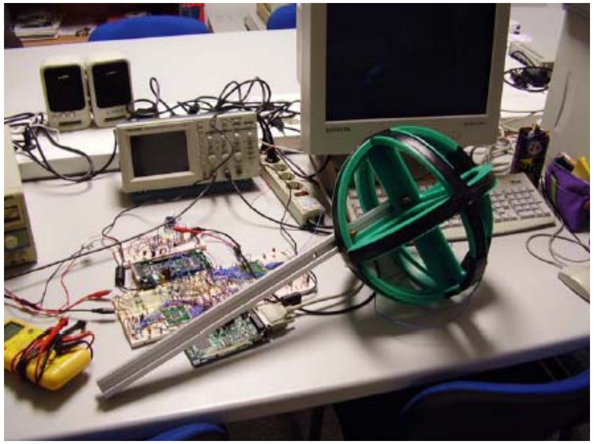
- Design and Development of an Electrical Wheelchair with Capability of Assisted Driving and Autonomous Navigation (2001-03)

Recent Projects

- Express: Automated Exploration Techniques for Rescue Applications (2004-07)
  - Road Tunnel Rescue

  - Snow Avalanches Rescue





Current Projects

- **URUS** (Ubiquitous Networking Robotics in Urban Settings), IST-1-045062-URUS-STP
  - Cooperation between Humans and Robots in Urban Environments
- **NERO**, DPI2006-07928
  - Networked Robotics
- **SLAM6DOF**, DPI2006-13578
  - Portable Systems for 3D Mapping
- **EVA**, DPI2006-15630-C02-02
  - Biological Human – Robot Interaction

PhD course 5007439

- Professors
  - José A. Castellanos (jacaste at unizar.es)
  - José Neira (jneira at unizar.es)

- Invited professor
  - Nando de Freitas (nando at cs.ubc.ca)
    - » Computer Science Department
    - » University of British Columbia, Canada

- Invited speakers
  - Rubén Martínez-Cantín (rmcantin at unizar.es)
  - Lina Paz Pérez (linapaz at unizar.es)
  - Pedro Piniés (ppinies at unizar.es)

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Timetable

Room A.22 (Ada Byron Building)

	Mo 16/04/07	Tu 17/04/07	We 18/04/07	Th 19/04/07	F 20/04/07
9:30	Presentation			Laboratory	SLAM research
10:00	Mobile robots				in Zaragoza
10:30					
11:00					
11:30	Localization and		Large-Scale SLAM		SLAM research
12:00	Mapping		Multivehicle SLAM		in Zaragoza
12:30					
13:00					
13:30					
15:00	Localization and	Data Association	Seminar 1	Seminar 2	Laboratory
15:30	Mapping	for Continuos SLAM,	Nando de Freitas	Nando de Freitas	
16:00		Loop Closing and		(Room A23)	
16:30		Relocation			
17:00					

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Reading List

- J. Borenstein, H. R. Everett and L. Feng, “Where am I ? Sensor and methods or mobile robot positioning”, TechRep, Univ. Michigan, 1996 (online).
- J. J. Leonard and H. F. Durrant-Whyte, “Directed sonar sensing for mobile robot navigation”, Kluwer Academic Publishers, 1992.
- J. A. Castellanos and J. D. Tardós, “Mobile robot localization and map building: A multisensor fusion approach”, Kluwer Academic Publishers, 1999.
- U. Nehmzow, “Mobile robotics: A practical introduction”, Applied Computing, Springer-Verlag, London, 2000.
- D. Kortenkamp et al (eds.), “AI and mobile robots”, AAAI Press, MIT Press, 1998
- R. Siegwart and I. R. Nourbakhsh, “Introduction to Autonomous Mobile Robots”, Bradford Books, 2004.
- …
- IEEE Transactions on Robotics
- IEEE Transactions on Robotics and Automation
- The International Journal of Robotics Research
- ...
- IEEE International Conference on Robotics and Automation (ICRA)
- IEEE/RSJ International Conference on Robots and Systems (IROS)
- Robotics, Science and Systems (RSS)
- International Symposium on Robotics Research (ISRR)
- ...
- EURON (www.euron.org)
- IFR (www.ifr.org)

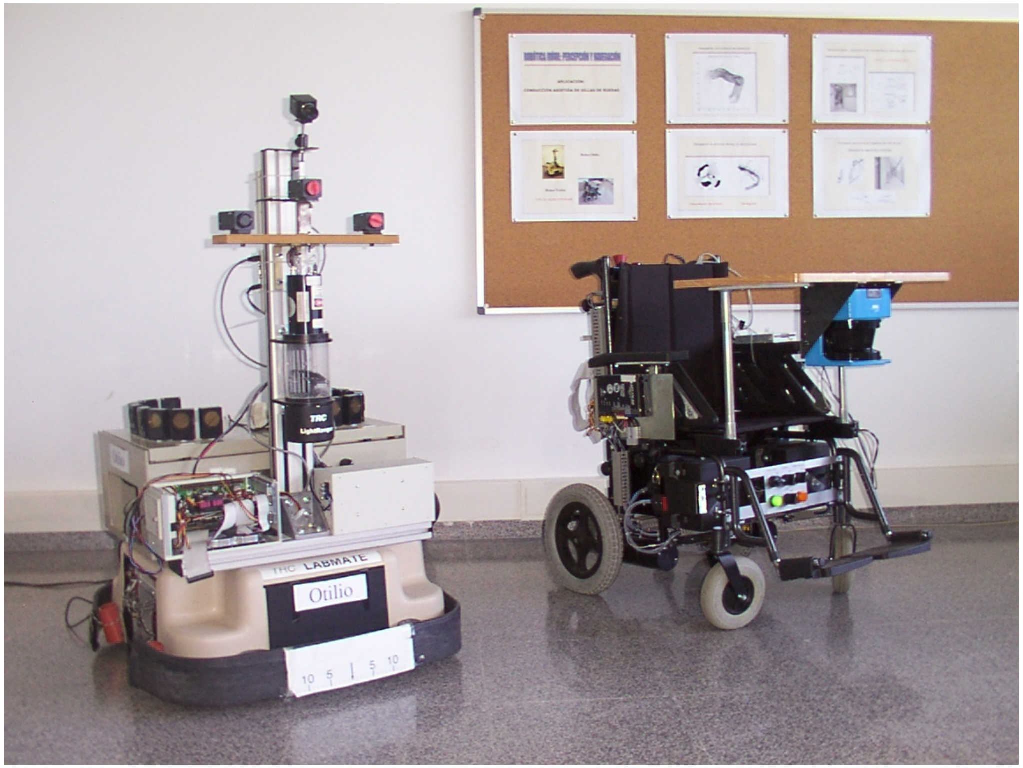
INDEX

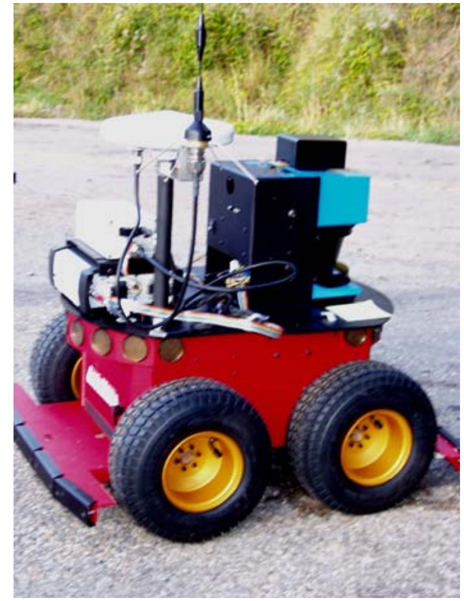
1. **Mobile Robots**
   1. **Common structures**
   2. **Perception-Decision-Action Loop**
   3. **Sensors for Mobile Robots**
      1. **Internal sensors: Odometry**
      2. **External sensors: Laser and Sonar**

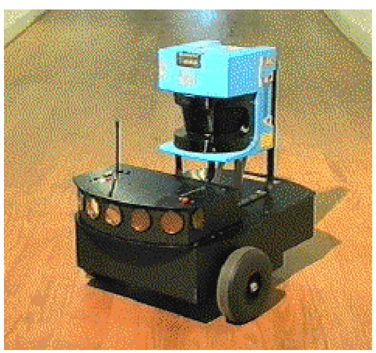
2. Localization and Mapping
   1. Overview. Kalman Filter
   2. Localization using Odometry
   3. Map-based Localization
   4. SLAM

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Mobile Robots

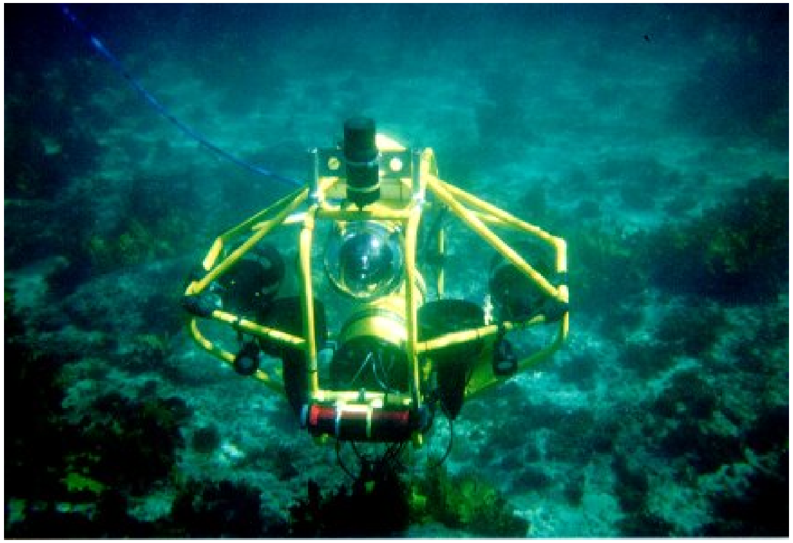






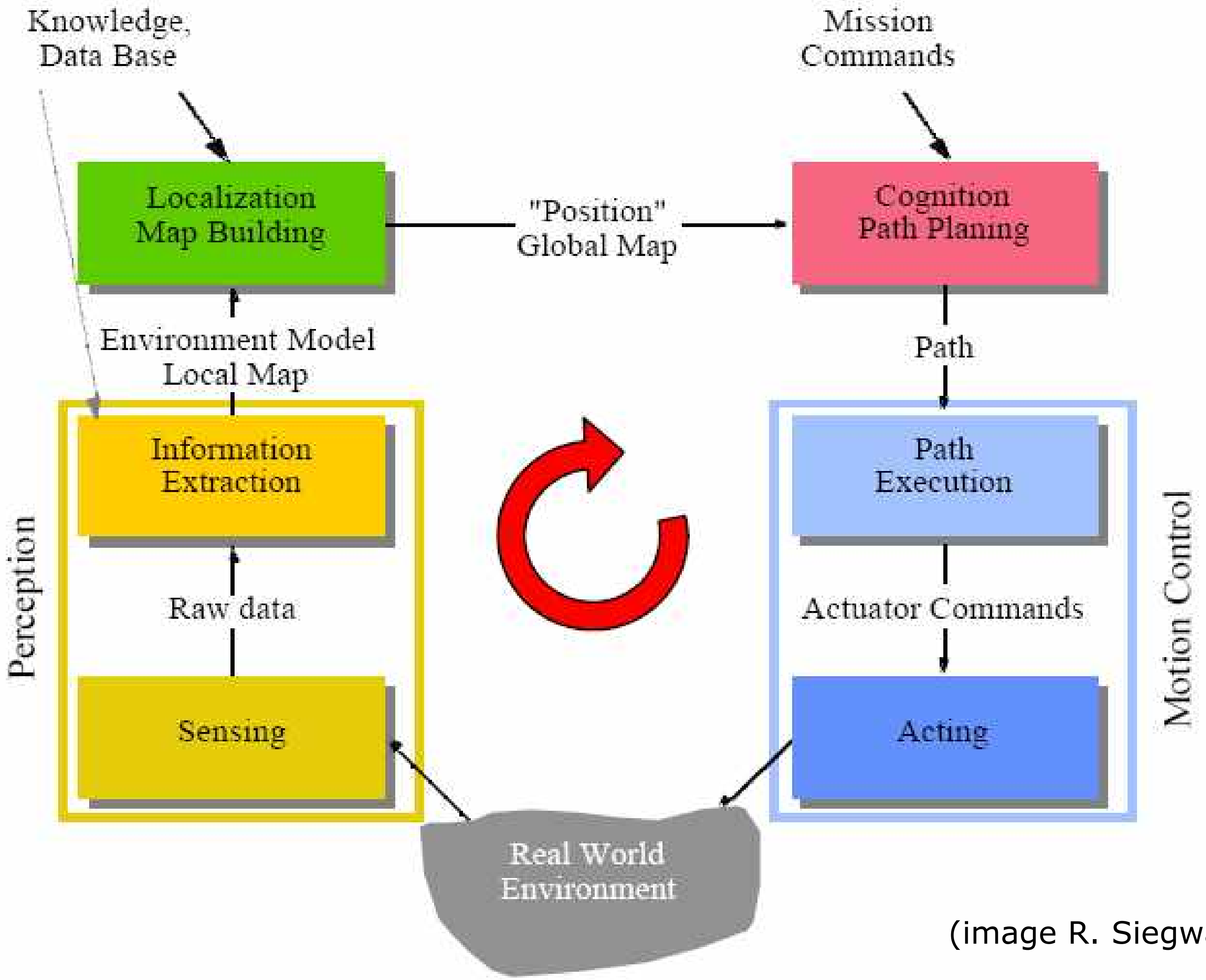
Mobile Robots

Mobile Robots



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Perception-Decision-Action Loop

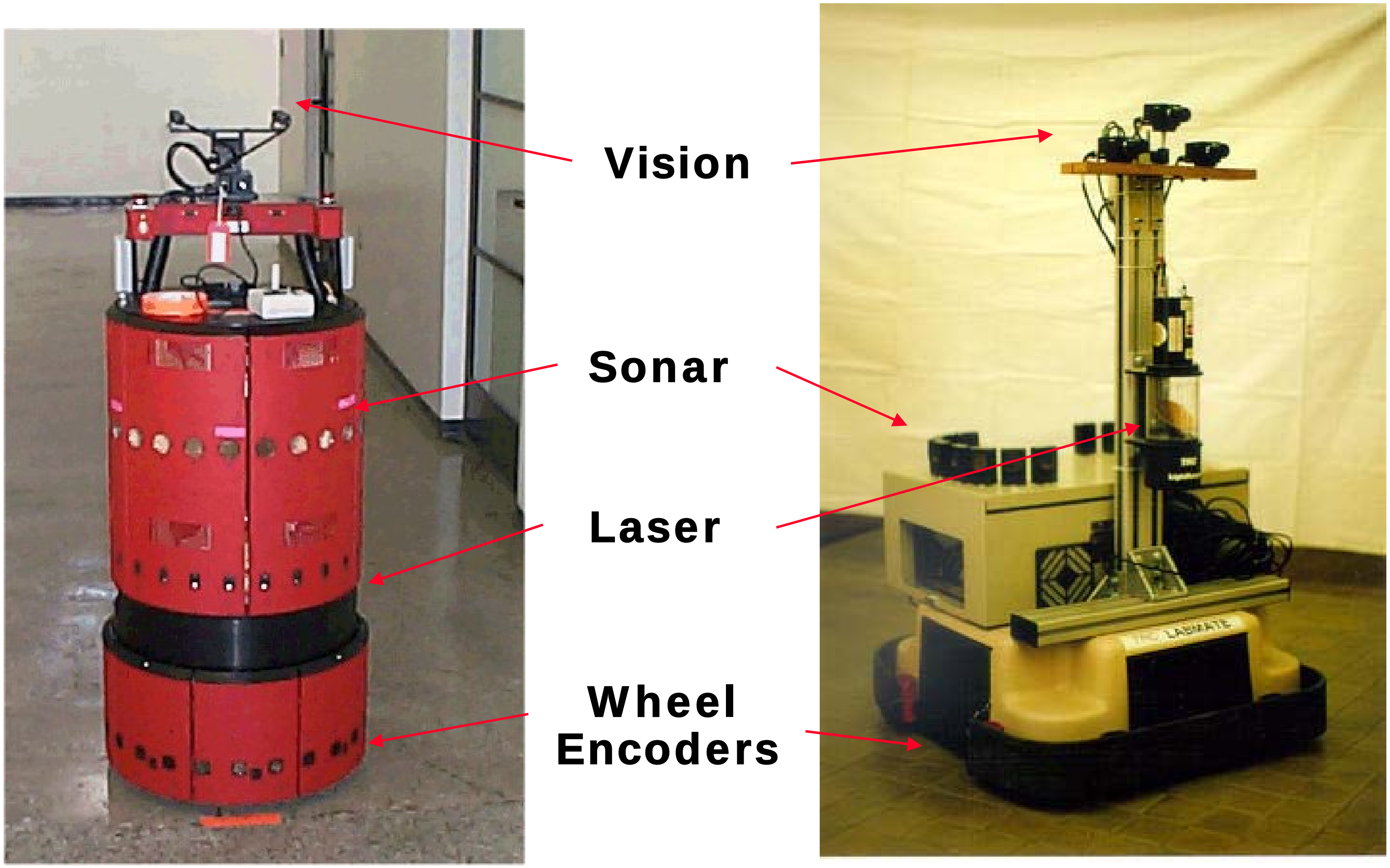


(image R. Siegwart)

Sensors for Mobile Robots

- Contact sensors: Bumpers
- Internal sensors (Dead-reckoning)
  - Wheel Encoders, Odometry
  - Accelerometers (spring-mounted masses)
  - Gyroscopes (spinning mass, laser light)
  - Compasses, inclinometers (earth magnetic field, gravity)
- Proximity sensors
  - Sonar (time of flight)
  - Radar (phase and frequency)
  - Laser range-finders (triangulation, ToF, phase)
  - Infrared (intensity)
- Visual sensors: Cameras
- Satellite-based sensors: GPS, Galileo?

Sensors for Mobile Robots



B21, MIT

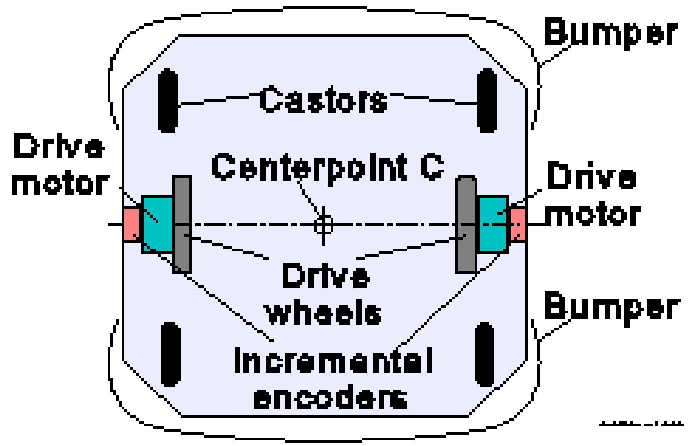
Otilio, UZ

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Internal sensors: Odometry

Usual configuration: **Differential Drive**

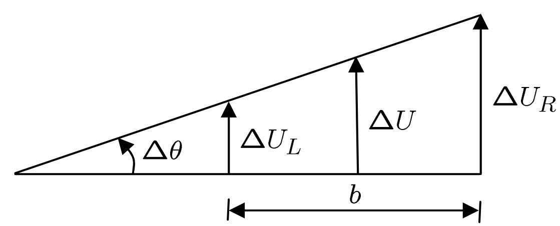
$$c_m = \frac{\pi D_n}{n C_e}$$



$$\begin{aligned} D_n &= \text{Nominal wheel diameter} \\ C_e &= \text{Encoder resolution} \\ n &= \text{Gear ratio} \end{aligned}$$

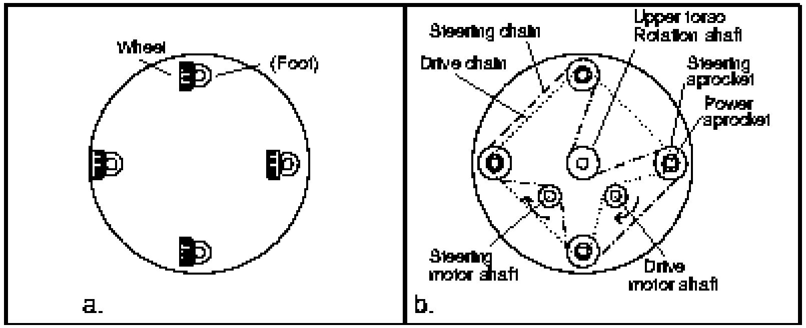
$$\left.\begin{aligned} \Delta U_L = c_m N_L \\ \Delta U_R = c_m N_R \end{aligned}\right\} \Rightarrow \Delta U = \frac{\Delta U_R + \Delta U_L}{2} \; ; \quad \Delta\theta = \frac{\Delta U_R - \Delta U_L}{b}$$

$$\left\{\begin{aligned} x_k &= x_{k-1} + \Delta U \cos(\theta_{k-1} + \Delta\theta) \\ y_k &= y_{k-1} + \Delta U \sin(\theta_{k-1} + \Delta\theta) \\ \theta_k &= \theta_{k-1} + \Delta\theta \end{aligned}\right.$$

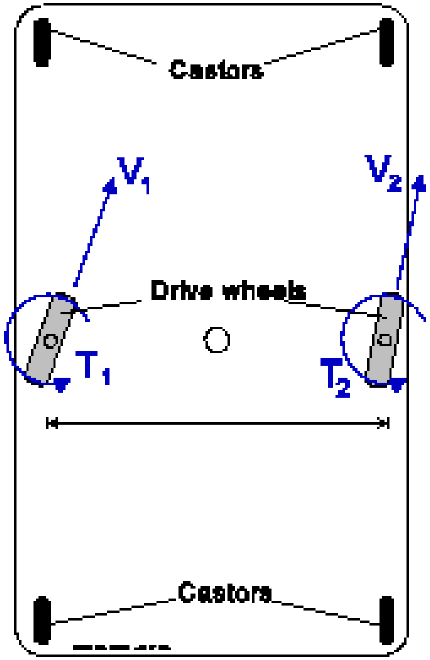


Odometry

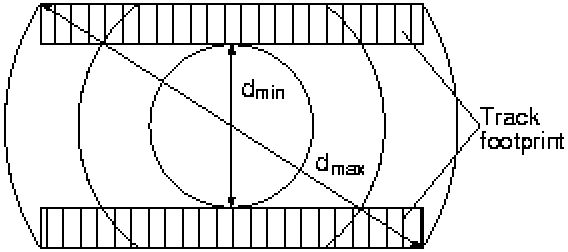
Synchro-drive



Multi-DoF vehicles

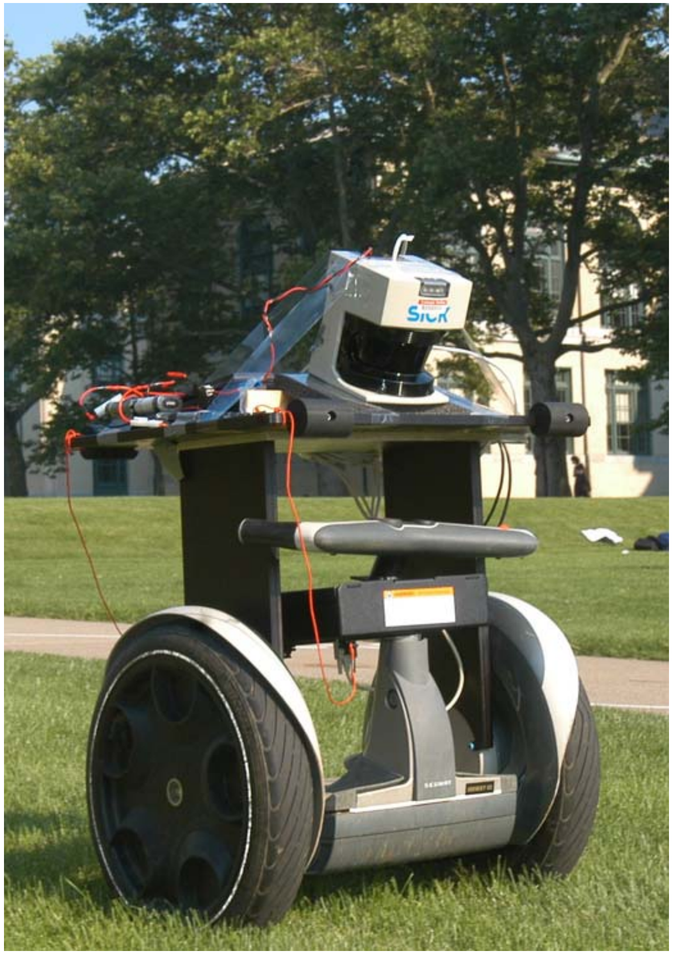


Tracked vehicles



Odometry

Shrimp III

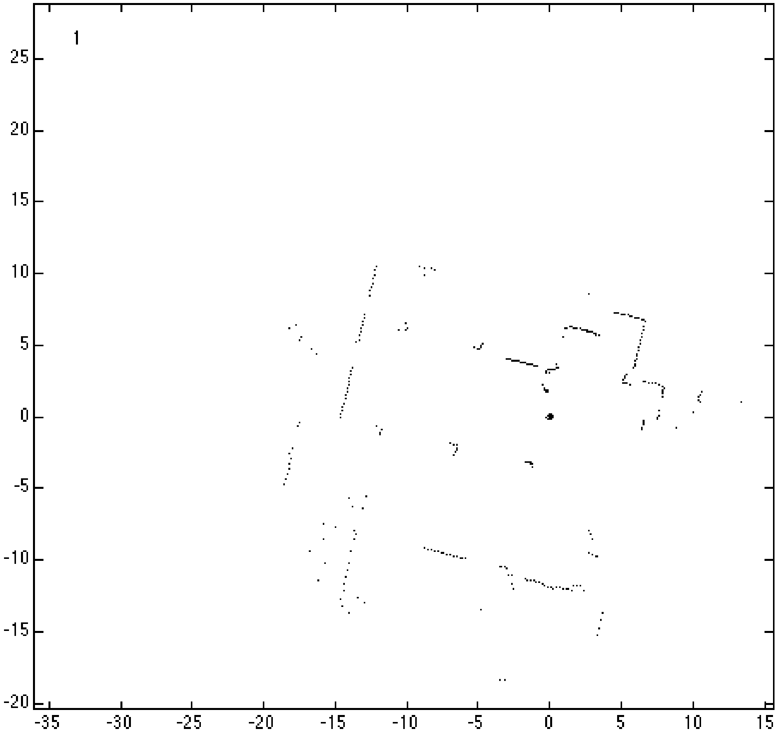


SegBot

Groundhog

Odometry: Limitations

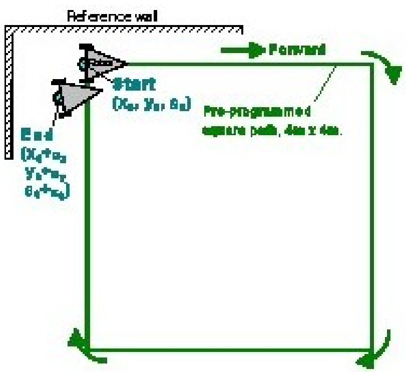
- Systematic errors
  - Unequal wheel diameters
  - Uncertain baseline
  - Wheel misalignment
  - Limited encoder resolution
  - Limited sampling resolution
- Accidental errors
  - Uneven floors
  - Unexpected objects
  - Wheel slippage

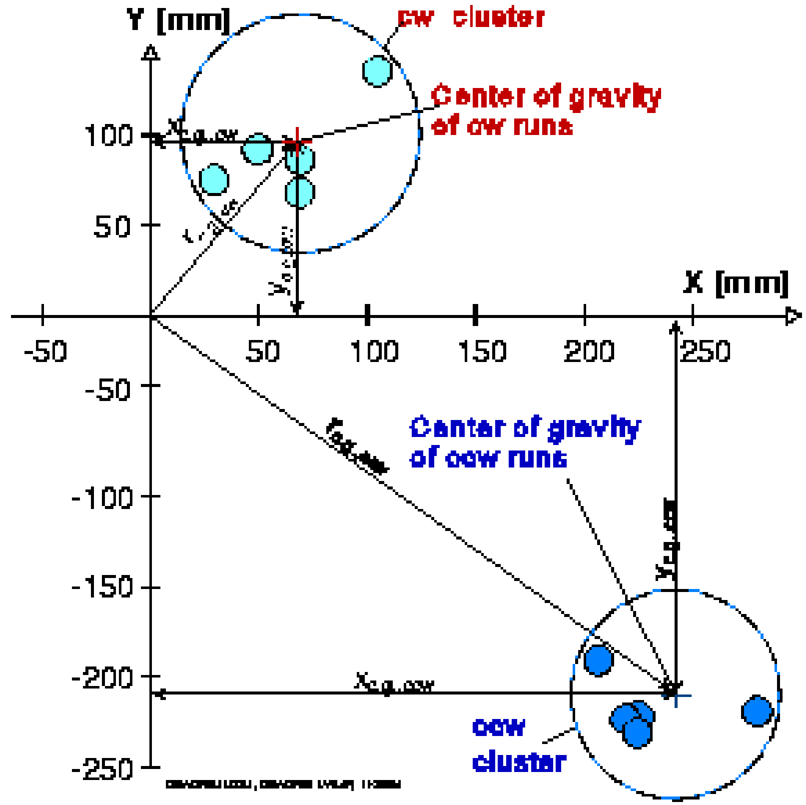


(image K.O.Arras)

UMBmark

- Square-shaped trajectory
  - Clock-wise (cw)
  - Counter clock-wise (ccw)





- Parameters of interest
  - Center of gravity (x, y)
  - Offset (r)
  - Precision

ROBOTICS UNIVERSIDAD DE ZARAGOZA

UMBmark

Correction of systematic errors for the TRC Labmate:

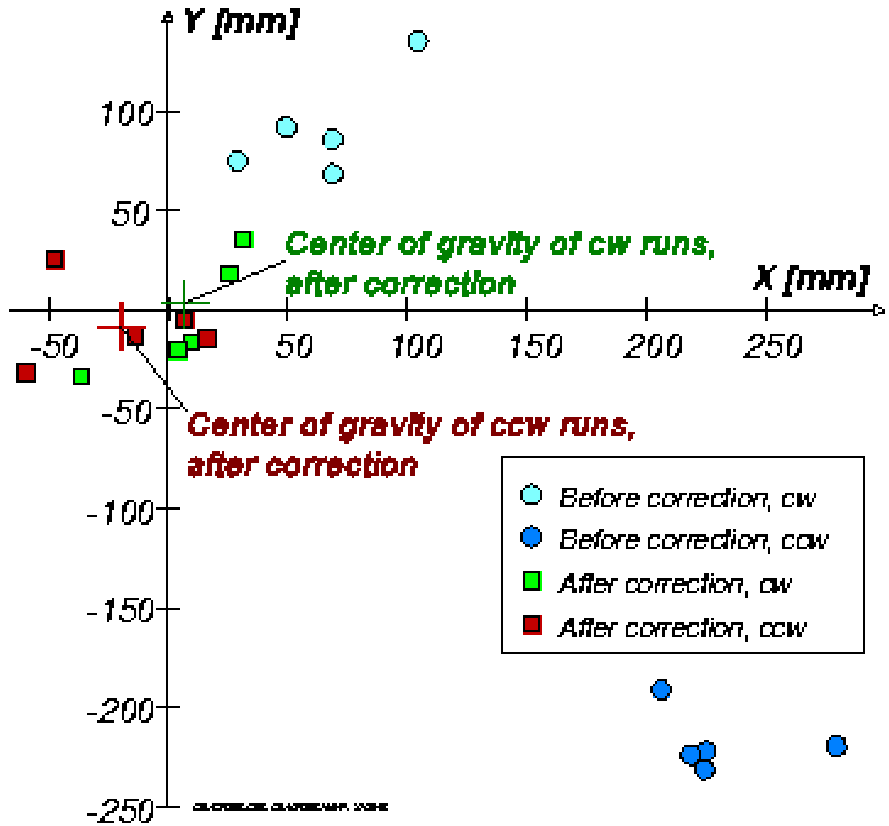


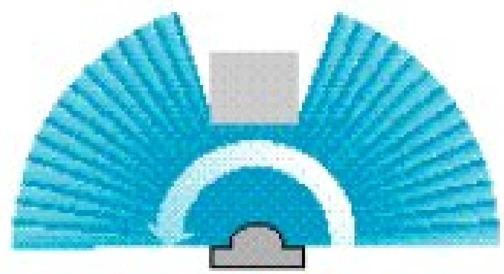
Figure 6.5: Position errors of a TRC LabMate after completing the UMBmark test (4 x 4 m).
Before calibration: b=340.00 mm, $D_R/D_L = 1.00000$
After calibration: b=336.17 mm, $D_R/D_L = 1.00084$

External sensors: Laser

TRC LightRanger

SICK LMS200
- 180º coverage
- 0.5º resolution
- 10 mm resolution
- up to 80 m
- up to 75 Hz
- indoor applications

180º



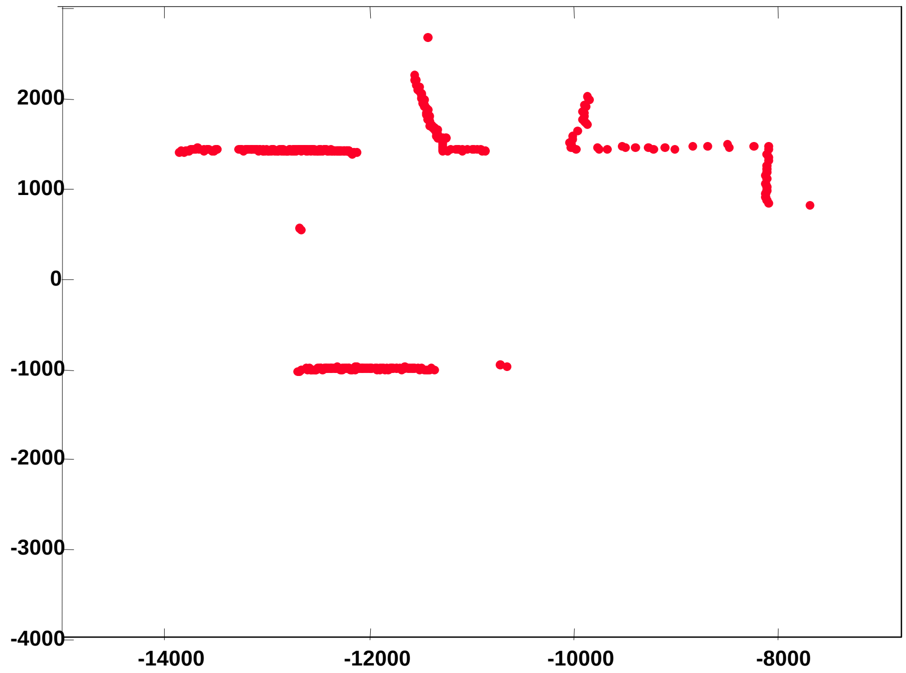
Raw data: $\{(\rho_1, \phi_1)^T, ..., (\rho_n, \phi_n)^T\}$

LMS291

PLS101/201

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Feature extraction for Laser



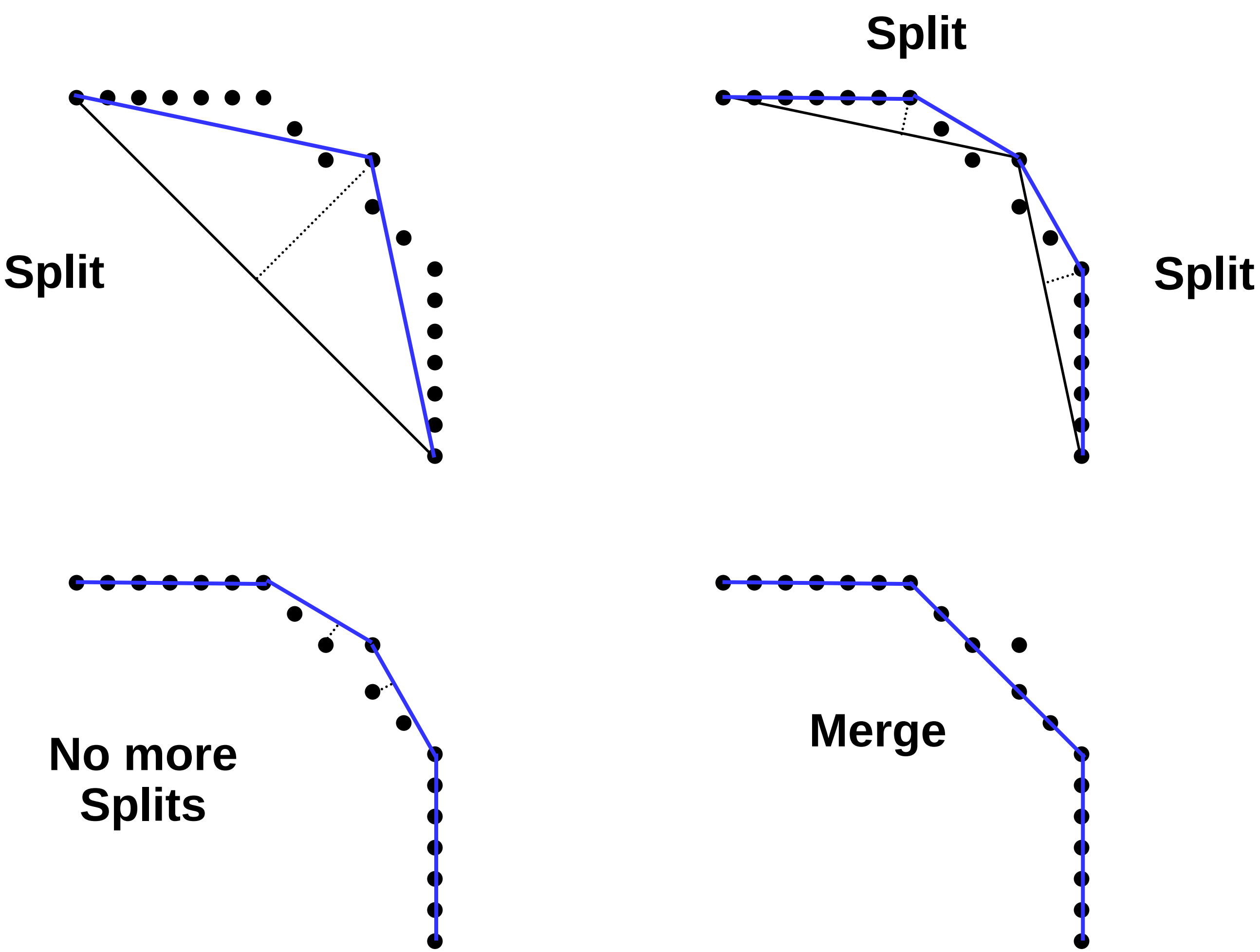
- Obtain line segments from a laser scan:
  - Segmentation
  - Line estimation

Split and merge:

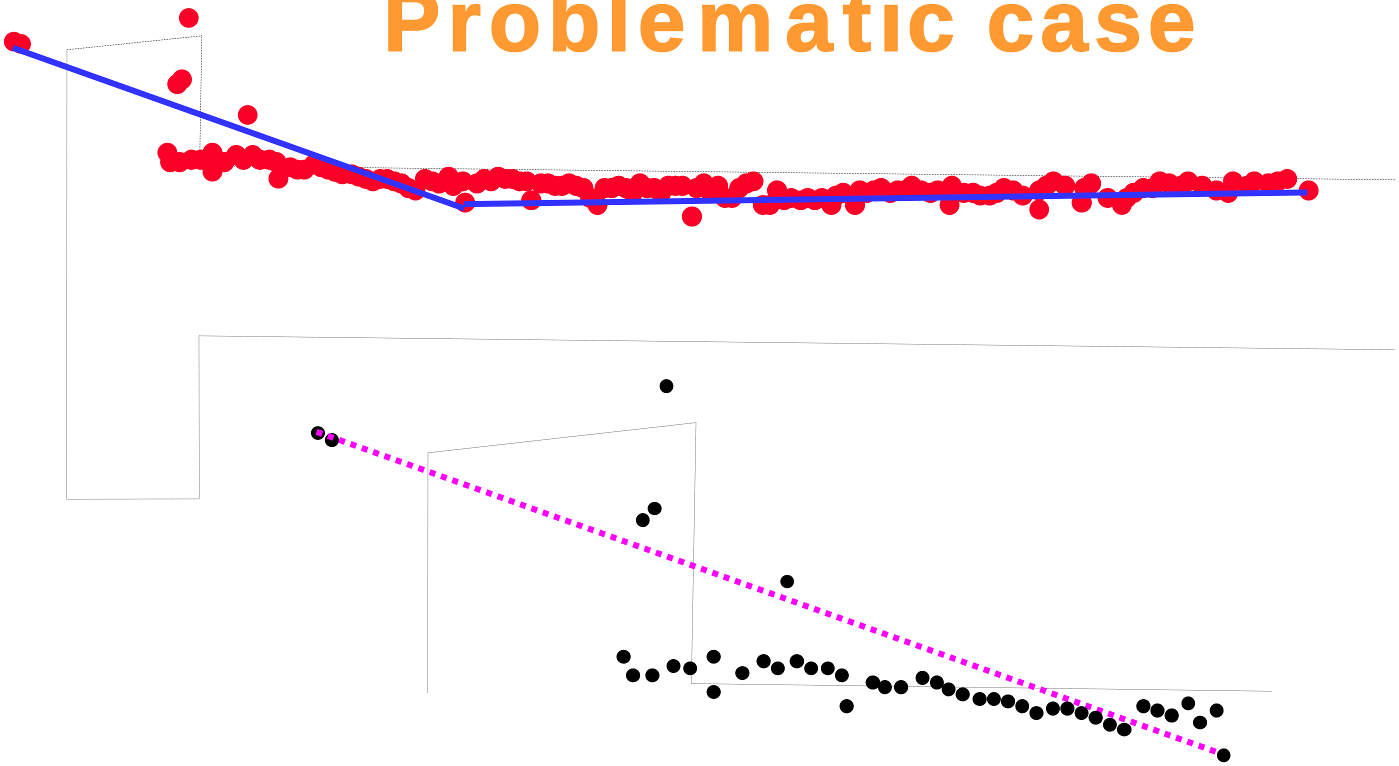
1. **Recursive Split:**
   1. Obtain the line passing by the two extreme points
   2. Obtain the point more distant to the line
   3. If distance > error_max, split and repeat with the left and right sub-scan
2. **Merge:**
   1. If two consecutive segments are close enough, obtain the common line and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



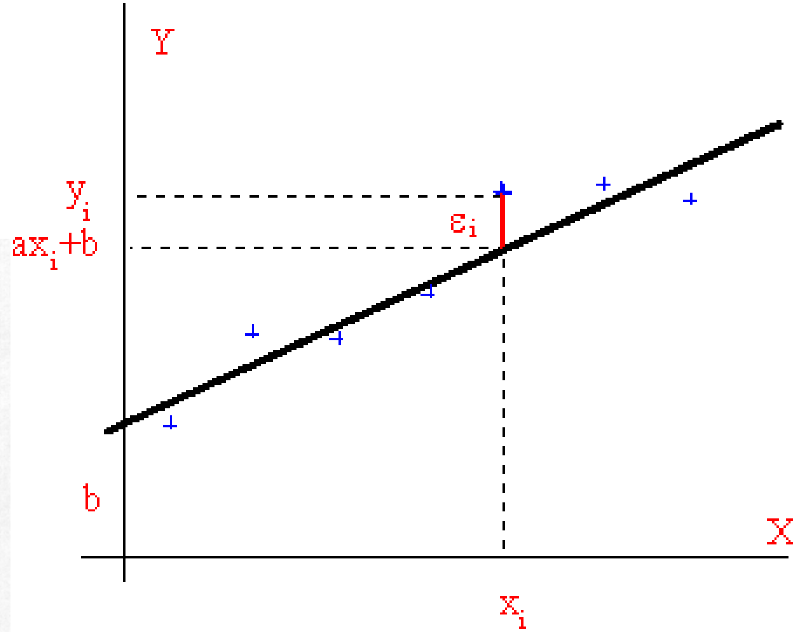
Problematic case

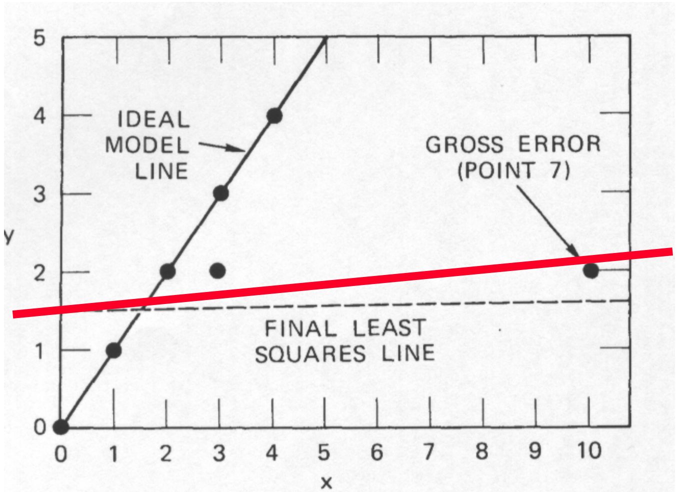


- **Split and merge:** uses only extreme points
- **Other options:**
  - TLS: Total Least squares (not much better)
  - RANSAC: RANdom SAmpling Consensus

RANSAC (Fischler y Bolles, 1981)

- TLS: A single "leverage point" produces a big estimation error
- Heuristics such as remove more discrepant points may also fail





POINT	x	y
1	0	0
2	1	1
3	2	2
4	3	2
5	3	3
6	4	4
7	10	2

SUCCESSIVE LEAST SQUARES APPROXIMATIONS		
ITERATION	DATA SET	FITTING LINE
1	1, 2, 3, 4, 5, 6, 7	1.48 + .16x
2	1, 2, 3, 4, 5, 7	1.25 + .13x
3	1, 2, 3, 4, 7	0.96 + .14x
4	2, 3, 4, 7	1.51 + .06x

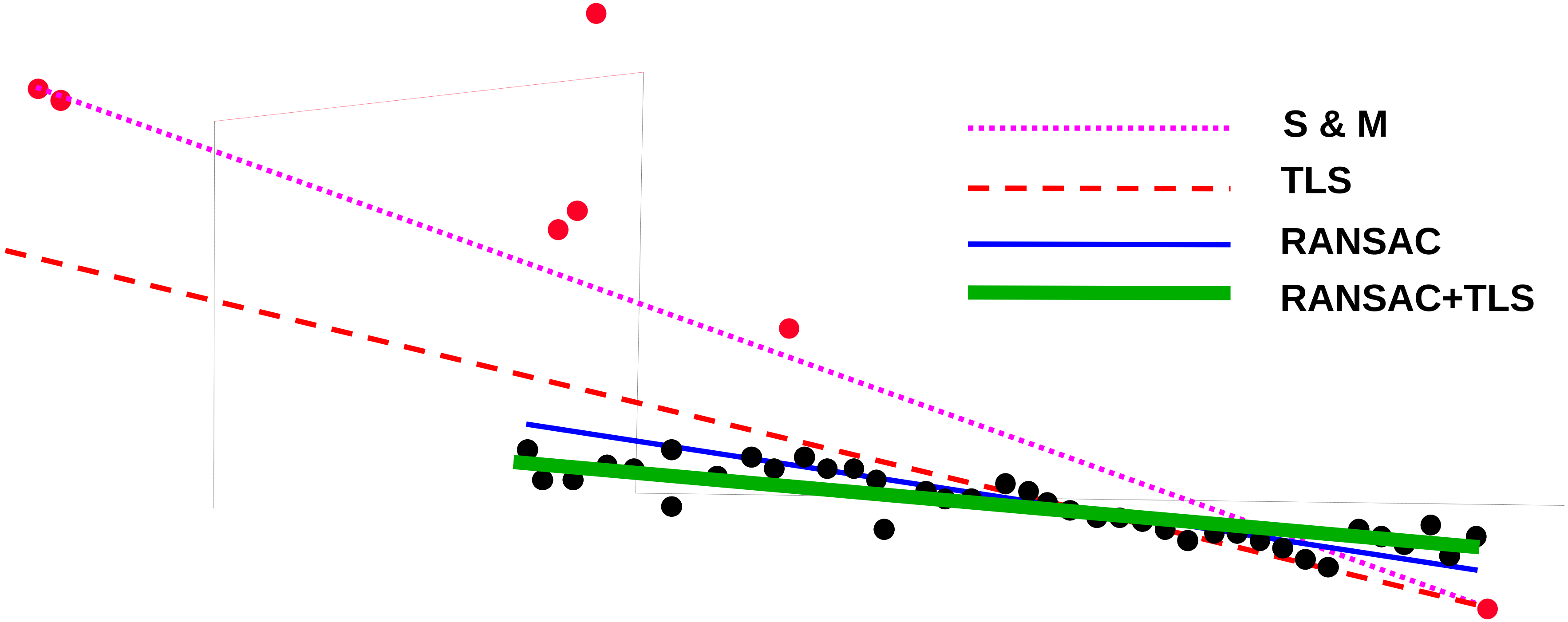
COMPUTATION OF RESIDUALS				
POINT	ITERATION 1 RESIDUALS	ITERATION 2 RESIDUALS	ITERATION 3 RESIDUALS	ITERATION 4 RESIDUALS
1	-1.48	-1.25	-.96*	—
2	-0.64	-0.38	-.10	-.57
3	-0.20	0.49	.76	.37
4	0.05	0.36	.63	.31
5	1.05	1.36*	—	—
6	1.89*	—	—	—
7	-1.06	-0.57	-.33	-.11

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*) > t, use S1* to compute (maybe using least squares) a new model M1*
  - If #(S1*) < t, randomly select another subset S2 and repeat
  - If, after some predetermined number of trials there is no consensus set with t points, return with failure

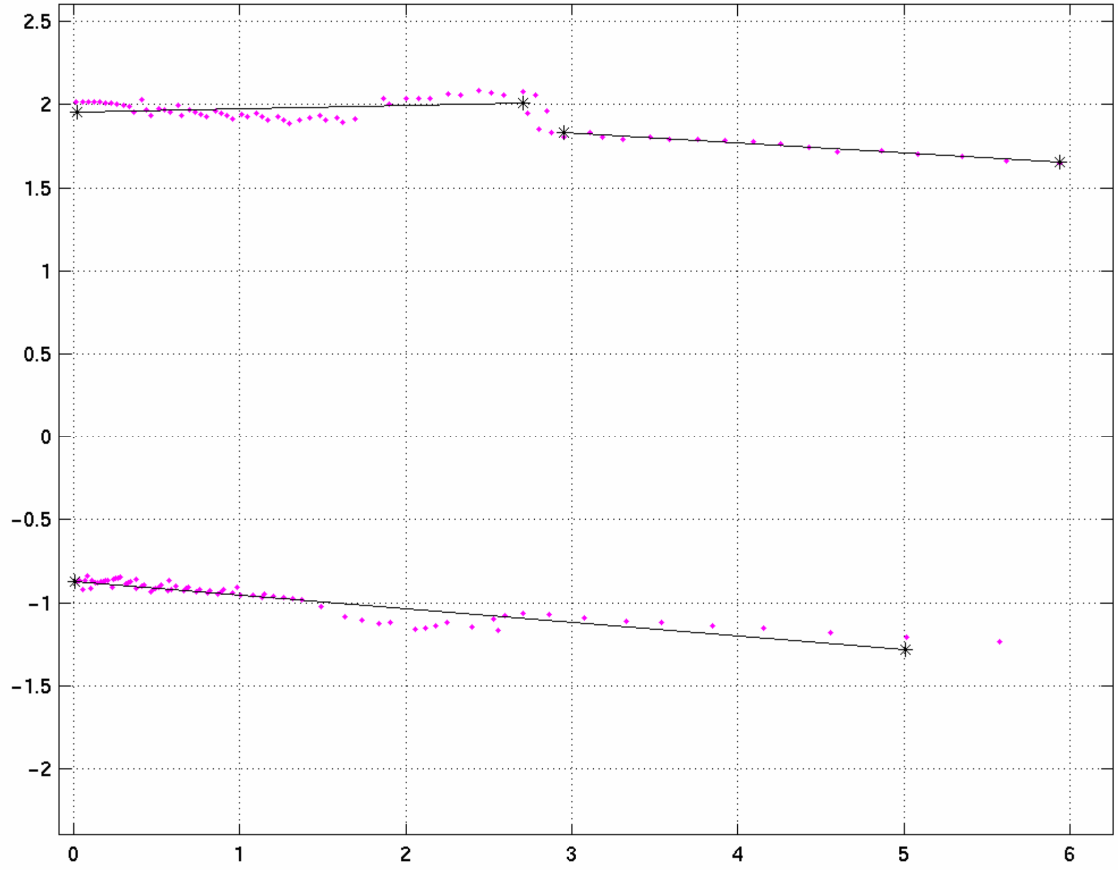
ROBOTICS UNIVERSIDAD DE ZARAGOZA

RANSAC



Laser

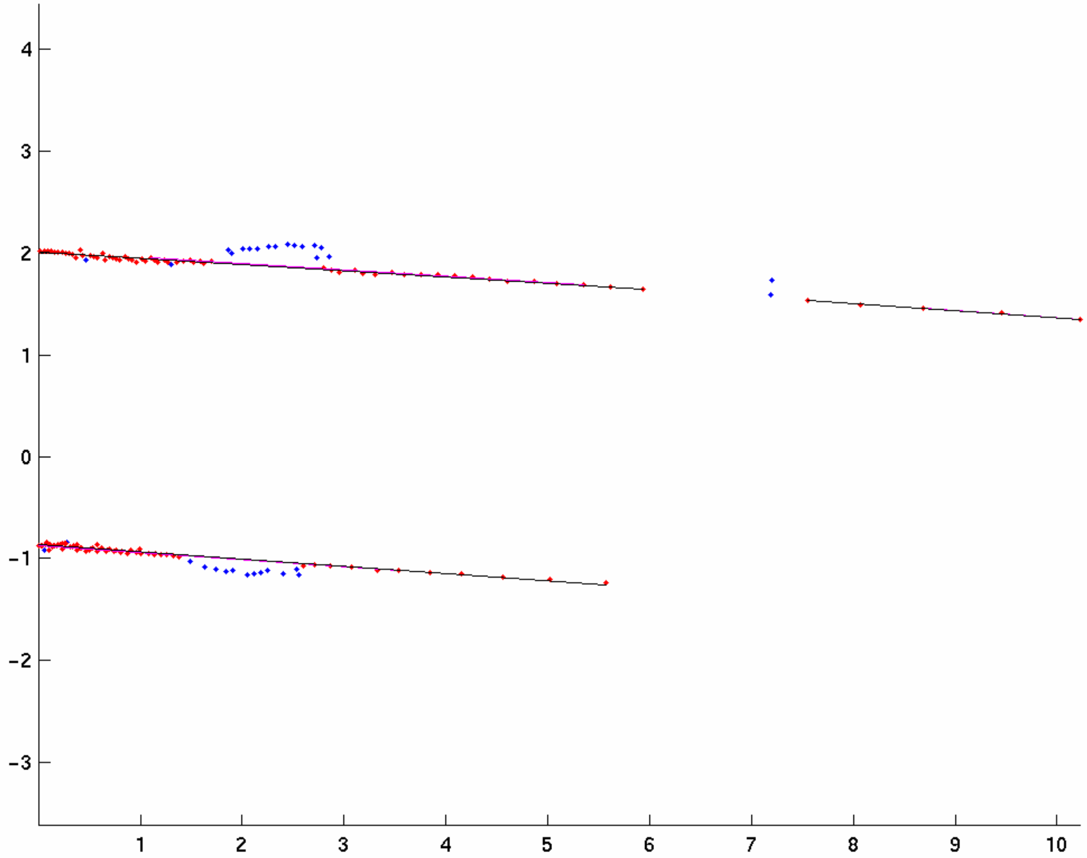
- Classical algorithms (Split & Merge):



Not robust to complex and/or spurious data

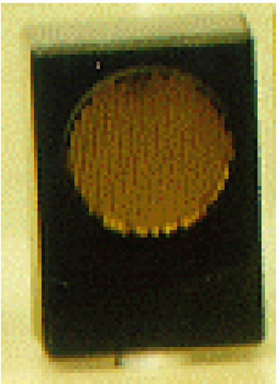
ROBOTICS
UNIVERSIDAD DE ZARAGOZA

RANSAC solution

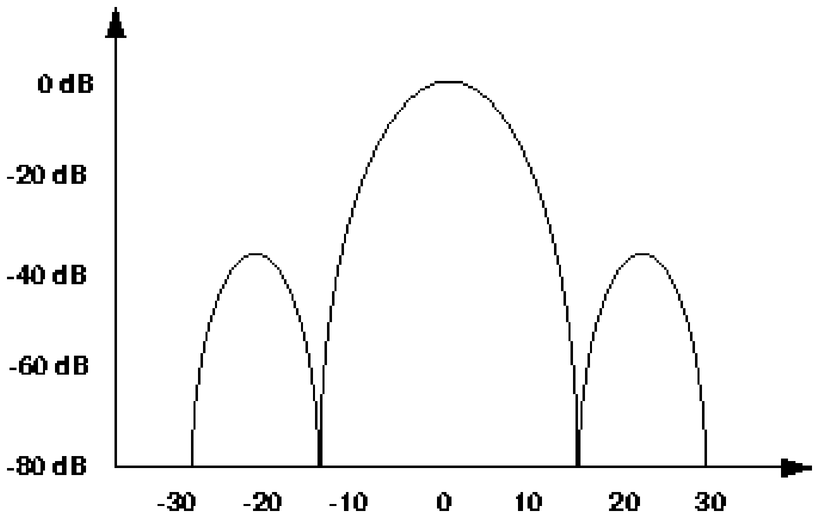


Robust statistics deal with spuriousness

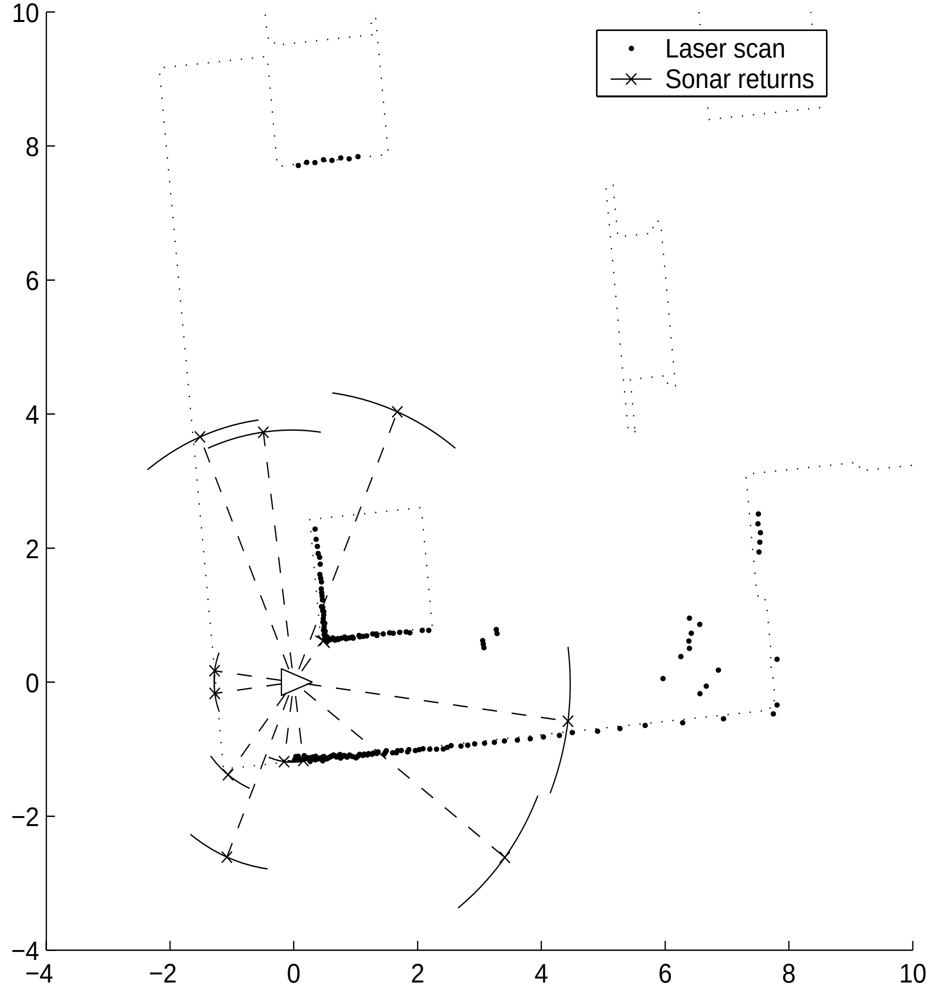
External sensors: Sonar



Polaroid US

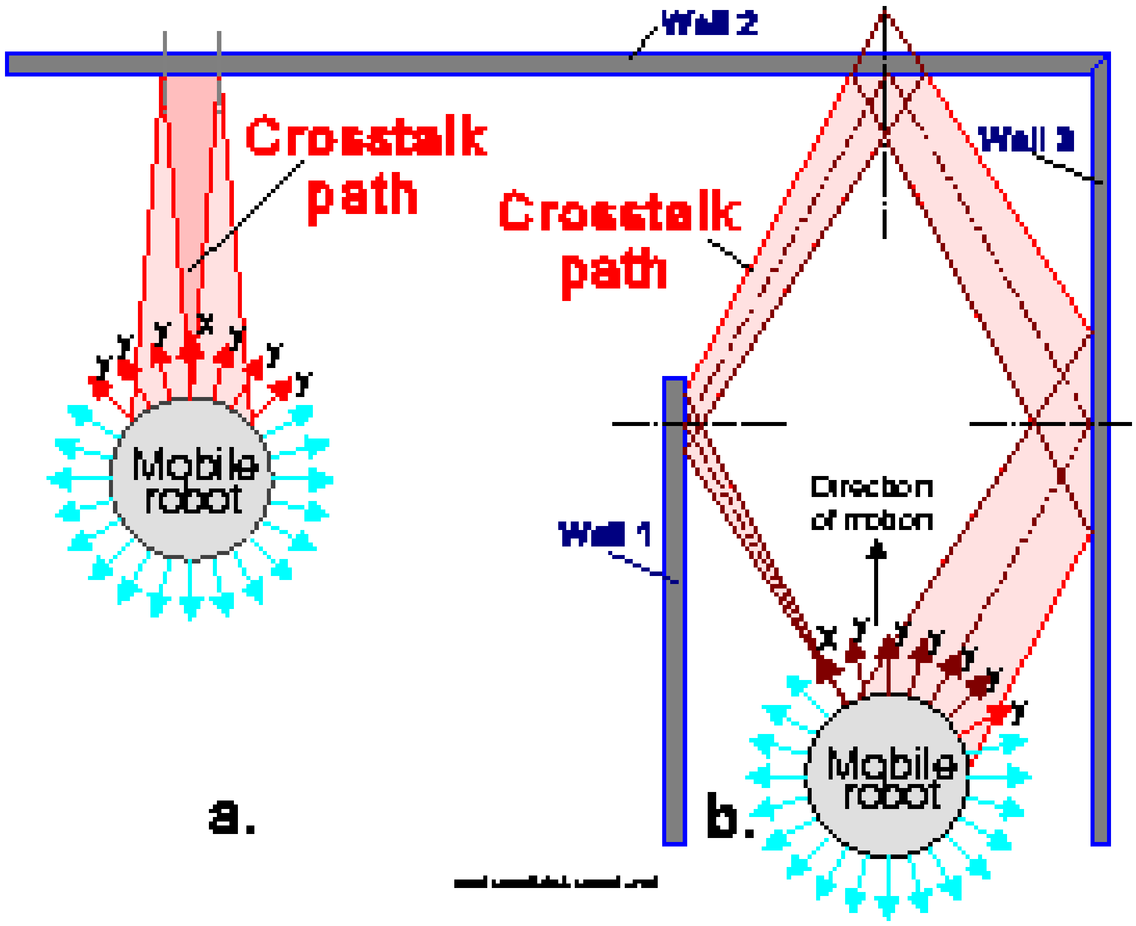


Frequency lobes

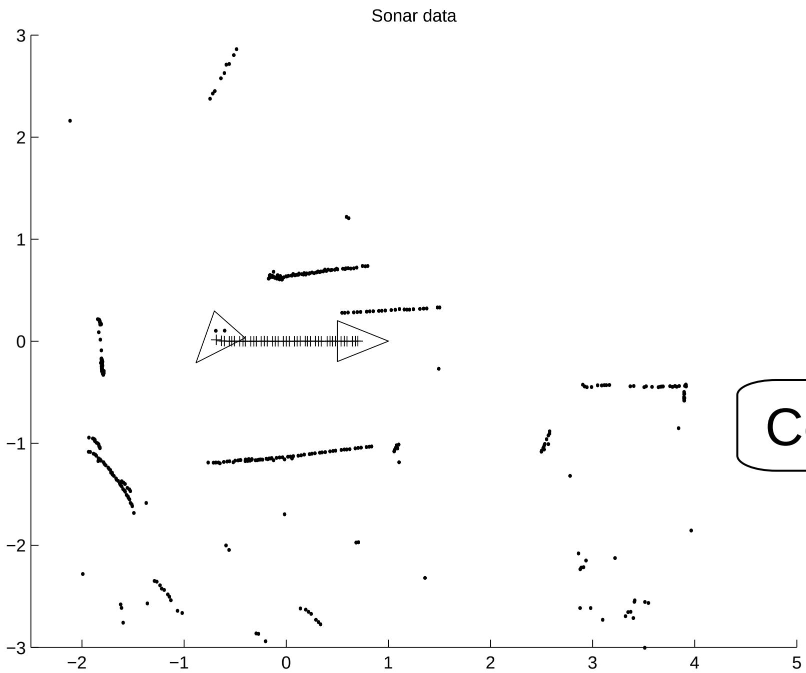


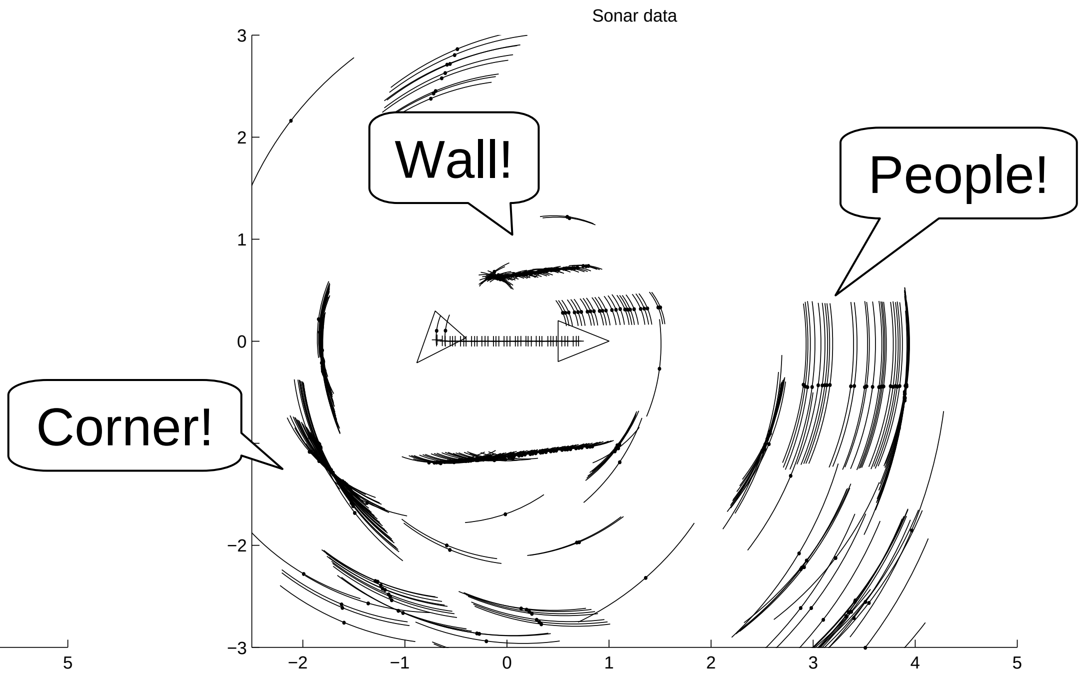
Very sparse and noisy data

Sonar


Wall 2
Crosstalk
path
Crosstalk
path
Wall 3
Wall 1
Direction
of motion
Mobile
robot
Mobile
robot
a.
b.

Move and build a local map

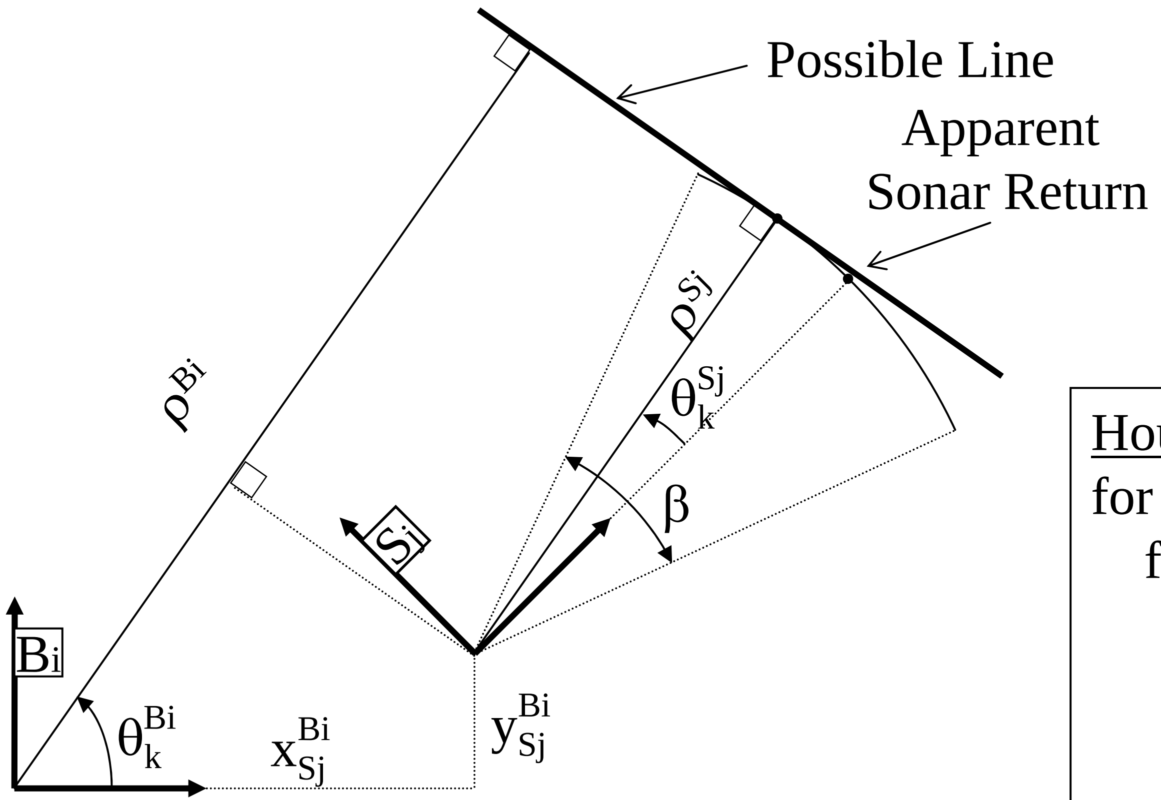




Exploit redundancy

Use a good sensor model

Sonar Model for Lines



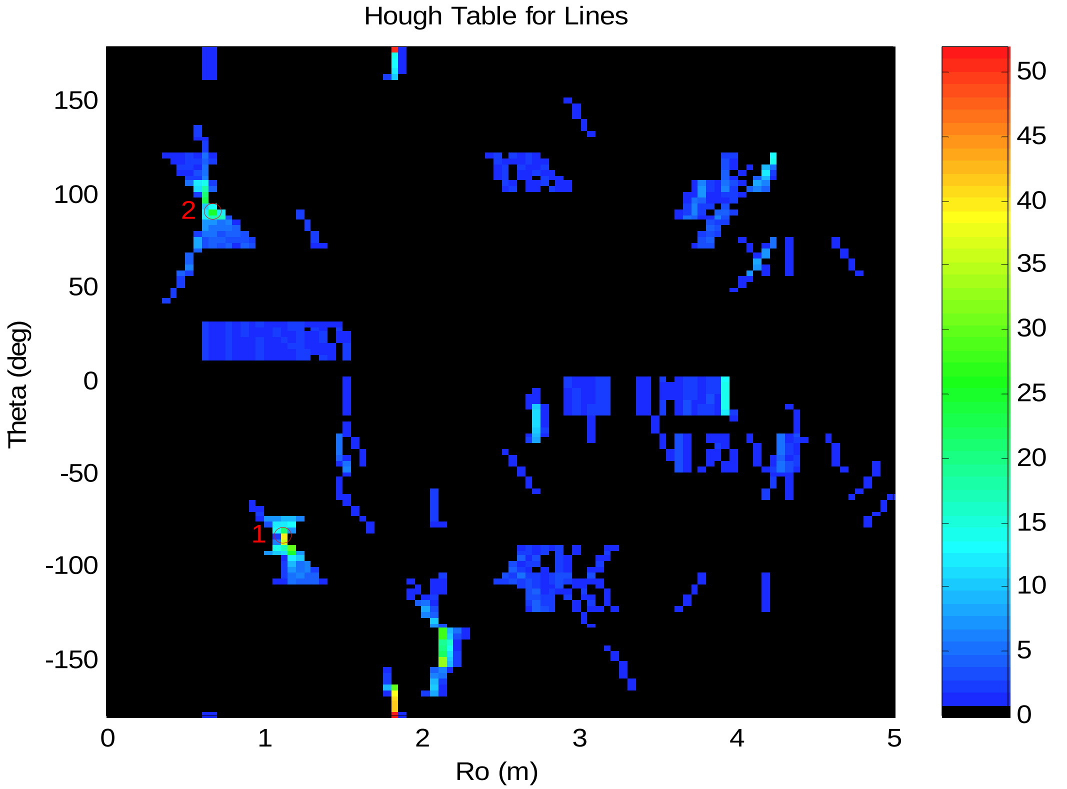
<u>Hough Voting</u>

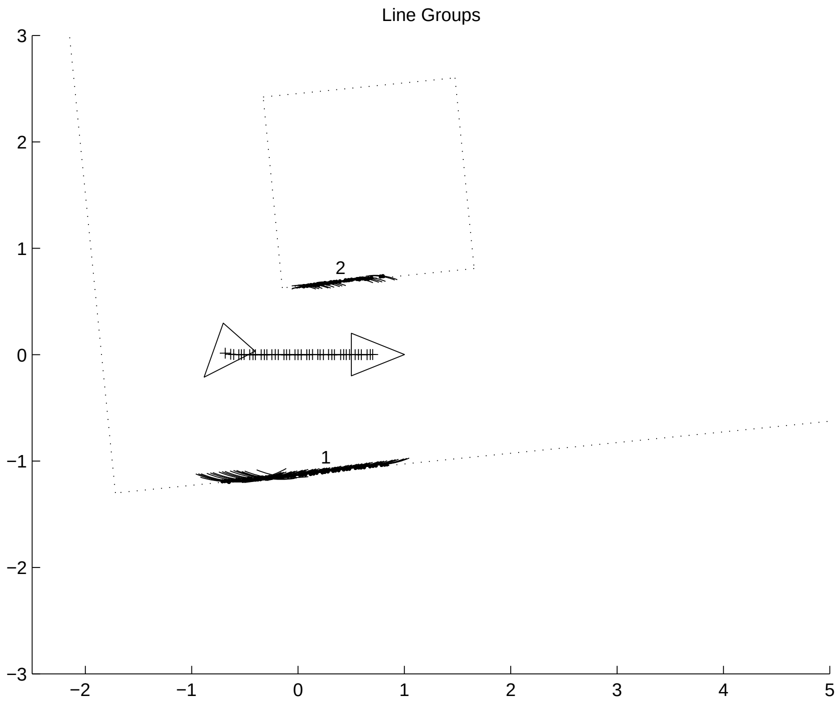
```
for i in 1..n_positions
    for j in 1..n_sensors
      Compute x^Bi_Sj
      for θ^Sj_k in -β/2..β/2 step δ
        Compute θ^Bj_k ρ^Bj_k
        Vote(θ^Bj_k, ρ^Bj_k)
      end
    end
end
```

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Hough Transform

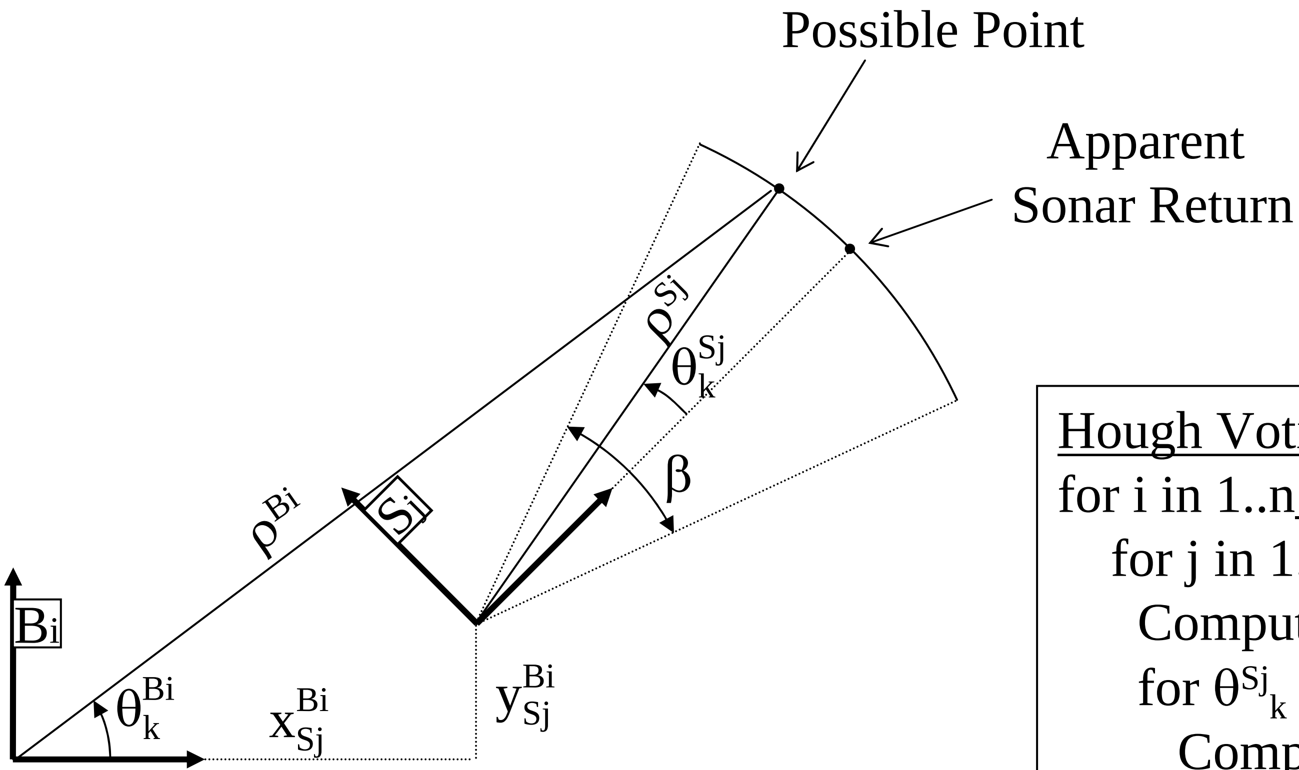
- Sonar returns vote for lines
- Look for local maxima





The Hough gives robust local data associations

Sonar Model for Points



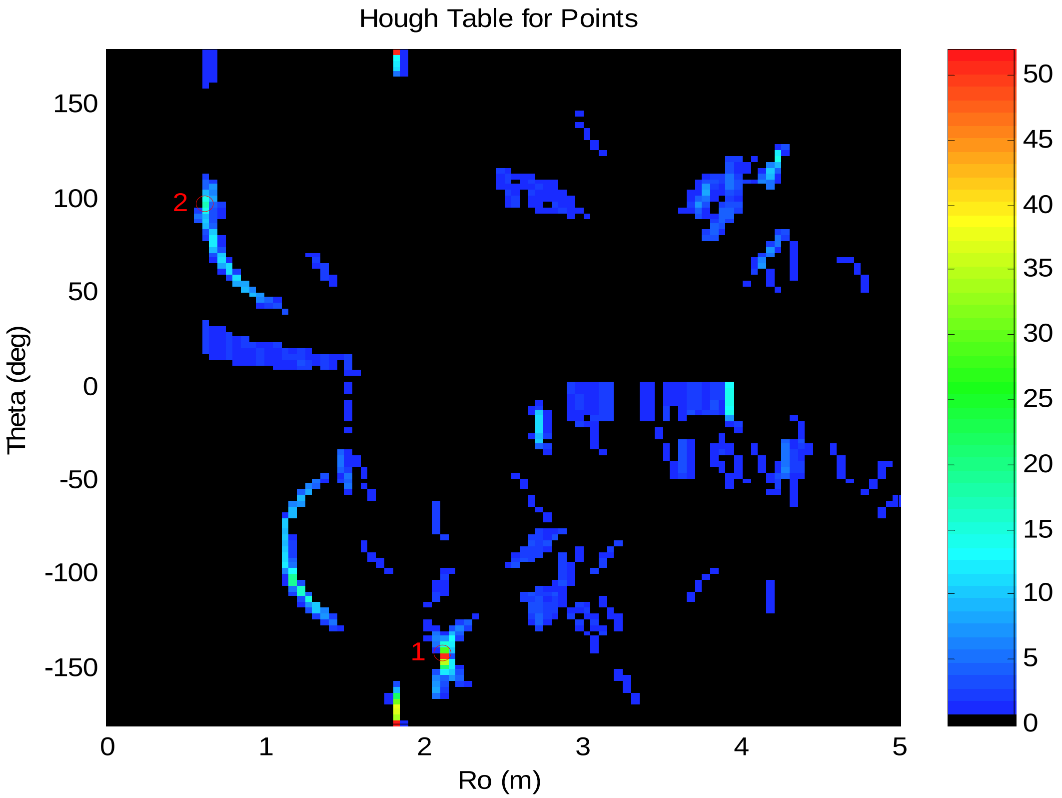
<u>Hough Voting</u>

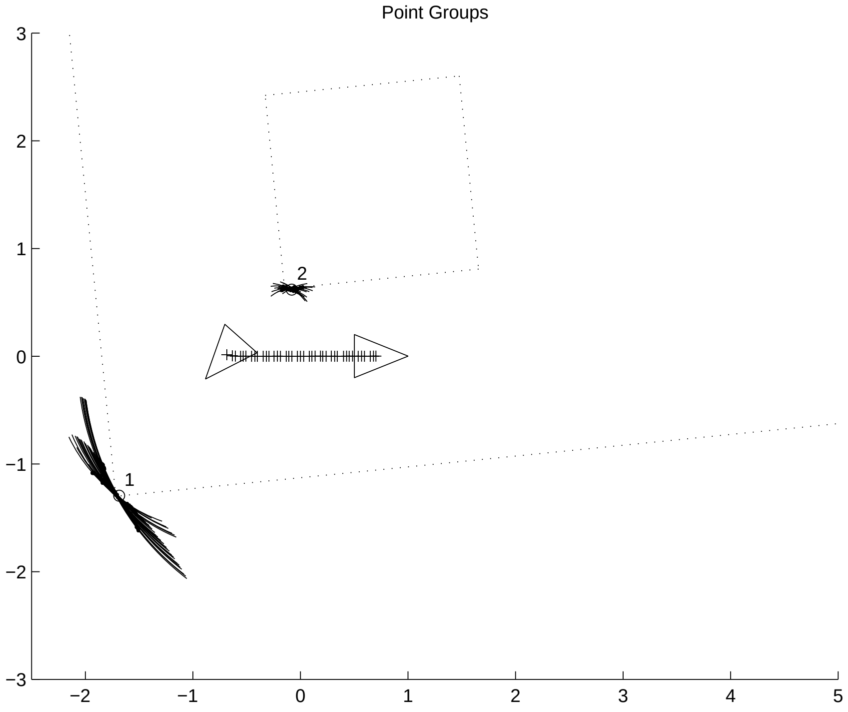
```
for i in 1..n_positions
    for j in 1..n_sensors
      Compute x^Bi_Sj
      for θ^Sj_k in -β/2..β/2 step δ
        Compute θ^Bj_k ρ^Bj_k
        Vote(θ^Bj_k, ρ^Bj_k)
      end
    end
end
```

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Hough Transform

- Sonar returns vote for points
- Look for local maxima





A local map is built using standard techniques

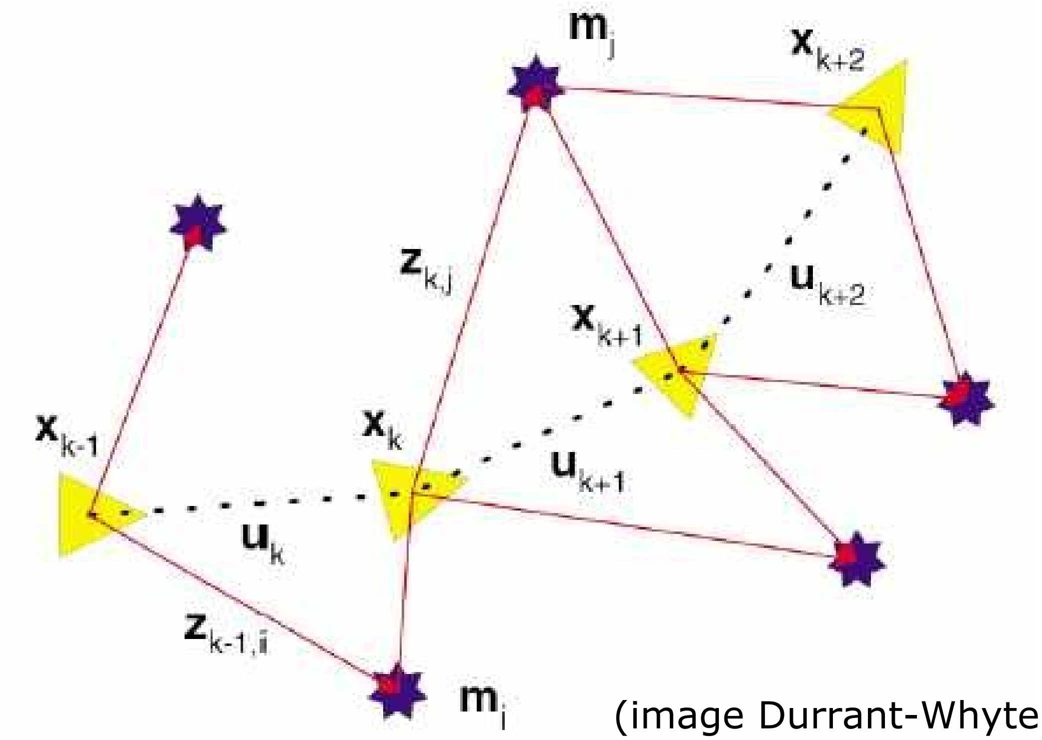
INDEX

1. Mobile Robots
   1. Common structures
   2. Perception-Decision-Action Loop
   3. Sensors for Mobile Robots
      1. Internal sensors: Odometry
      2. External sensors: Laser and Sonar

2. **Localization and Mapping**
   1. **Overview. Kalman Filter**
   2. **Localization using Odometry**
   3. **Map-based Localization**
   4. **SLAM**

2. Localization and Mapping

- M: Environment features
- U: Control inputs
- X: Vehicle locations
- Z: Sensor measurements



(image Durrant-Whyte)

Probabilistic view-point

- State x_k= Multivariate random variable
- Current measurement z_k
- Set of measurements Z_k = { z_1, z_2, ..., z_k }

$$\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{Z}_k)$$

- Recursive Bayesian Estimation

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})d\mathbf{x}_{k-1}$$

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})}$$

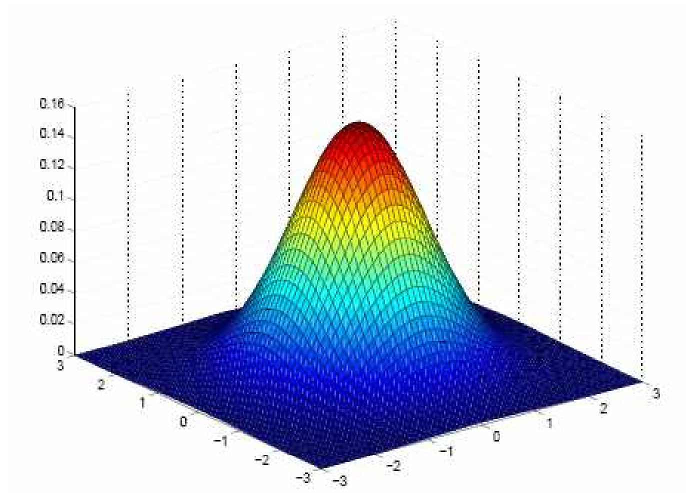
Gaussianity assumption

- State x = Multivariate Gaussian rv

$$\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$$

- Mean

$$\hat{\mathbf{x}} = E[\mathbf{x}] = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \vdots \\ \hat{\mathbf{x}}_n \end{bmatrix}$$



2D Gaussian rv

- Covariance matrix

$$\mathbf{P} = E[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^T] = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \dots & \mathbf{P}_{1n} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \dots & \mathbf{P}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{n1} & \mathbf{P}_{n2} & \dots & \mathbf{P}_{nn} \end{pmatrix}$$

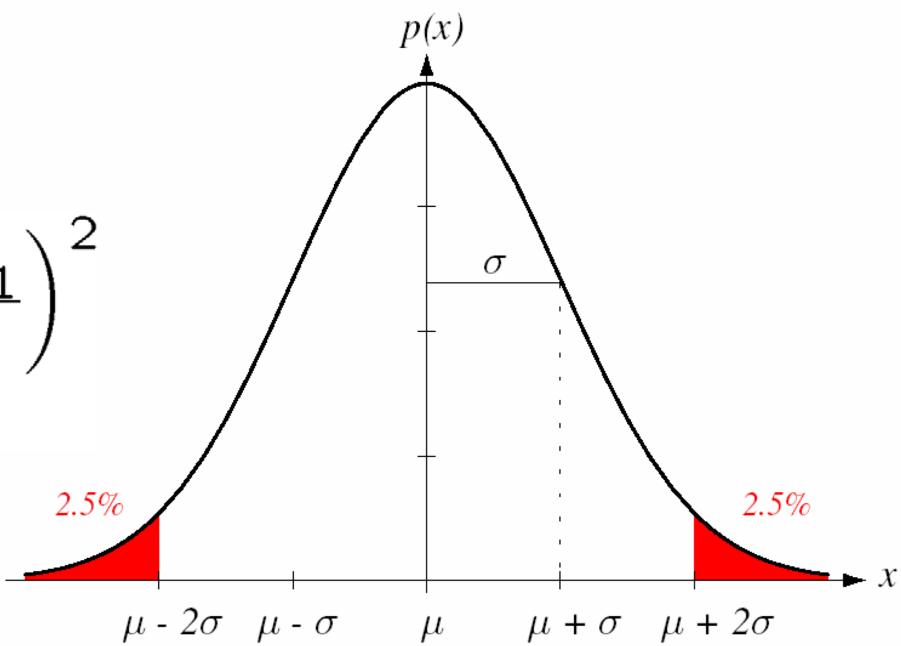
ROBOTICS UNIVERSIDAD DE ZARAGOZA

A Simple Example

- Gaussianity assumption:

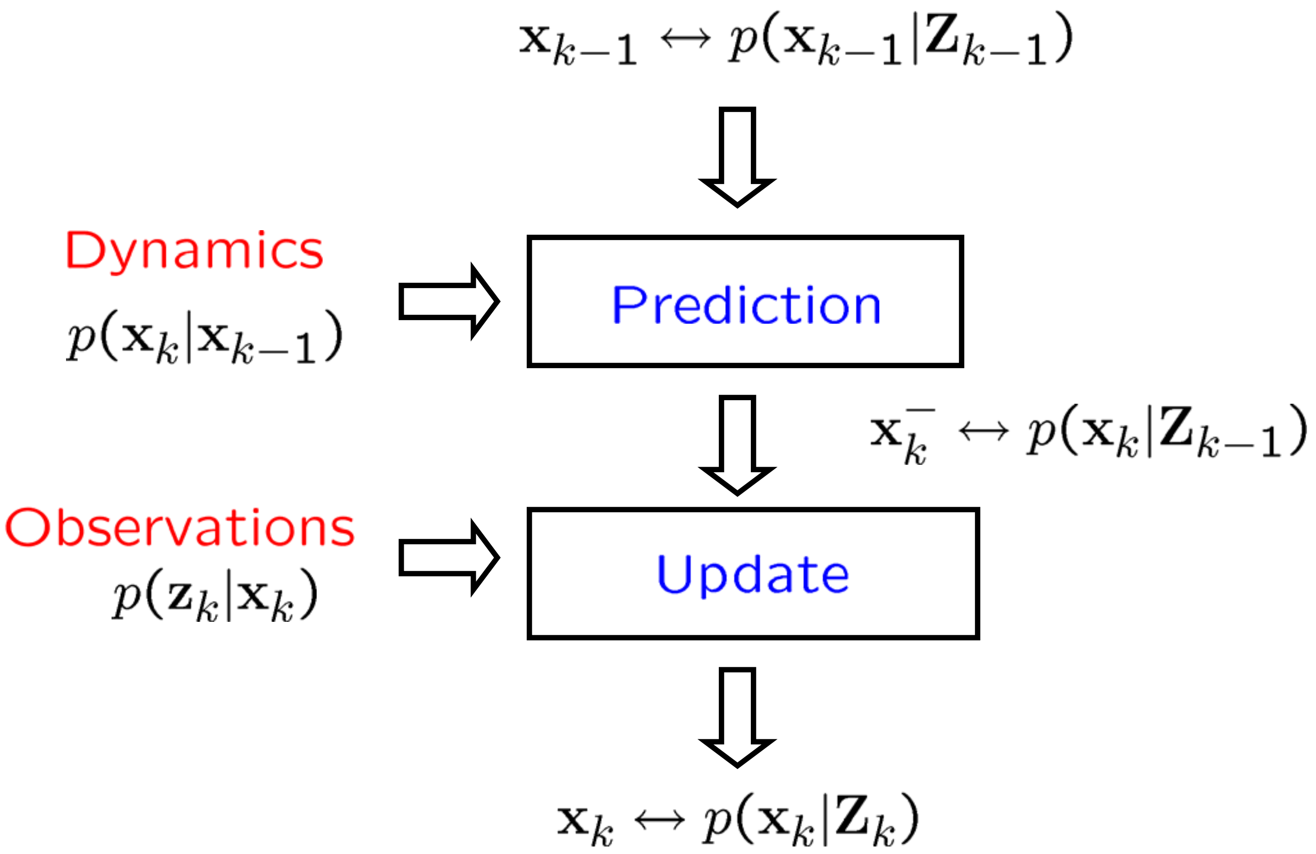
$$\tilde{d}_1 \sim N(\mu_1, \sigma_1^2)$$

$$p(\tilde{d}_1) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{1}{2}\left(\frac{\tilde{d}_1 - \mu_1}{\sigma_1}\right)^2}$$



$$Pr\left\{\mid \tilde{d}_1 - \mu_1 \mid \leq \sigma_1\right\} \simeq 0.68$$
$$Pr\left\{\mid \tilde{d}_1 - \mu_1 \mid \leq 2\sigma_1\right\} \simeq 0.95$$
$$Pr\left\{\mid \tilde{d}_1 - \mu_1 \mid \leq 3\sigma_1\right\} \simeq 0.997$$

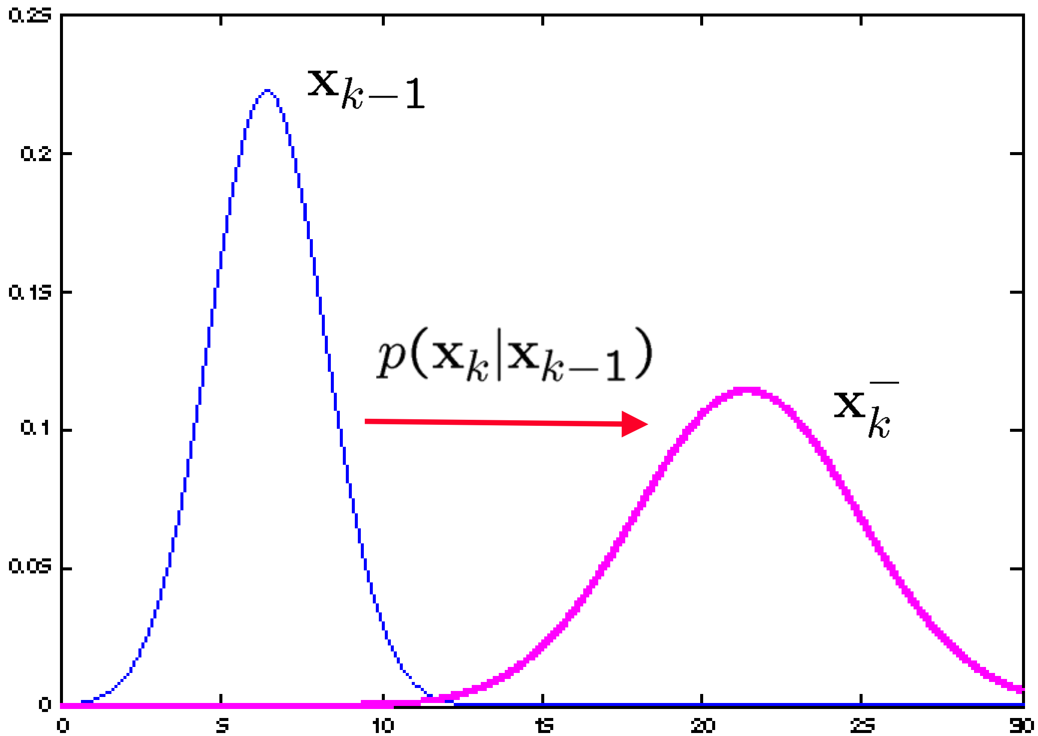
Kalman Filter



Nonlinear models: Extended / Unscented Kalman Filter (EKF / UKF)

1D Kalman Filter

- Prediction step:

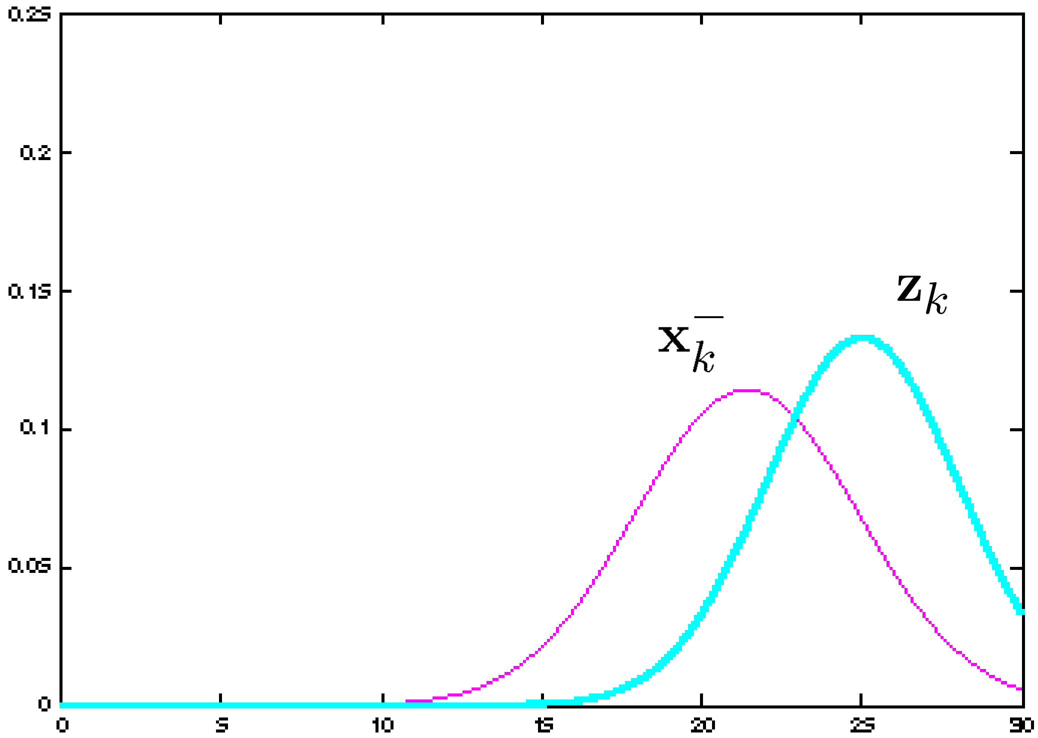


$$\mathbf{x}_k^- = \mathbf{f}_k(\mathbf{x}_{k-1}) + \mathbf{v}_k \Rightarrow \begin{cases} \mathbf{x}_k^- = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_k & ; \text{ linear} \\ \mathbf{x}_k^- \simeq \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_k & ; \text{ nonlinear} \end{cases}$$

$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$$

1D Kalman Filter

- Measurement:



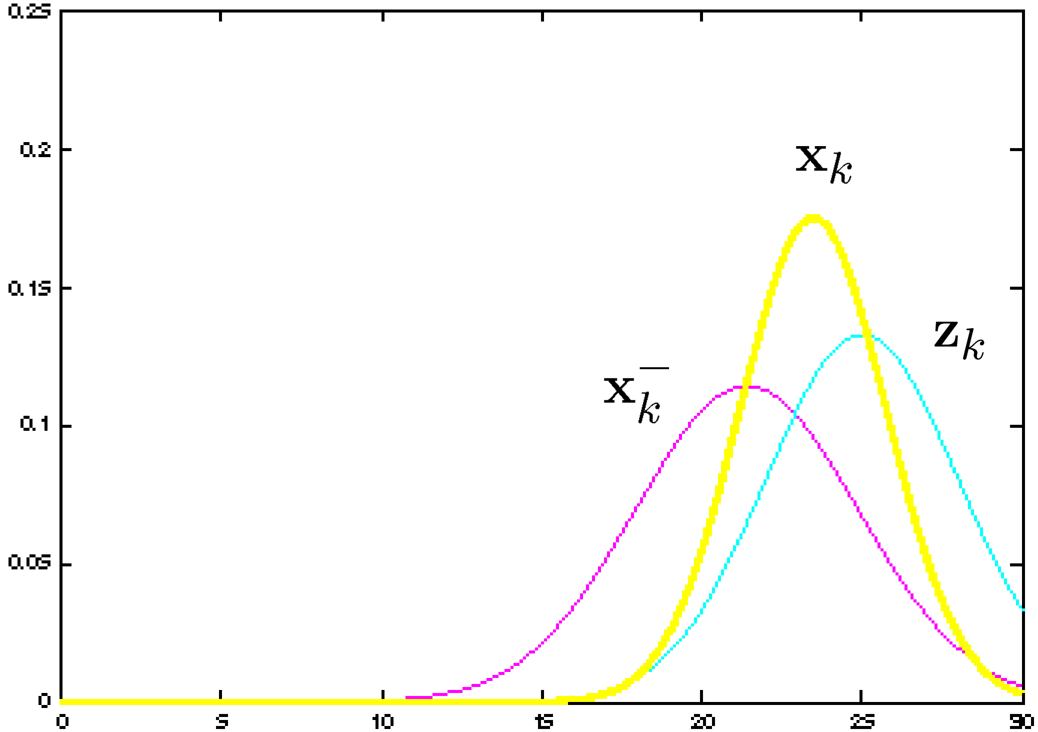
$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k^-) + \mathbf{w}_k \Rightarrow \begin{cases} \mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k^- + \mathbf{w}_k & ; \quad \text{linear} \\ \mathbf{z}_k \simeq \mathbf{H}_k \mathbf{x}_k^- + \mathbf{w}_k & ; \quad \text{nonlinear} \end{cases}$$

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$$

ROBOTICS UNIVERSIDAD DE ZARAGOZA

1D Kalman Filter

- Update step:



$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)$$

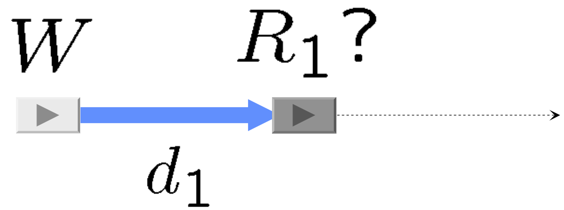
$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-$$

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

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Localization using Odometry

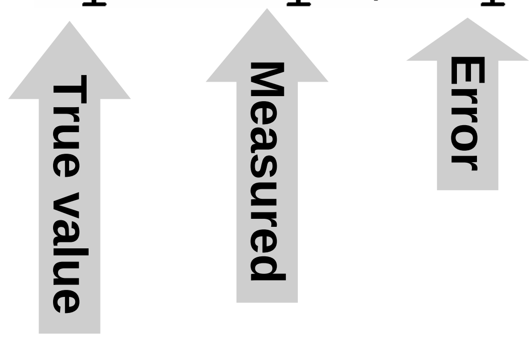
- A robot moves in a 1D space:



Uncertainty in Robot Position

- Odometry:

$$d_1 = \hat{d}_1 + \tilde{d}_1$$



- Uncertainty model:

$$\begin{aligned} E(\tilde{d}_1) &= \mu_1 \\ \mathtt{Var}(\tilde{d}_1) &= \sigma_1^2 \end{aligned}$$

Robot Odometry: Example

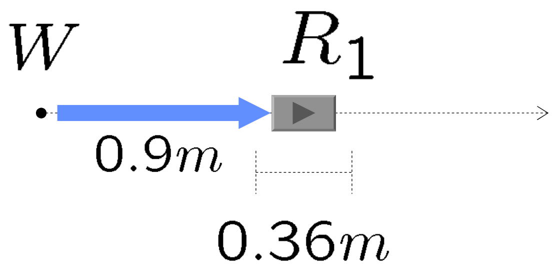
- Odometry error model: $d_1 = \hat{d}_1 + \tilde{d}_1$

$$\tilde{d}_1 \sim N(\mu_1, \sigma_1^2)$$

$$\mu_1 = 0$$

$$\sigma_1 = 0.1 \cdot \hat{d}_1$$

- Example: move 0.9m



$$\hat{x}_{WR_1} = \hat{d}_1 = 0.9m$$

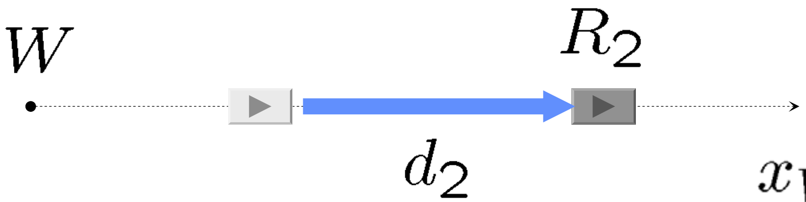
$$\sigma_{x_{WR_1}} = 0.09m$$

$$Pr\left\{ |\tilde{d}_1| \le 0.18m \right\} \simeq 0.95$$

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Robot Odometry: Example

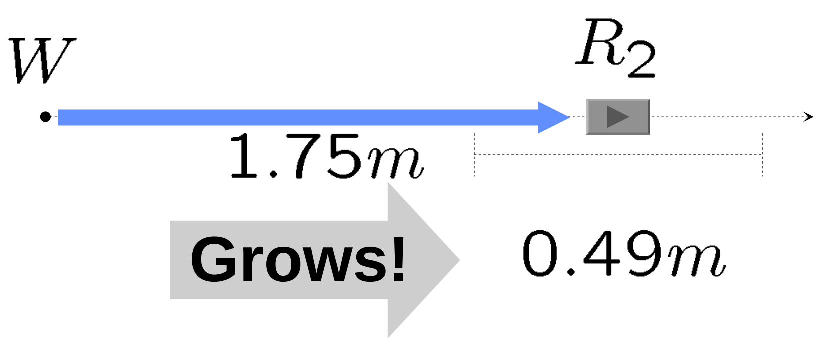
- Robot moves again 0.85m: $\hat{d}_2 = 0.85m$



$$x_{WR_2} = \hat{d}_1 + \tilde{d}_1 + \hat{d}_2 + \tilde{d}_2$$

$$\begin{aligned} E(\tilde{d}_1 + \tilde{d}_2) &= \mu_1 + \mu_2 = 0 \\ \mathtt{Var}(\tilde{d}_1 + \tilde{d}_2) &= \sigma_1^2 + \sigma_2^2 \\ &= 0.1^2 \left(\hat{d}_1^2 + \hat{d}_2^2\right) \end{aligned}$$

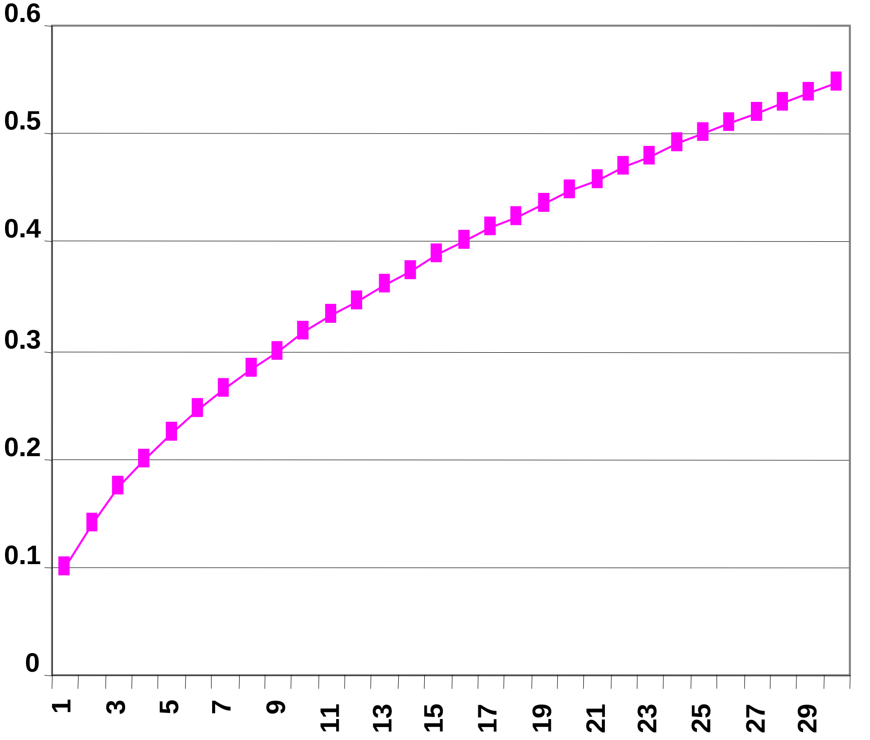
- New estimation:



$$\begin{aligned} \hat{x}_{WR_2} &= 1.75m \\ \sigma_{x_{WR_2}} &\simeq 0.12m \end{aligned}$$

Odometry

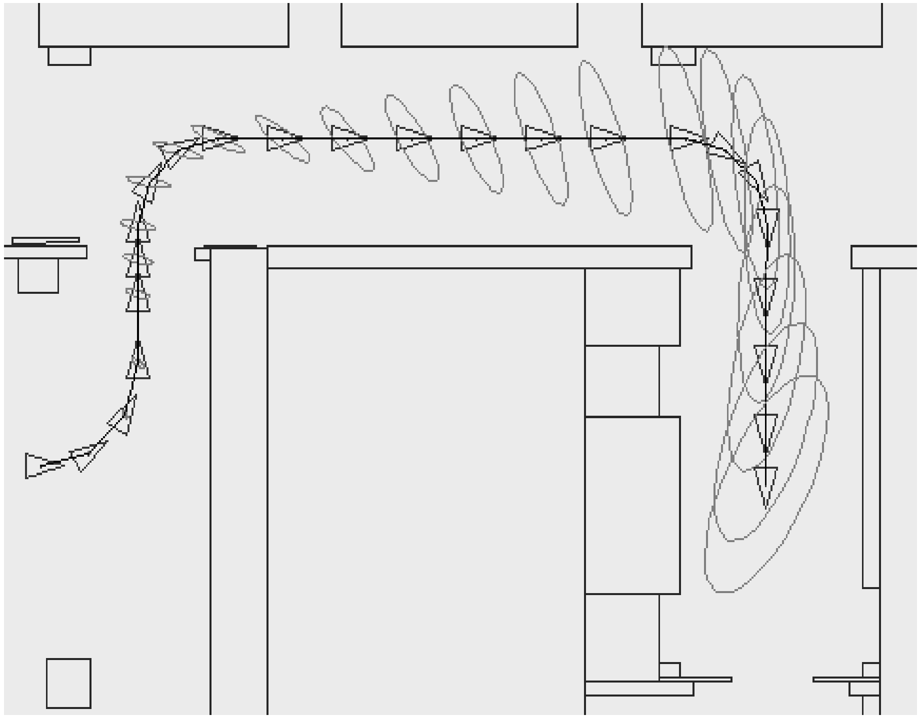
Odometry error grows unbounded



We are assuming error independence

Odometry

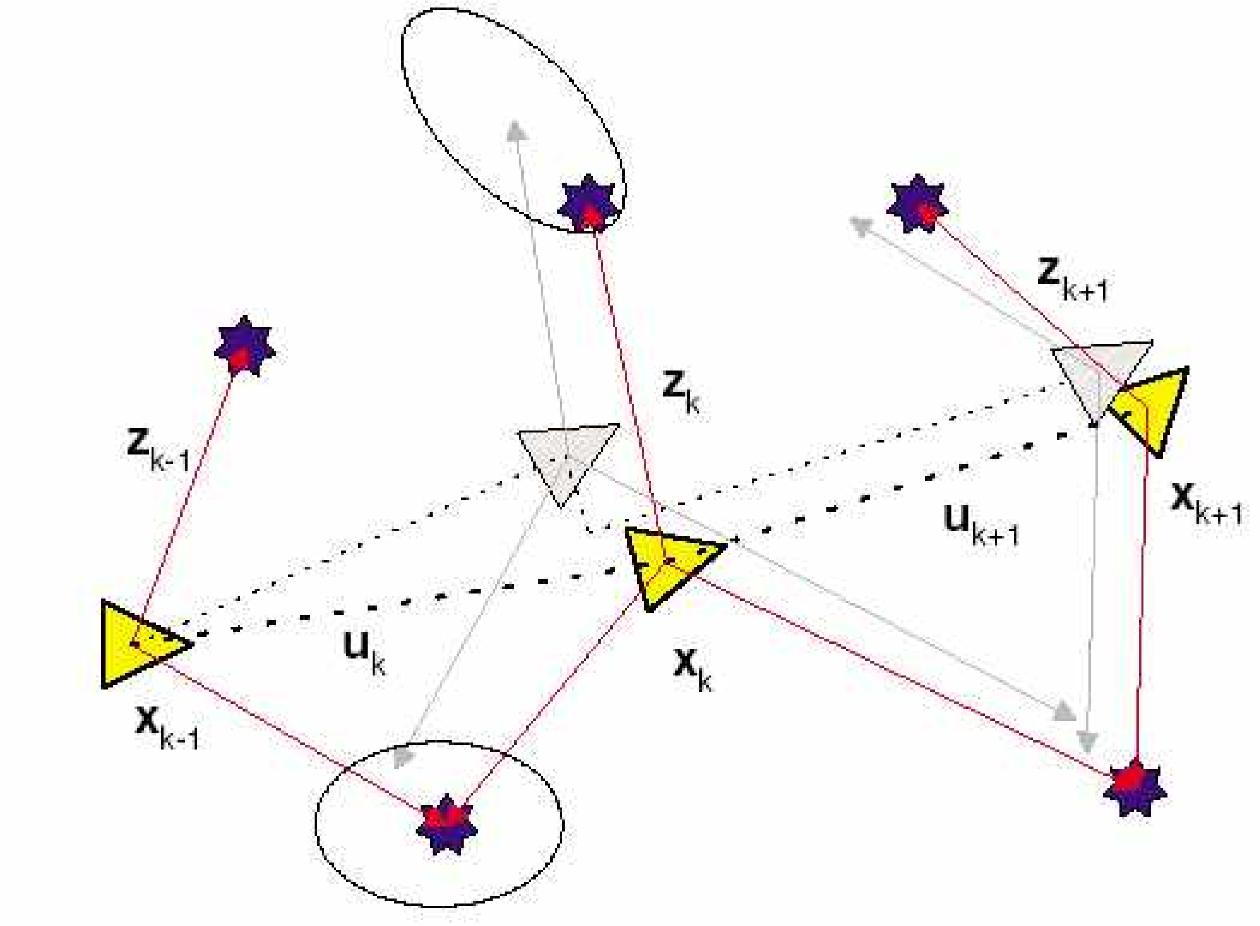
Uncertainty in 2D



How can we avoid unbounded uncertainty increase?

3. Map-based Localization

- Given M, U, Z
- Compute X



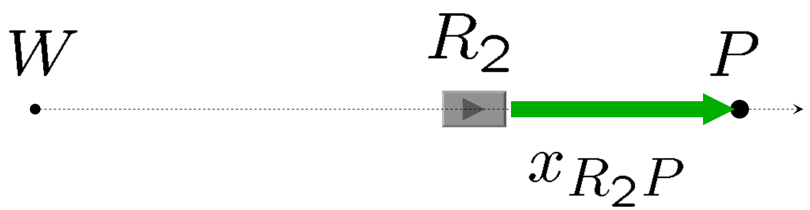
(image Durrant-Whyte)

Map-based Localization

- Assume that we know the exact location of a point P:

$$x_{WP} = 3m$$

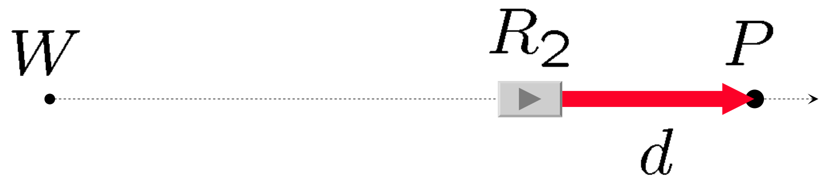
- We can predict the position of the point P relative to the robot position:



$$\begin{aligned}\hat{x}_{R_2P} &= -\hat{x}_{WR_2} + x_{WP} \\ &= 1.25m \\ \sigma_{x_{R_2P}} &= \sigma_{x_{WR_2}} \\ &\simeq 0.12m\end{aligned}$$

Environment Perception

- Assume that the robot has a sensor that measures the distance to the point, with some uncertainty, proportional to the distance:



Sensor measure:

$$\begin{aligned} \hat{d} &= 1.1m \\ \mu_d &= 0 \\ \sigma_d &= 0.01 \cdot \hat{d} \\ &= 0.011m \end{aligned}$$

Predicted value:

$$\begin{aligned} \hat{x}_{R_2P} &= 1.25m \\ \sigma_{x_{R_2P}} &\simeq 0.12m \end{aligned}$$

We can get a better estimation of robot position

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Kalman Filtering

- State to estimate: robot location

$$\begin{aligned} x &= x_{WR_2} \\ P &= \mathtt{Cov}(x) \end{aligned}$$

- Prediction: given by odometry

$$\begin{aligned} \hat{x}^- &= \hat{x}_{WR_2} = 1.75m \\ P^- &= \sigma^2_{x_{WR_2}} = 0.0144m^2 \end{aligned}$$

- Sensor measurement: distance to the point

$$\begin{aligned} z &= \hat{d} + \mu_d = 1.1m \\ R &= \sigma_d^2 = 0.000121m^2 \end{aligned}$$

Kalman Filtering

- Measurement equation:

$$z = d = -x_{WR_2} + x_{WP}$$

- Filter gain: $K = P^-(P^- + R)^{-1} = 0.991667241$

- Estimation update:

$$\begin{aligned} \hat{x} &= \hat{x}^- + K(1.1 - 1.25) \\ &= 1.899m \end{aligned}$$

$$\begin{aligned} P &= (I - K)P^- \\ &= 0.000120m^2 \\ \sigma &= 0.011m \end{aligned}$$

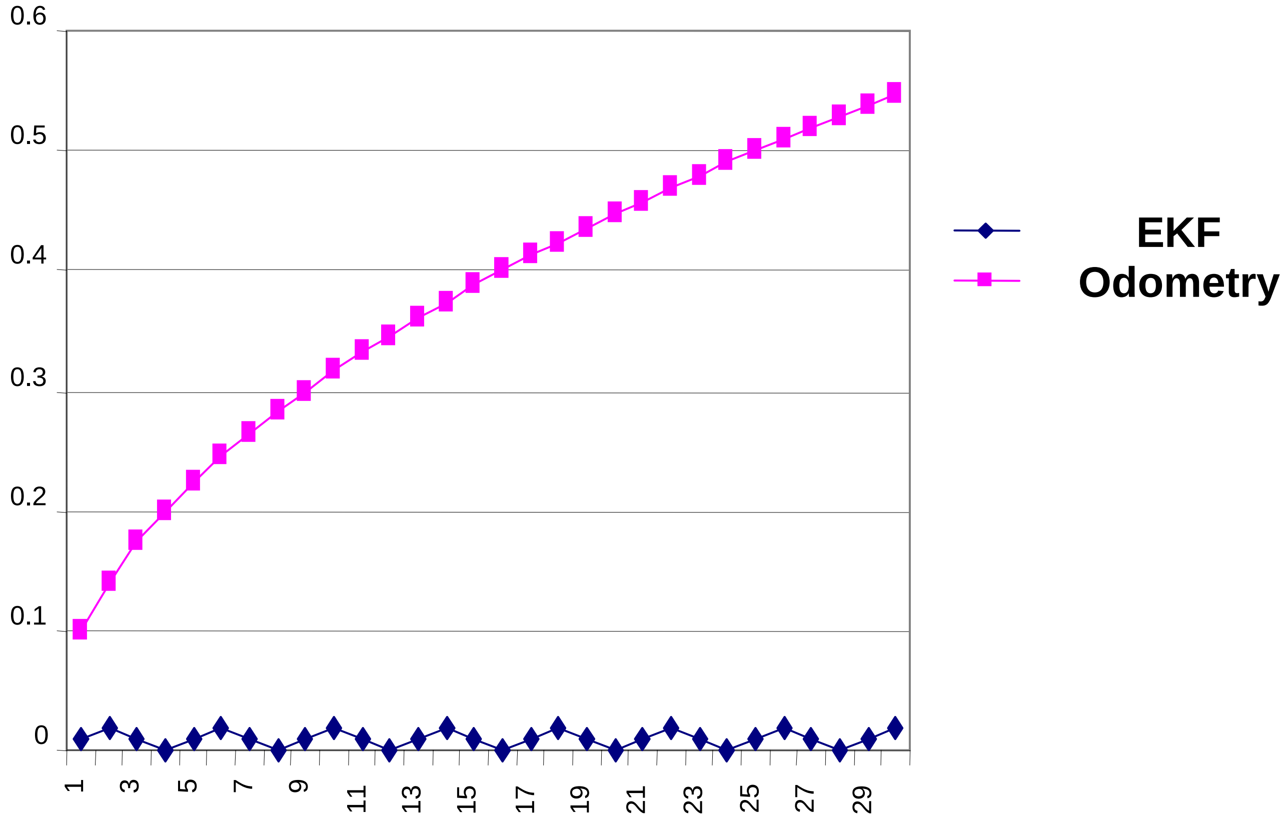
- Compare with:

$$\hat{x}^- = 1.75m \qquad \sigma^- = 0.12m$$

Better Robot Estimation

ROBOTICS UNIVERSIDAD DE ZARAGOZA

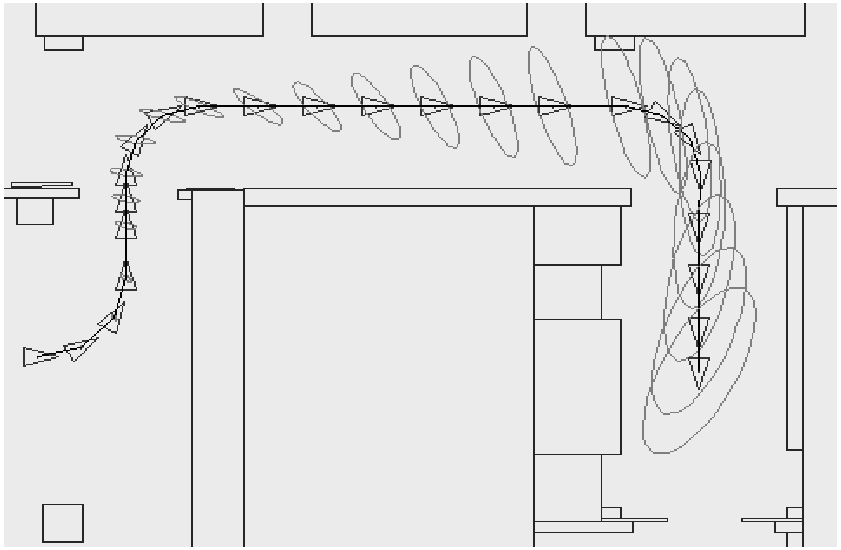
Bounded Robot Error

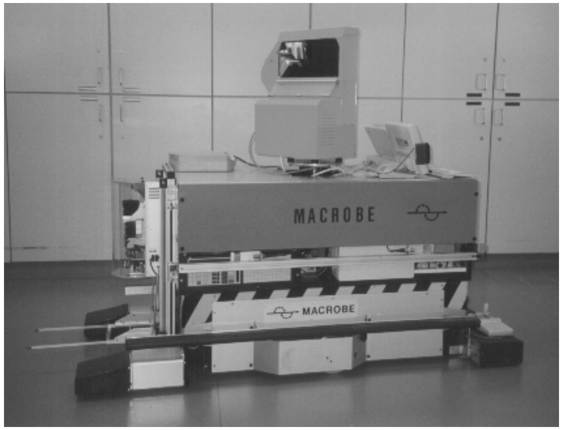


Moving close to point P

Real Example: MACROBE

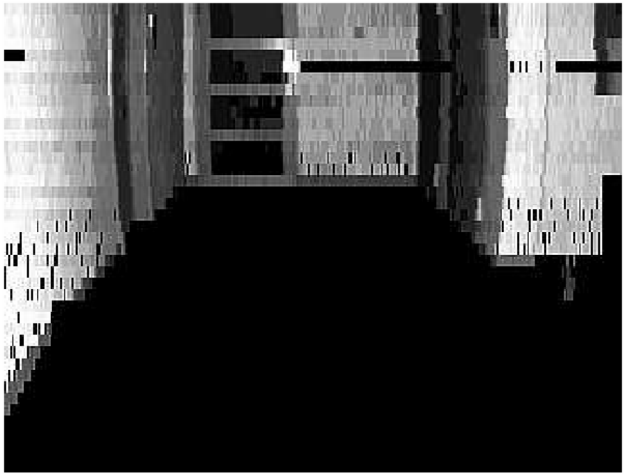
- Cumulative odometry



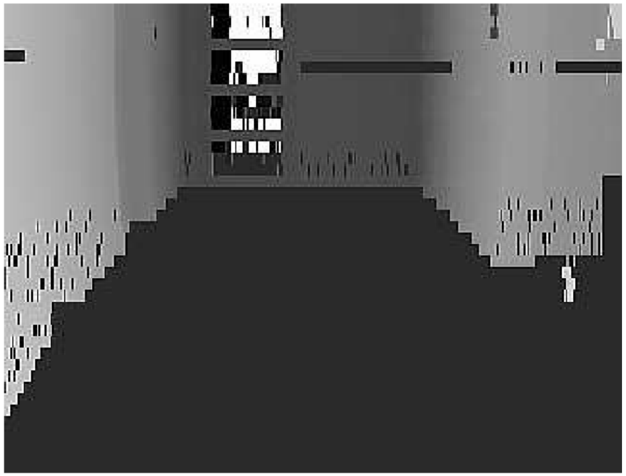


LSR-TUM, Germany

- A priori Map + Perception

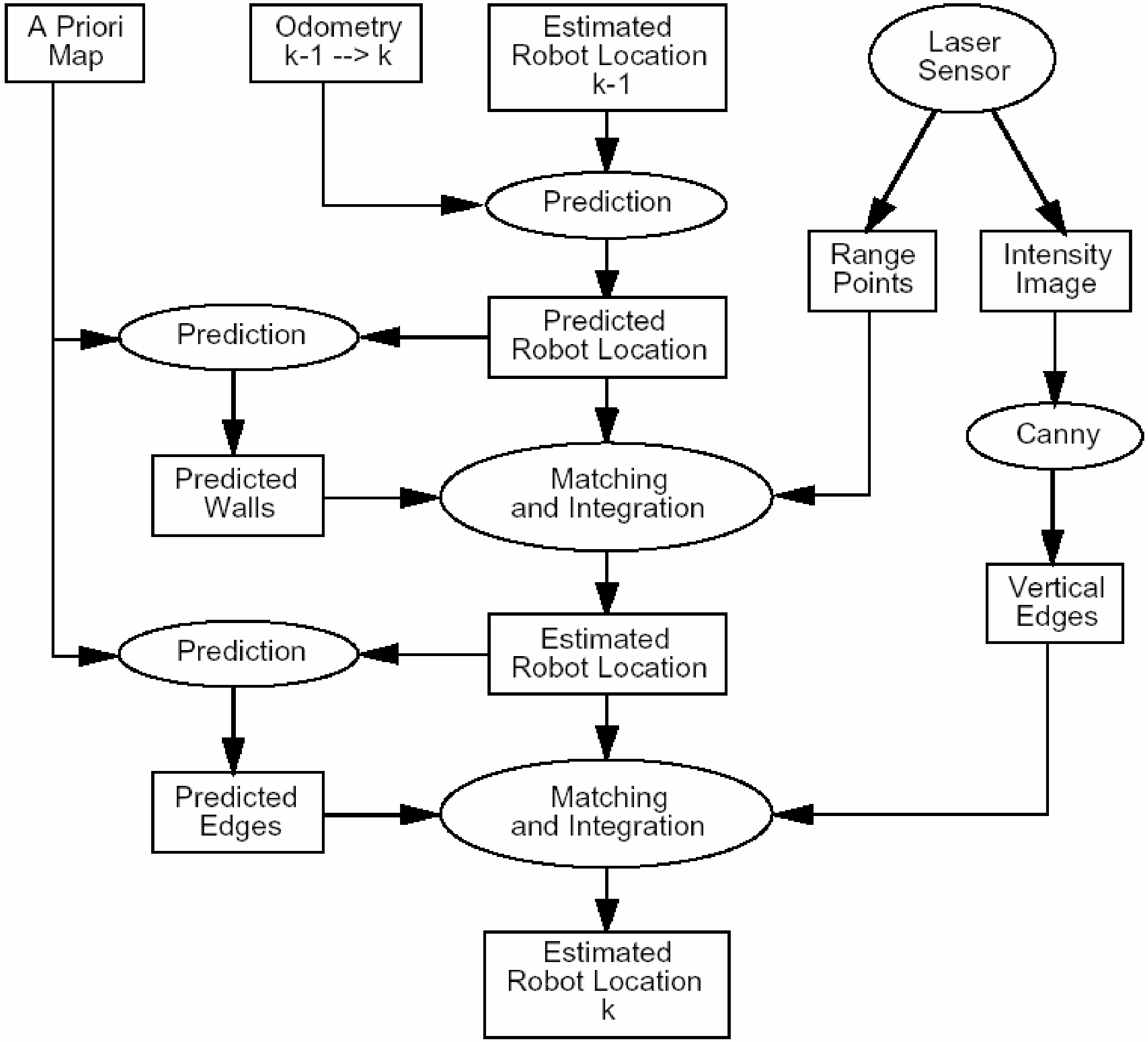


Vision

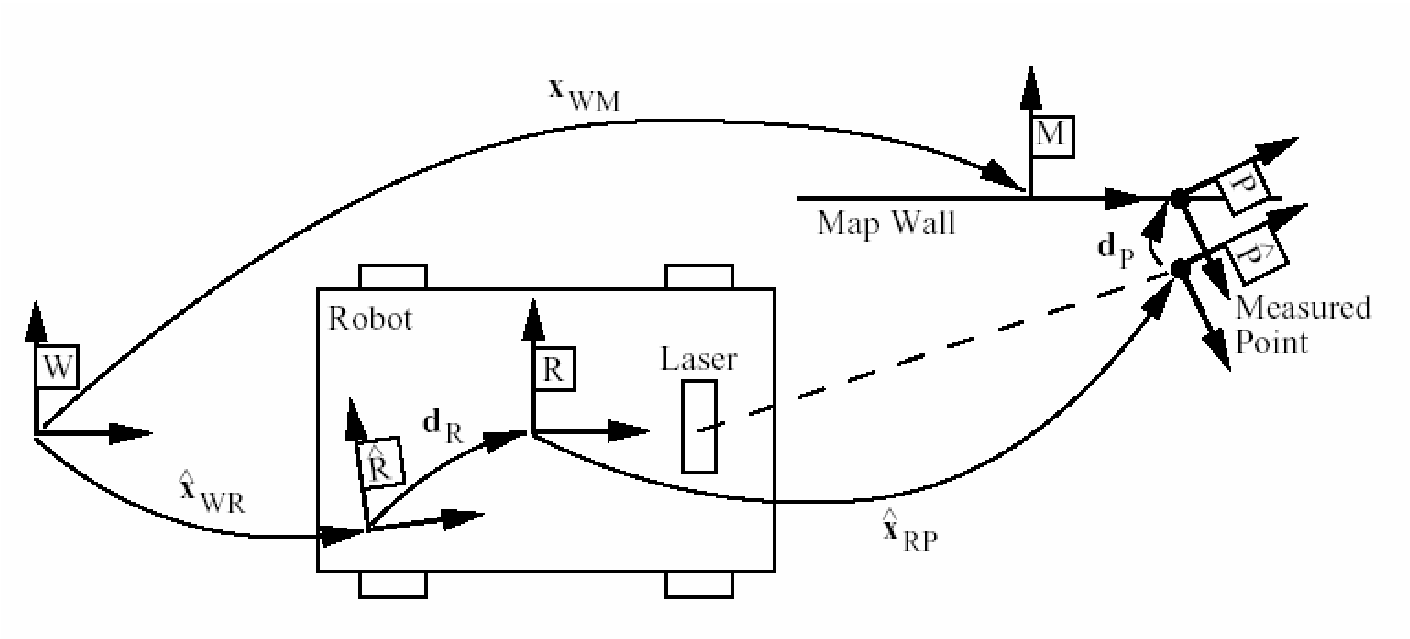


Laser

Robot Localization



Robot Localization: Laser points

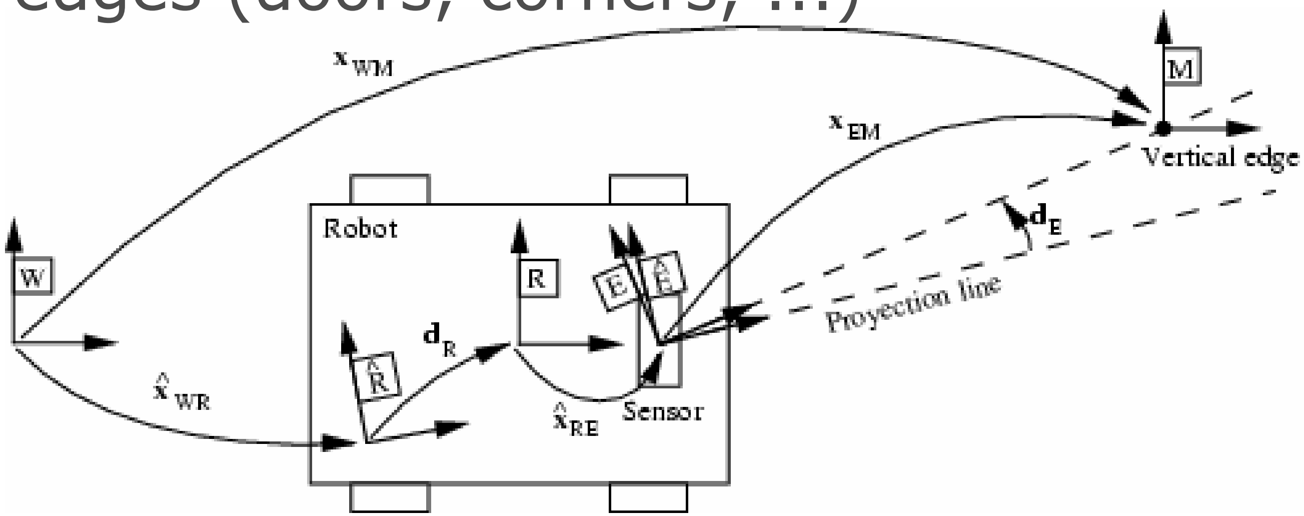


- Measurement equation: Point belongs to the wall

$$\begin{aligned} f(\mathbf{x},\ \mathbf{y}) &= y_{MP} \\ &= 0 \end{aligned}$$

Robot Localization: Vision

- Vertical edges (doors, corners, ...)



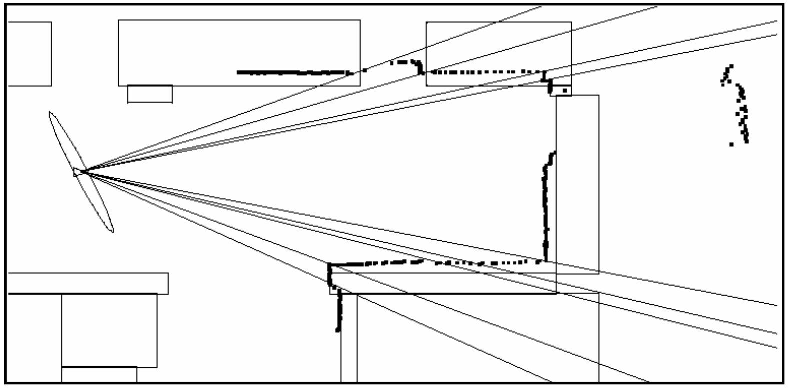
Measurement equation: Edge (2D point) on the projection line

$$\begin{aligned} f(\mathbf{x},\ \mathbf{y}) &= y_{EM} \\ &= 0 \end{aligned}$$

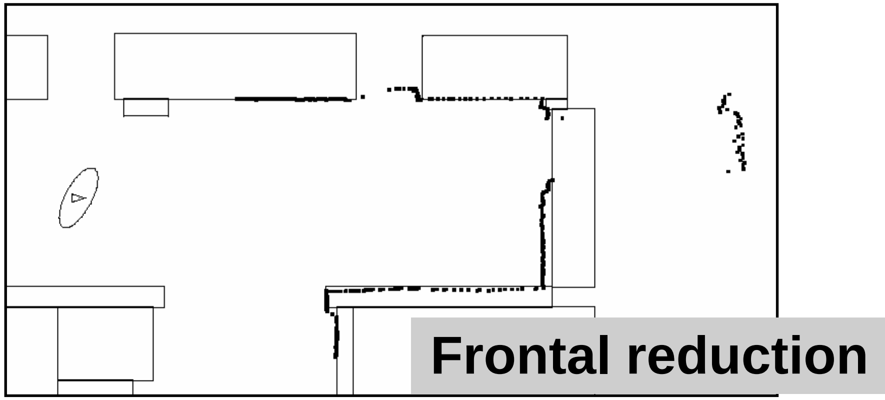
Multisensor Robot Localization

- **Correction** of localization errors.
- **Uncertainty reduction**

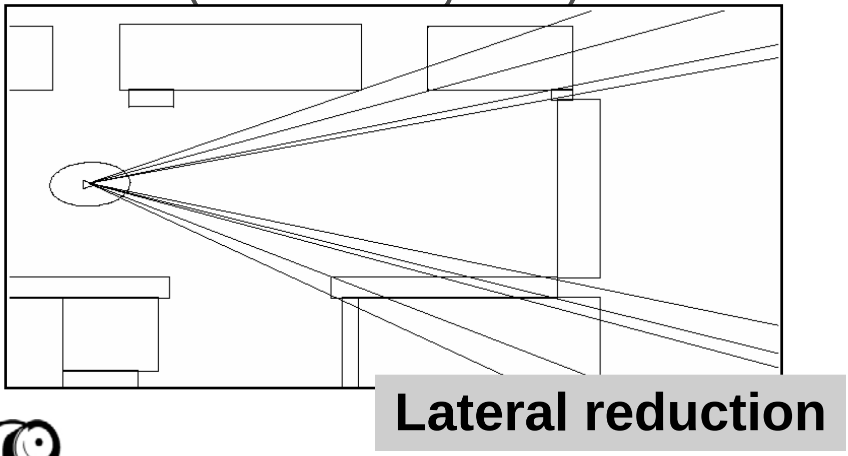
Prediction and sensor data:



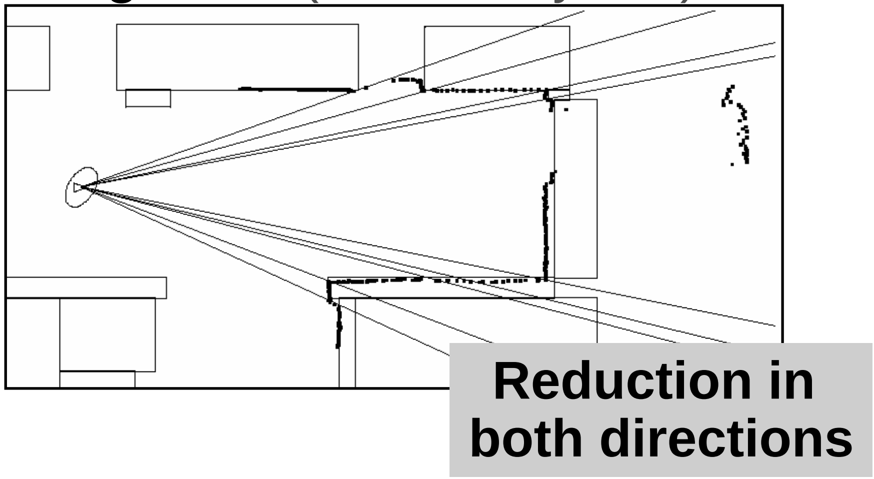
Laser (uncertainty x30)



Vision (uncertainty x30)



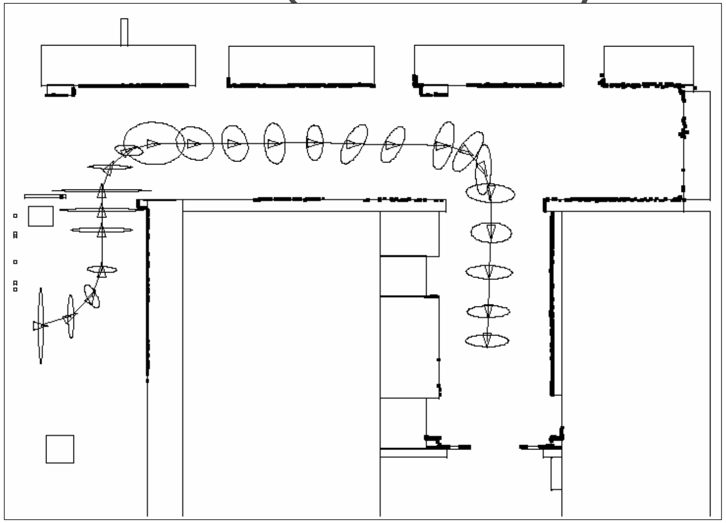
Integration (uncertainty x30)



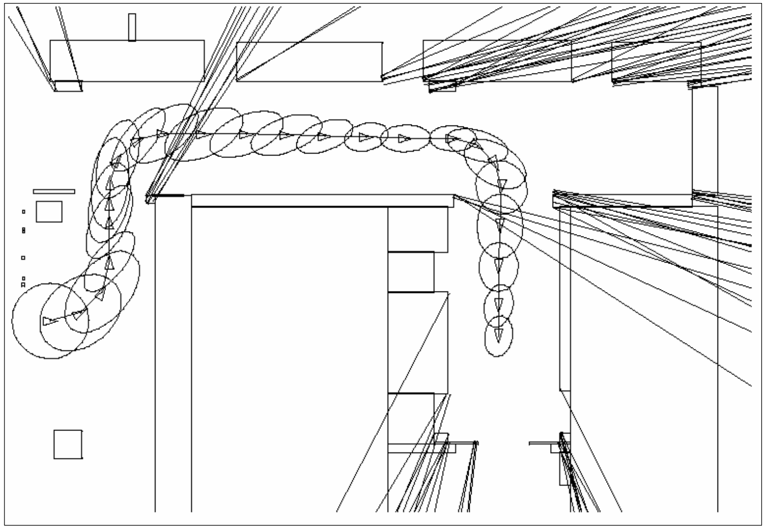
ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Multisensor Robot Localization

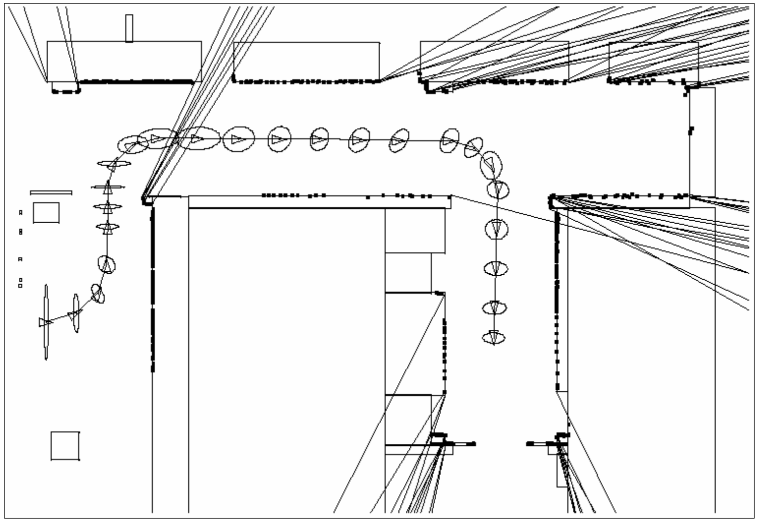
- Results (uncertainty x30)



Laser



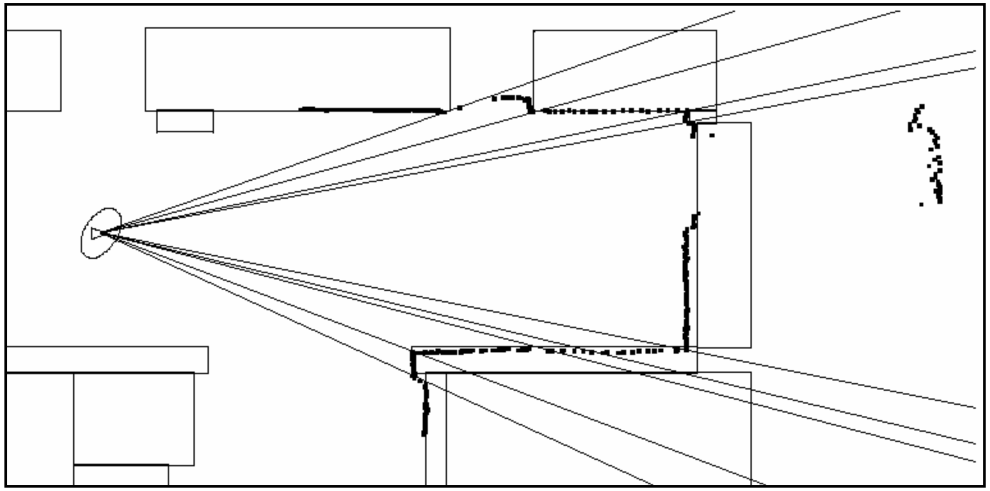
Vision



Laser + vision

The need for SLAM

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

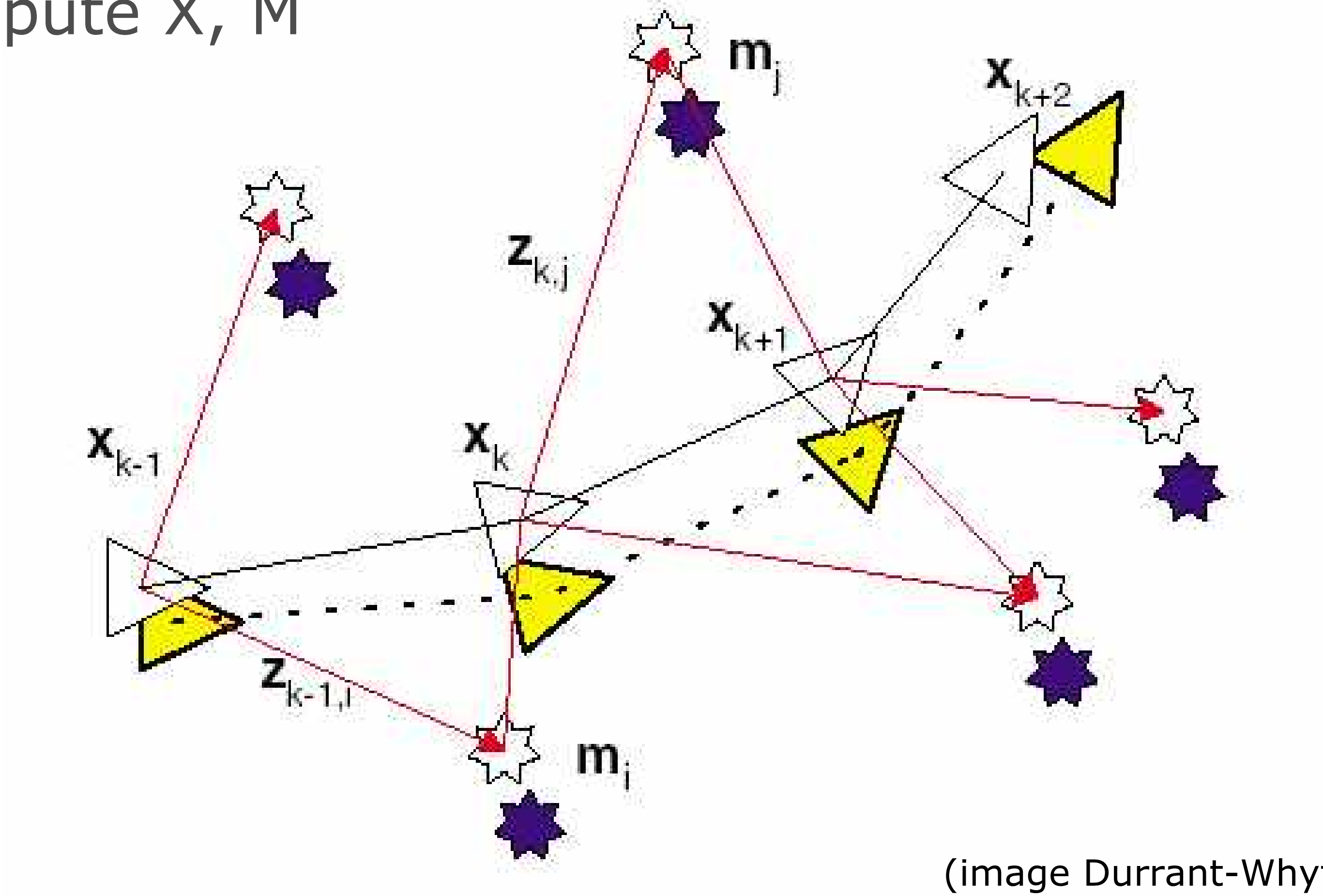


➔ Autonomous Map Building

4. SLAM

Simultaneous Localization And Mapping:

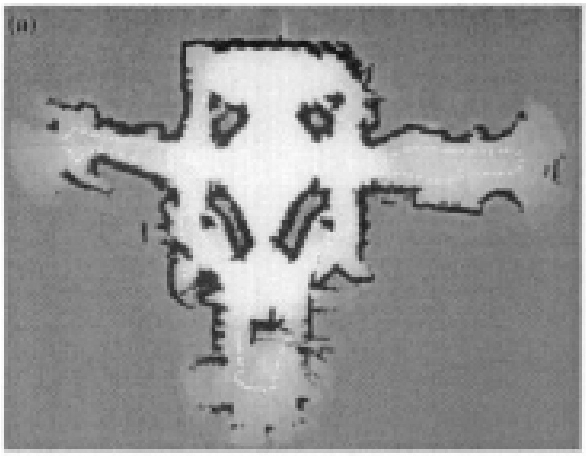
- Given U, Z
- Compute X, M

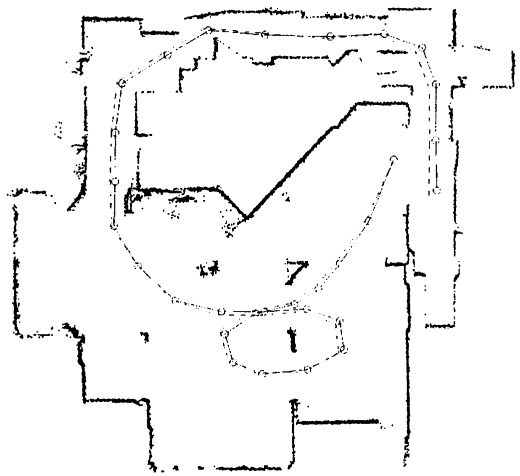


(image Durrant-Whyte)

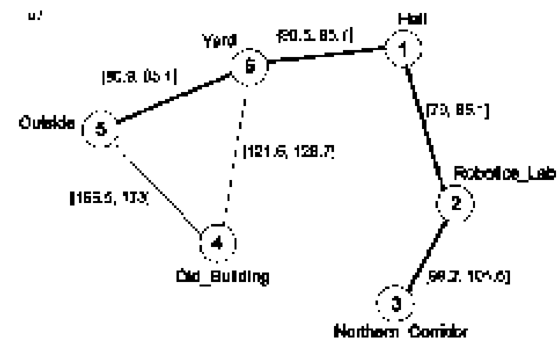
SLAM

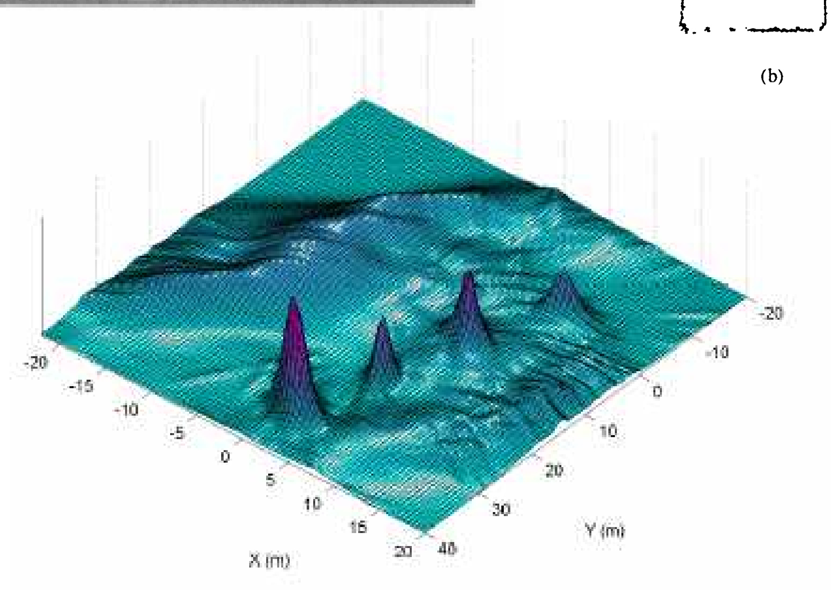
Mapping examples

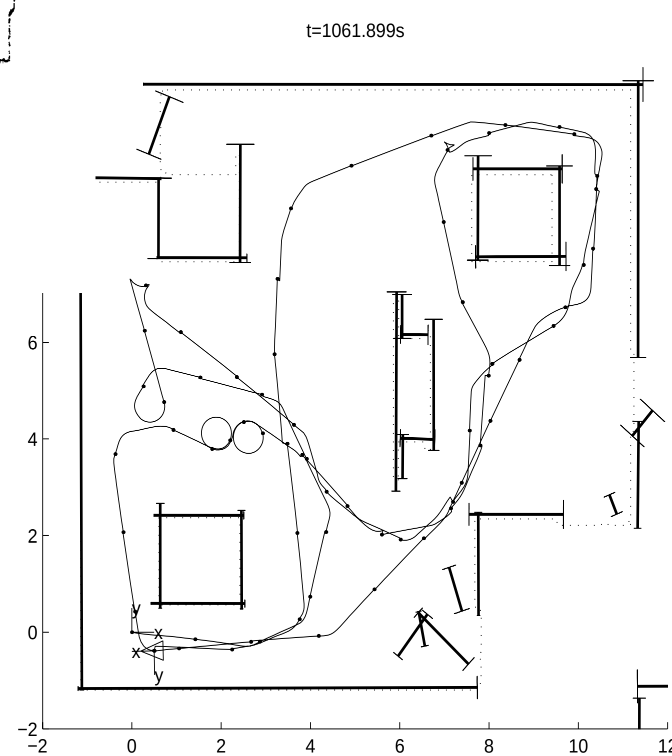




(b)

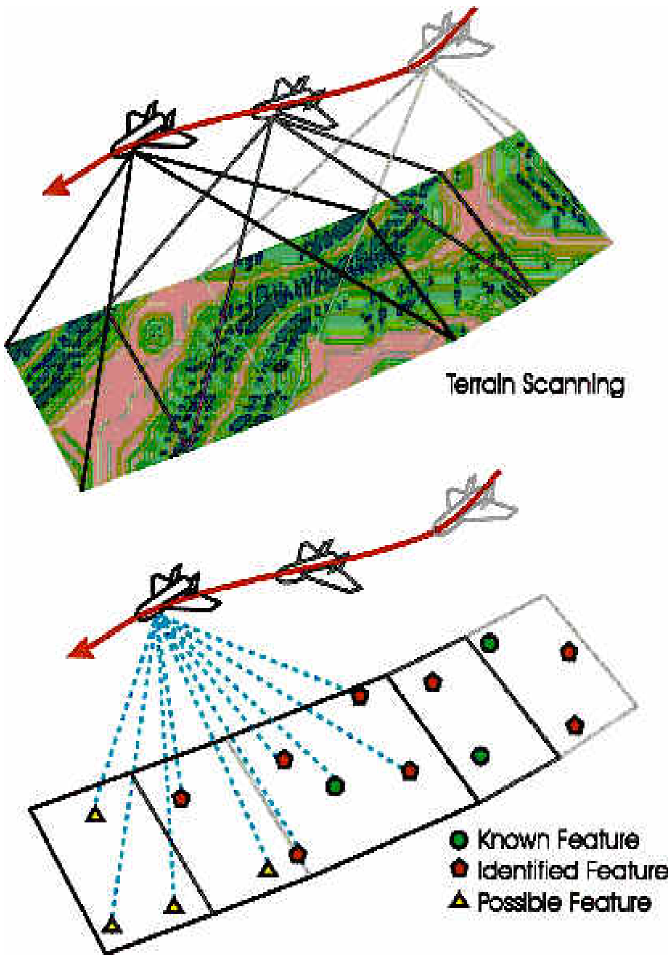


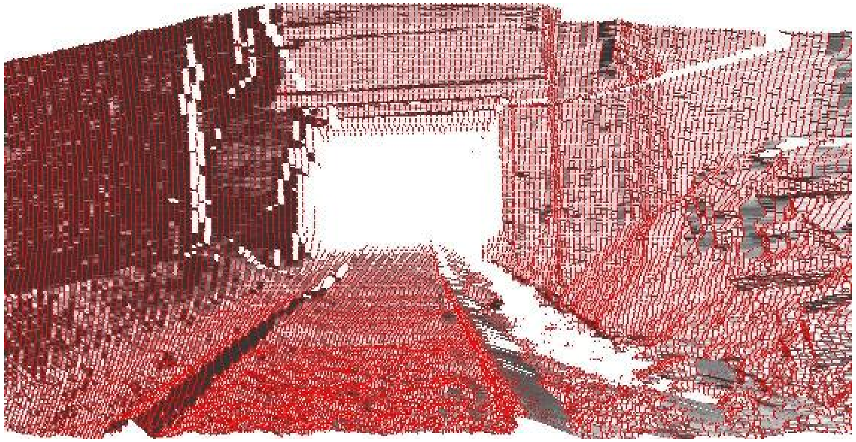


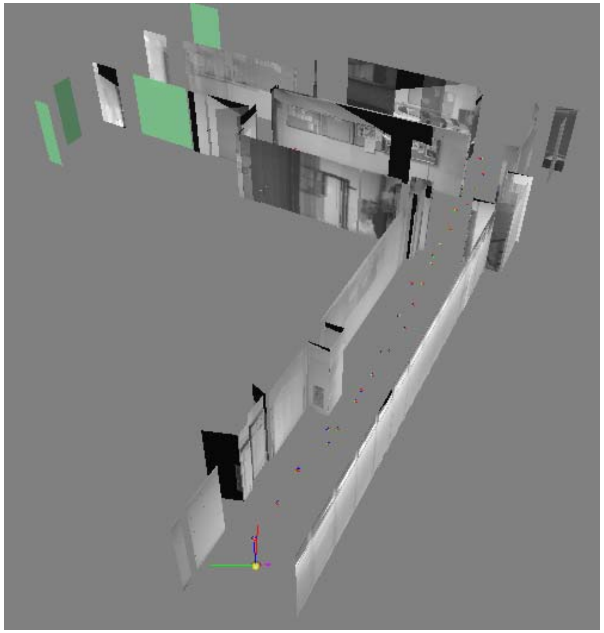


SLAM

Mapping examples







SLAM: EKF approach

- Algorithm:
  - Given the robot and map estimates at instant k
  - Move the robot and predict its location (EKF prediction)
  - Get sensor data (laser, sonar, vision,..) and extract features
  - Associate observed features with map features
  - Update the map and robot estimates at instant k+1 (EKF update)
  - Add new features to the map
- First proposed by:

R. Smith and P. Cheeseman, “On the Representation and Estimation of Spatial Uncertainty”, Int. J. Robotics Research 5(4), pp. 56-68, 1986.

R. Smith, M. Self and P. Cheeseman, “A Stochastic Map for Uncertain Spatial Relationships”, In O. Faugeras and G. Giralt (eds.), Robotics Research, The Fourth Int. Symp., pp. 467-474. The MIT Press, 1988.

SLAM: EKF approach

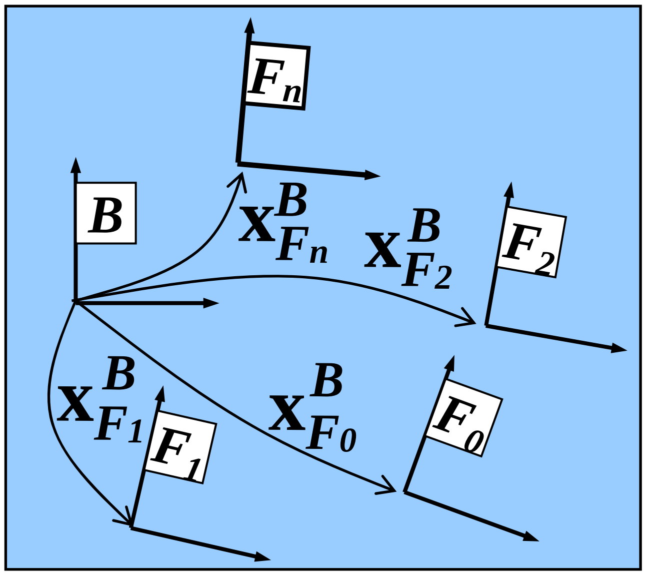
- Environment information related to a set of elements:

$$\mathcal{F} = \{B, F_0, F_1, \ldots, F_n\}$$

$F_0 =$ Vehicle

- represented by a map:

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



$$\hat{\mathbf{x}}^B_{\mathcal{F}} = \begin{bmatrix} \hat{\mathbf{x}}^B_{F_0} \\ \vdots \\ \hat{\mathbf{x}}^B_{F_n} \end{bmatrix}$$

$$\mathbf{P}^B_{\mathcal{F}} = \begin{bmatrix} \mathbf{P}^B_{F_0F_0} & \cdots & \mathbf{P}^B_{F_0F_n} \\ \vdots & \ddots & \vdots \\ \mathbf{P}^B_{F_nF_0} & \cdots & \mathbf{P}^B_{F_nF_n} \end{bmatrix}$$

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Transformations in 2D

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

Homogeneous matrix:

$$\mathbf{H} = \begin{pmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} \cos\phi_1 & -\sin\phi_1 & x_1 \\ \sin\phi_1 & \cos\phi_1 & y_1 \\ 0 & 0 & 1 \end{pmatrix}$$

Inversion:

$$\mathbf{H}^{-1} = \begin{pmatrix} \mathbf{R}^{\mathbf{T}} & -\mathbf{R}^T\mathbf{p} \\ 0 & 1 \end{pmatrix}$$

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

Transformations 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:

$$\mathbf{H}_3 = \mathbf{H}_1 \mathbf{H}_2 = \begin{pmatrix} \mathbf{R}_1 \mathbf{R}_2 & \mathbf{p}_1 + \mathbf{R}_1 \mathbf{p}_2 \\ \mathbf{0} & 1 \end{pmatrix}$$

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

Transformations in 2D

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

Jacobians:

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

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

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

Map Features in 2D

Points:

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

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

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

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

Lines:

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

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

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

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

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:

$$\mathbf{x}^B_{\mathcal{F}_{k|k-1}} \simeq \hat{\mathbf{x}}^B_{\mathcal{F}_{k|k-1}} + \mathbf{F}_k(\mathbf{x}^B_{\mathcal{F}_{k-1}} - \hat{\mathbf{x}}^B_{\mathcal{F}_{k-1}}) + \mathbf{G}_k(\mathbf{x}^{R_{k-1}}_{R_k} - \hat{\mathbf{x}}^{R_{k-1}}_{R_k})$$

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

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

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

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

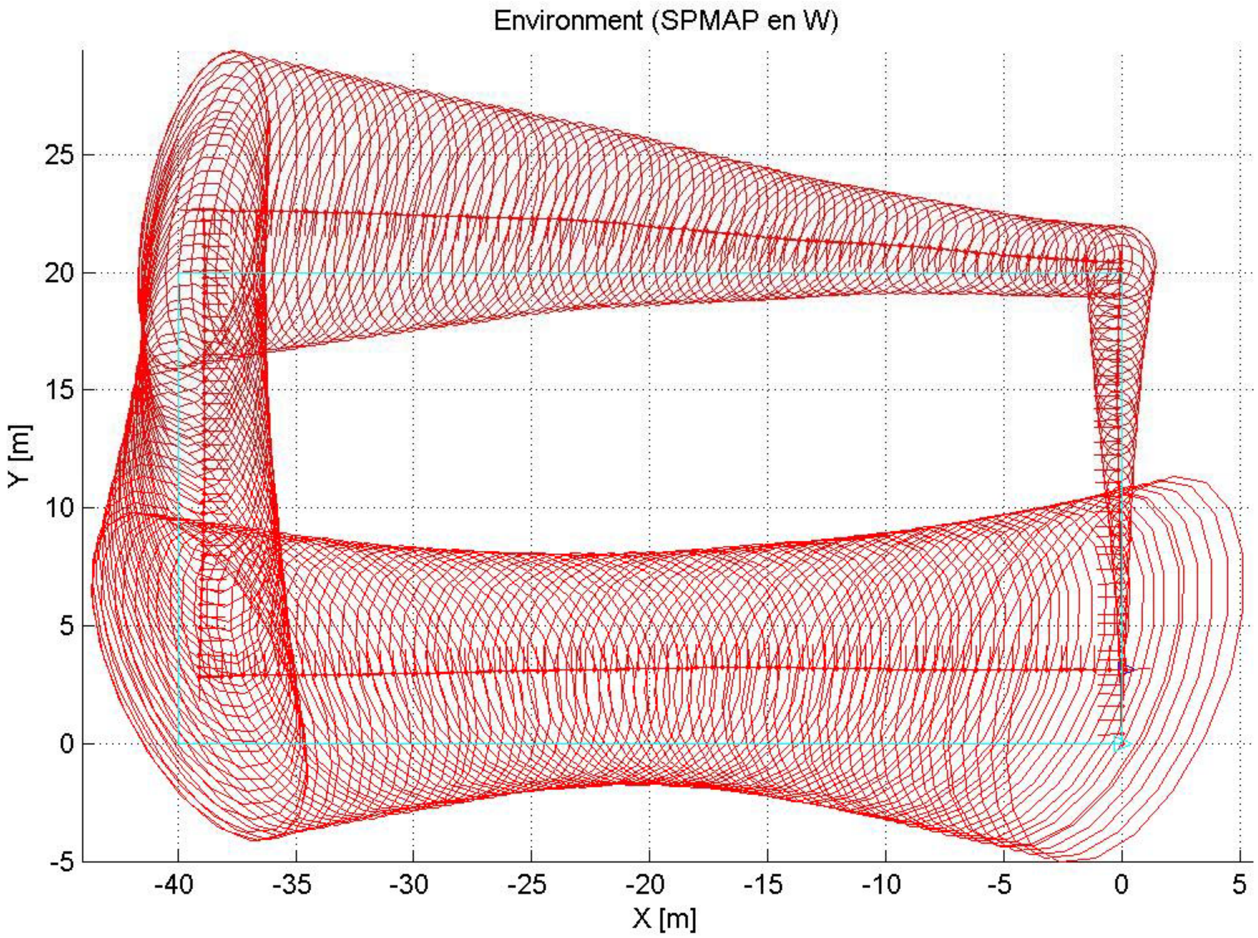
EKF-SLAM: Robot Motion

$$\mathbf{P}^B_{\mathcal{F}_{k-1}} = \begin{pmatrix} \mathbf{P}_{F_0F_0} & \mathbf{P}_{F_0F_1} & \cdots & \mathbf{P}_{F_0F_n} \\ \mathbf{P}^T_{F_0F_1} & \mathbf{P}_{F_1F_1} & \cdots & \mathbf{P}_{F_1F_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^T_{F_0F_n} & \mathbf{P}^T_{F_1F_n} & \cdots & \mathbf{P}_{F_nF_n} \end{pmatrix}$$

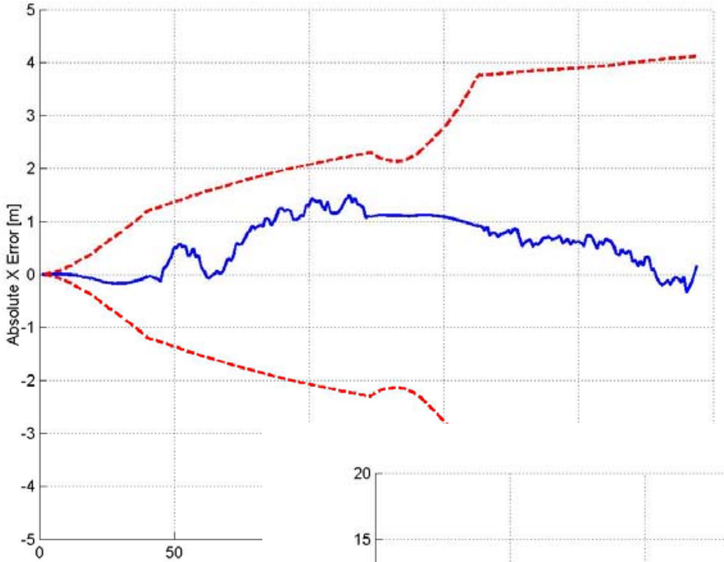
$$\mathbf{J}_1 = \mathbf{J}_{1\oplus}\left\{\hat{\mathbf{x}}^B_{R_{k-1}}, \hat{\mathbf{x}}^{R_{k-1}}_{R_k}\right\} \quad ; \quad \mathbf{J}_2 = \mathbf{J}_{2\oplus}\left\{\hat{\mathbf{x}}^B_{R_{k-1}}, \hat{\mathbf{x}}^{R_{k-1}}_{R_k}\right\}$$

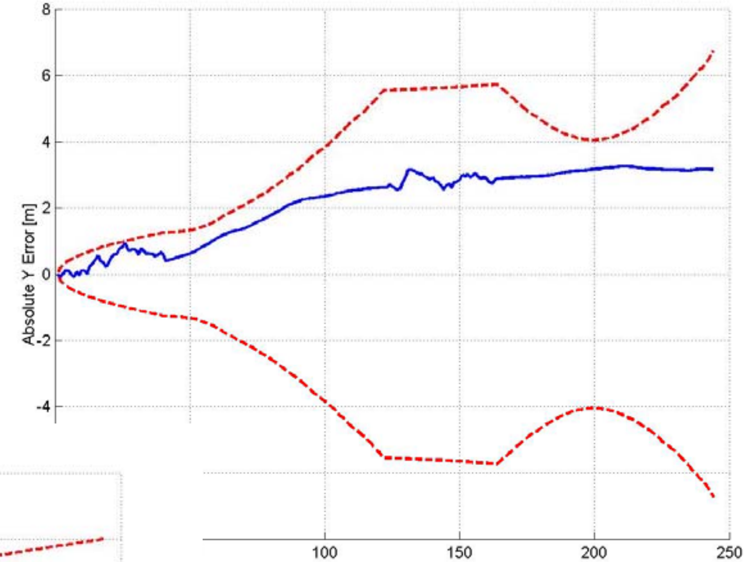
$$\mathbf{P}^B_{\mathcal{F}_{k|k-1}} = \begin{pmatrix} \mathbf{J}_1\mathbf{P}_{F_0F_0}\mathbf{J}_1^T + \mathbf{J}_2\mathbf{Q}_k\mathbf{J}_2^T & \mathbf{J}_1\mathbf{P}_{F_0F_1} & \cdots & \mathbf{J}_1\mathbf{P}_{F_0F_n} \\ \mathbf{J}_1^T\mathbf{P}^T_{F_0F_1} & \mathbf{P}_{F_1F_1} & \cdots & \mathbf{P}_{F_1F_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{J}_1^T\mathbf{P}^T_{F_0F_n} & \mathbf{P}^T_{F_1F_n} & \cdots & \mathbf{P}_{F_nF_n} \end{pmatrix}$$

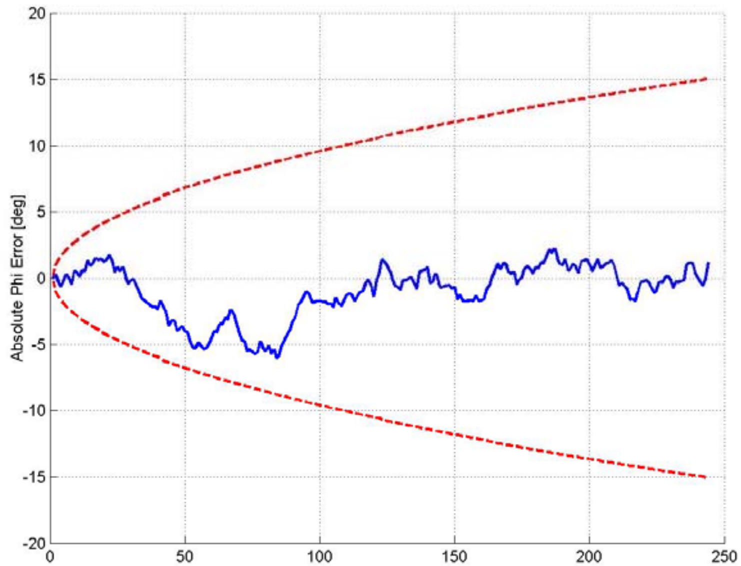
EKF-SLAM: Robot Motion



EKF-SLAM: Robot Motion







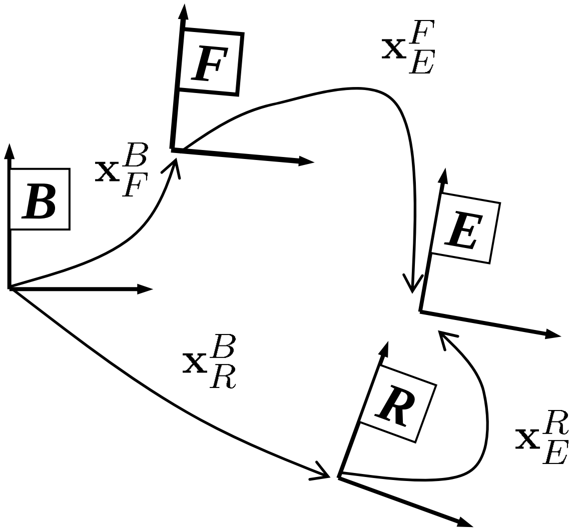
Error$\pm 2\sigma$ (prob. 0.95)

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: Feature 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:

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

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

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: Feature Observations

Linearization:

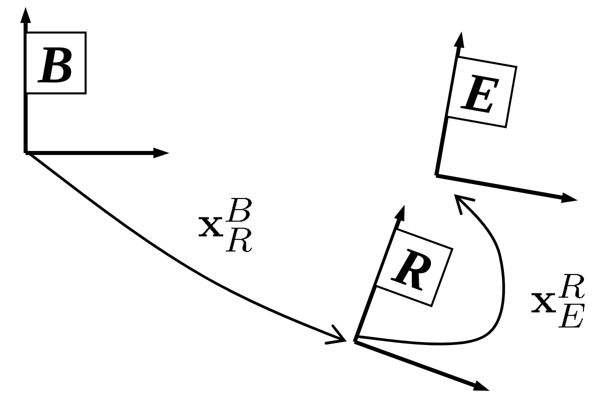
$$\begin{aligned} \mathbf{z}_k &\simeq \mathbf{h}_k(\hat{\mathbf{x}}^B_{\mathcal{F}_{k|k-1}}) + \mathbf{H}_k(\mathbf{x}^B_{\mathcal{F}_k} - \hat{\mathbf{x}}^B_{\mathcal{F}_{k|k-1}}) + \mathbf{w}_k \\ \mathbf{H}_k &= \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}^B_{\mathcal{F}_k}}\right|_{(\hat{\mathbf{x}}^B_{\mathcal{F}_{k|k-1}})} \end{aligned}$$

EKF map update:

$$\begin{aligned} \hat{\mathbf{x}}^B_{\mathcal{F}_k} &= \hat{\mathbf{x}}^B_{\mathcal{F}_{k|k-1}} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}^B_{\mathcal{F}_{k|k-1}})) \\ \mathbf{P}^B_{\mathcal{F}_k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}^B_{\mathcal{F}_{k|k-1}} \\ \mathbf{K}_k &= \mathbf{P}^B_{\mathcal{F}_{k|k-1}}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}^B_{\mathcal{F}_{k|k-1}}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \end{aligned}$$

EKF-SLAM: Adding features

$$\mathbf{x}_E^B = \mathbf{x}_{R_k}^B \oplus \mathbf{x}_E^{R_k}$$



Observation model (white noise):

$$\begin{aligned} \mathbf{x}_E^{R_k} &= \hat{\mathbf{x}}_E^{R_k} + \mathbf{w}_k \\ E[\mathbf{w}_k] &= \mathbf{0} \\ E[\mathbf{w}_k \mathbf{w}_j^T] &= \delta_{kj} \mathbf{R}_k \end{aligned}$$

Linearized equation:

$$\mathbf{x}_{\mathcal{F}_{k+}}^B \simeq \hat{\mathbf{x}}_{\mathcal{F}_{k+}}^B + \mathbf{F}_k(\mathbf{x}_{\mathcal{F}_k}^B - \hat{\mathbf{x}}_{\mathcal{F}_k}^B) + \mathbf{G}_k(\mathbf{x}_E^{R_k} - \hat{\mathbf{x}}_E^{R_k})$$

$$\hat{\mathbf{x}}_{\mathcal{F}_{k+}}^B = \begin{bmatrix} \hat{\mathbf{x}}_{R_k}^B \\ \hat{\mathbf{x}}_{F_{1,k}}^B \\ \vdots \\ \hat{\mathbf{x}}_{F_{n,k}}^B \\ \hat{\mathbf{x}}_{R_k}^B \oplus \hat{\mathbf{x}}_E^{R_k} \end{bmatrix}$$

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

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{J}_{2\oplus}\left\{\hat{\mathbf{x}}_{R_k}^B, \hat{\mathbf{x}}_E^{R_k}\right\} \end{bmatrix}$$

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

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: Adding features

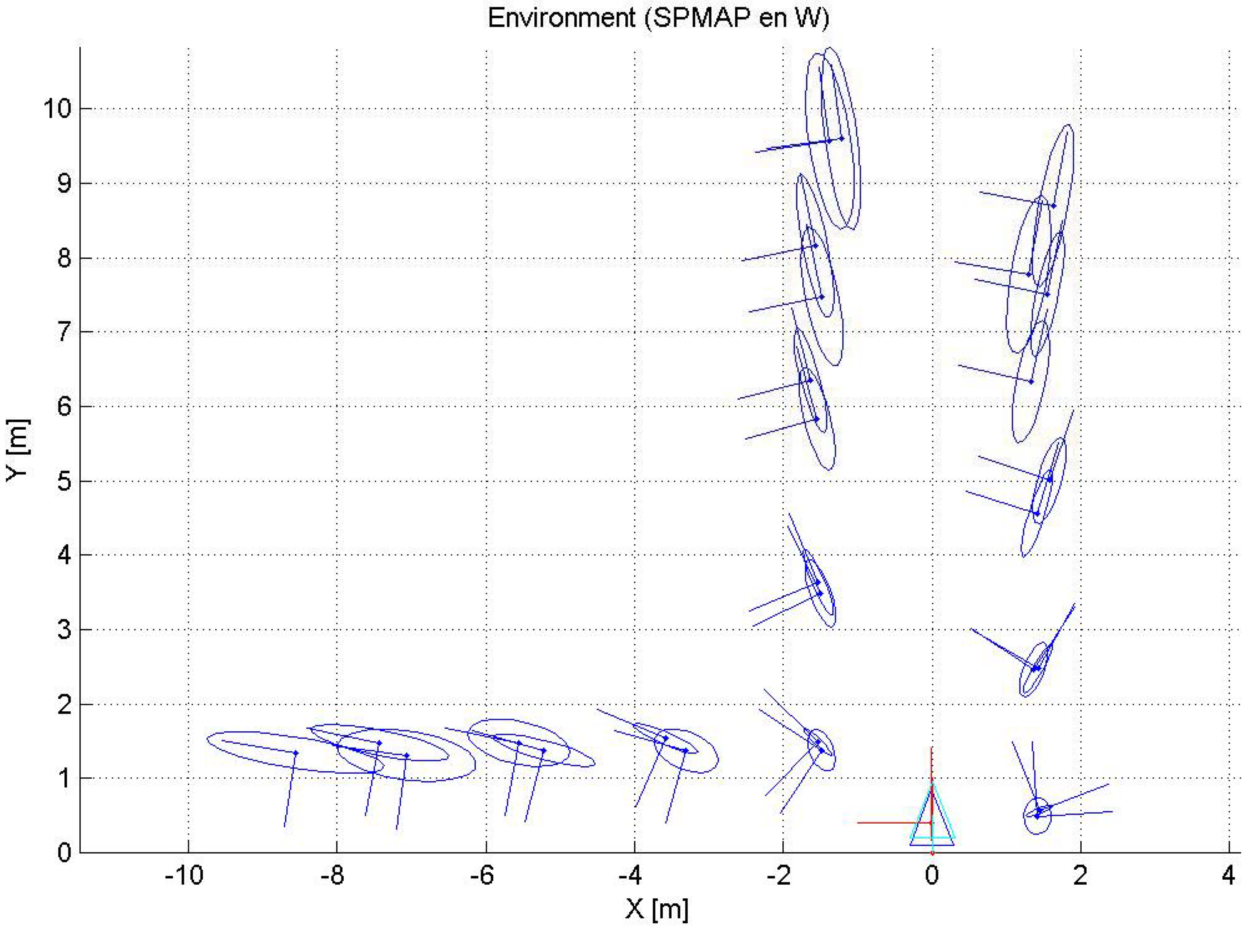
$$\mathbf{P}^B_{\mathcal{F}_k} = \begin{pmatrix} \mathbf{P}_{F_0F_0} & \mathbf{P}_{F_0F_1} & \cdots & \mathbf{P}_{F_0F_n} \\ \mathbf{P}^T_{F_0F_1} & \mathbf{P}_{F_1F_1} & \cdots & \mathbf{P}_{F_1F_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^T_{F_0F_n} & \mathbf{P}^T_{F_1F_n} & \cdots & \mathbf{P}_{F_nF_n} \end{pmatrix}$$

$$\mathbf{J}_1 = \mathbf{J}_{1\oplus}\left\{\hat{\mathbf{x}}^B_{R_k}, \hat{\mathbf{x}}^{R_k}_E\right\} \quad ; \quad \mathbf{J}_2 = \mathbf{J}_{2\oplus}\left\{\hat{\mathbf{x}}^B_{R_k}, \hat{\mathbf{x}}^{R_k}_E\right\}$$

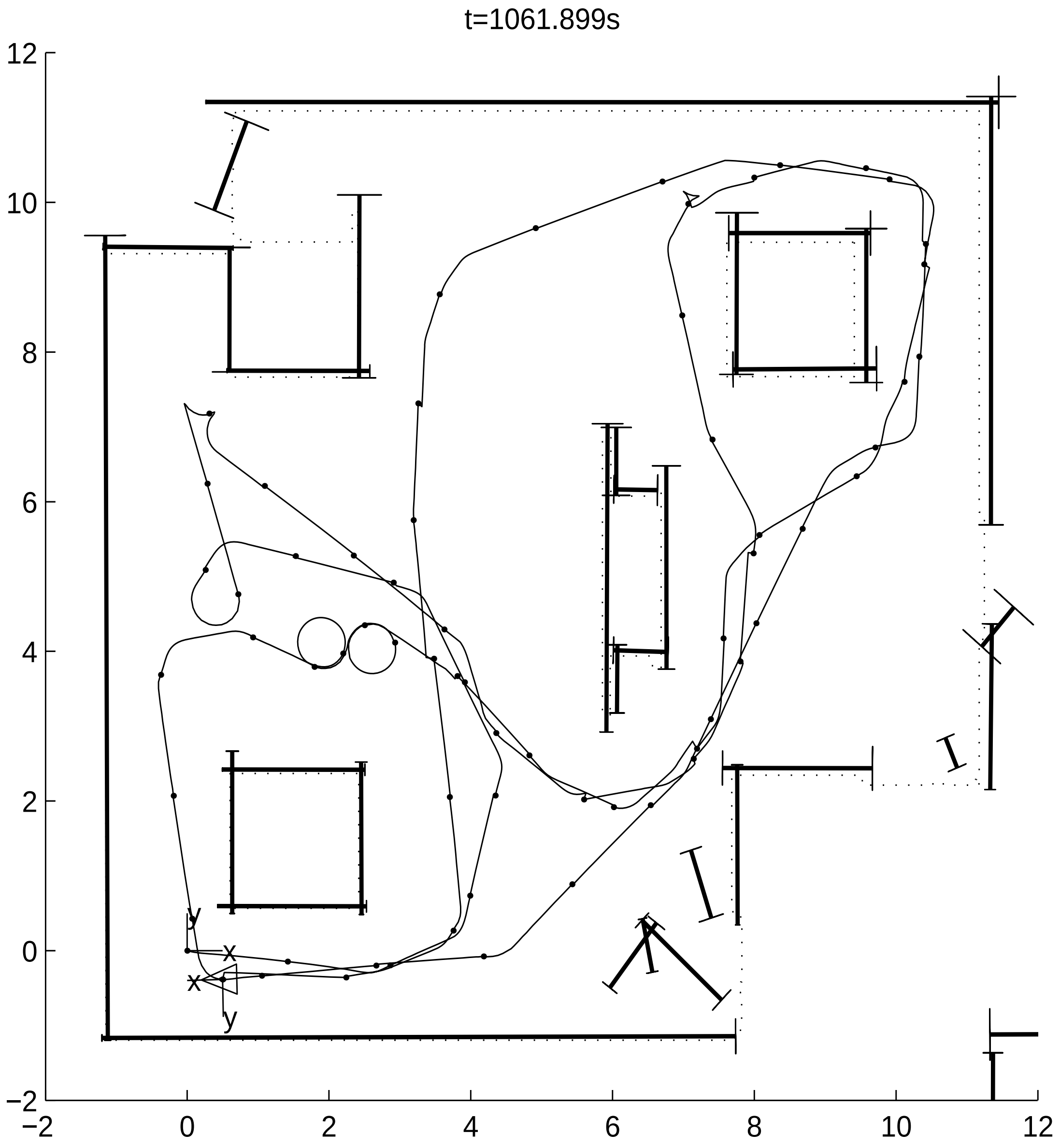
$$\mathbf{P}^B_{\mathcal{F}_{k+}} = \begin{pmatrix} \mathbf{P}_{F_0F_0} & \mathbf{P}_{F_0F_1} & \cdots & \mathbf{P}_{F_0F_n} & \mathbf{P}_{F_0}\mathbf{J}_1^T \\ \mathbf{P}^T_{F_0F_1} & \mathbf{P}_{F_1F_1} & \cdots & \mathbf{P}_{F_1F_n} & \mathbf{P}^T_{F_0F_1}\mathbf{J}_1^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{P}^T_{F_0F_n} & \mathbf{P}^T_{F_1F_n} & \cdots & \mathbf{P}_{F_nF_n} & \mathbf{P}^T_{F_0F_n}\mathbf{J}_1^T \\ \mathbf{J}_1\mathbf{P}_{F_0F_0} & \mathbf{J}_1\mathbf{P}_{F_0F_1} & \cdots & \mathbf{J}_1\mathbf{P}_{F_0F_n} & \mathbf{J}_1\mathbf{P}_{F_0F_0}\mathbf{J}_1^T + \mathbf{J}_2\mathbf{R}_k\mathbf{J}_2^T \end{pmatrix}$$

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

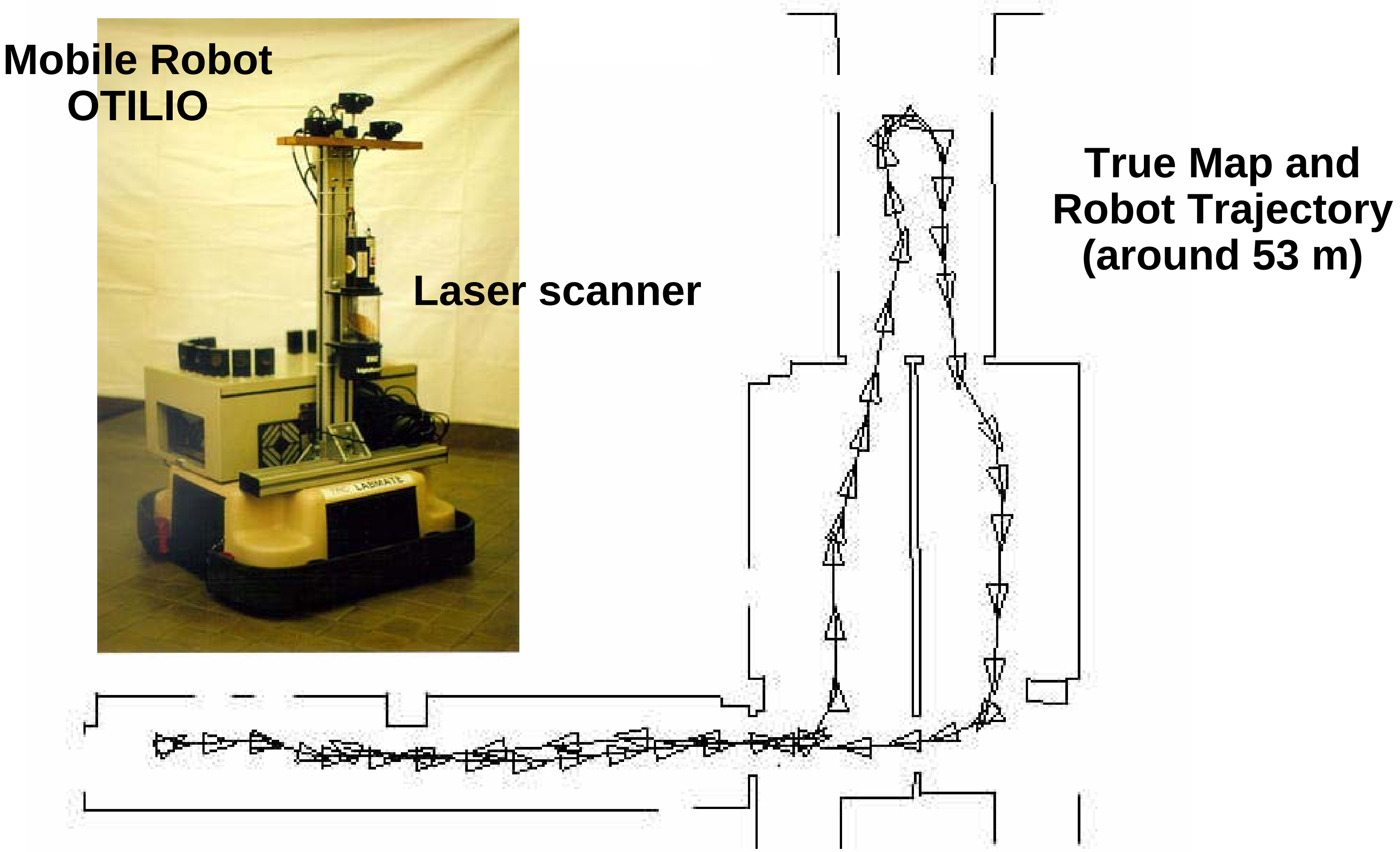
SLAM



EKF- SLAM step by step



Important SLAM Properties

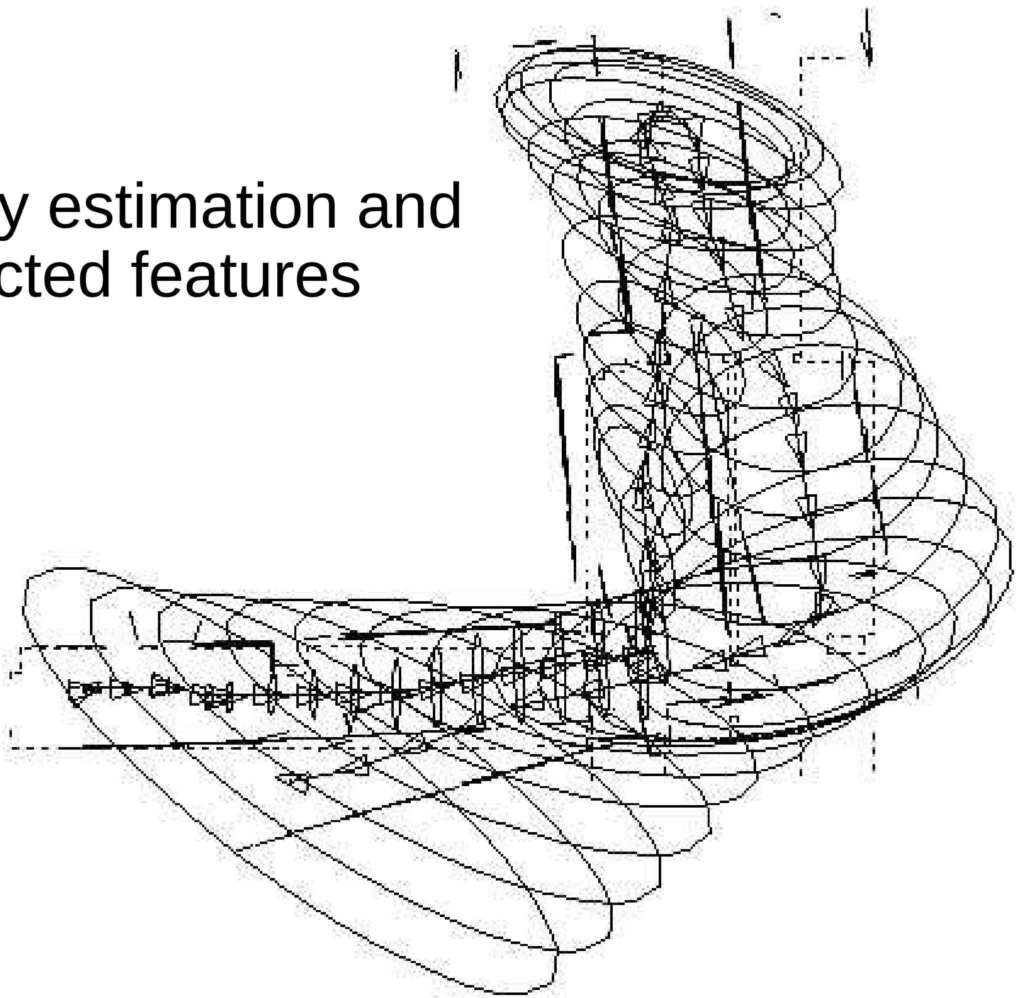


J.A. Castellanos, J.M.M. Montiel, J. Neira and J.D. Tardós.: The SPmap: A Probabilistic Framework for Simultaneous Localization and Map Building, IEEE Trans. Robotics and Automation, vol. 15 no. 5, pp. 948-953,Oct 1999

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

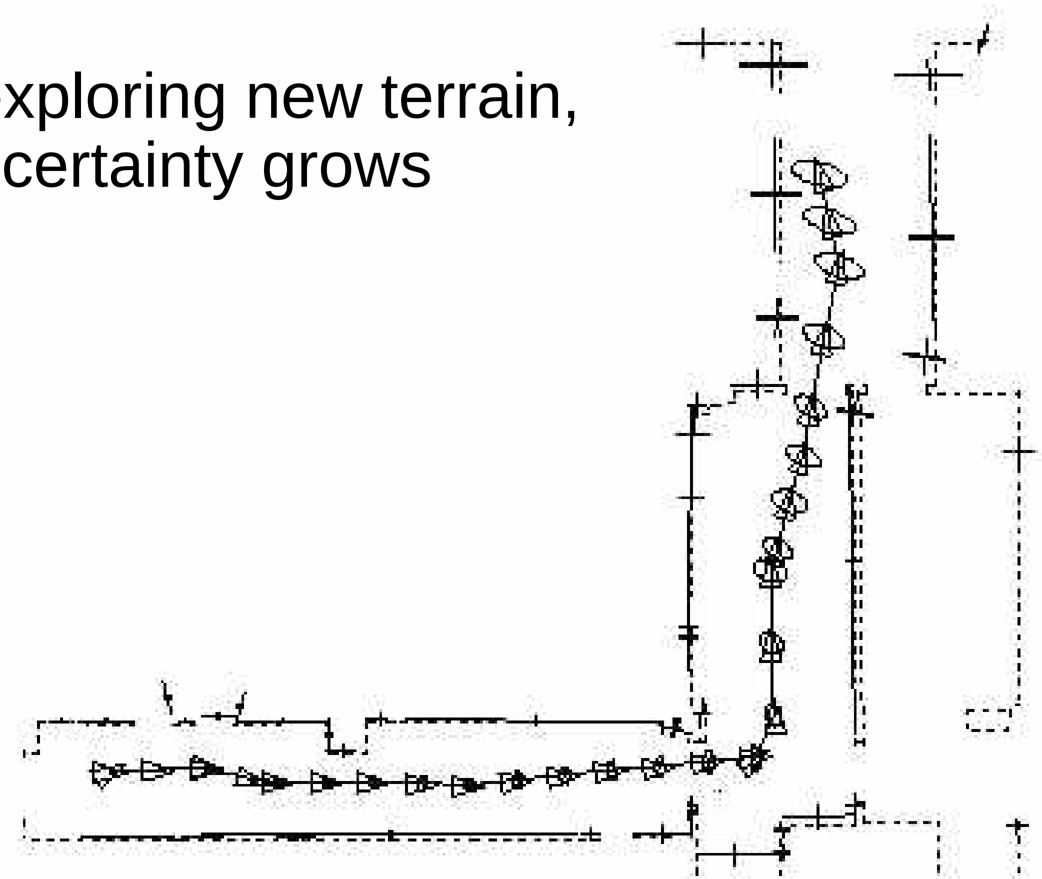
Odometry

Odometry estimation and extracted features



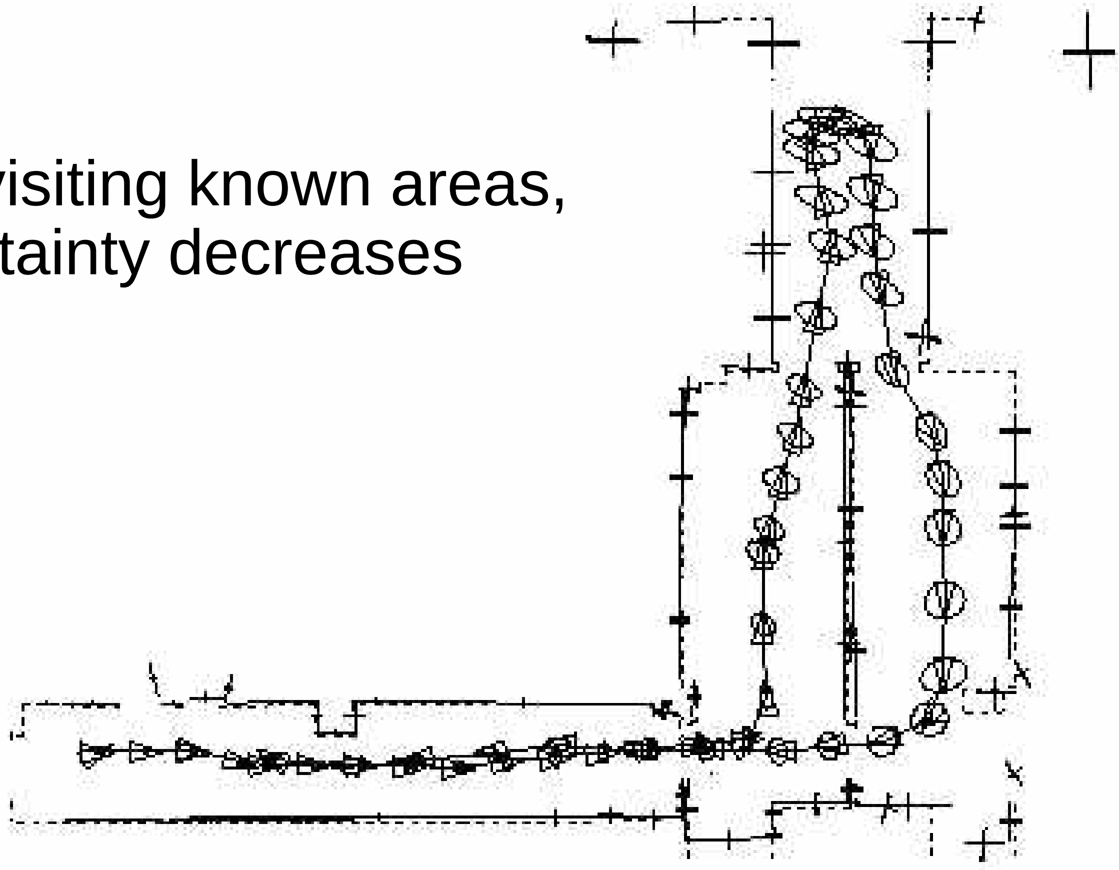
Exploration

When exploring new terrain, uncertainty grows



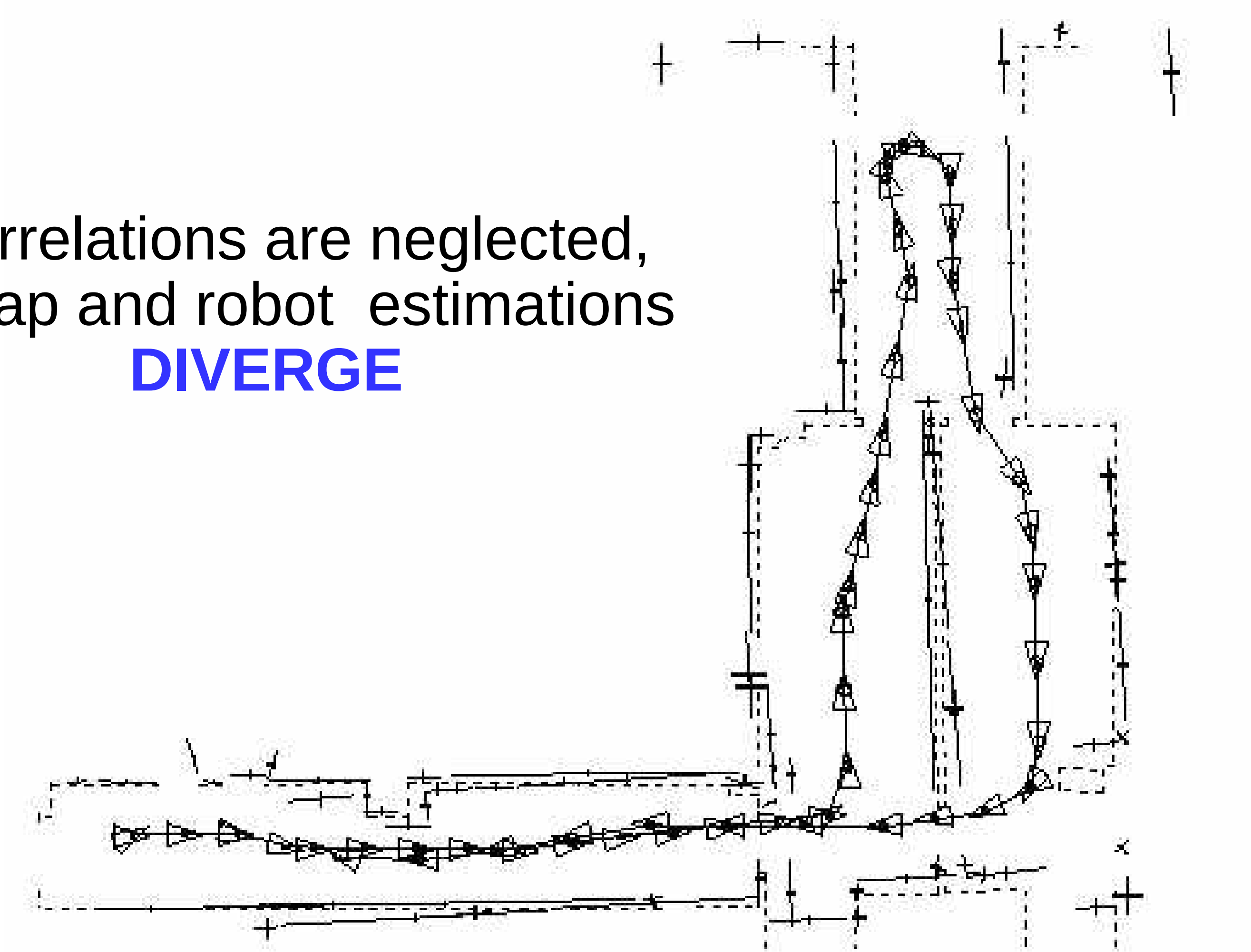
Revisiting (Loop closing 30m)

When revisiting known areas,
uncertainty decreases

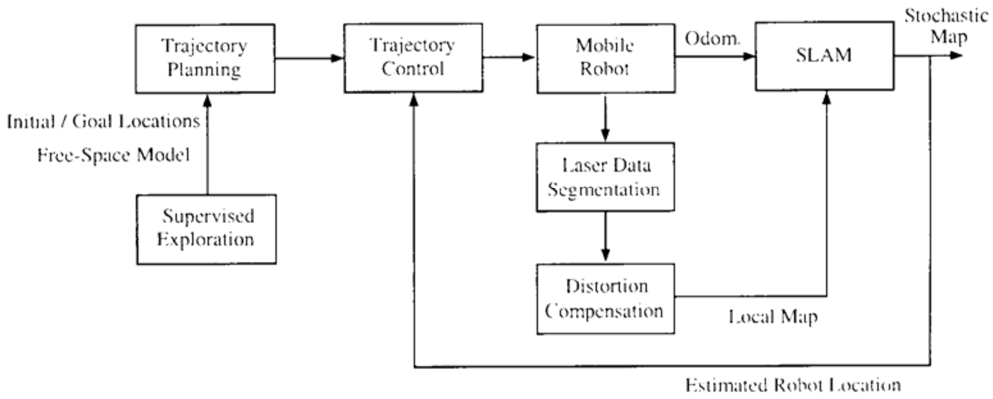


The Importance of Correlations

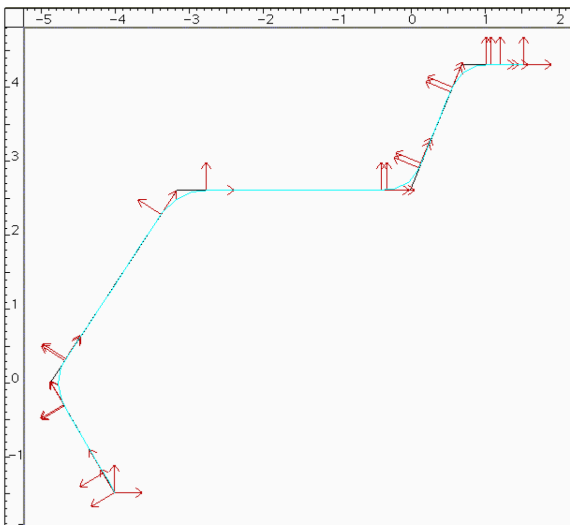
If correlations are neglected,
the map and robot estimations
DIVERGE

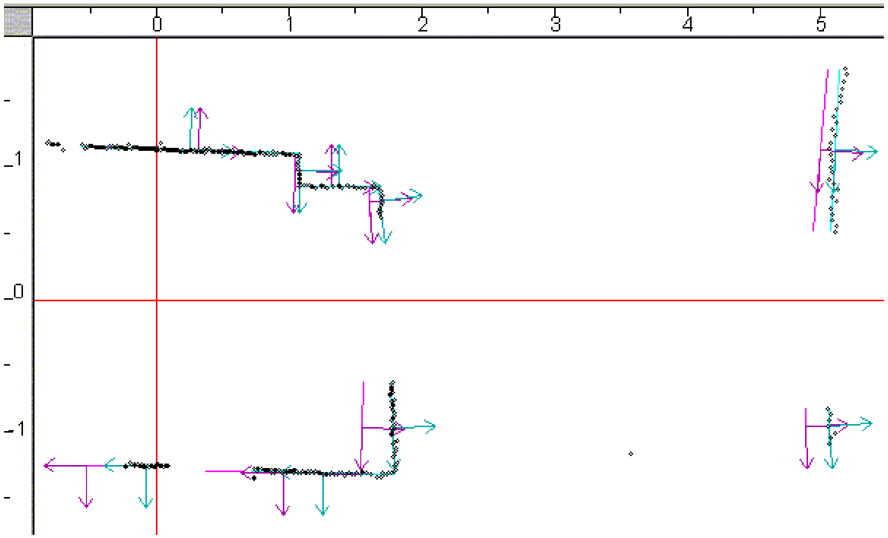


Real-time SLAM



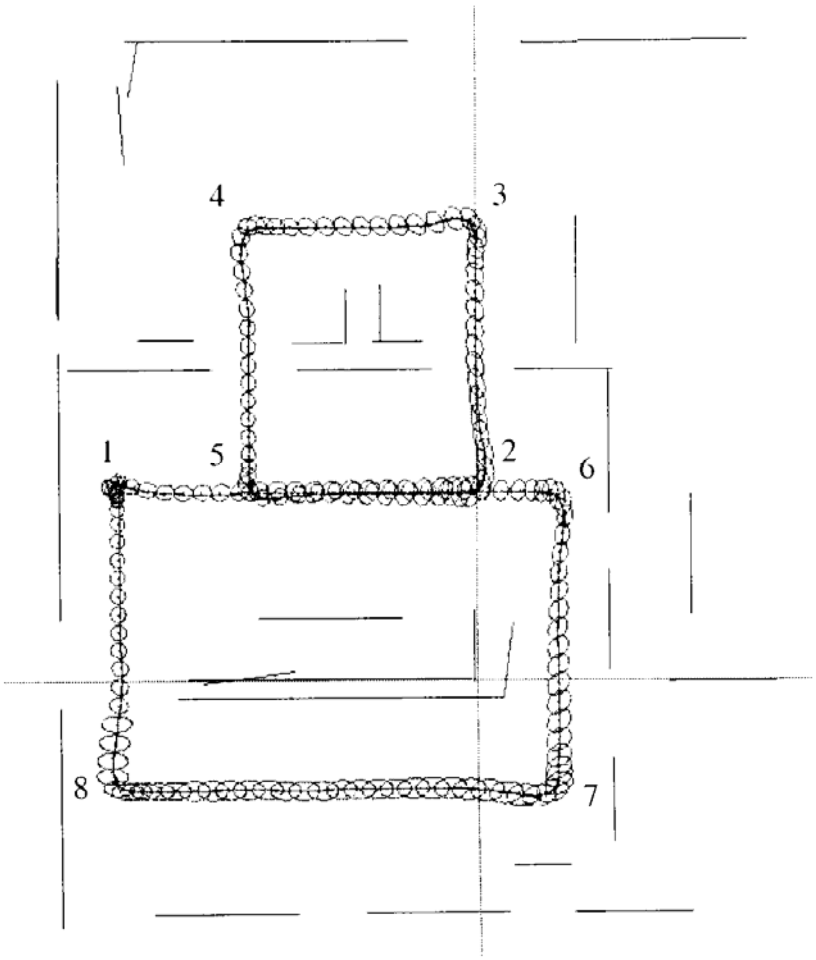
Sampling Period	1 sec
Look-ahead Time-Horizon	2 sec
Nominal Linear Velocity	0.3 m sec^{-1}
Maximum Linear Velocity	1 m sec^{-1}
Nominal Angular Velocity	$\pi/6$ rad sec^{-1}
Maximum Angular Velocity	$\pi/2$ rad sec^{-1}
Maximum Linear Acceleration	1 m sec^{-2}
Maximum Arc Segment Error	0.1 m
Laser Angular Velocity	20π rad sec^{-1}
Laser Response Time	0.145 sec

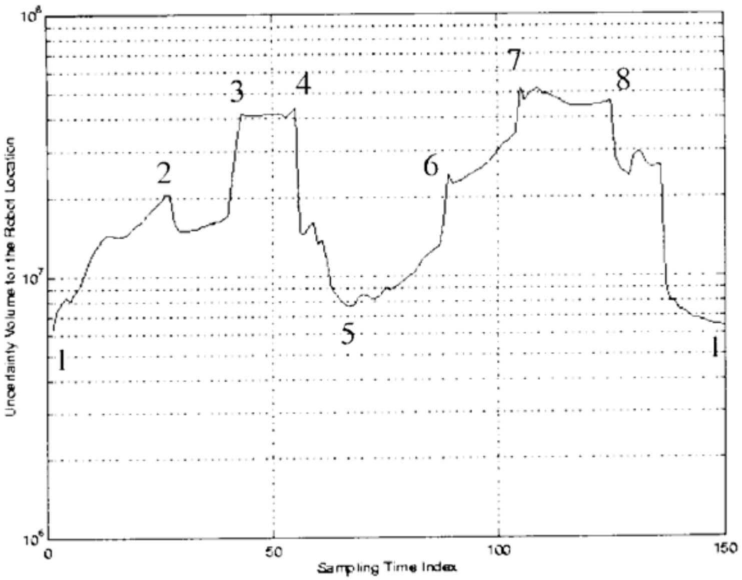




Real-time SLAM

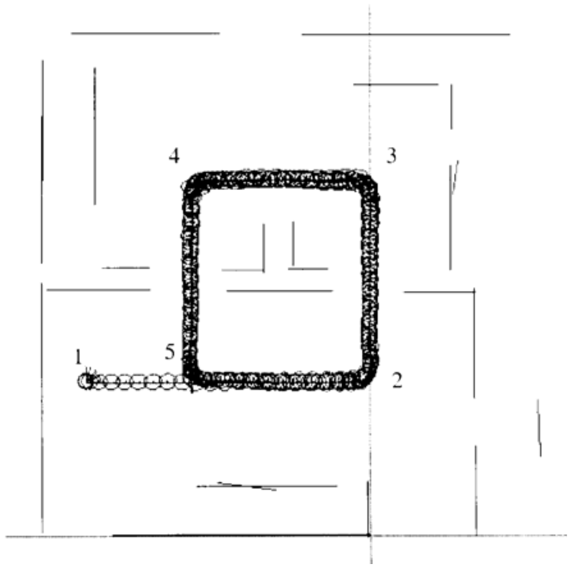
Eight-shaped trajectory

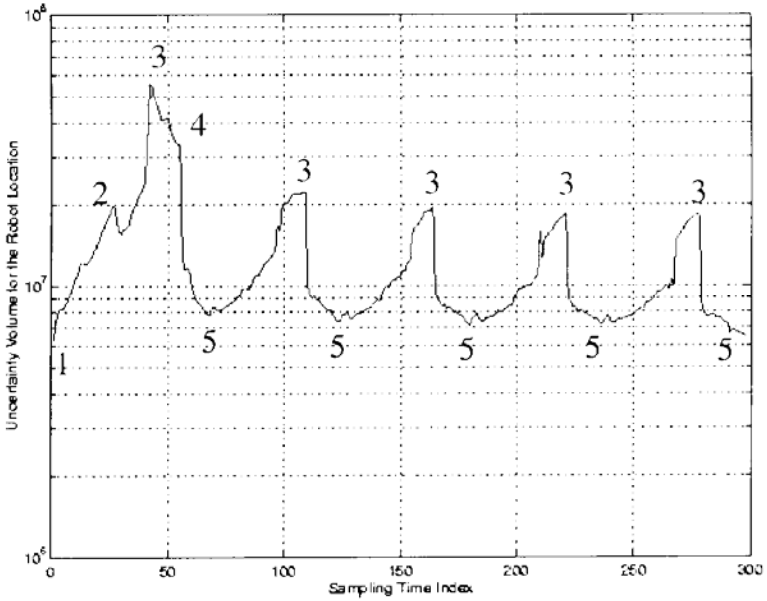


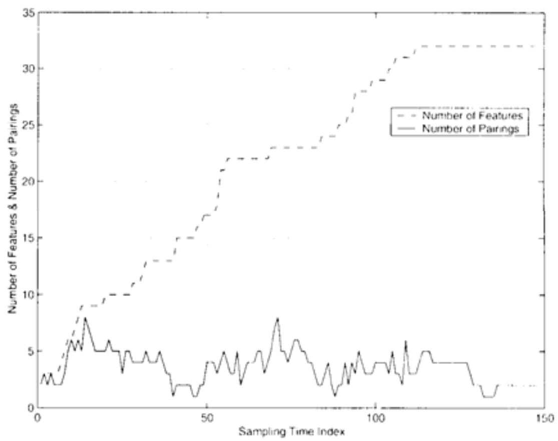


Real-time SLAM

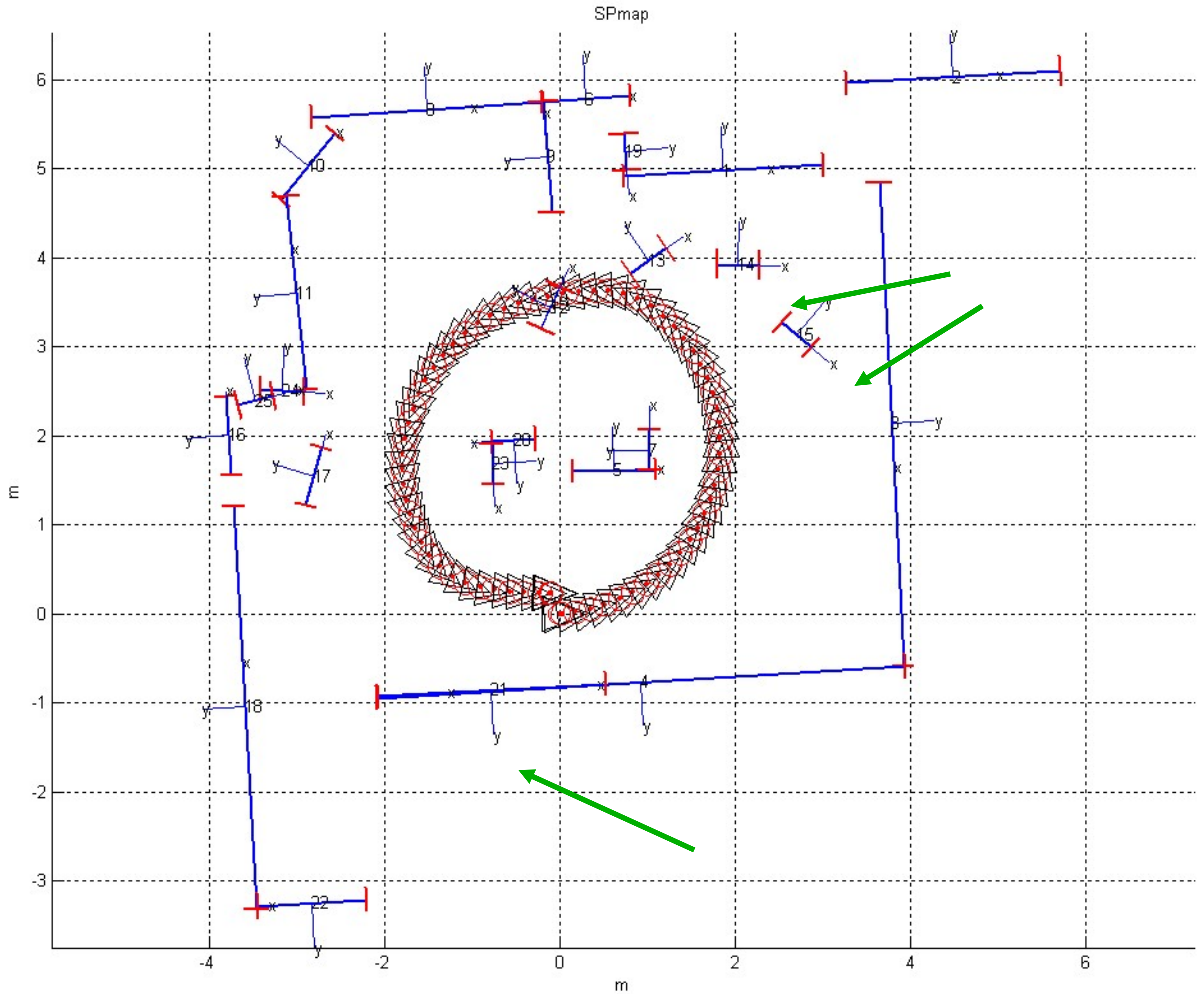
Multi-loop trajectory





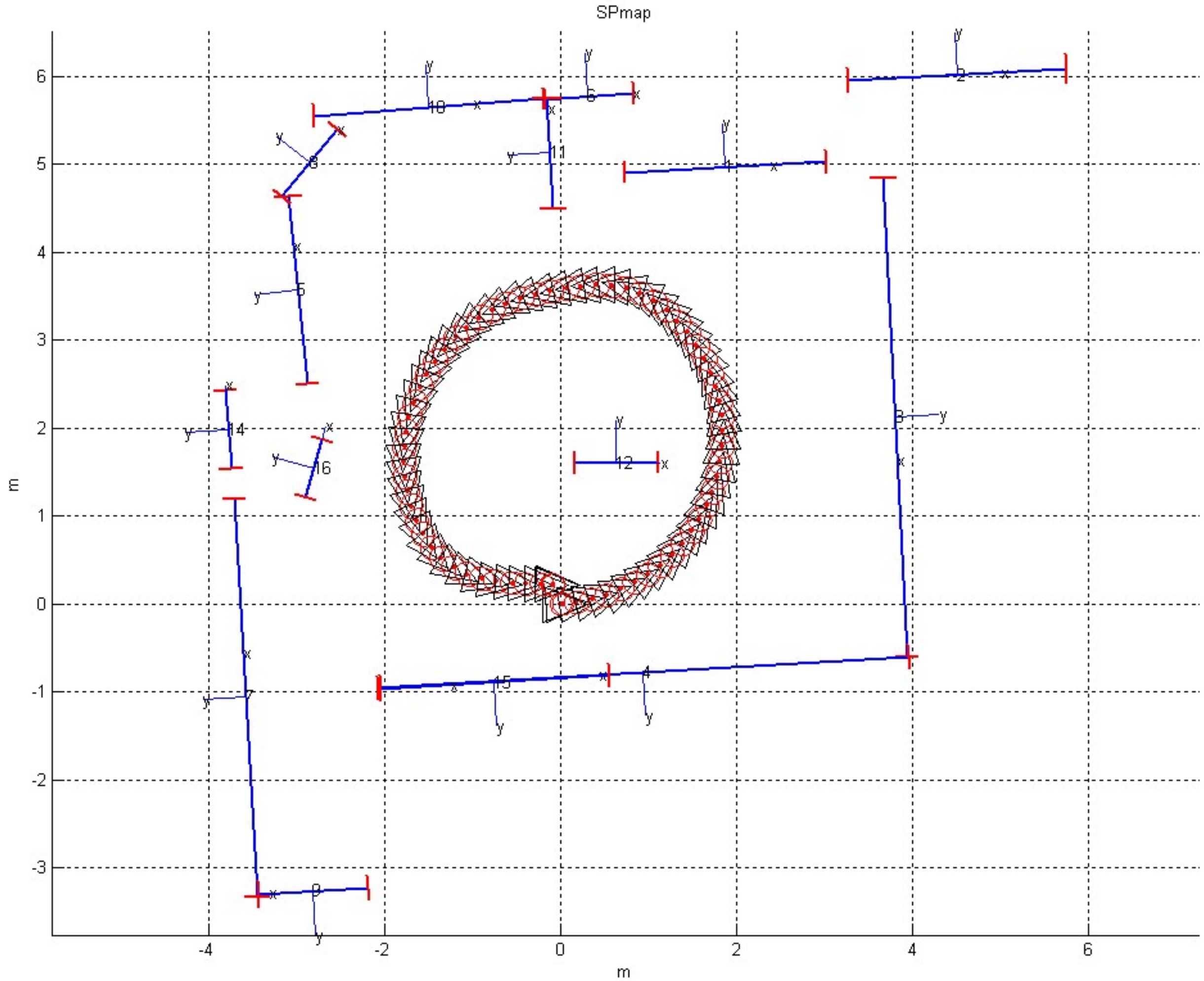


Outliers

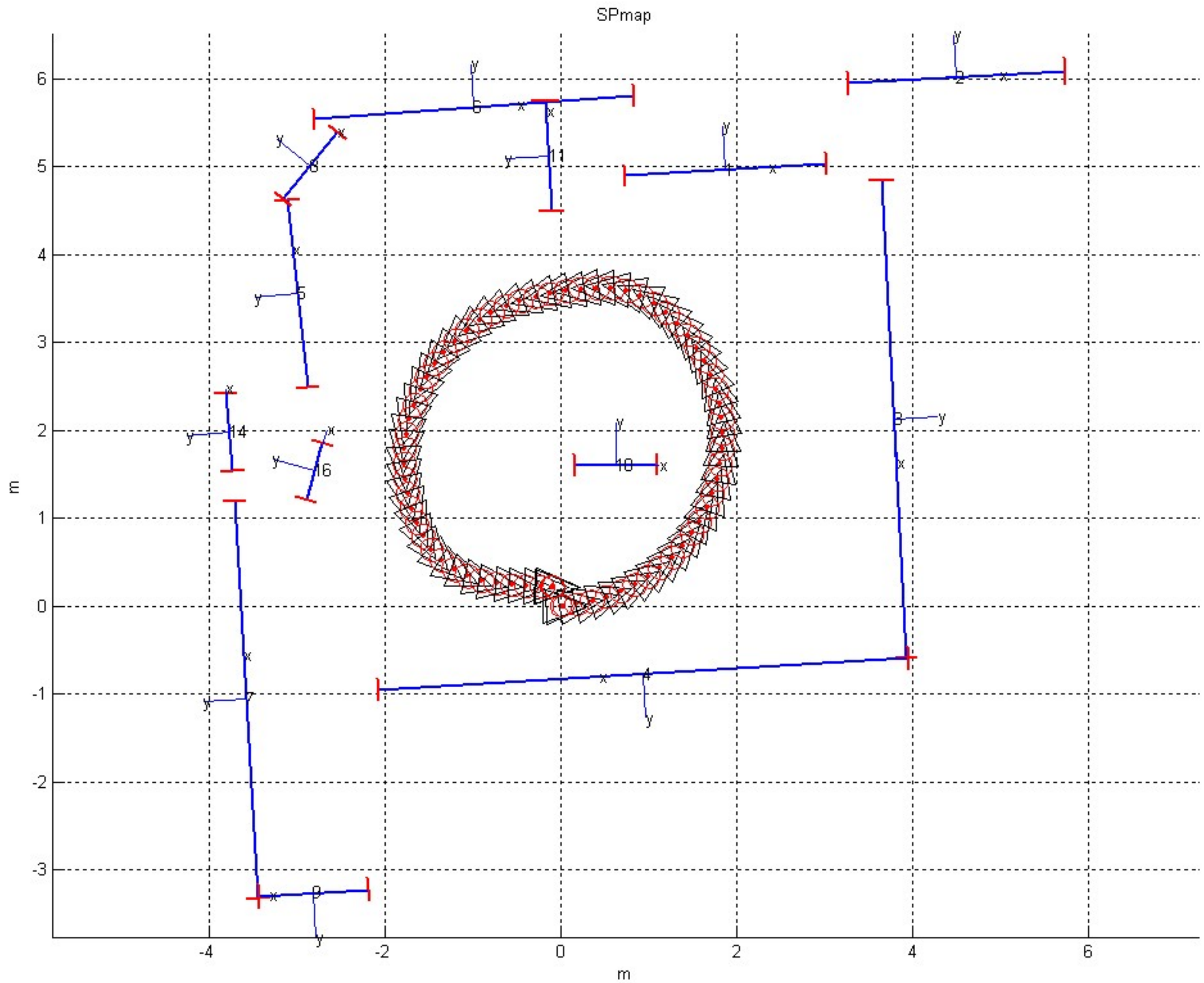


ROBOTICS
UNIVERSIDAD DE ZARAGOZA

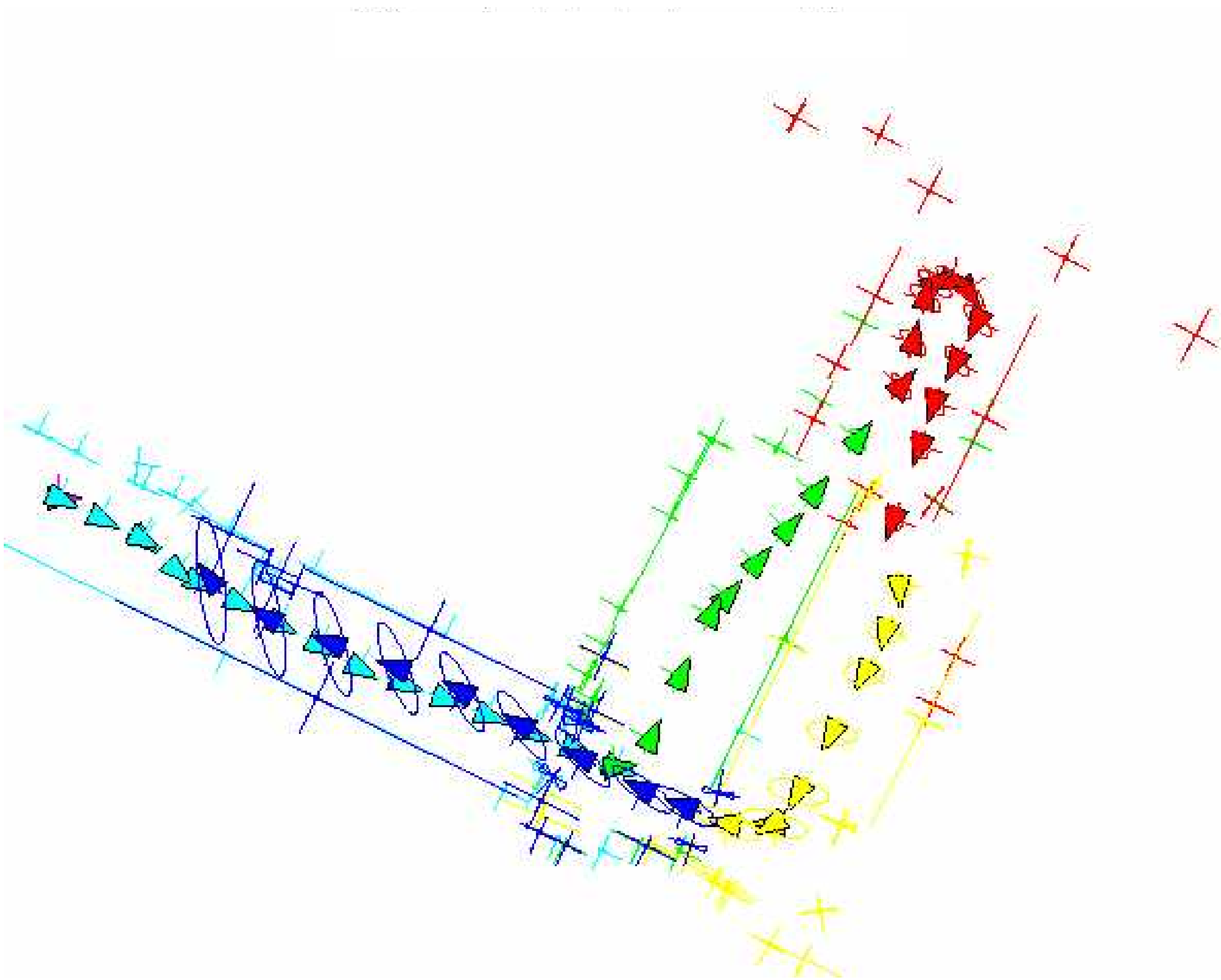
Persistent Features



Map Management



Large-Scale SLAM



EKF-SLAM properties

- Nice "convergence" properties of $\mathbf{P}_k^B$
  - 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

[Dissanayake et al. 01]

- But SLAM is a nonlinear problem
  - The inherent approximations due to linearizations can lead to divergence (inconsistency) of the EKF

[Jazwinski 70]

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

The Consistency Problem

True map value $\mathbf{x}_k^B$

EKF-SLAM estimation $\hat{\mathbf{x}}_k^B$, $\mathbf{P}_k^B$

- An estimator is consistent if:

$$E\left[\mathbf{x}_k^B - \hat{\mathbf{x}}_k^B\right] = \mathbf{0}$$

Unbiased

$$E\left[\left(\mathbf{x}_k^B - \hat{\mathbf{x}}_k^B\right)\left(\mathbf{x}_k^B - \hat{\mathbf{x}}_k^B\right)^T\right] = \mathbf{P}_k^B$$

The Mean Square Error matches the filter computed Covariance

  - Pessimistic covariance is OK (but not too pessimistic)
  - Optimistic covariance = Inconsistency = Filter divergence

Consistency Testing

1. Normalized Estimation Error Squared NEES

$$\text{NEES} = \left(\mathbf{x}_k^B - \hat{\mathbf{x}}_k^B\right)^T \left(\mathbf{P}_k^B\right)^{-1} \left(\mathbf{x}_k^B - \hat{\mathbf{x}}_k^B\right)$$

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

True map required
→ Simulations

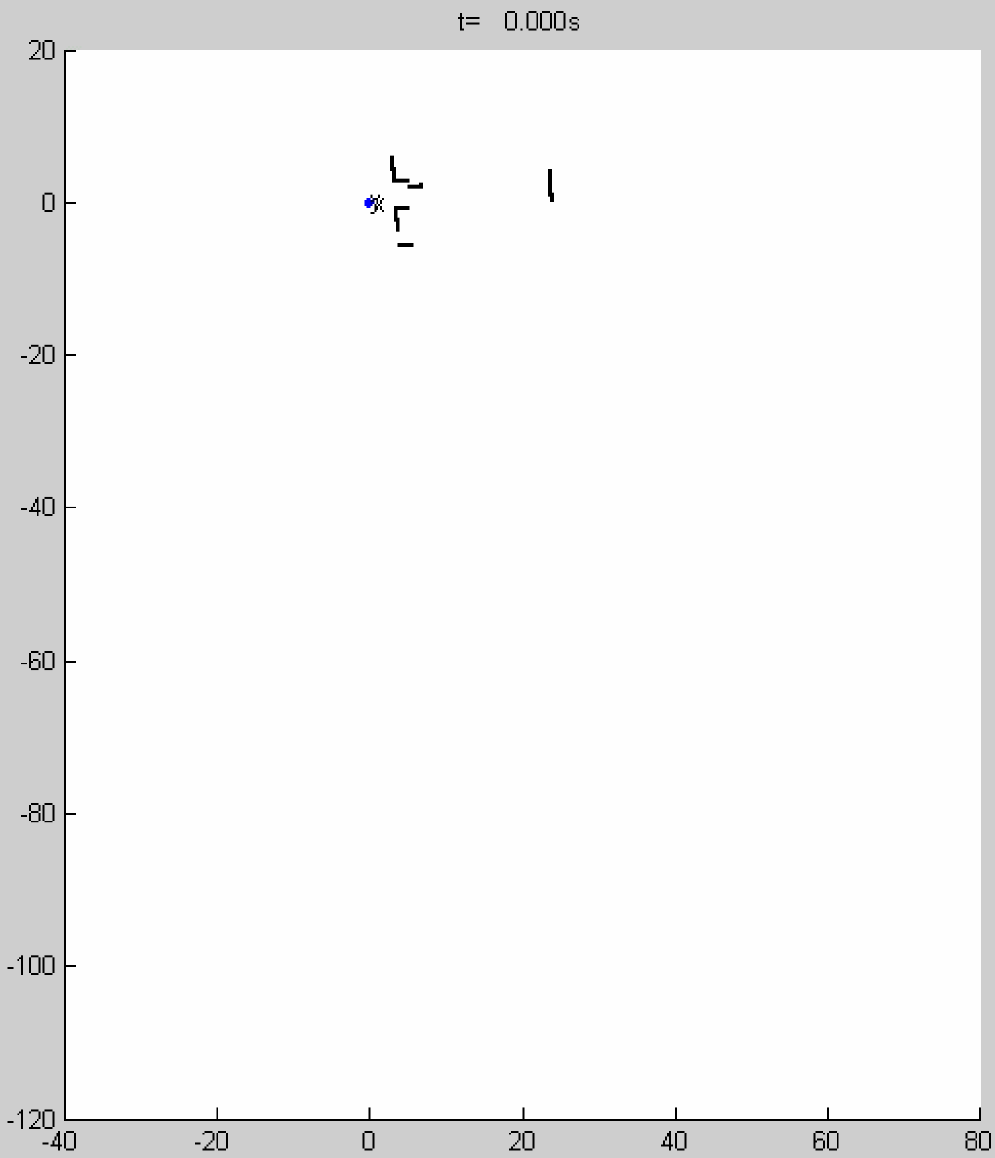
2. Innovation test (observation *i* → map feature *j*)

$$\text{NIS} = \left(\mathbf{z}_i - \mathbf{h}_j(\hat{\mathbf{x}}_k^B)\right)^T \left(\mathbf{H}_j \mathbf{P}_k^B \mathbf{H}_j^T + \mathbf{R}_i\right)^{-1} \left(\mathbf{z}_i - \mathbf{h}_j(\hat{\mathbf{x}}_k^B)\right)$$

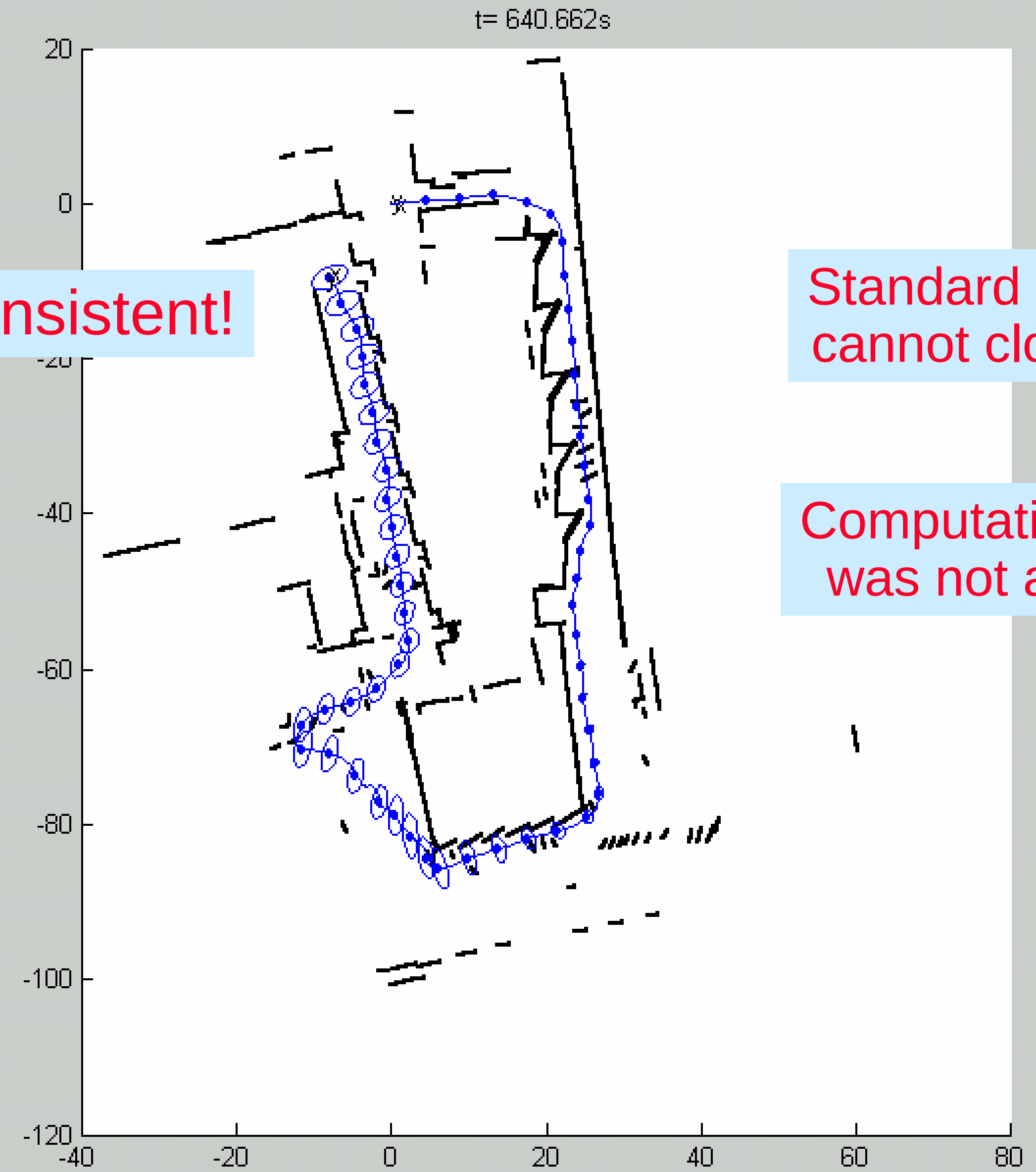
$$\text{NIS} \leq \chi^2_{d,1-\alpha}$$

Critical when
closing big loops

EKF-SLAM: Real Example



EKF-SLAM: Real Example



Inconsistent!

Standard data association cannot close a 250m loop

Computational complexity was not a problem here!

EKF-SLAM: Analytical Linearization

- Nonlinear model:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \quad ; \quad \mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P}_\mathbf{x})$$

- Linearization (1st order Taylor):

$$\mathbf{y} \simeq \mathbf{f}(\hat{\mathbf{x}}) + \left.\frac{\partial \mathbf{f}}{\partial t}\right|_{\mathbf{x}=\hat{\mathbf{x}}} (\mathbf{x} - \hat{\mathbf{x}})$$

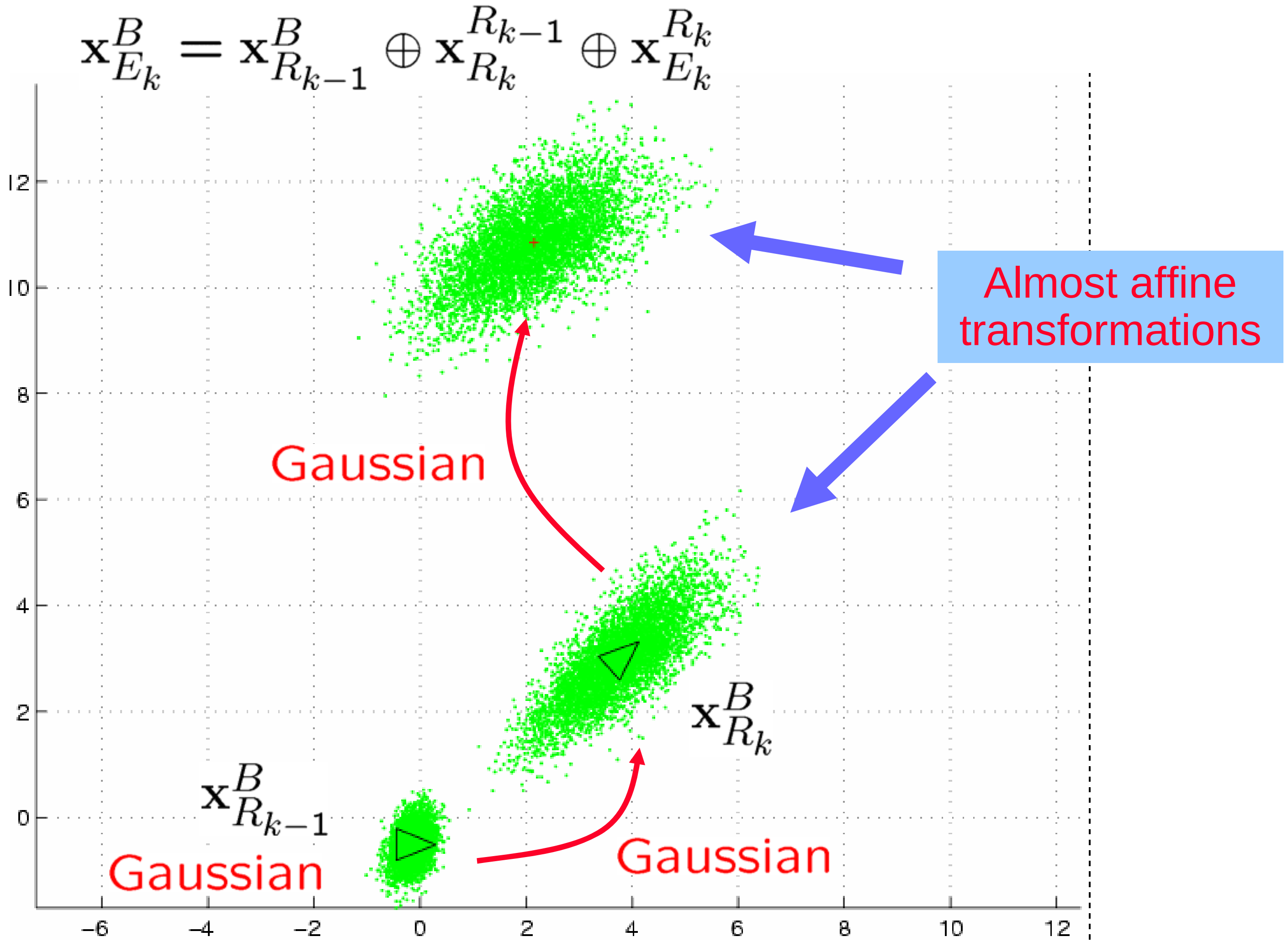
- Transformed moments:

$$\hat{\mathbf{y}} \simeq \mathbf{f}(\hat{\mathbf{x}}) \quad ; \quad \mathbf{P}_\mathbf{y} \simeq \left(\left.\frac{\partial \mathbf{f}}{\partial t}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \right) \mathbf{P}_\mathbf{x} \left(\left.\frac{\partial \mathbf{f}}{\partial t}\right|_{\mathbf{x}=\hat{\mathbf{x}}} \right)^T$$

- Assumptions:
  - Smoothness (i.e. discontinuities not handled)
  - Deterministic state vector (i.e. randomness of x not considered)

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

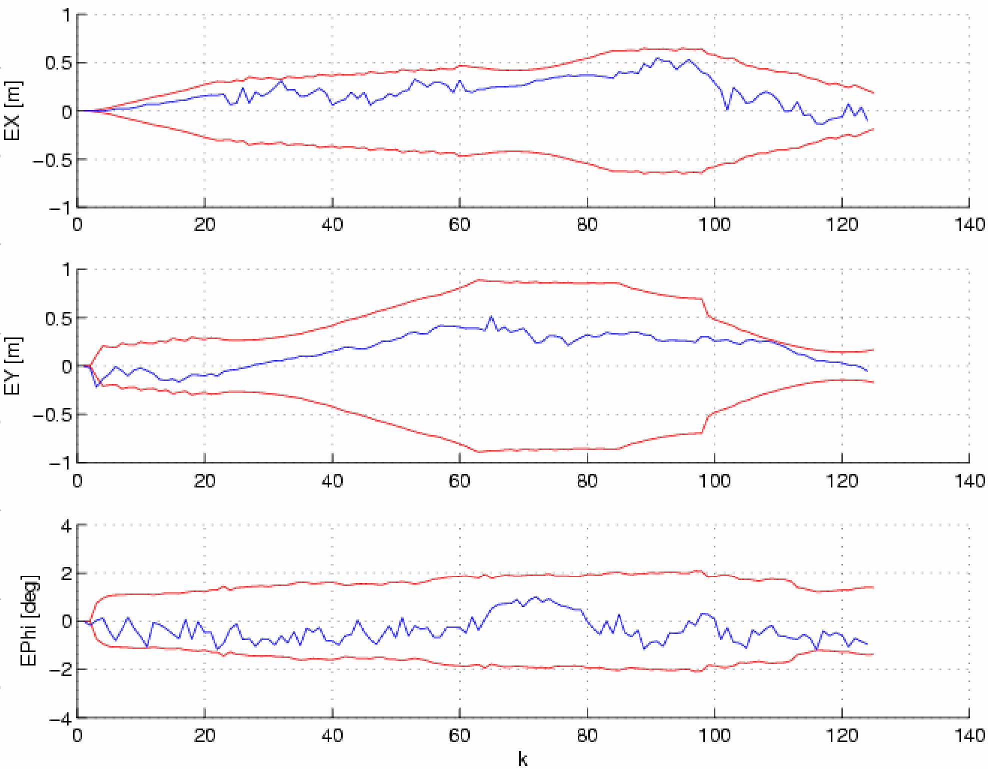
EKF-SLAM: Low observation noise



EKF-SLAM: Low observation noise

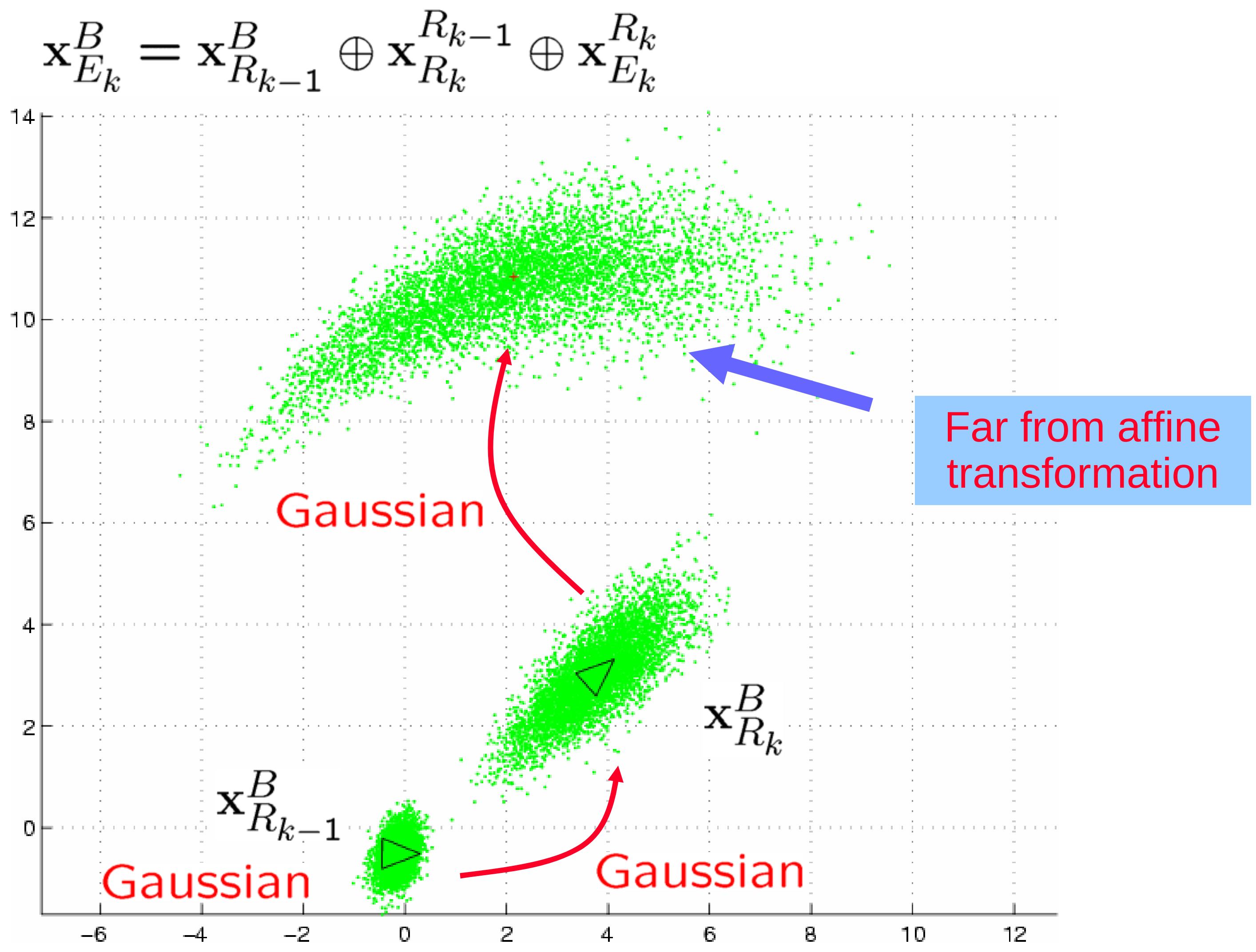
- Simulated 120-m loop trajectory, point features, known data association, true gaussian noise

$1 - \text{NEES}/\chi^2_{r,1-\alpha}$



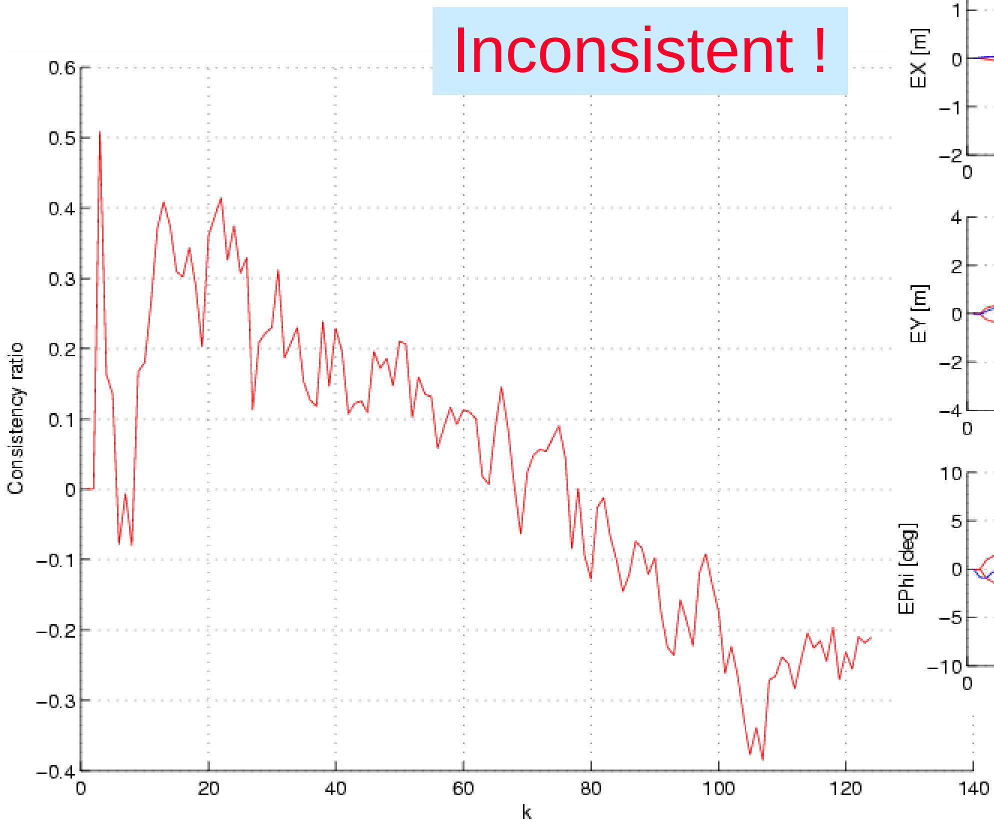
Errors $\pm\ 2\sigma$ bounds

EKF-SLAM: High observation noise

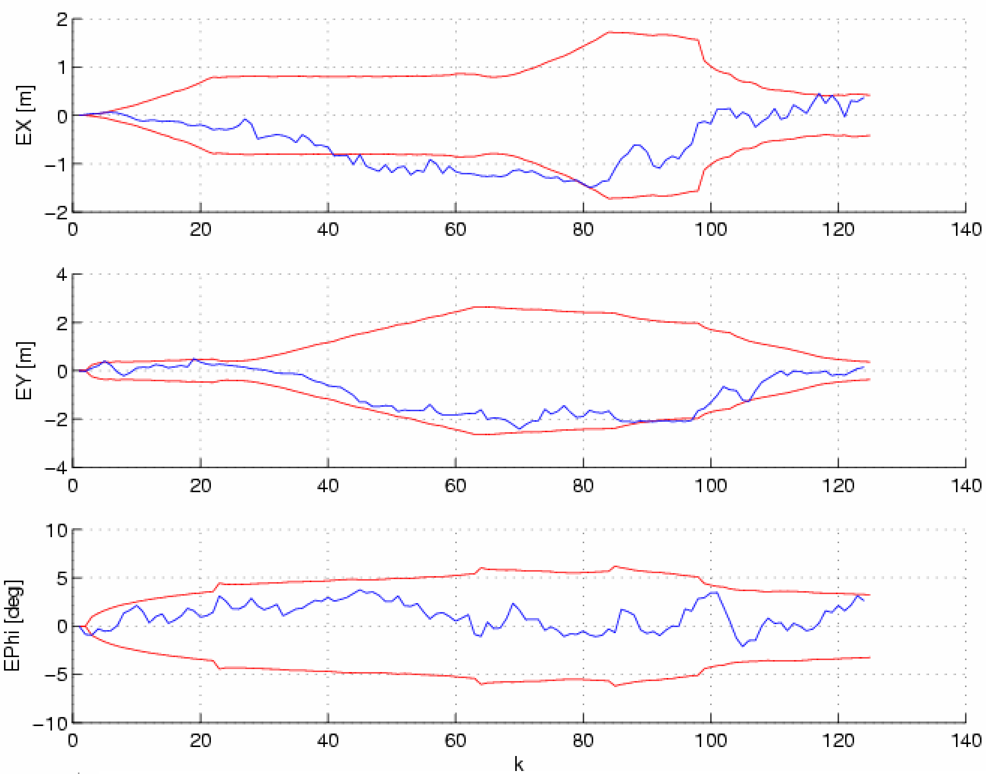


EKF-SLAM: High observation noise

- Simulated 120-m loop trajectory, …



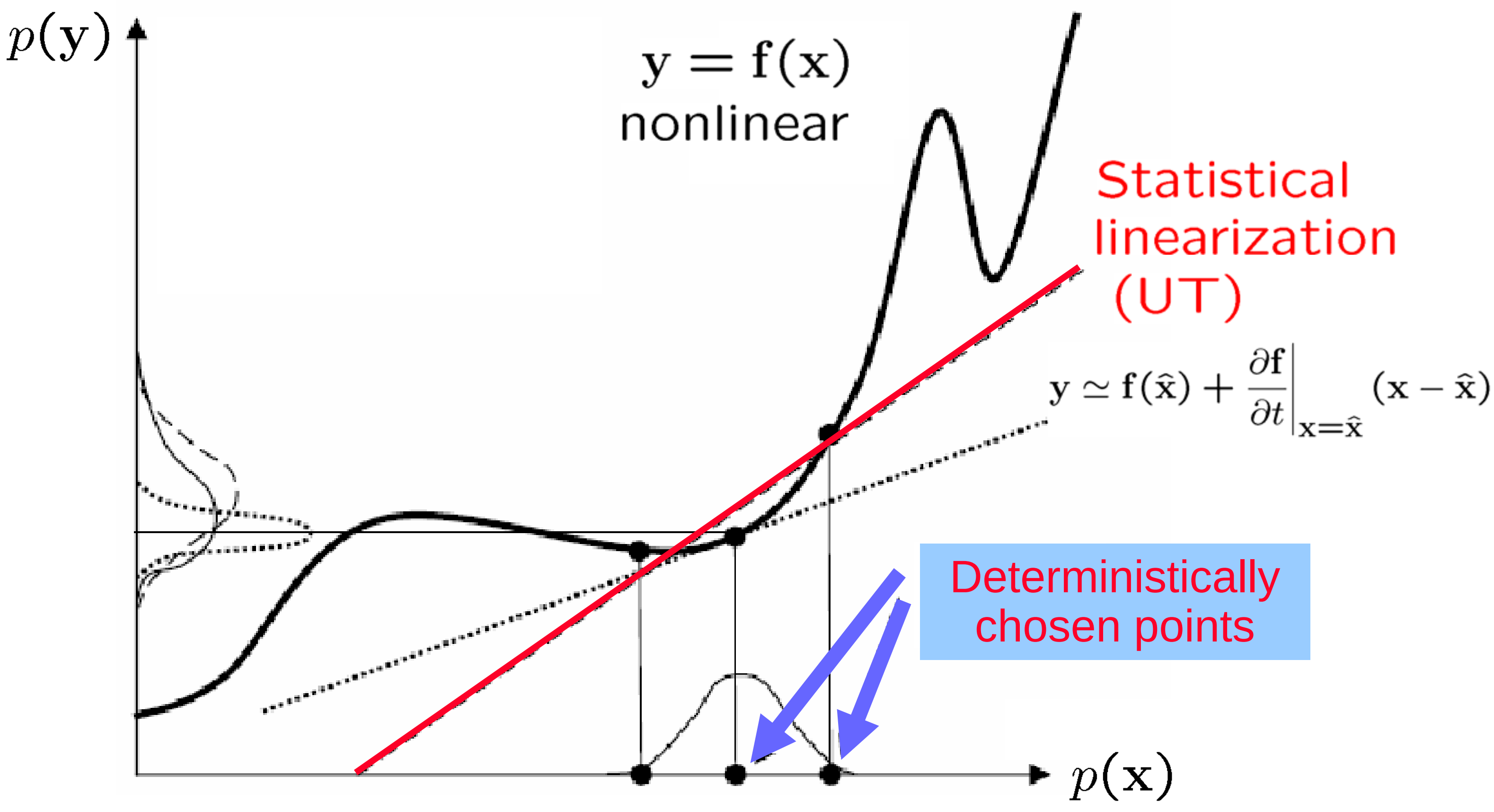
$1 - \mathrm{NEES}/\chi^2_{r,1-\alpha}$



Errors $\pm\ 2\sigma$ bounds

Unscented SLAM: Idea

- Same framework as EKF with improved linearization
- Takes randomness of the state into account



Adapted from [van der Merwe 04]

Unscented SLAM

- Nonlinear model:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \;\; ; \quad \text{Known pdf of } \mathbf{x} \;\; ; \quad \text{e.g. } \mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{P}_\mathbf{x})$$

- Transformation of moments:

$$\text{pdf}(\mathbf{x}) \rightarrow \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\} \;,\; \mathbf{x}^{(j)} \equiv \text{sigma-point}$$

$$\hat{\mathbf{y}} \simeq \sum_{j=1}^{N} \omega_m^{(j)} \mathbf{y}^{(j)}, \quad \mathbf{y}^{(j)} = \mathbf{f}(\mathbf{x}^{(j)})$$

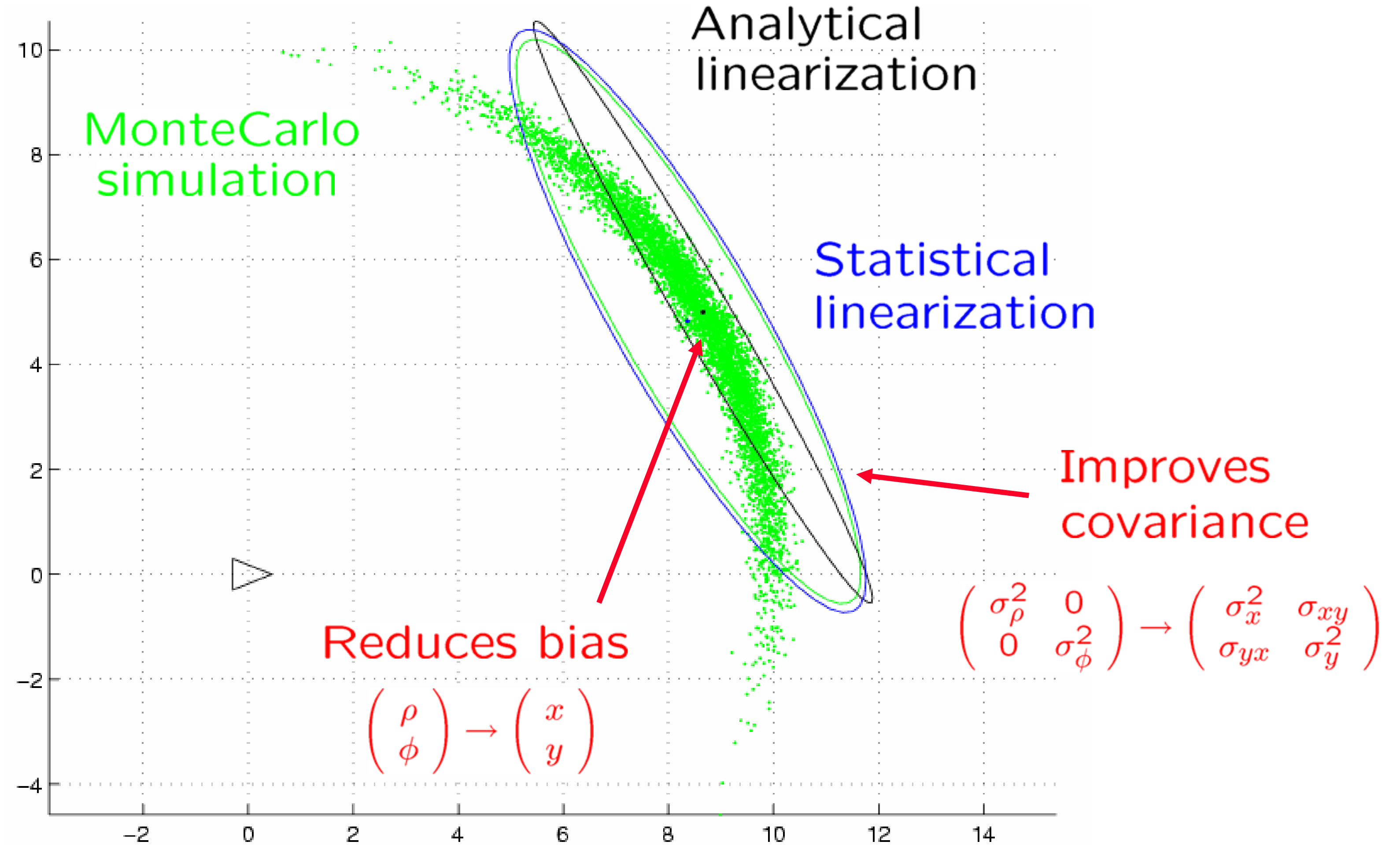
$$\mathbf{P}_\mathbf{y} \simeq \sum_{j=1}^{N} \omega_c^{(j)} (\mathbf{y}^{(j)} - \hat{\mathbf{y}})(\mathbf{y}^{(j)} - \hat{\mathbf{y}})^T$$

- Accuracy depends on:
  - Number of sigma-points, e.g. 2n+1 with n=dim(x)
  - Regression weights

[Julier et al. 97]

Unscented SLAM: p2c

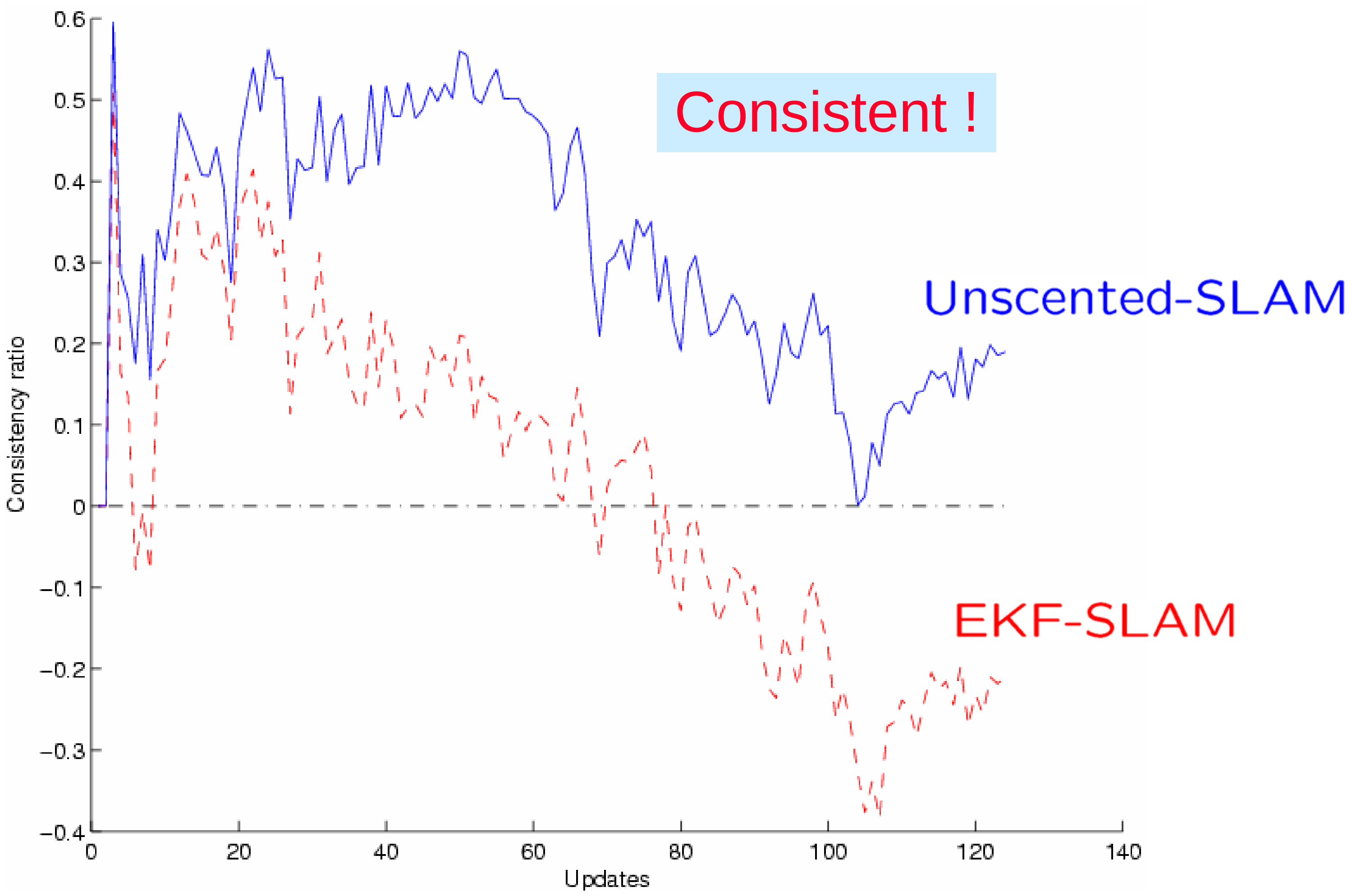
- Polar to Cartesian



Adapted from [Julier et al. 97]

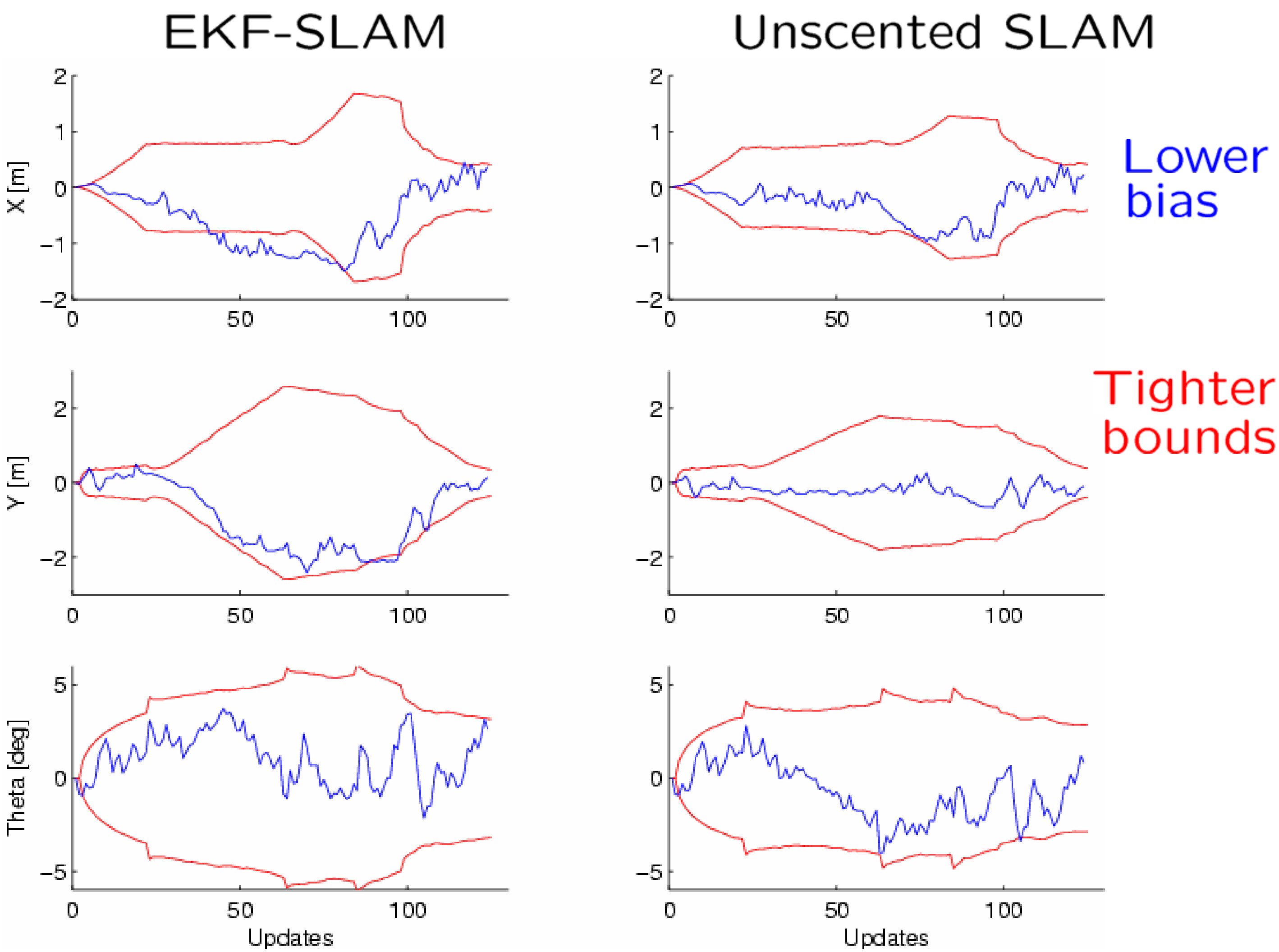
Unscented SLAM: Example (1)

- Simulated 120-m loop trajectory, …



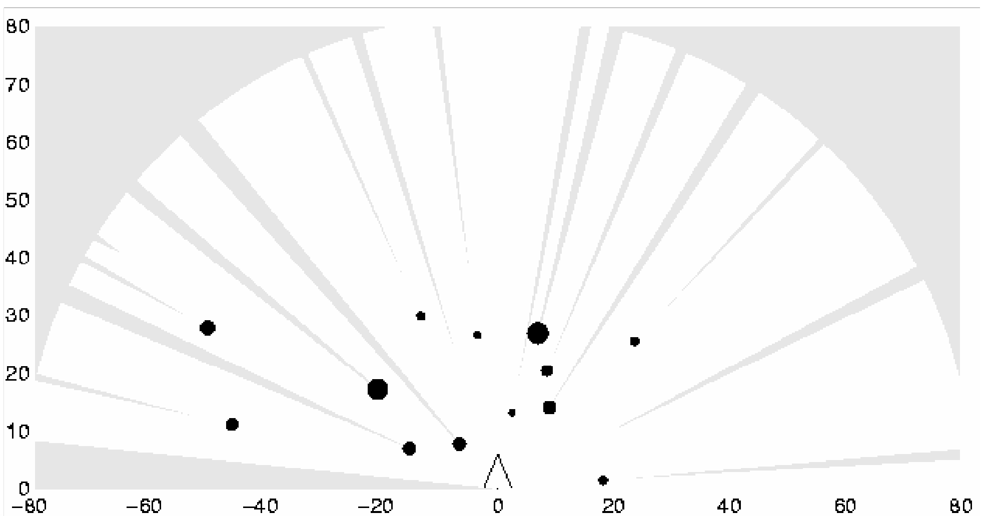
Unscented SLAM: Example (1)

- Simulated 120-m loop trajectory



Unscented SLAM: Example (2)

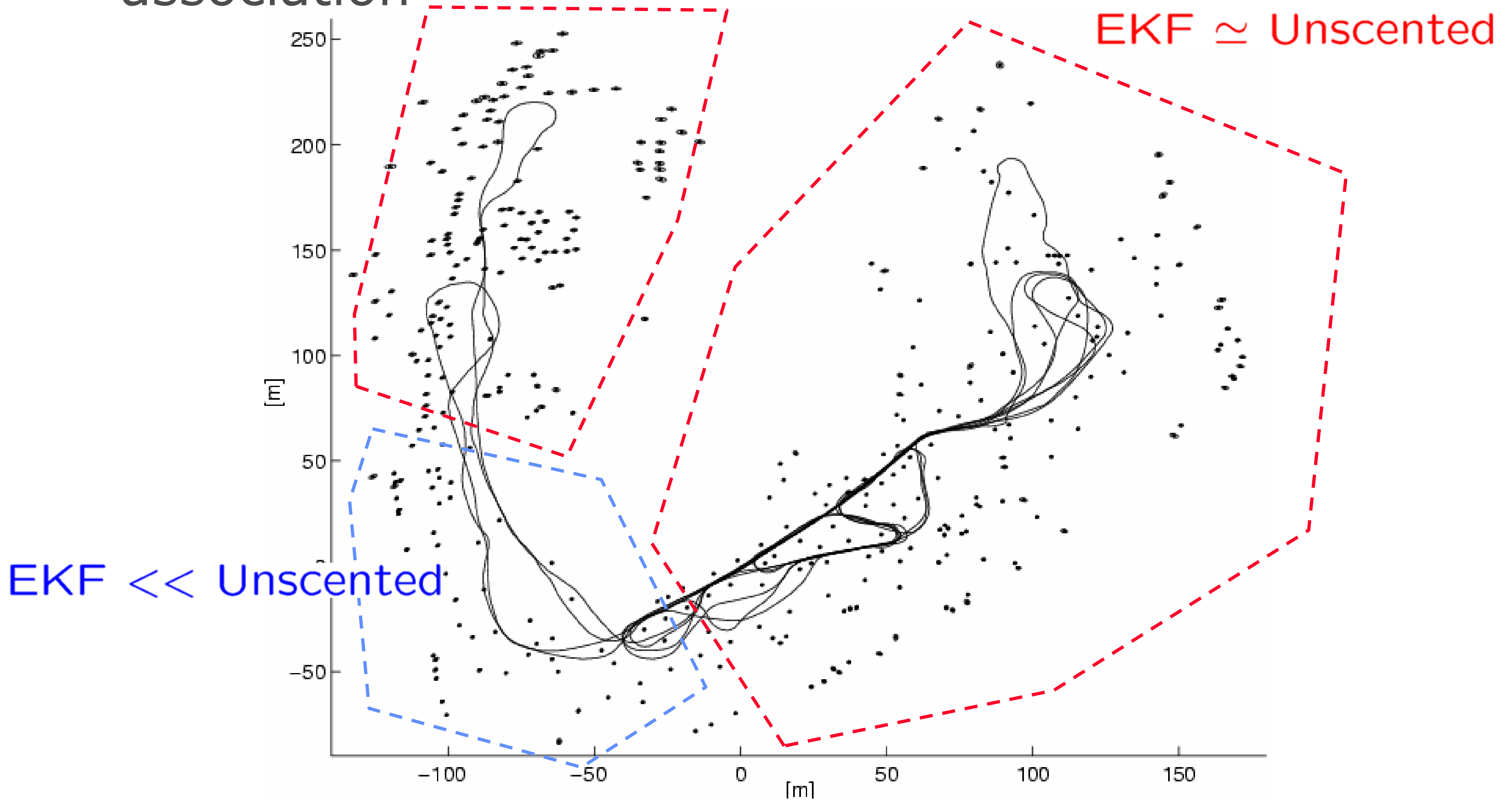
- Vehicle with SICK Laser



- Victoria Park, Sydney (Guivant and Nebot)

Unscented SLAM: Example (2)

- Feasible / Not real-time / Simplifies data association



Conclusions EKF vs UKF

- EKF-SLAM is consistent for:
  - The linear case (e.g. 1D robot)
  - Nonlinear + Low uncertainty

- Unscented SLAM:
  - Improves consistency for nonlinear + large uncertainty
  - Reduces bias
  - Provides more precise uncertainty bounds
  - But, further research is required

Asociación de datos para SLAM continuo, cerrado de bucles y relocalización

José Neira
Universidad de Zaragoza

jneira@unizar.es
http://www.cps.unizar.es/~jneira/

Outline

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

2. **Feature extraction**

3. **Data association in continuous SLAM**

4. **The loop closing problem**

5. **The global localization problem**

6. **Appendix**

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

1. Introduction

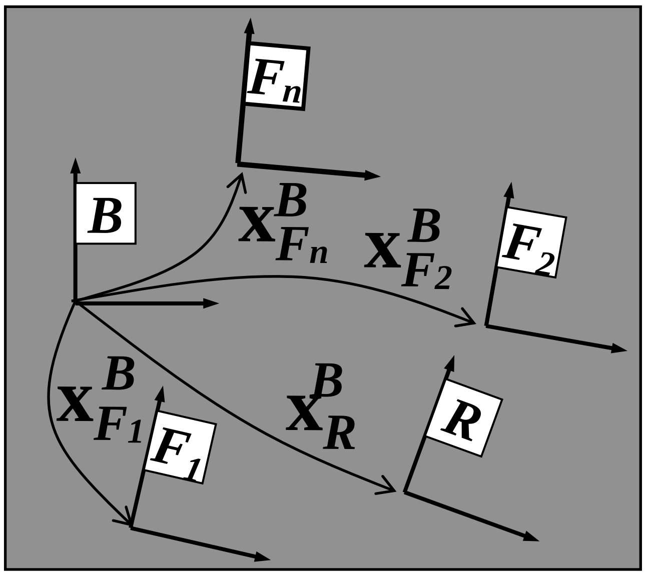
The basic EKF-SLAM algorithm

- Environment information related to a set of elements:

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

- represented by a **stochastic map**:

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



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

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

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

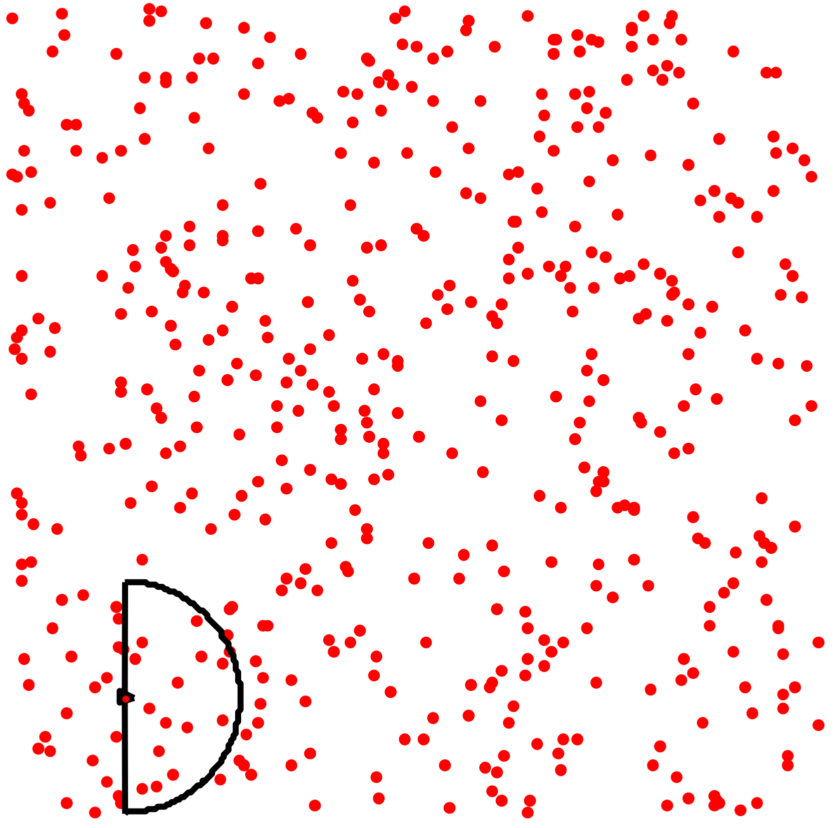
end for

For details on the rest of the algorithm, see appendix

ROBOTICS UNIVERSIDAD DE ZARAGOZA

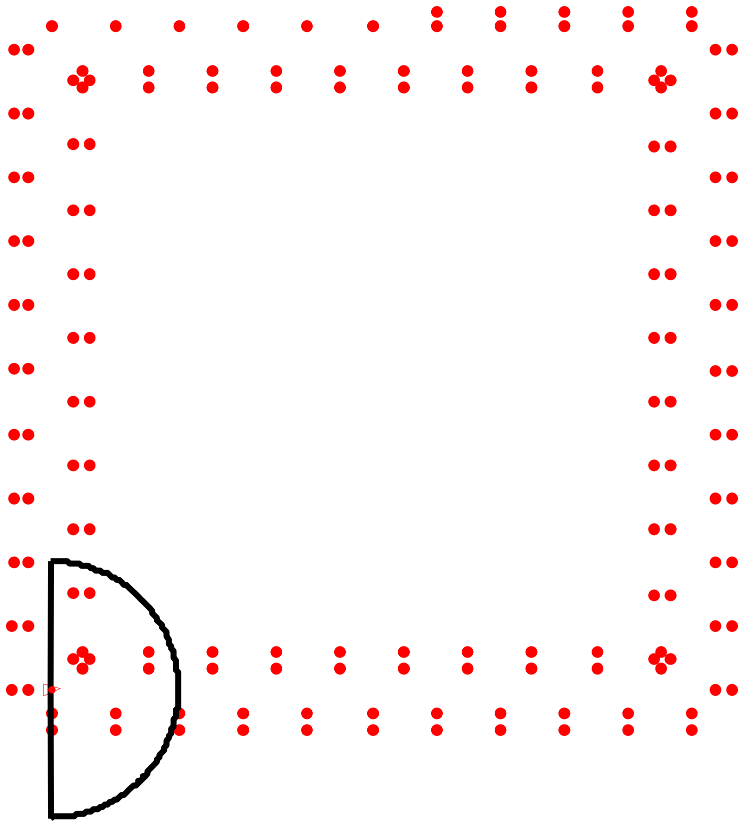
Without loss of generality...

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



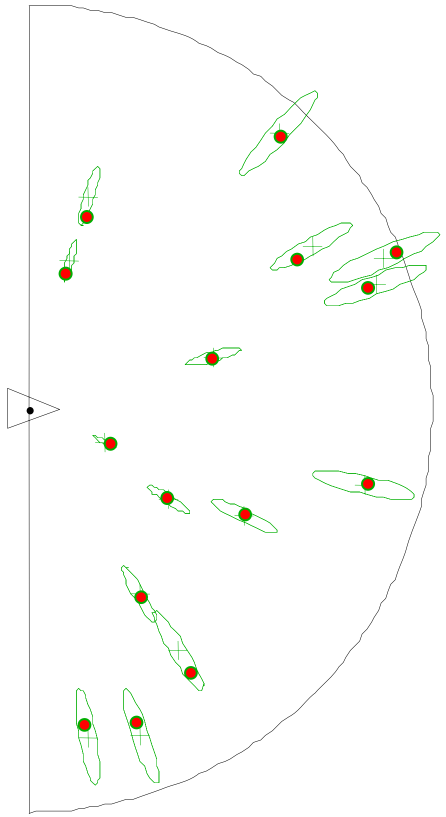
Without loss of generality...

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



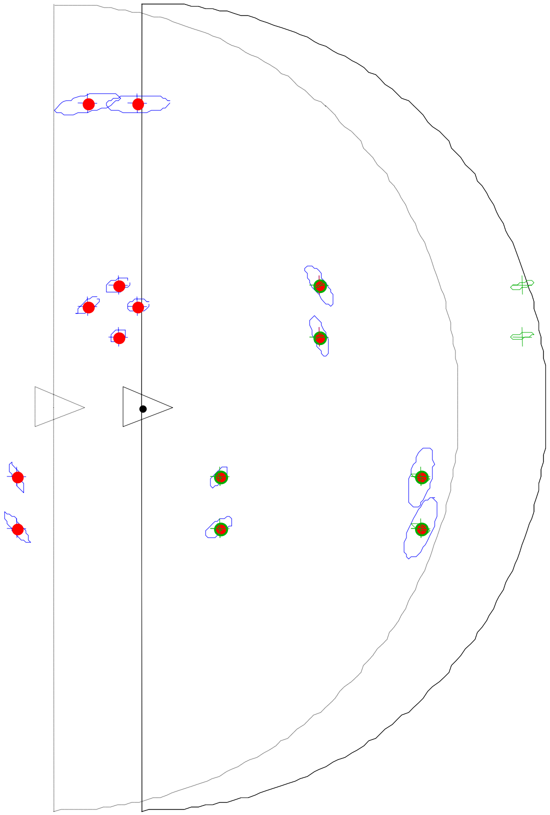
Without loss of generality...

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

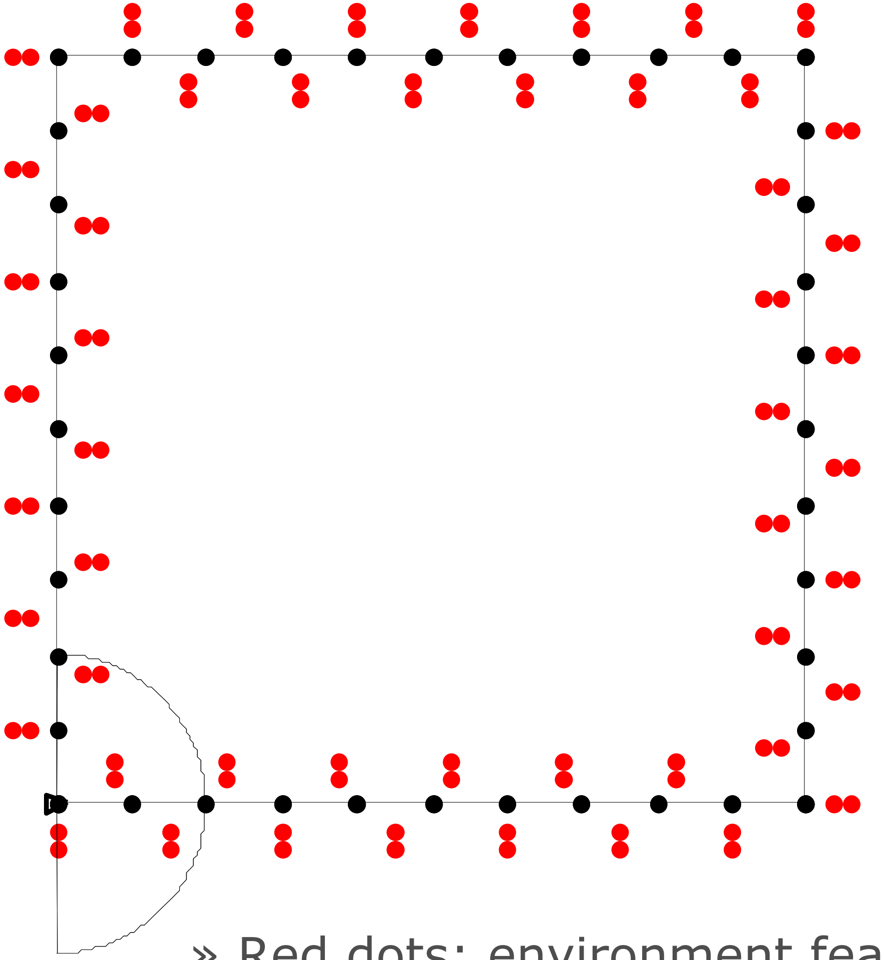


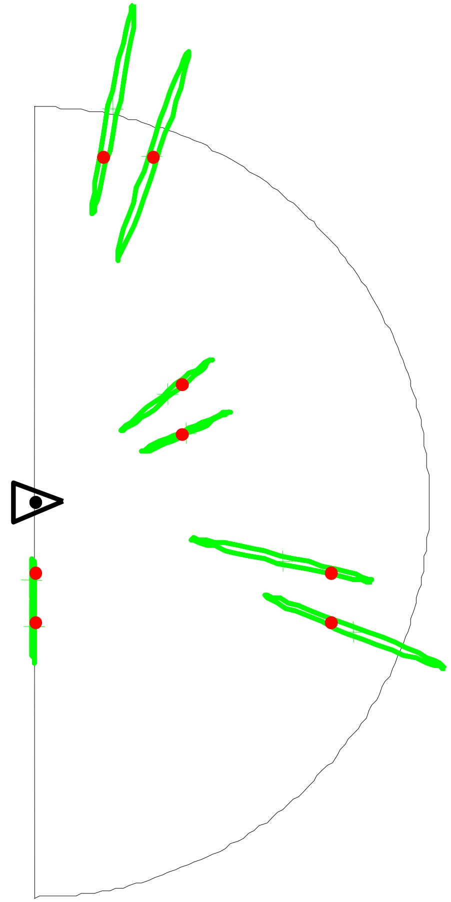
Without loss of generality...

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



Example: SLAM in a cloister





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

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

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}

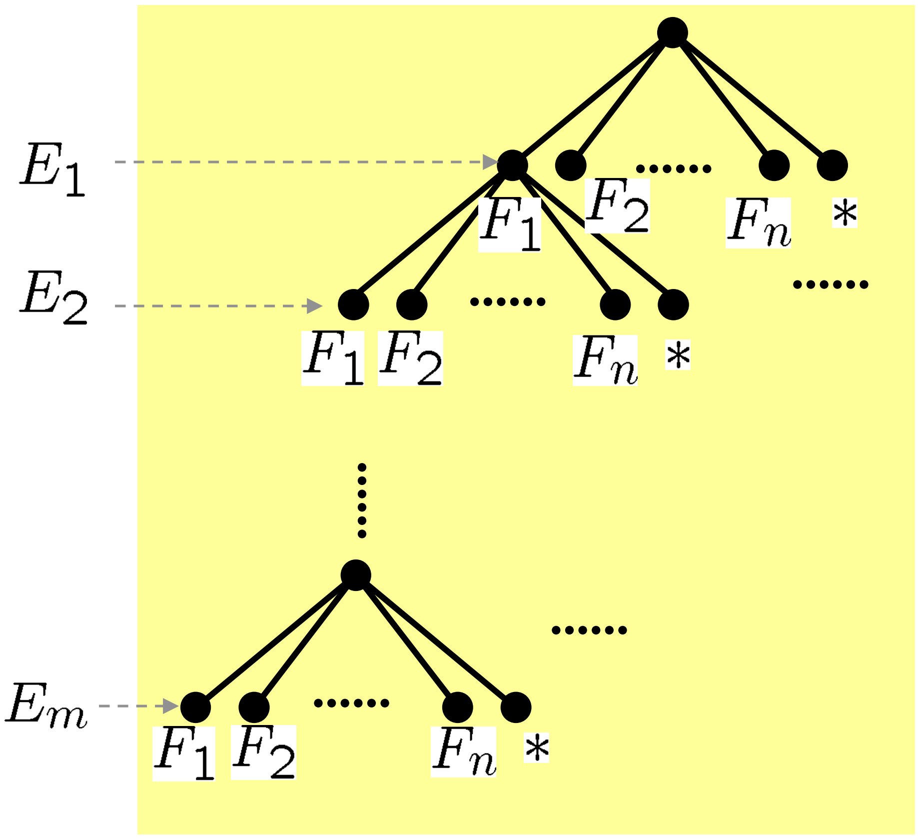
$$\mathcal{H}_m = [j_1 \ldots j_i \ldots j_m]$$

$$E_i \swarrow \; \searrow F_{j_i}$$

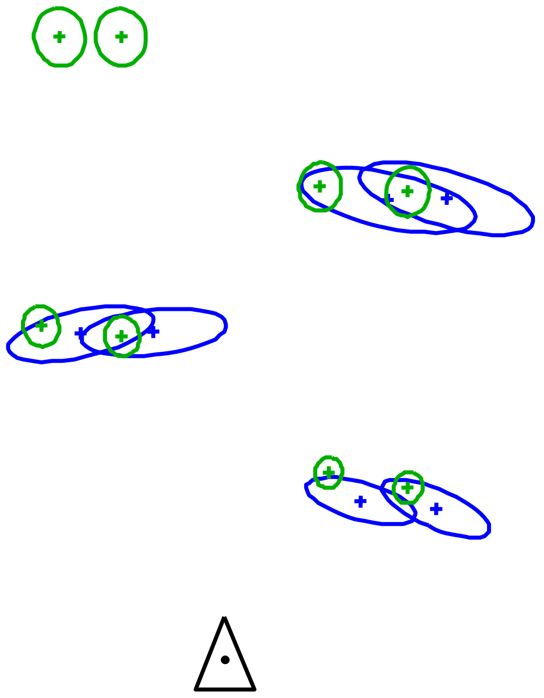
- Non matched observations: $j_i = 0$

The Correspondence Space

Interpretation tree
(Grimson et al. 87):



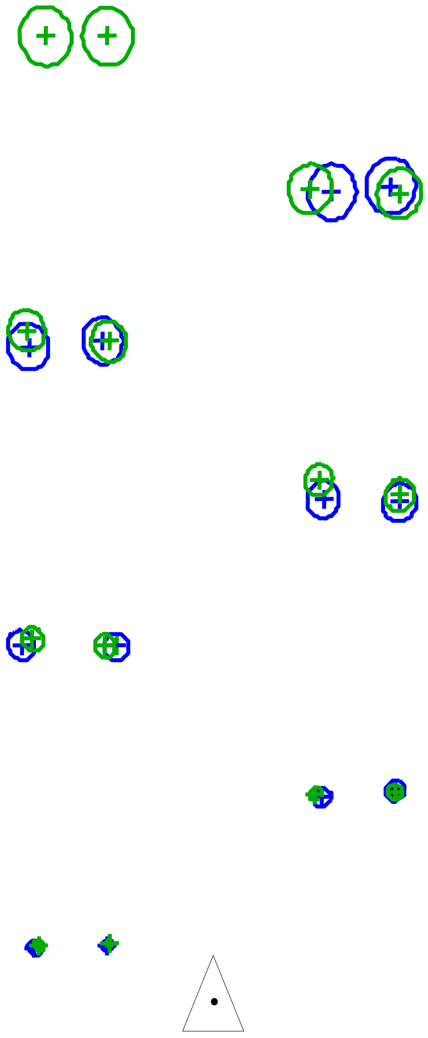
$(n+1)^m$ possible hypotheses



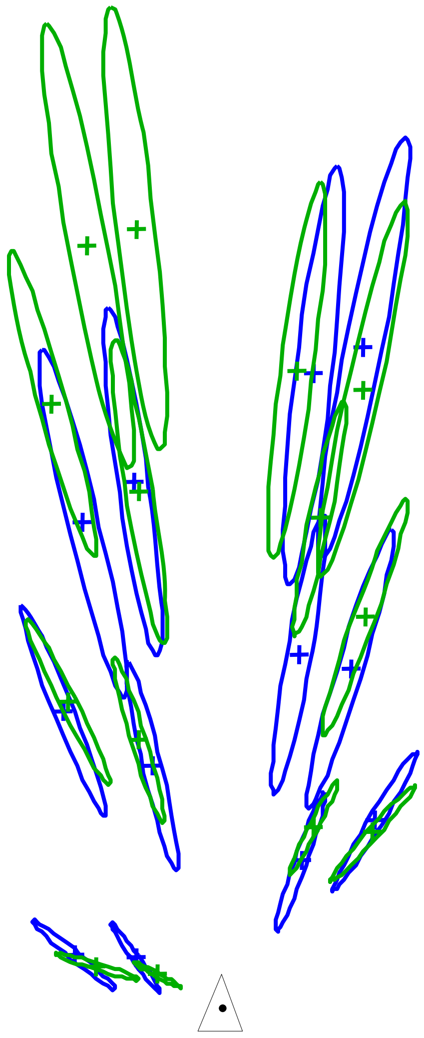
Green points: measurements
Blue Points: predicted features

Why data association is difficult

- Low sensor error

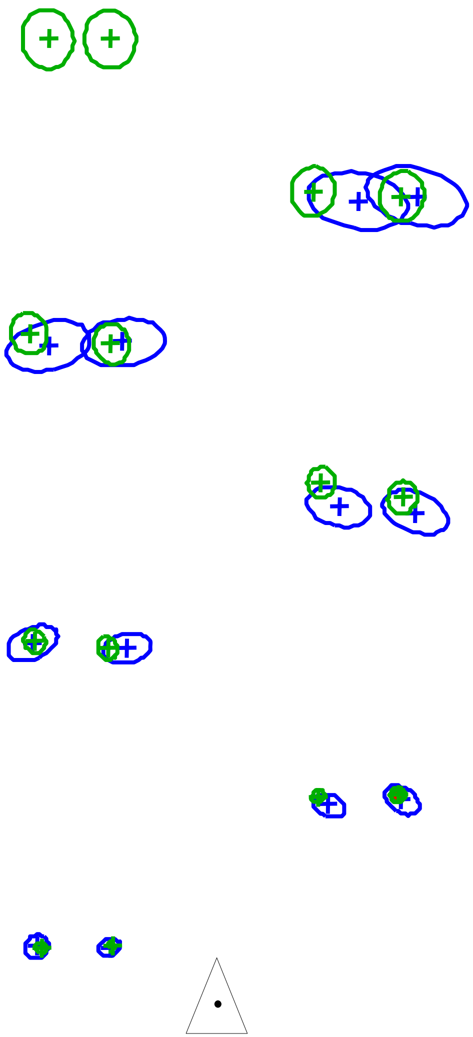


- High sensor error

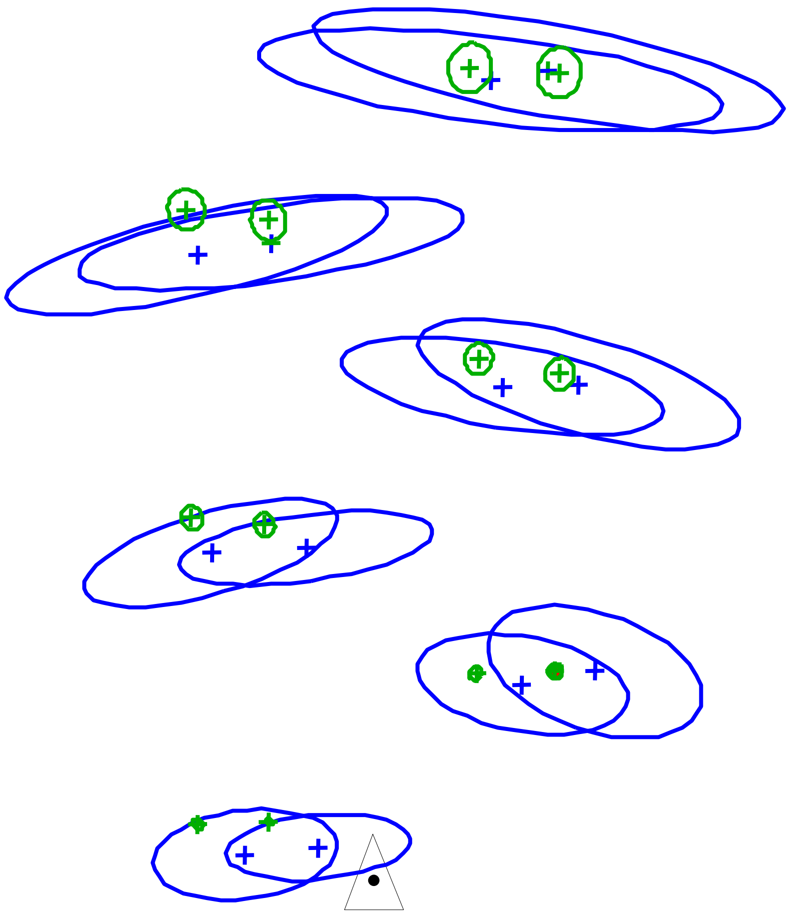


Why data association is difficult

- Low odometry error



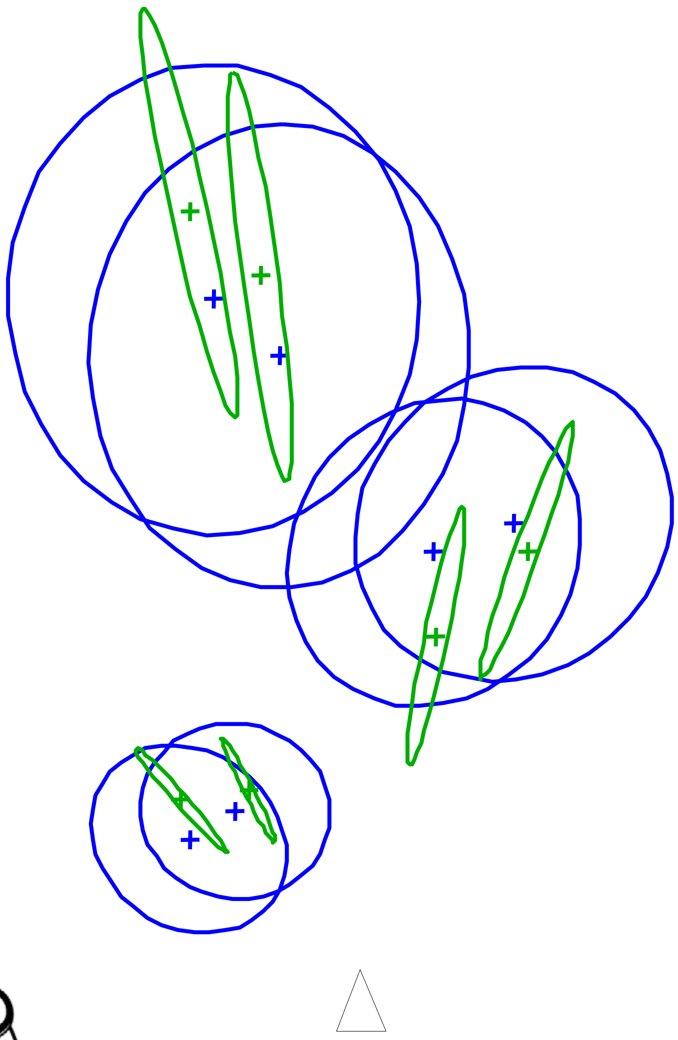
- High odometry error

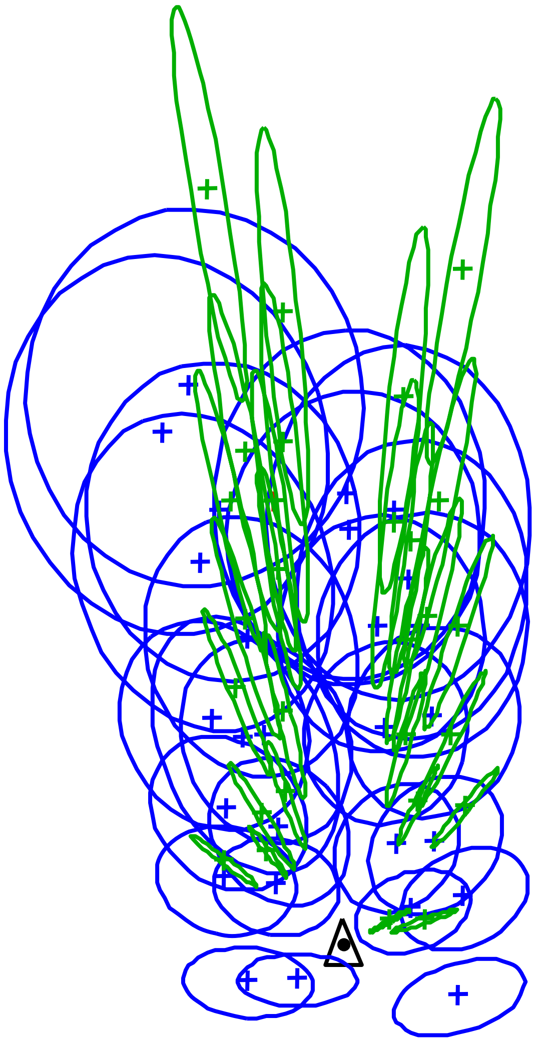


ROBOTICS
UNIVERSIDAD DE ZARAGOZA

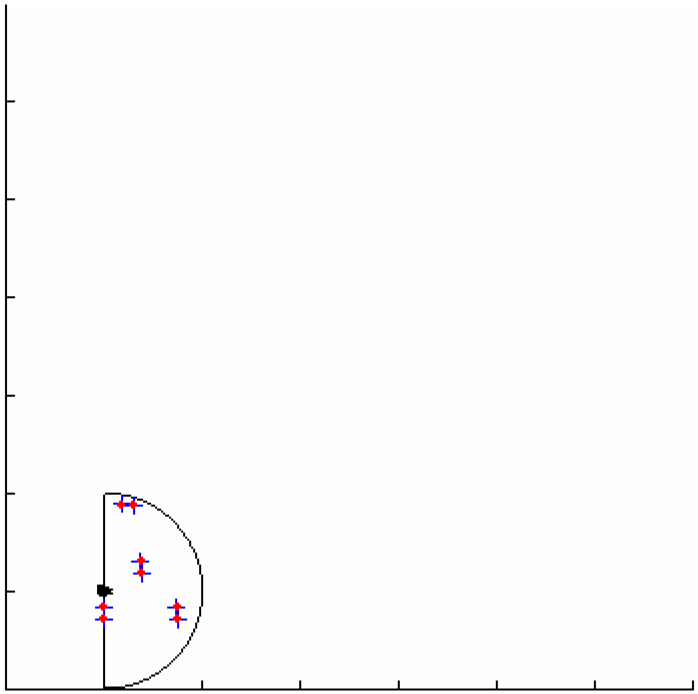
Why data association is difficult

- Low feature density
- High feature density

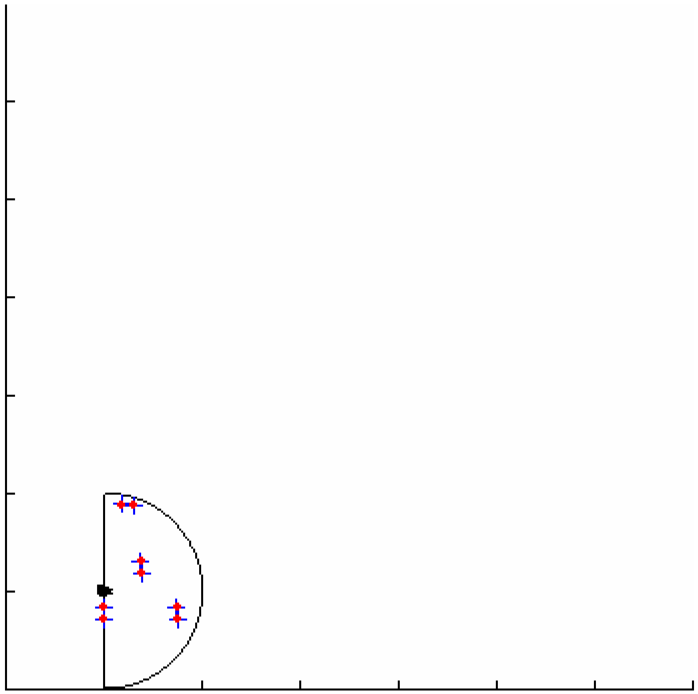




How important is data association?

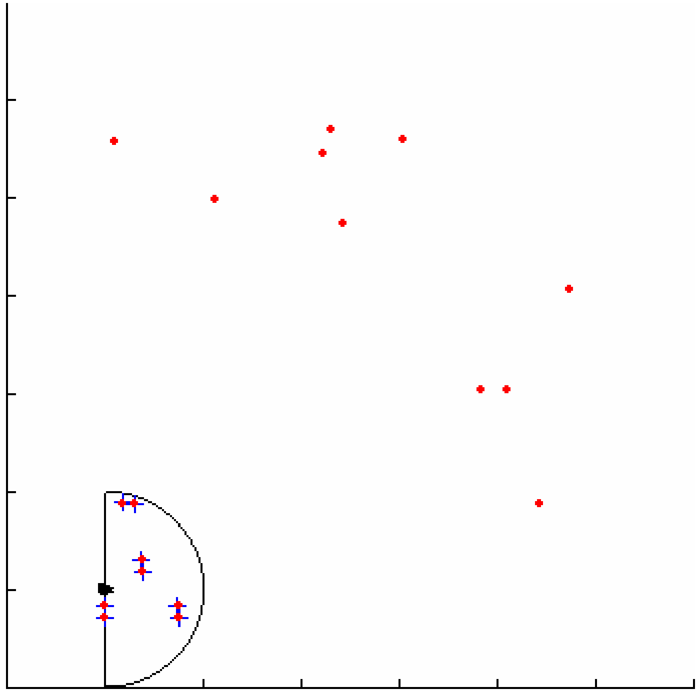


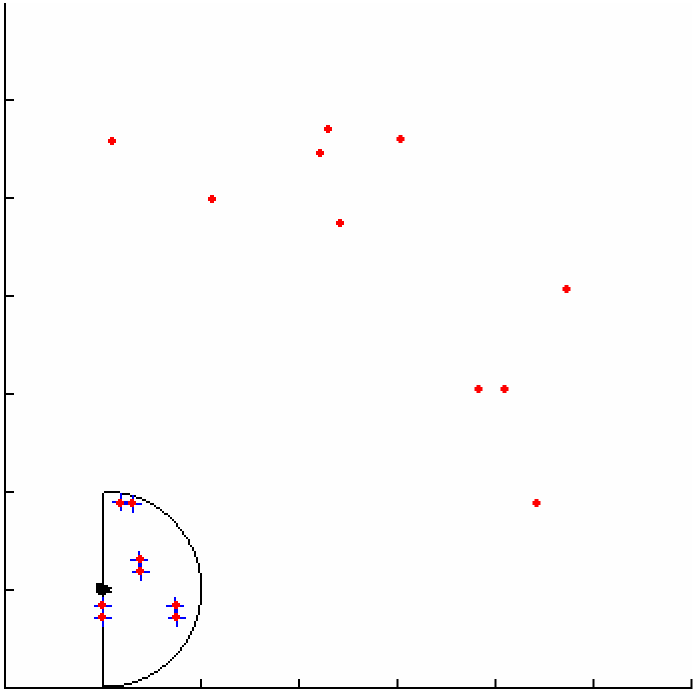
A good algorithm



A **bad** algorithm

Why it's difficult?





A good algorithm

A **bad** algorithm

Importance of Data Association

- EKF update:

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

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

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

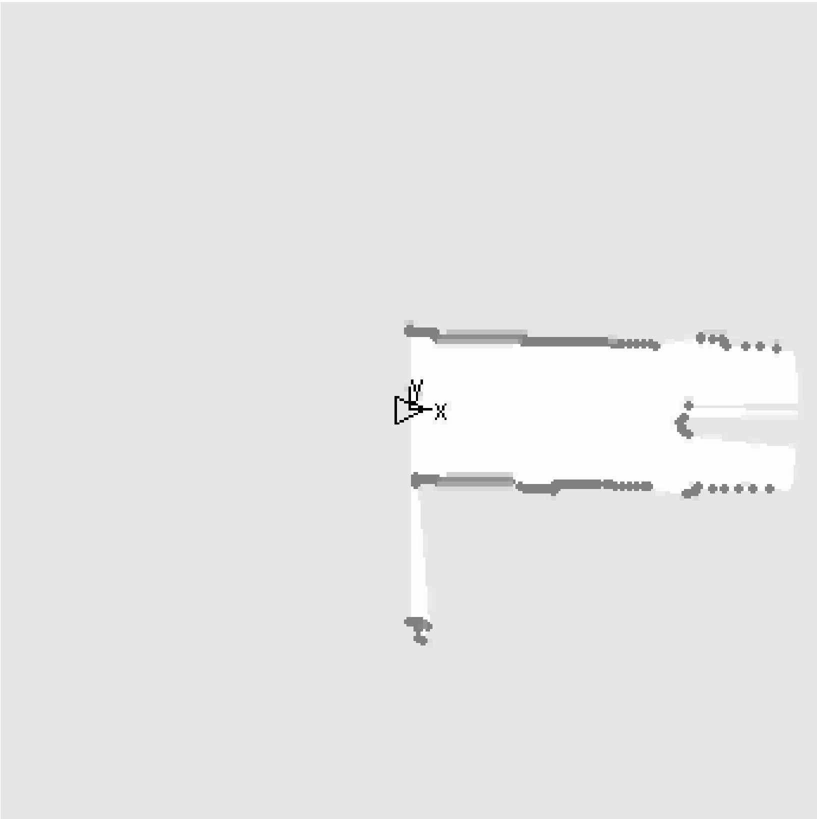
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}^B \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1}^B \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

- If the association of $\boldsymbol{E_i}$ with feature $\boldsymbol{F_j}$ is.....

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

2. Feature Extraction

Feature extraction: Laser



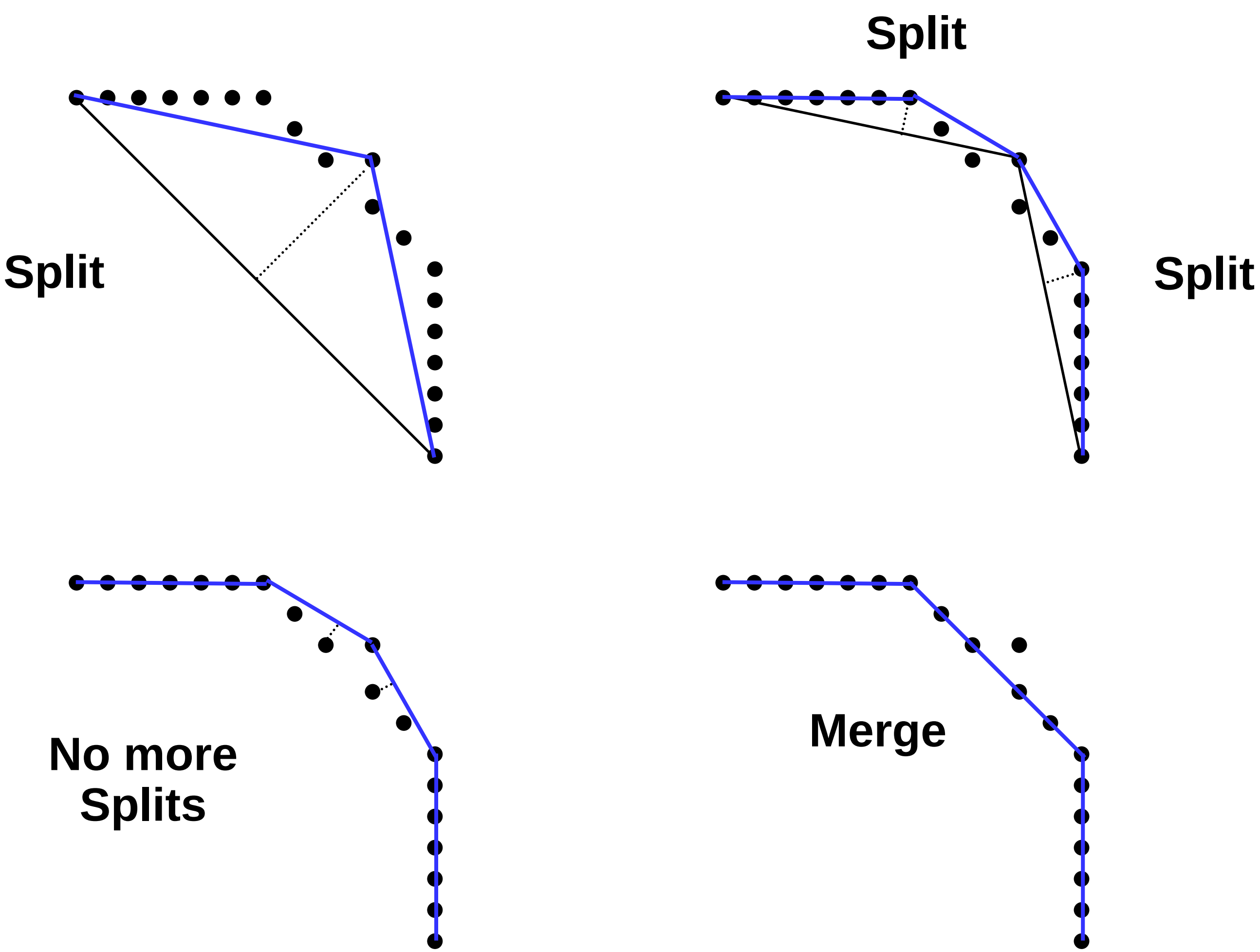
- Obtain line segments from a laser scan:
  - Segmentation
  - Line estimation

Split and merge:

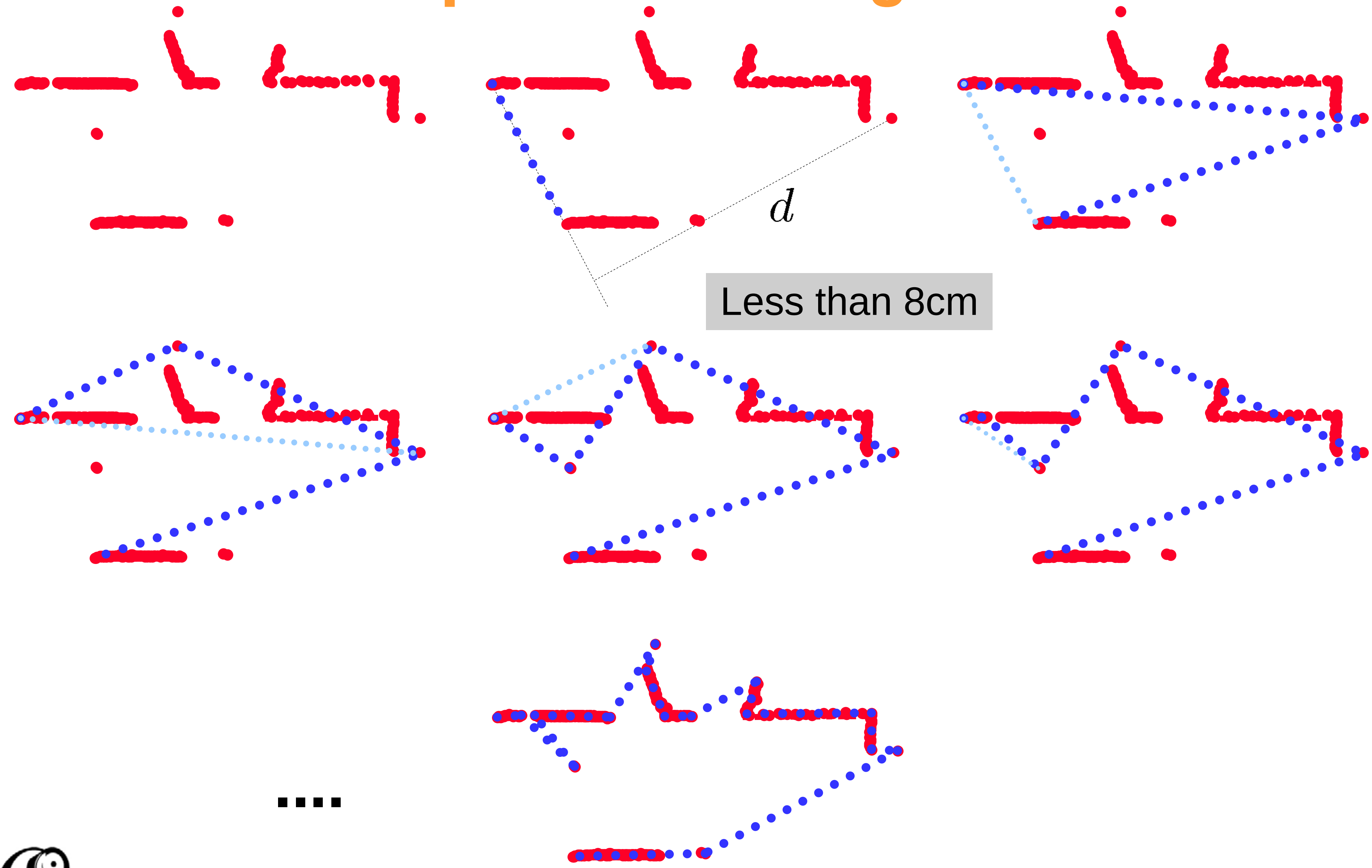
1. **Recursive Split:**
   1. Obtain the line passing by the two extreme points
   2. Obtain the point more distant to the line
   3. If distance > error_max, split and repeat with the left and right sub-scan
2. **Merge:**
   1. If two consecutive segments are close enough, obtain the common line and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

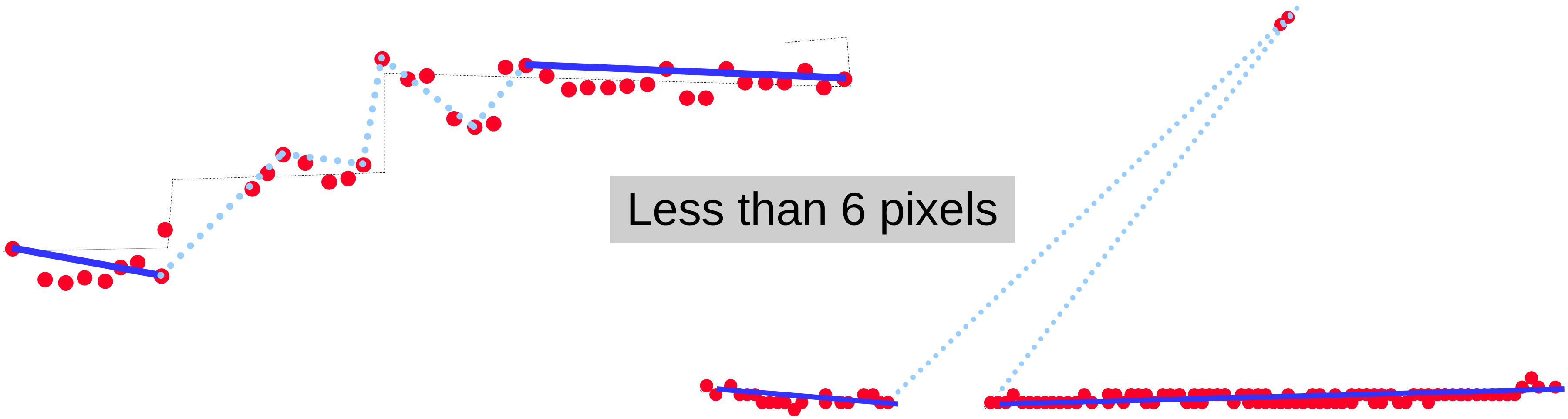


Split and Merge

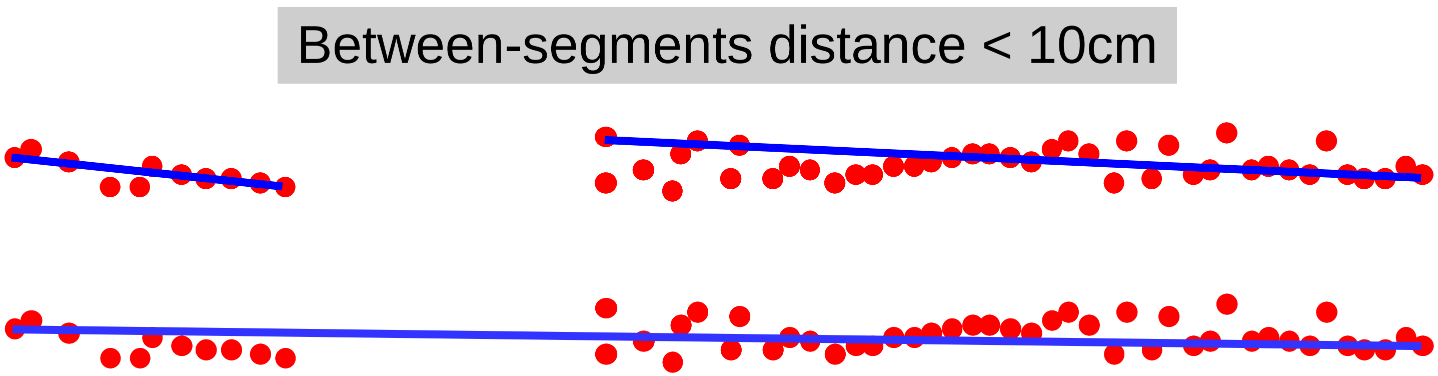


Split and Merge

- Elimination of small segments:

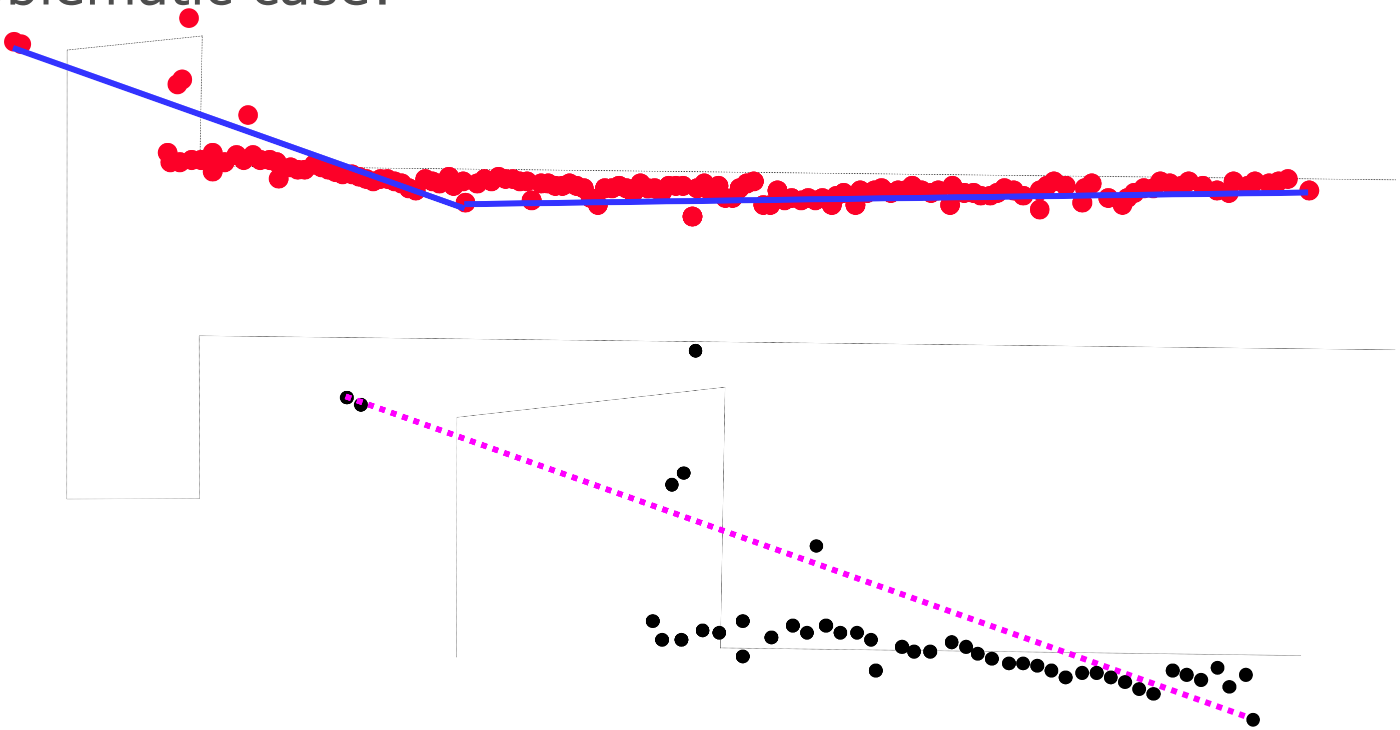


- Segment fusion:



Split and Merge

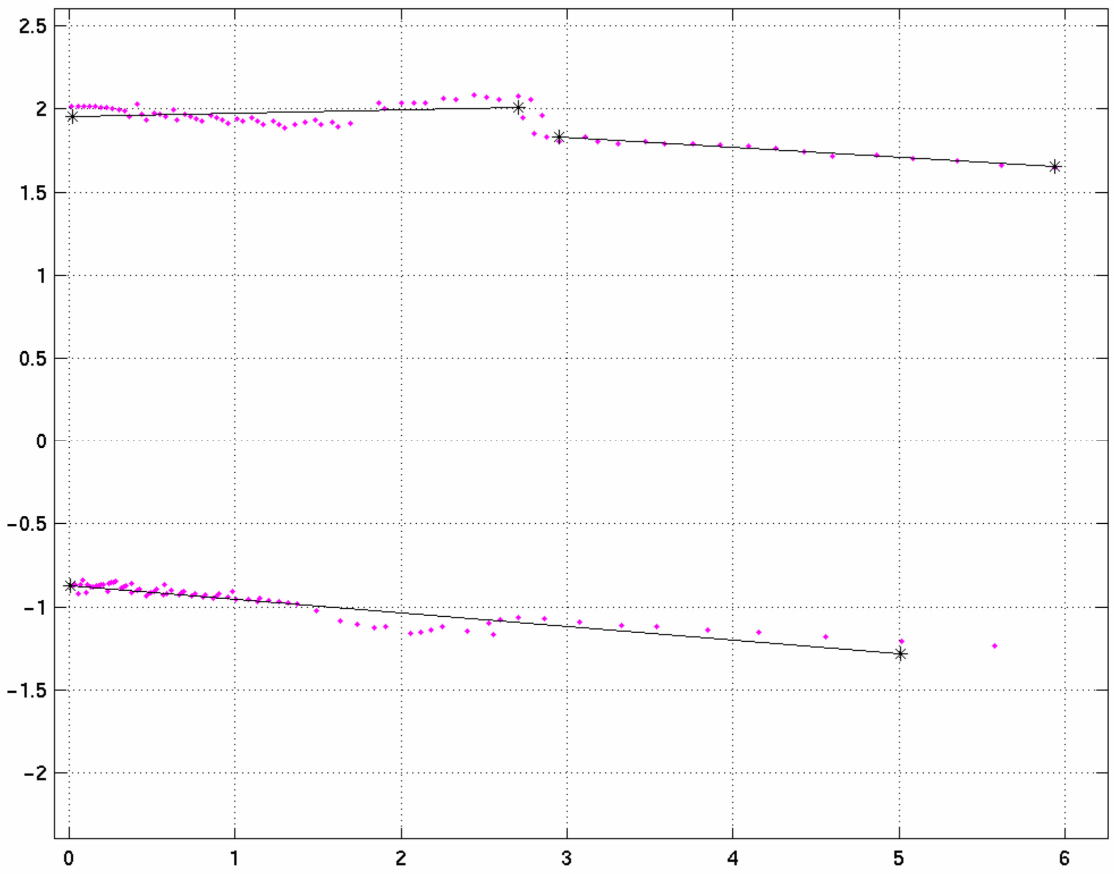
- Problematic case:



- **Split and merge:** only the extreme points are used
- **Alternatives:**
  - Linear regression
  - RANSAC
  - Hough transform

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Split and Merge



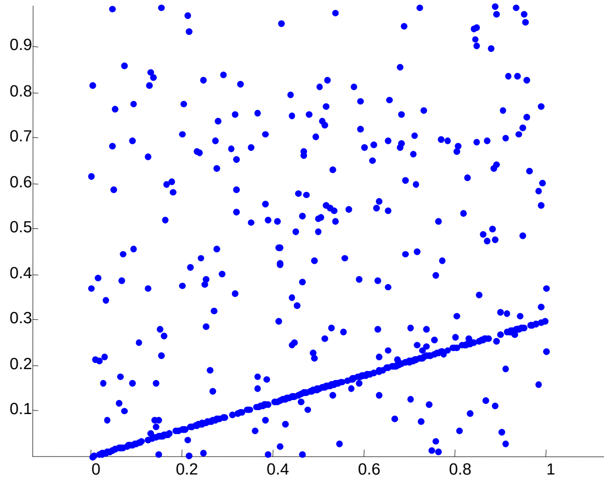
Not robust to complex and/or spurious data

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

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

Alternative: RANSAC



z acceptable probability of failure

t tries ¿?

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

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

p no. of points

n points to build model

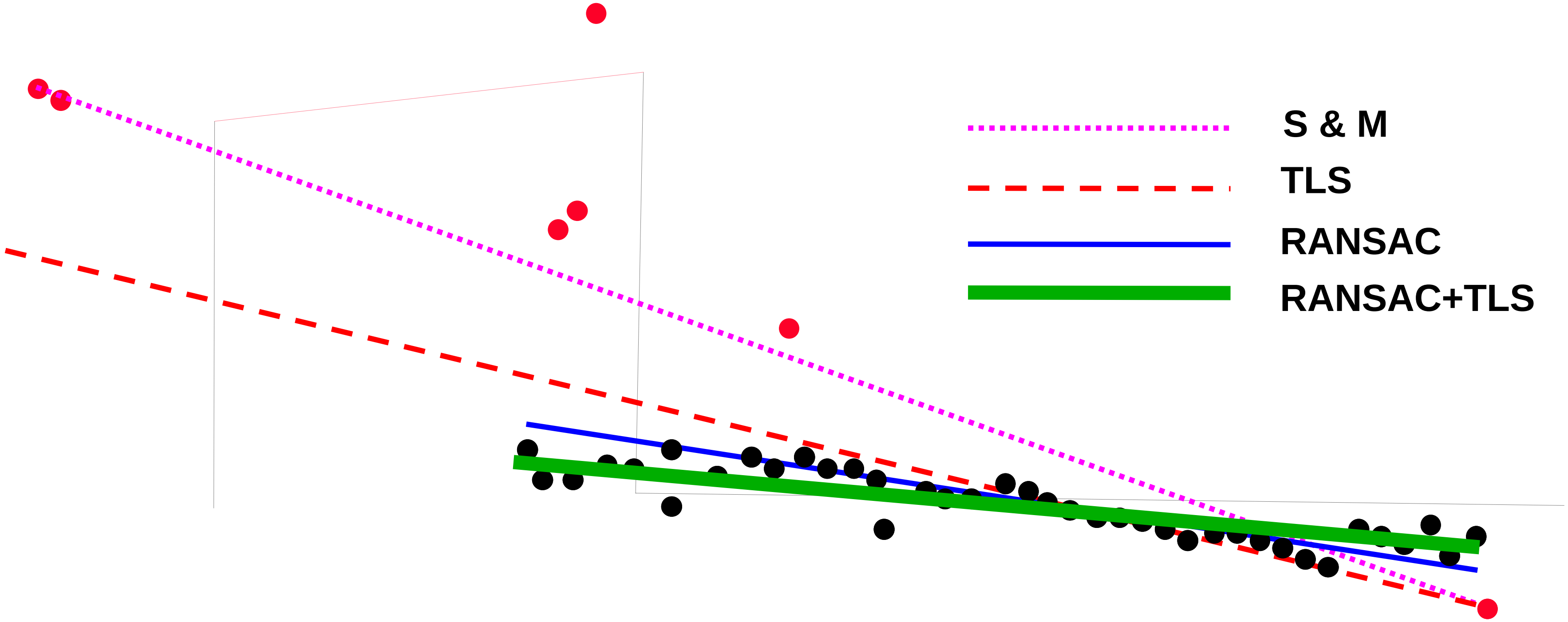
w probability that a point is good

$O(p^n)$_possible_models

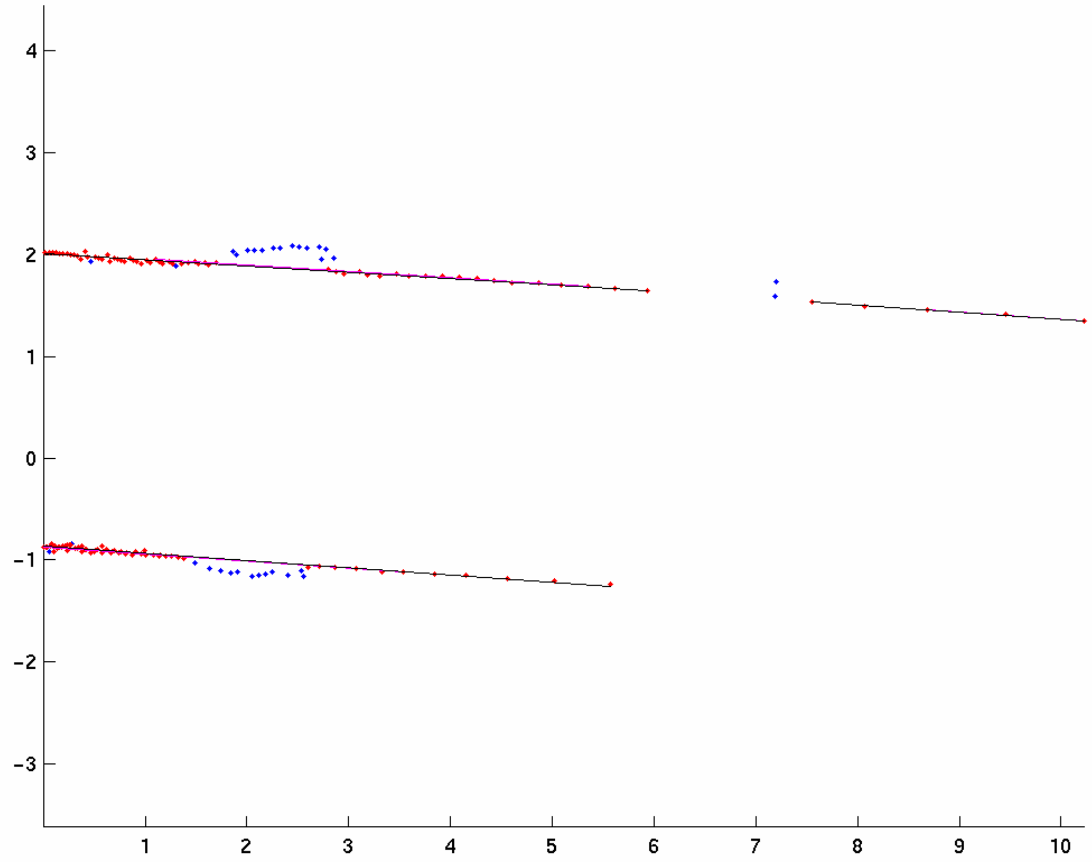
w	0,5
n	2
z	0,05
t	11

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

RANSAC

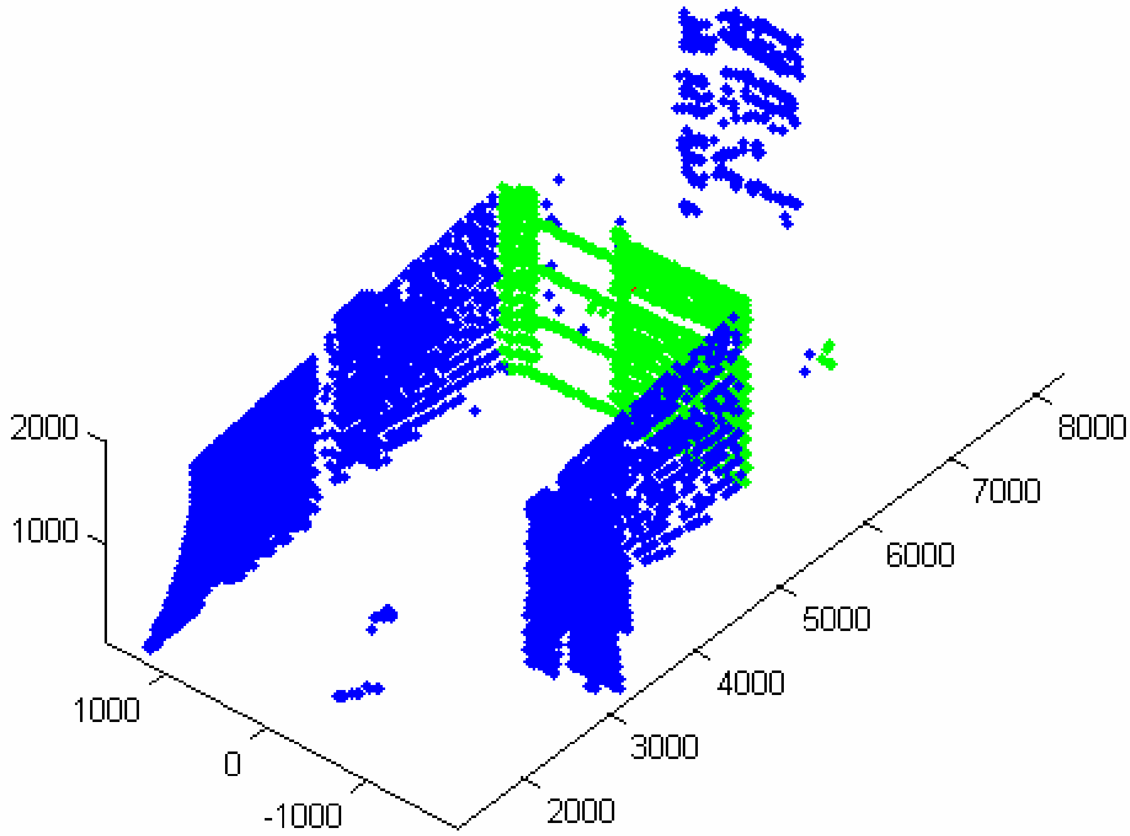


RANSAC



Robust statistics deal with spuriousness

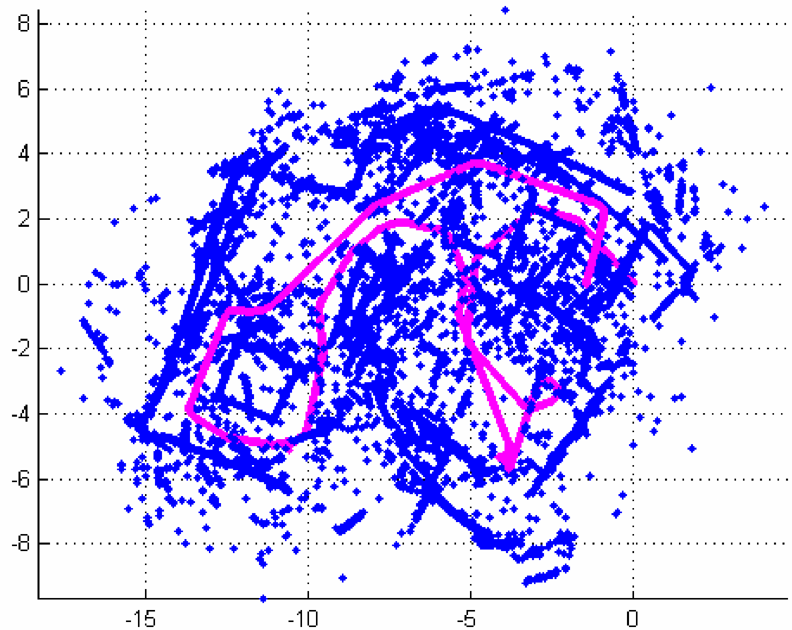
RANSAC for 3D planes

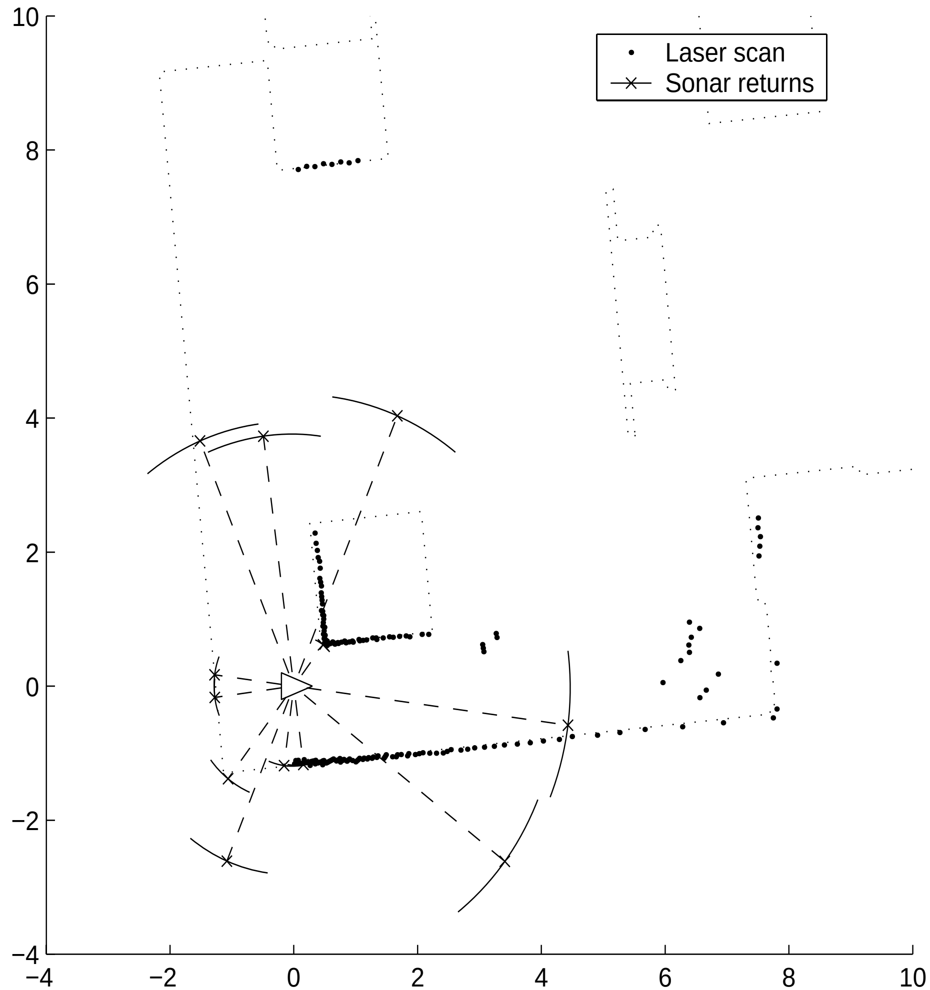


P.M. Newman, J.J. Leonard, J. Neira and J.D. Tardós: **Explore and Return: Experimental Validation of Real Time Concurrent Mapping and Localization**. IEEE Int. Conf. Robotics and Automation, May, 2002

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

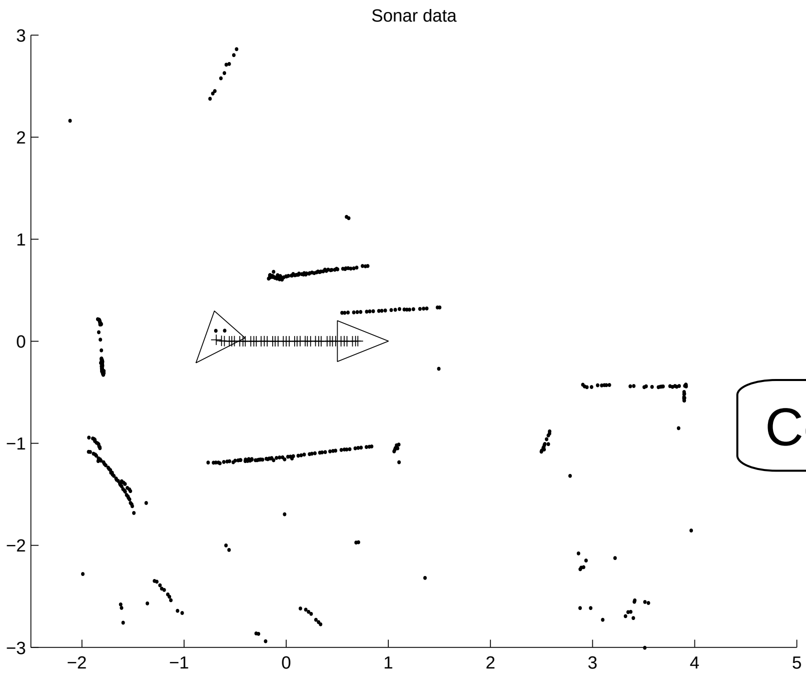
Sonar

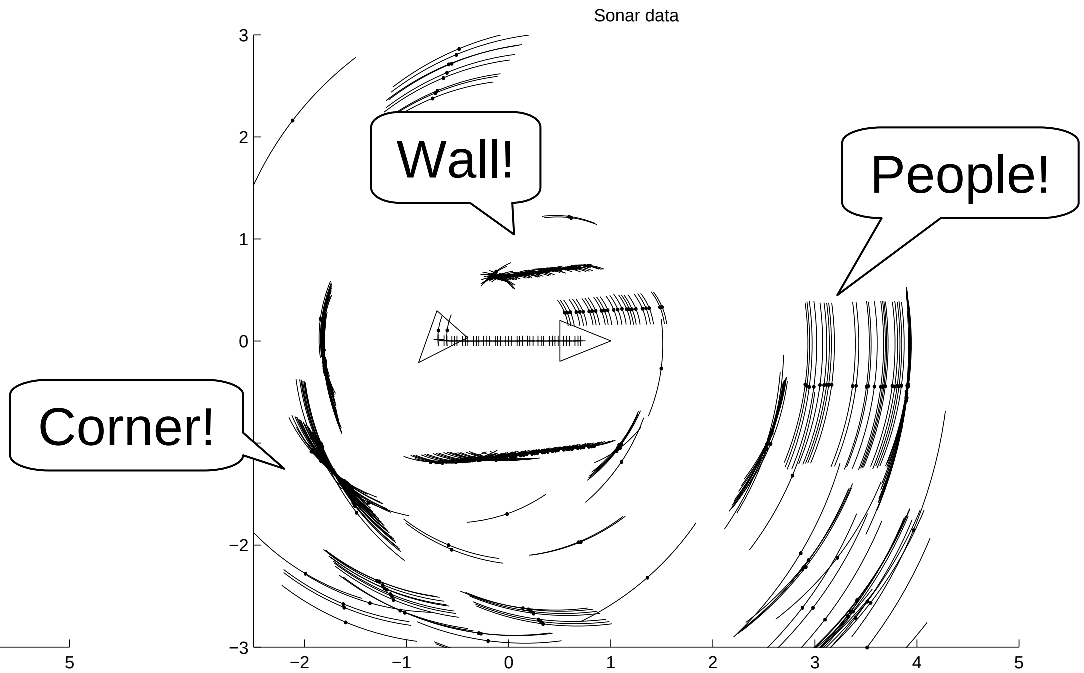




Very sparse and noisy data

Move and build a local map

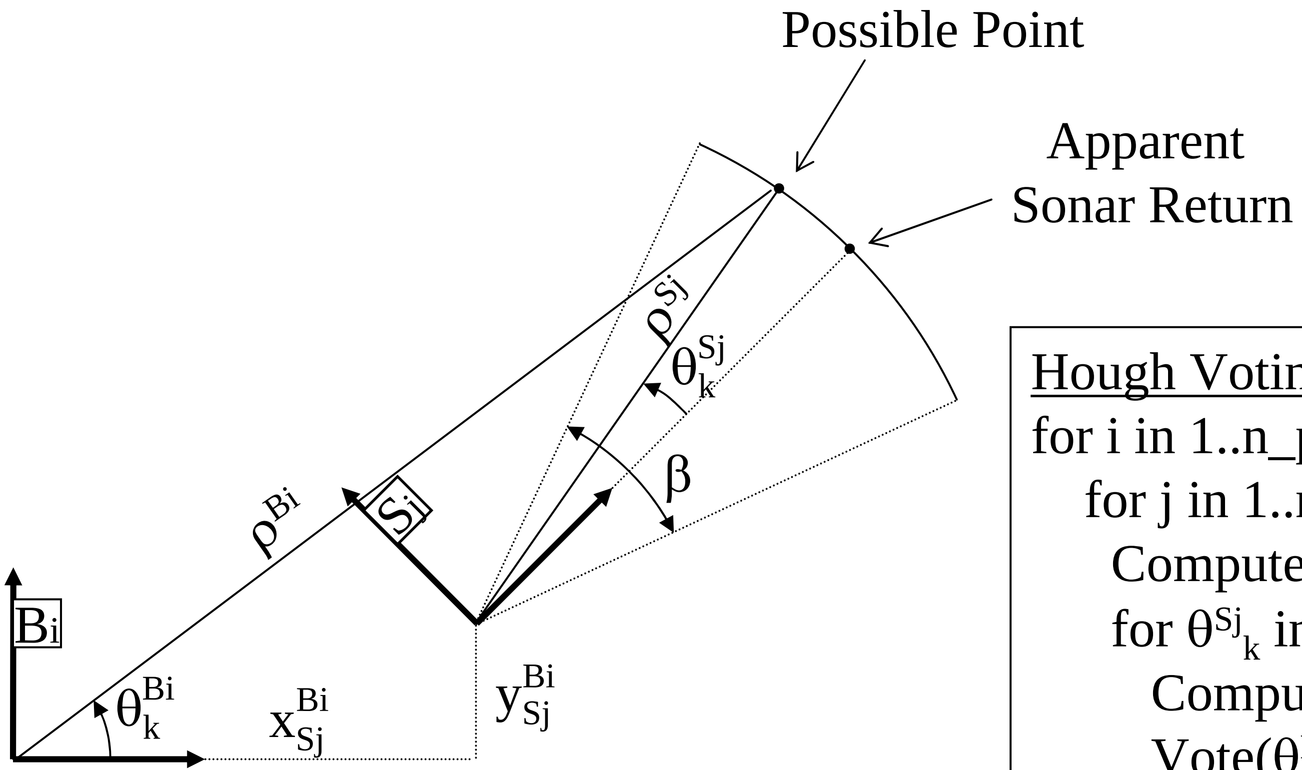




Exploit redundancy

Use a good sensor model

Sonar Model for Points

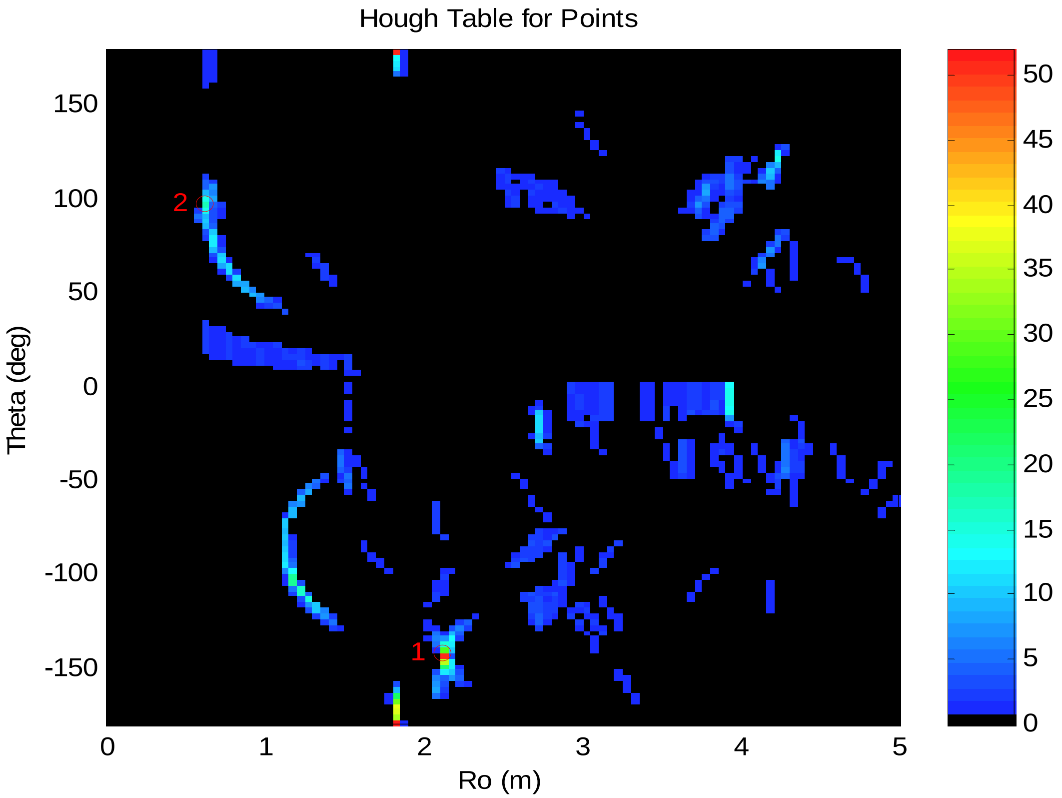


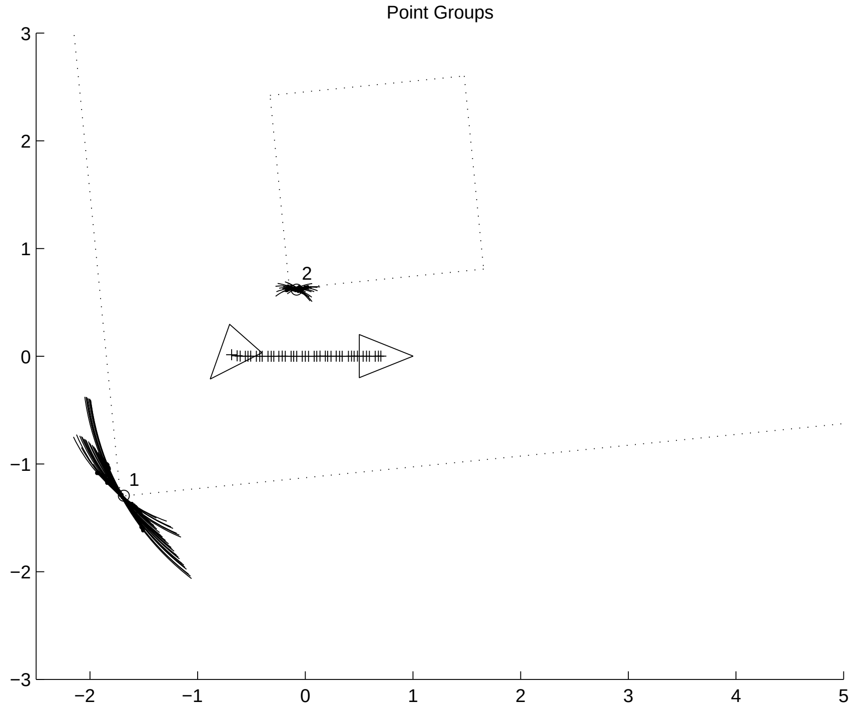
<u>Hough Voting</u>

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

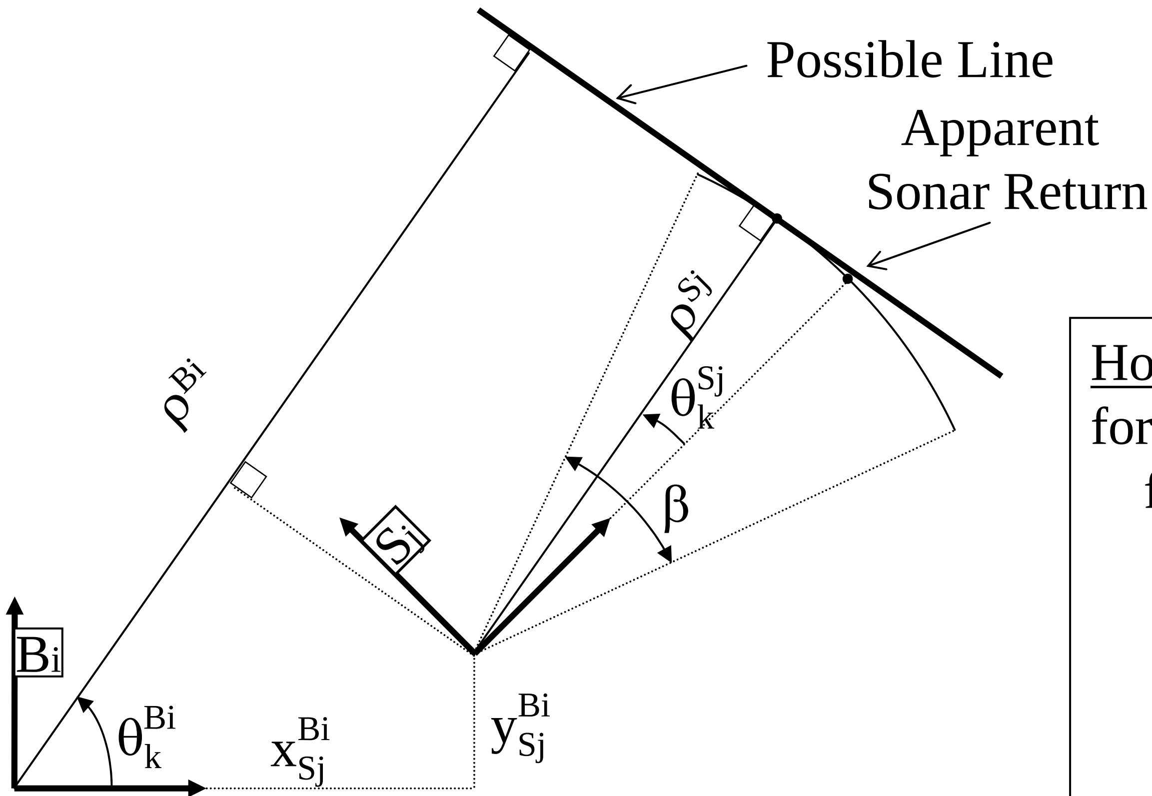
- Sonar returns **vote** for points
- Look for local maxima





ROBOTICS UNIVERSIDAD DE ZARAGOZA

Sonar Model for Lines

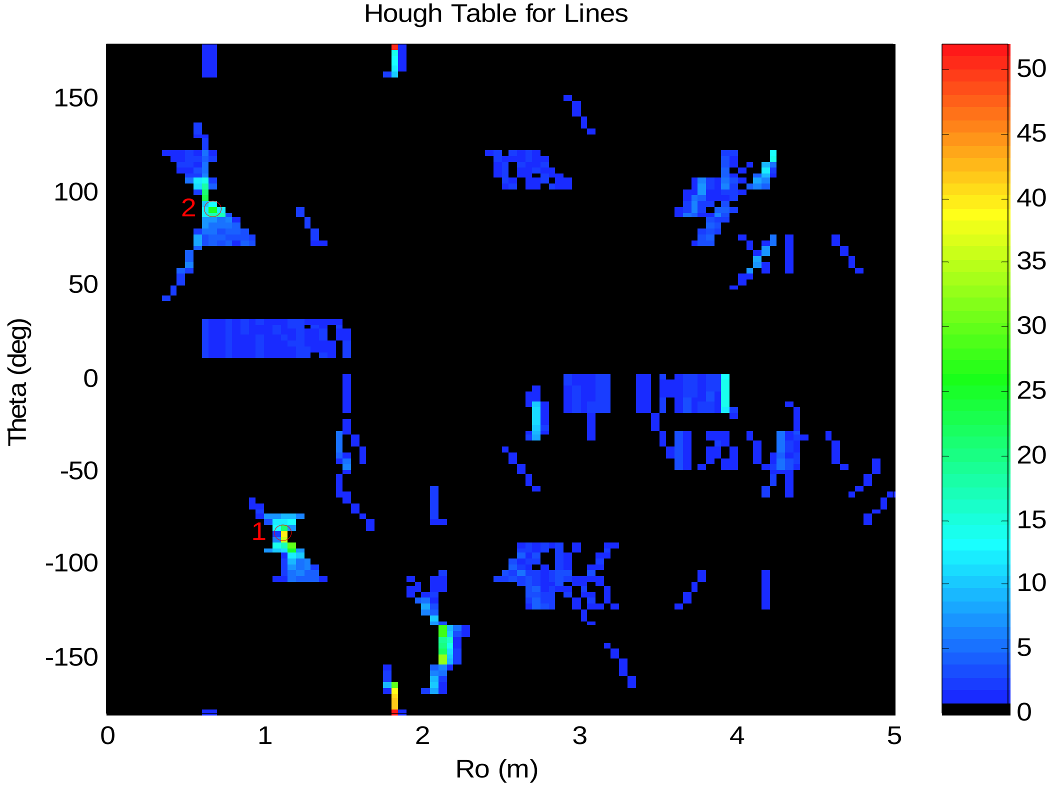


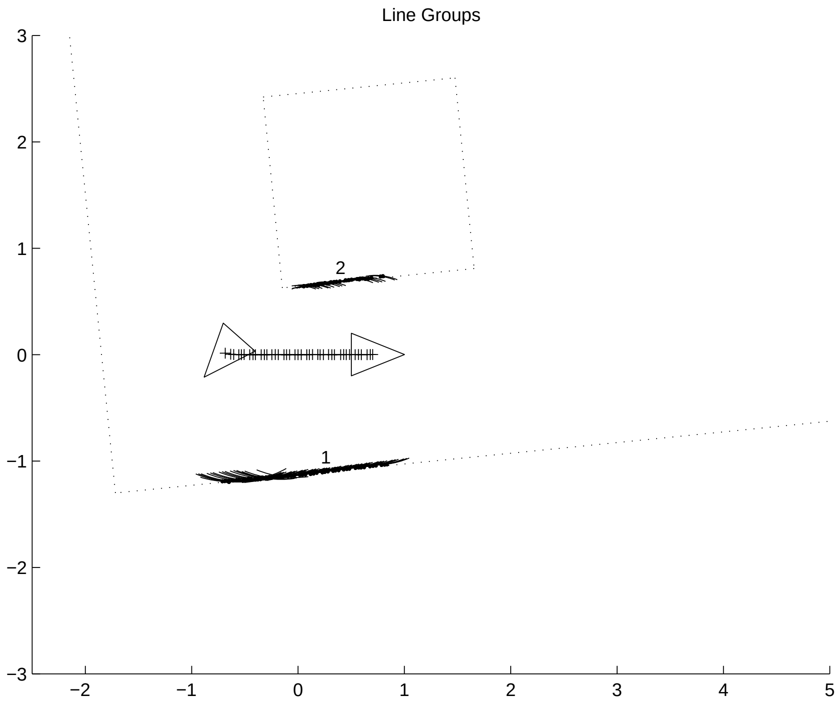
<u>Hough Voting</u>

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

Hough Transform: Lines

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



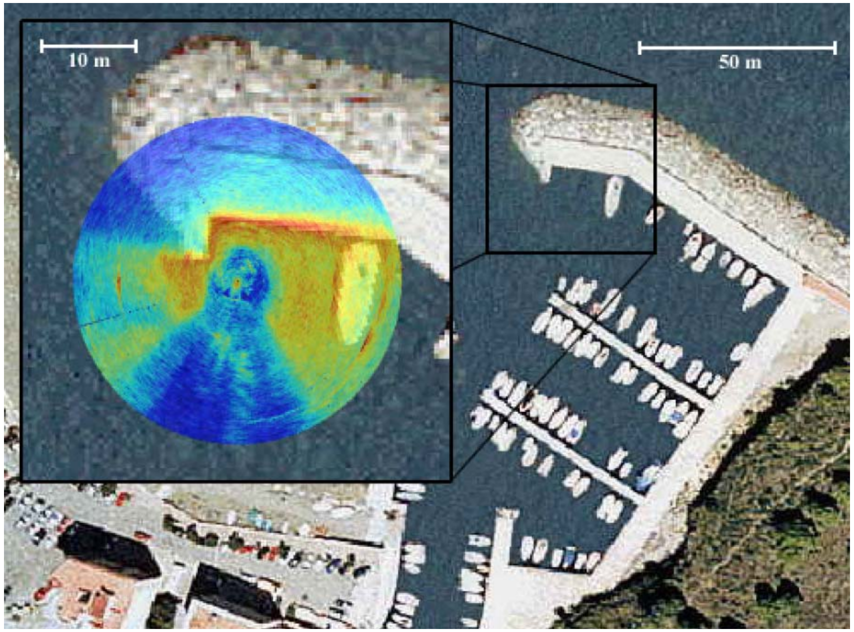


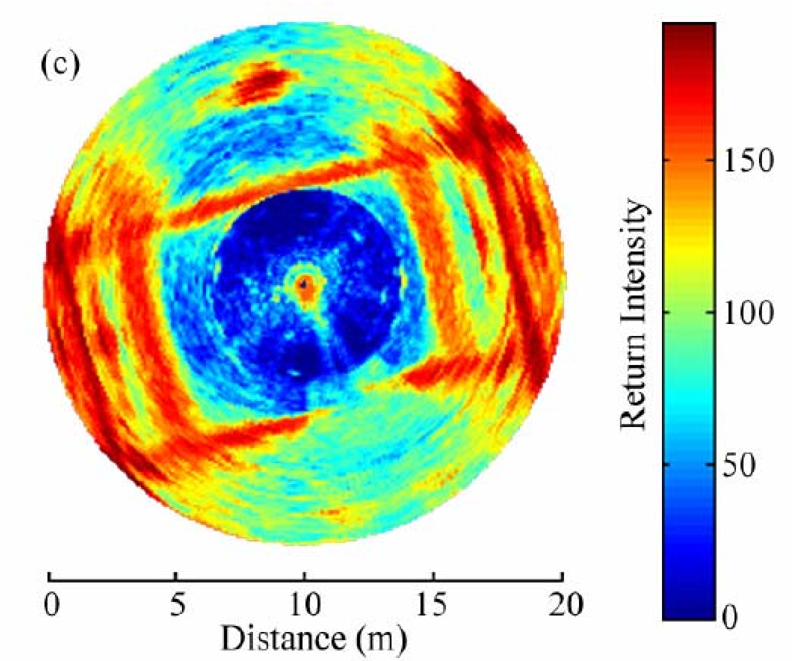
The Hough gives robust **local** data associations

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Hough Transform

- Imaging sonar





Hough Transform

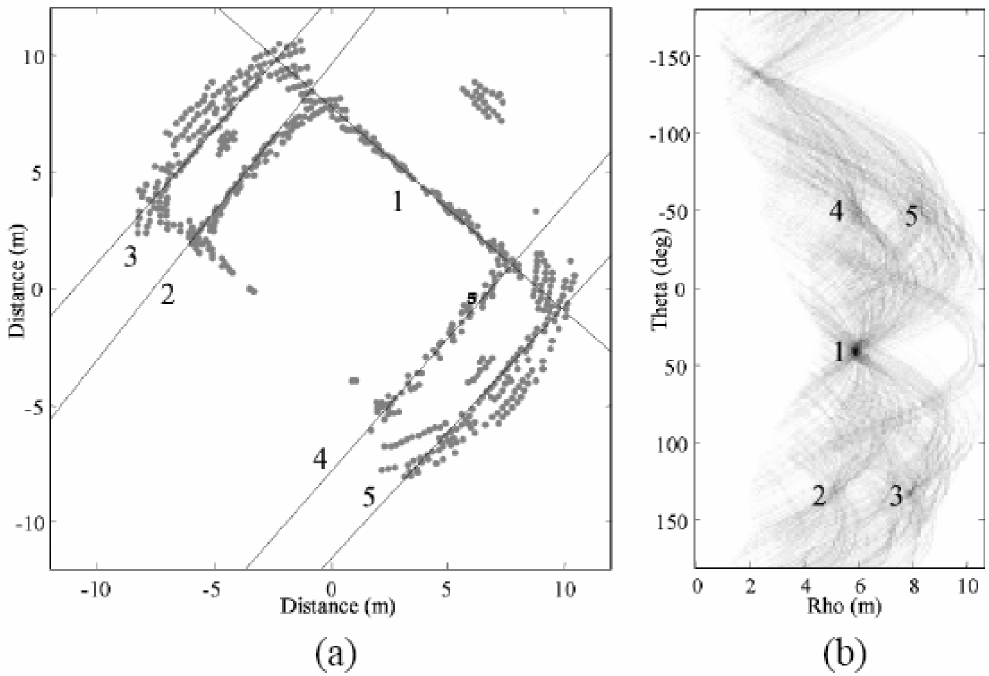
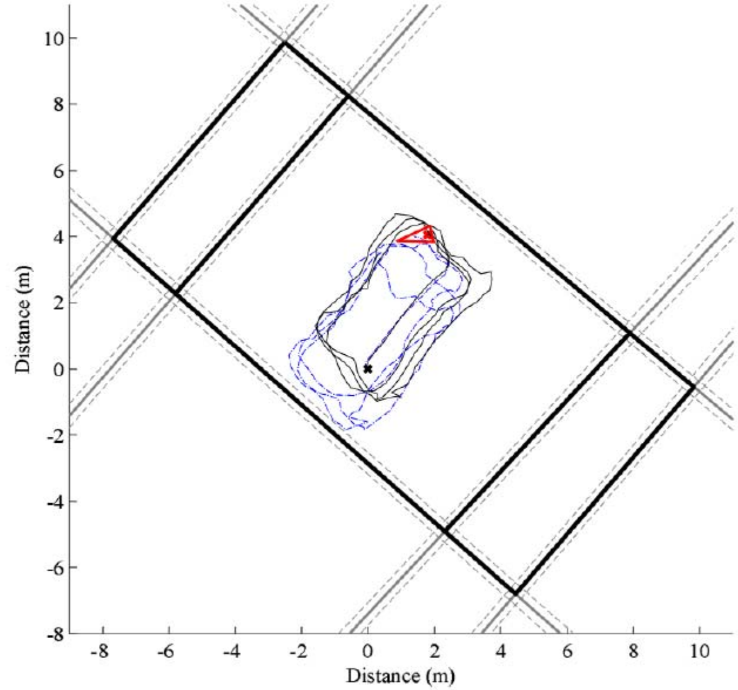


Fig. 3. Hough transform for line detection. (a) High echo-amplitude returns and the winning lines. (b)The obtained Hough voting space



D. Ribas, P. Ridao, J. Neira, J.D. Tardós, **SLAM using an Imaging Sonar for Partially Structured Underwater Environments**, The 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (to appear).

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

3. Data association in continuous SLAM

Individual Compatibility

- Measurement equation for observation E_i and feature F_j

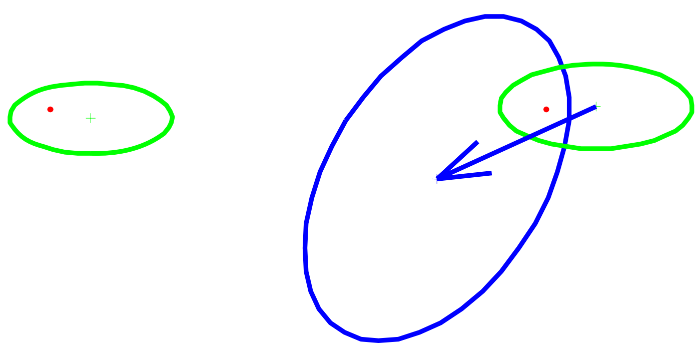
$$\mathbf{z}_i = \mathbf{h}_{ij}(\mathbf{x}^B) + \mathbf{w}_i$$
$$\mathbf{z}_i \simeq \mathbf{h}_{ij}(\hat{\mathbf{x}}^B) + \mathbf{H}_{ij}(\mathbf{x}^B - \hat{\mathbf{x}}^B)$$

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

- ***E_i*** and ***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)

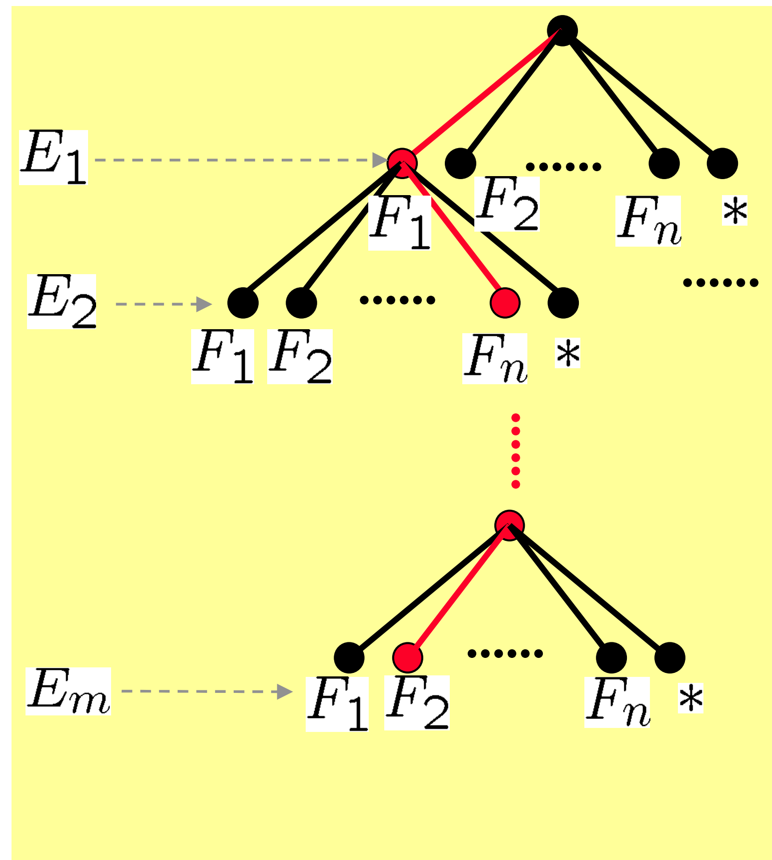


Nearest Neighbor

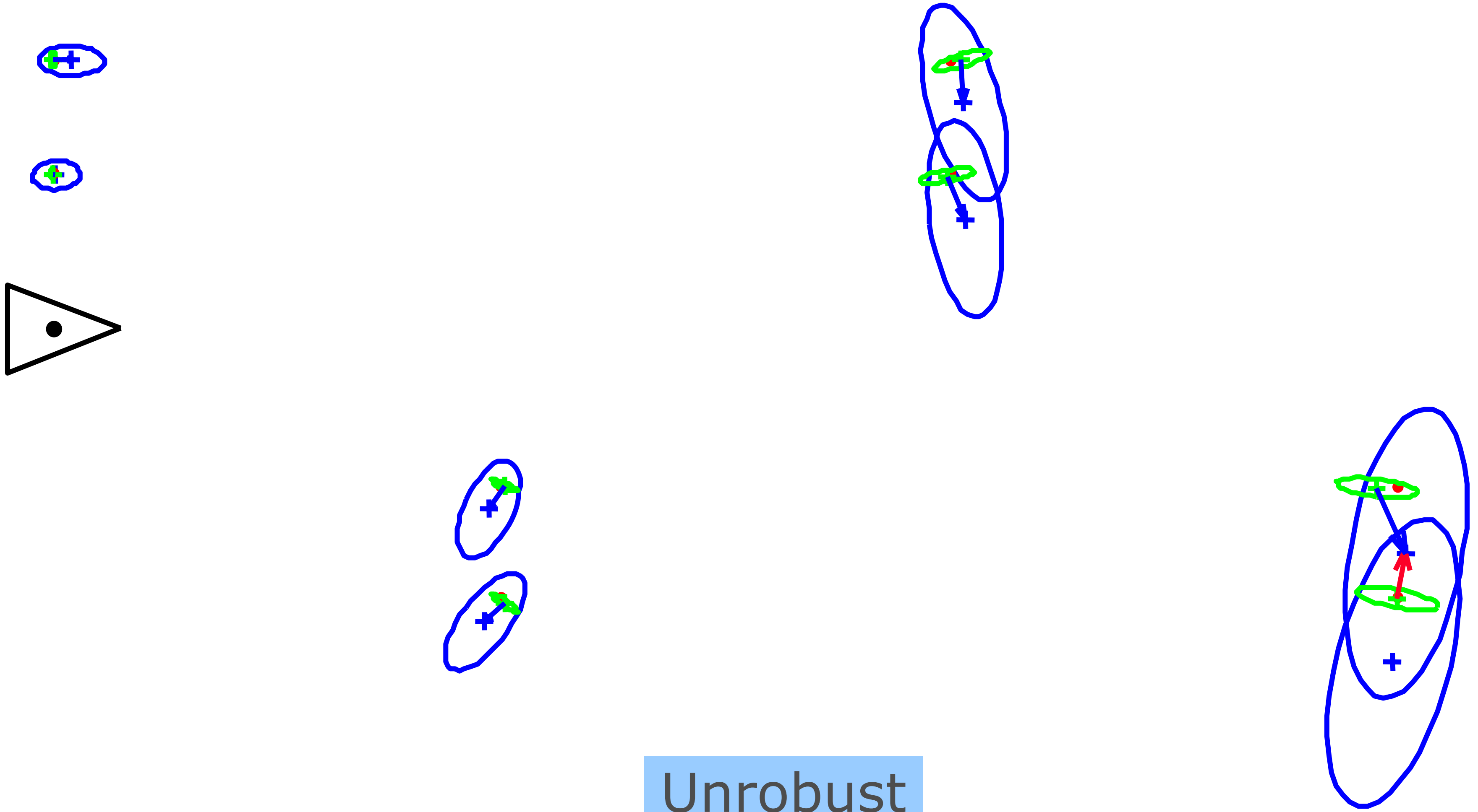
Algorithm 2 Individual Compatibility Nearest Neighbor ICNN $(E_{1\cdots m}, F_{1\cdots n})$

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

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

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

d = length(z)

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Joint Compatibility Branch and Bound

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

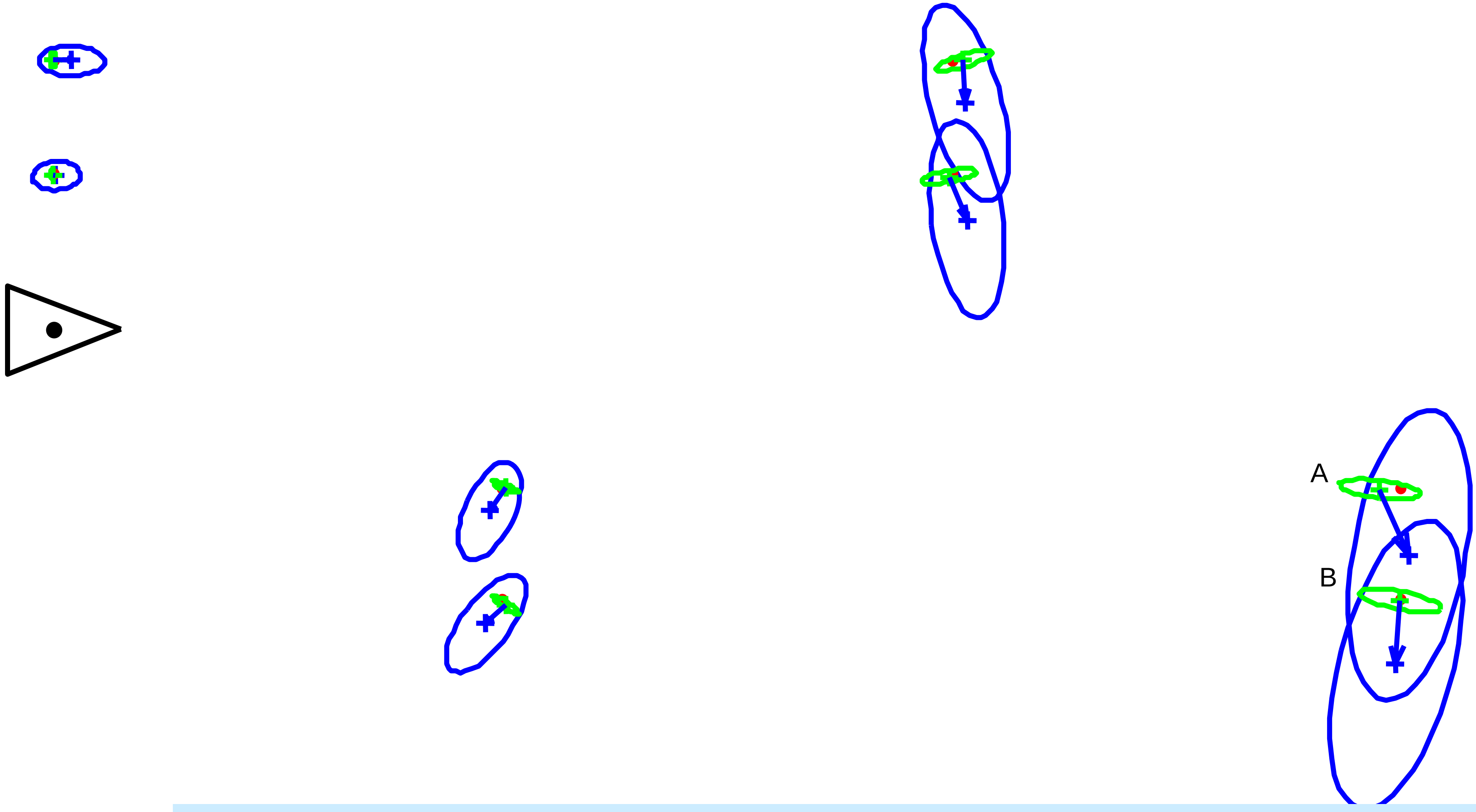
```
procedure JCBB (H, i):  -- find pairings for observation E_i

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**

ROBOTICS UNIVERSIDAD DE ZARAGOZA

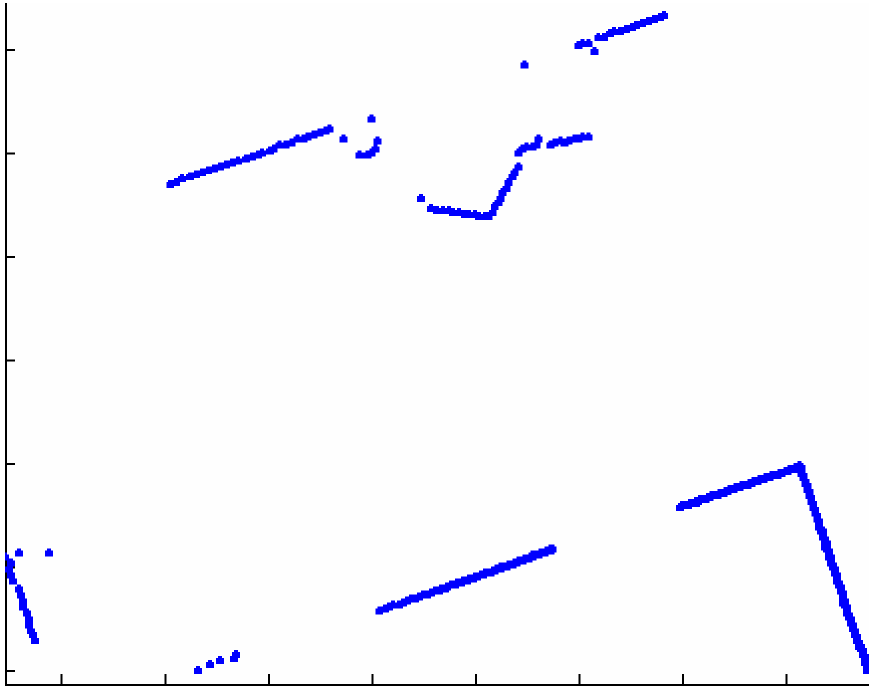
The Fallacy of the Nearest Neighbor

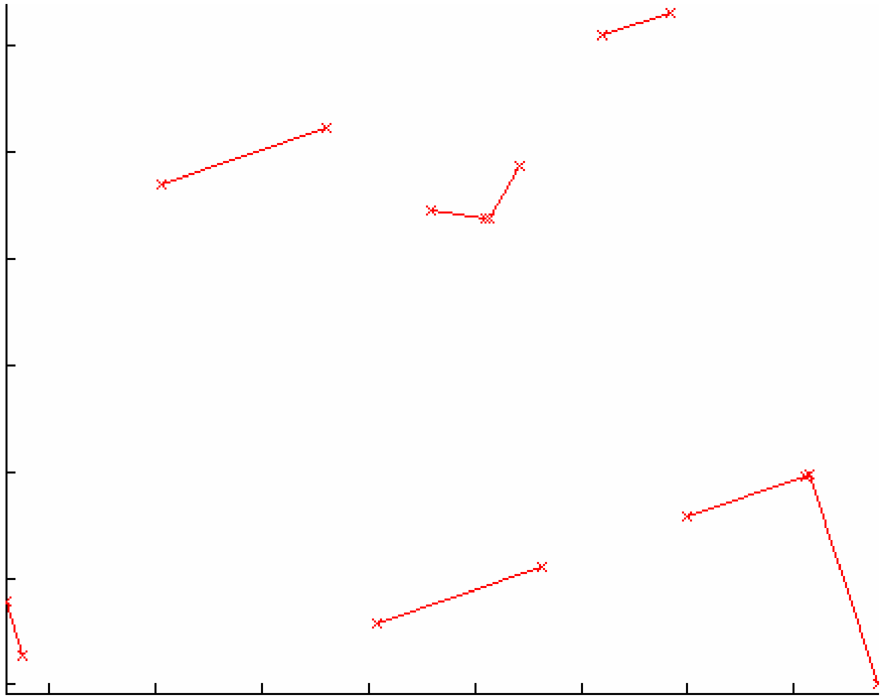


J. Neira, J.D. Tardós. **Data Association in Stochastic Mapping using the Joint Compatibility Test** IEEE Trans. Robotics and Automation, Vol. 17, No. 6, Dec 2001, pp 890 –897

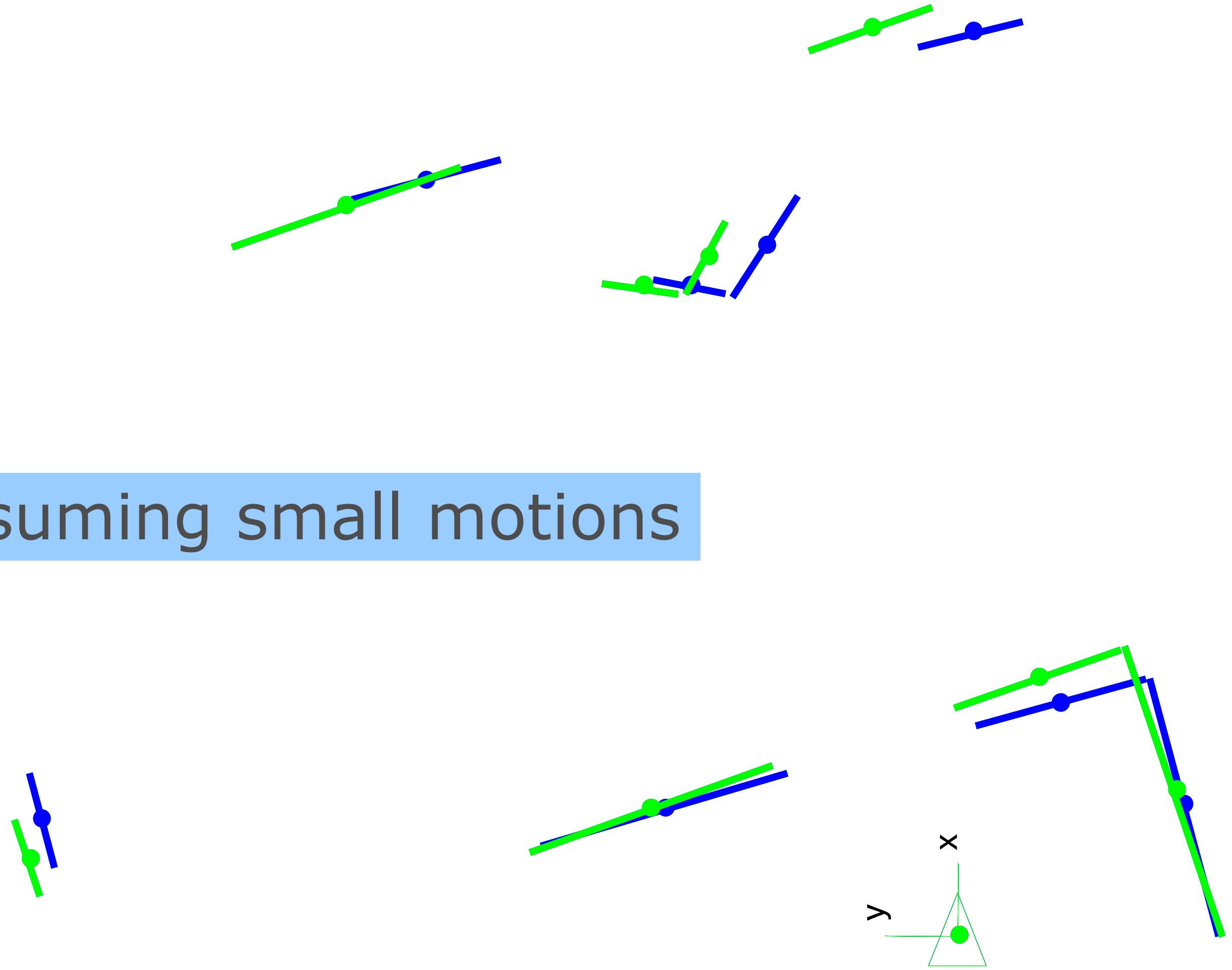
SLAM without odometry

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



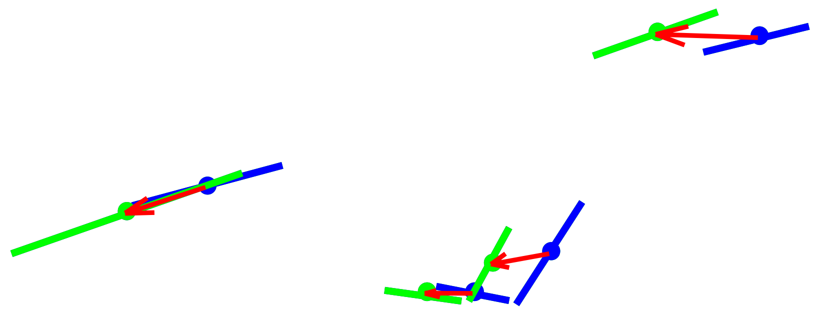


SLAM without odometry

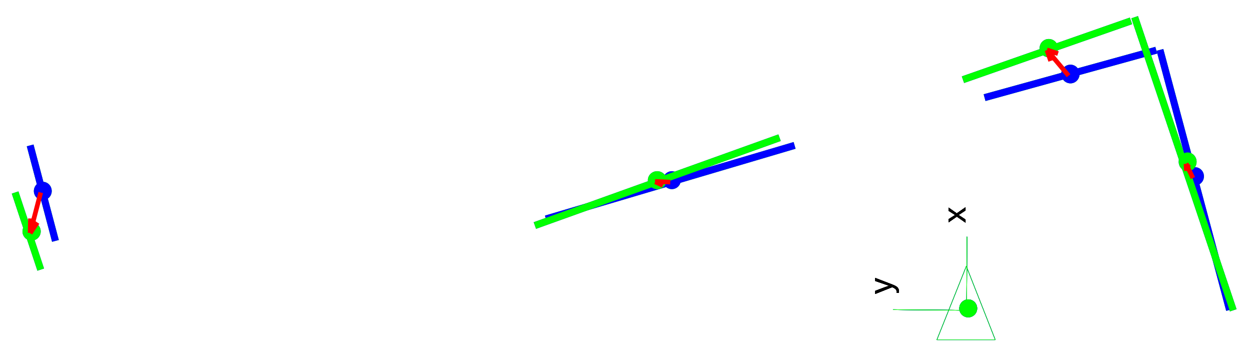


Assuming small motions

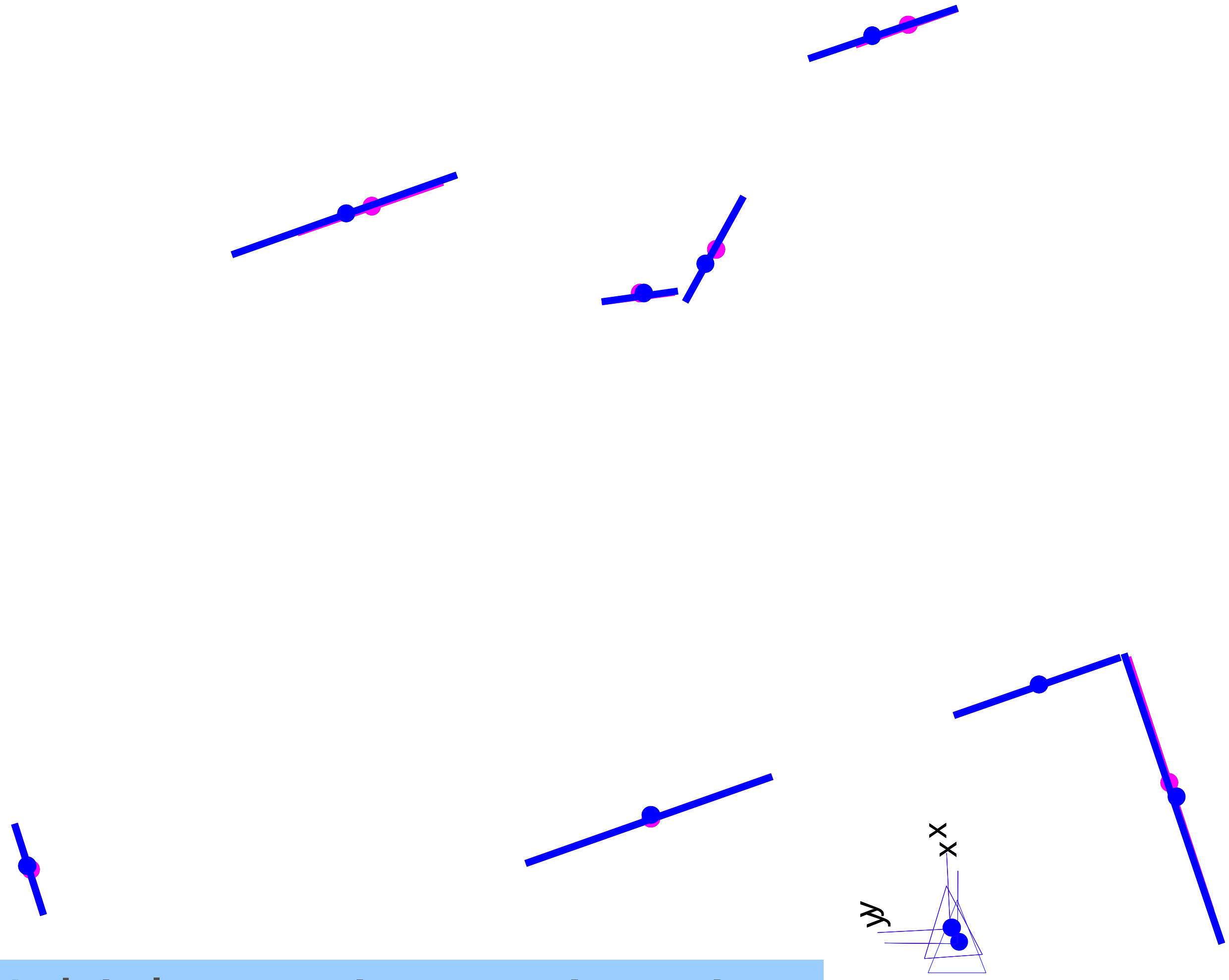
SLAM without odometry



Data association using Joint Compatibility

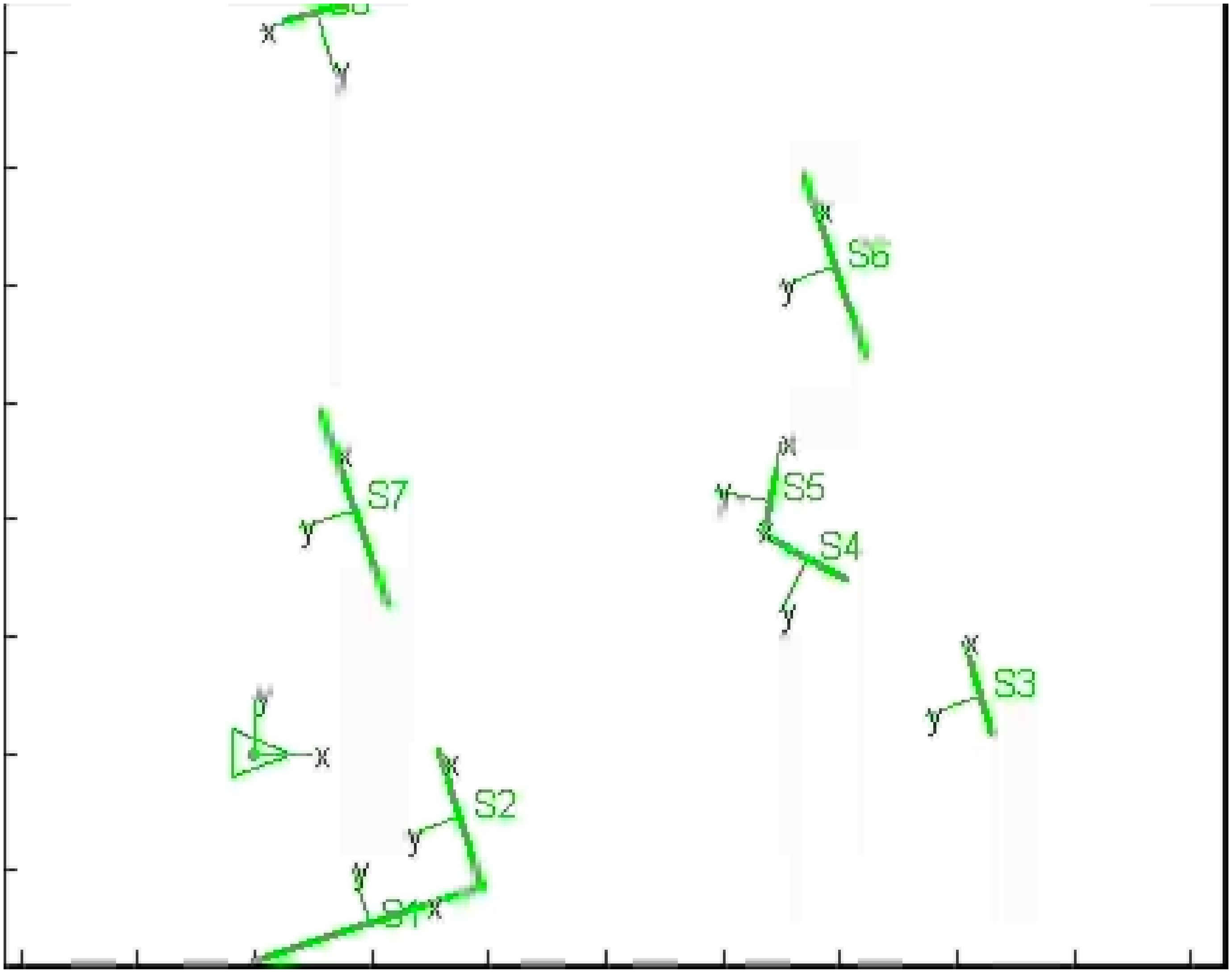


SLAM without odometry

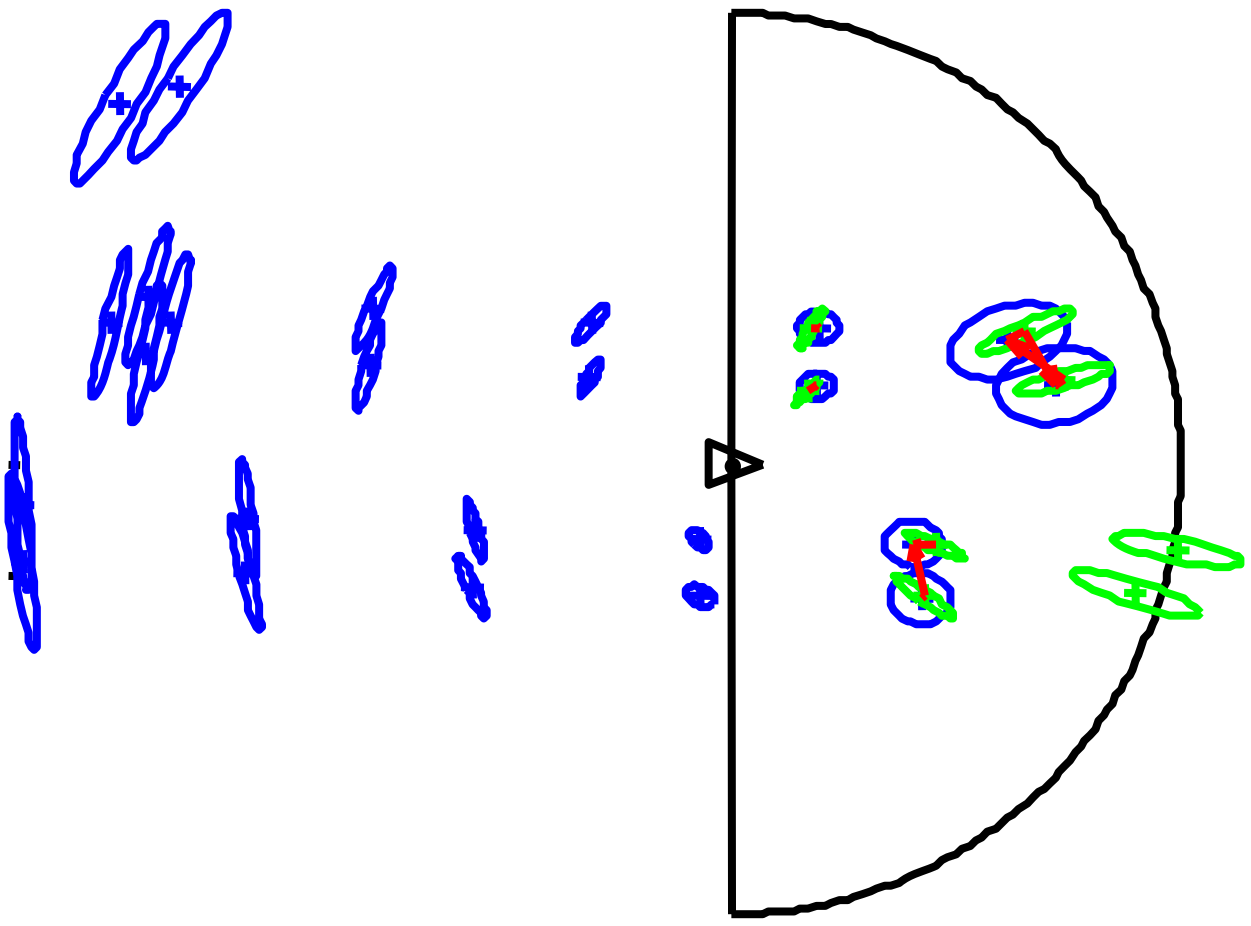


Vehicle motion estimation

SLAM without odometry

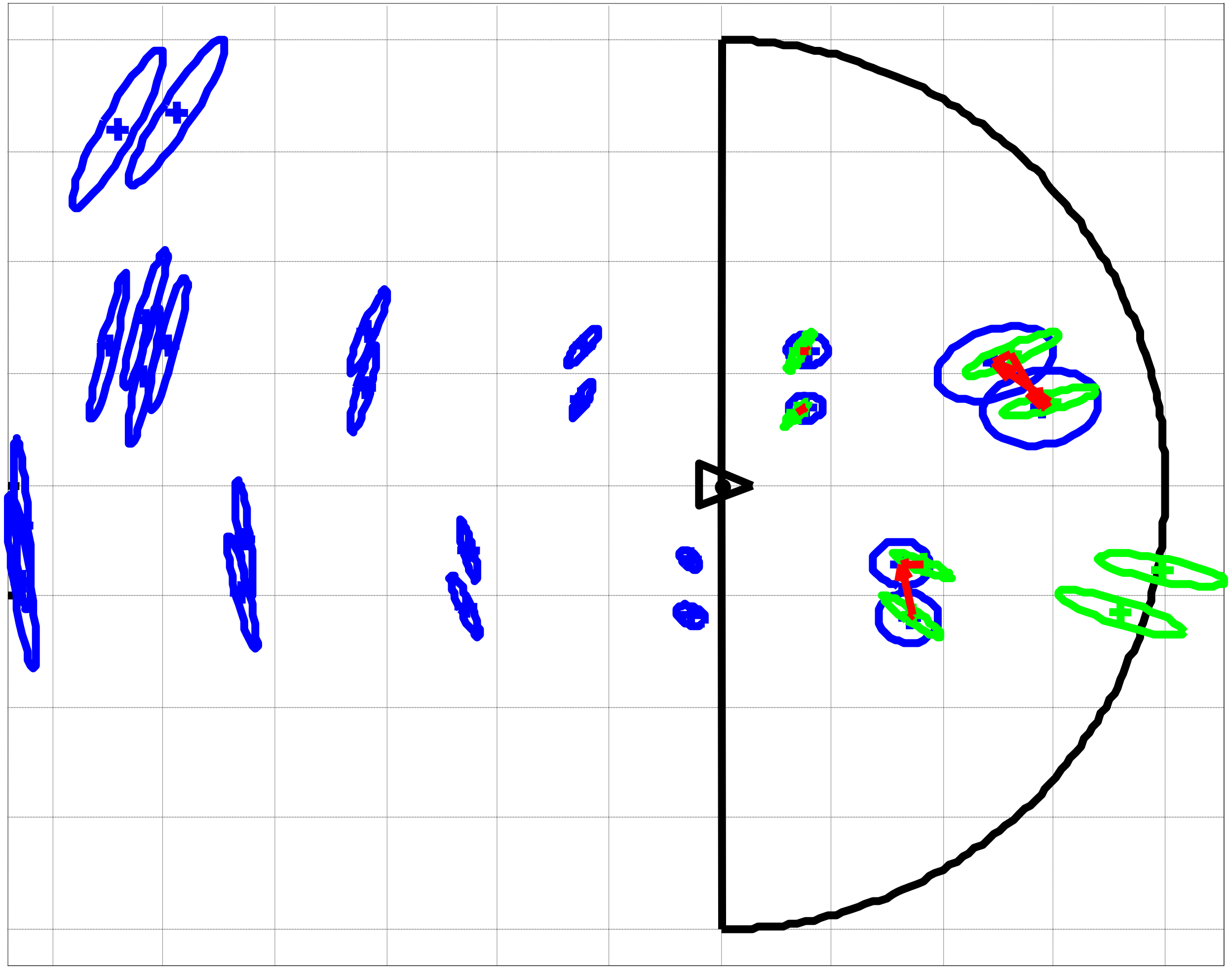


Continuous data association



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

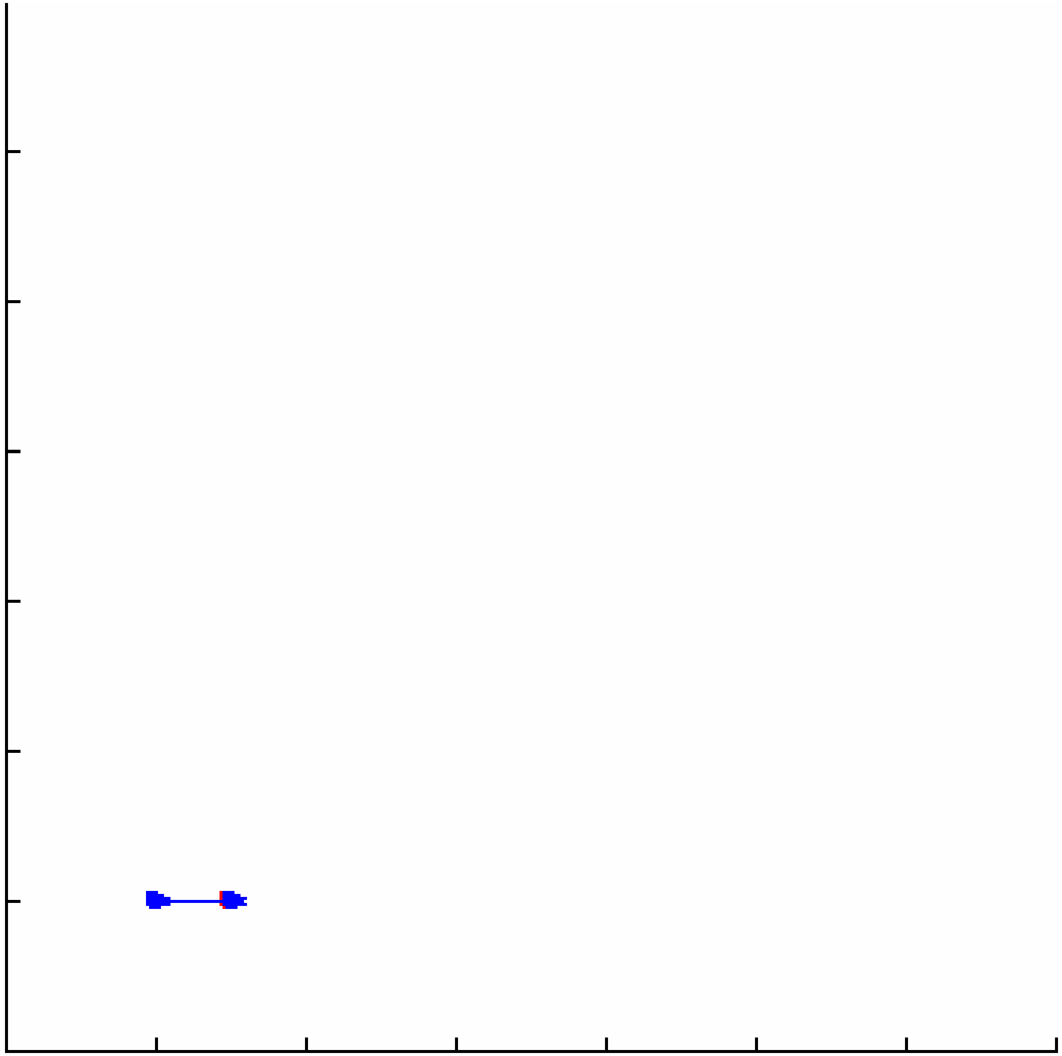
Map tessellation



Individual compatibility can be *O(1)*

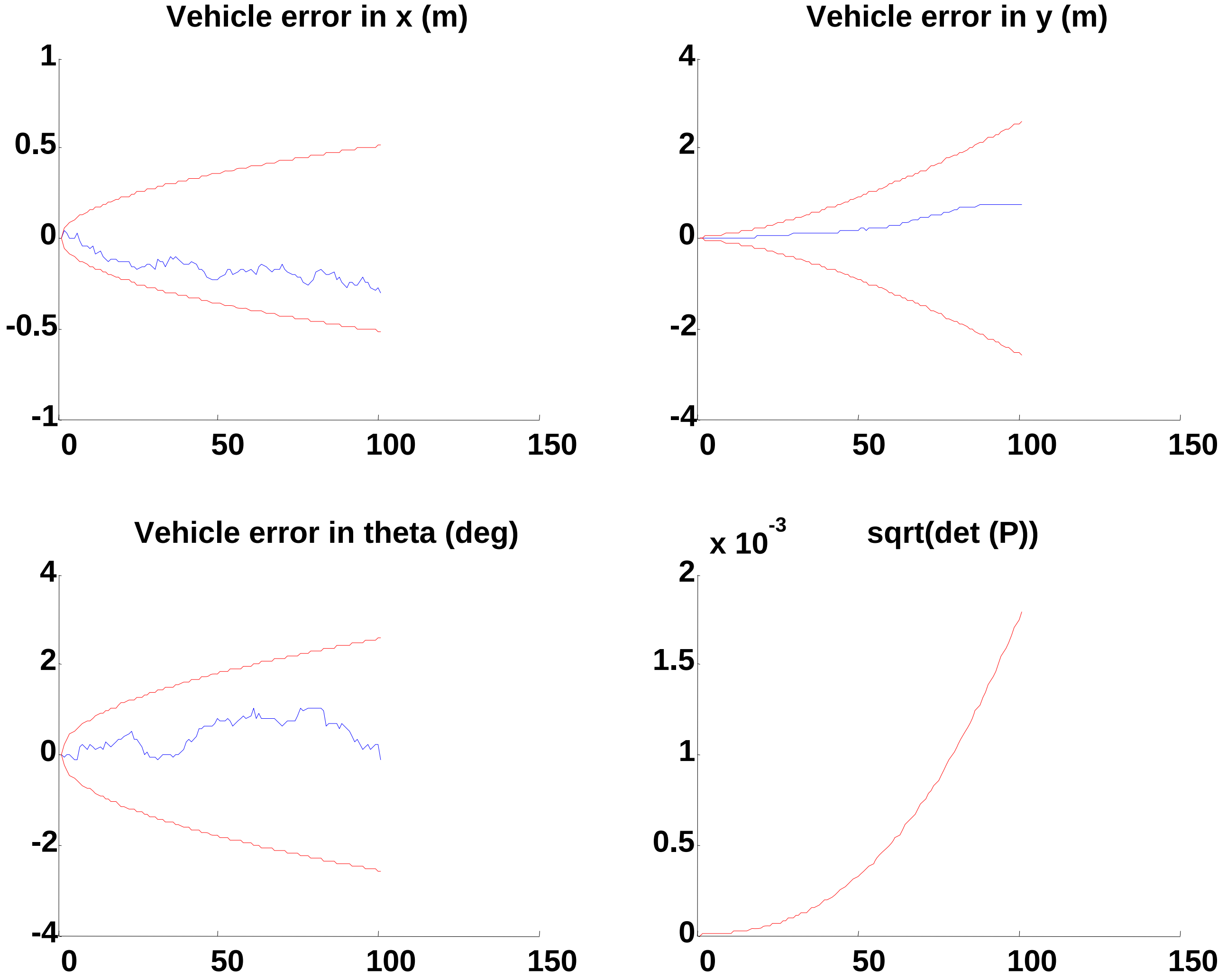
4. The loop closing problem

Why we do SLAM

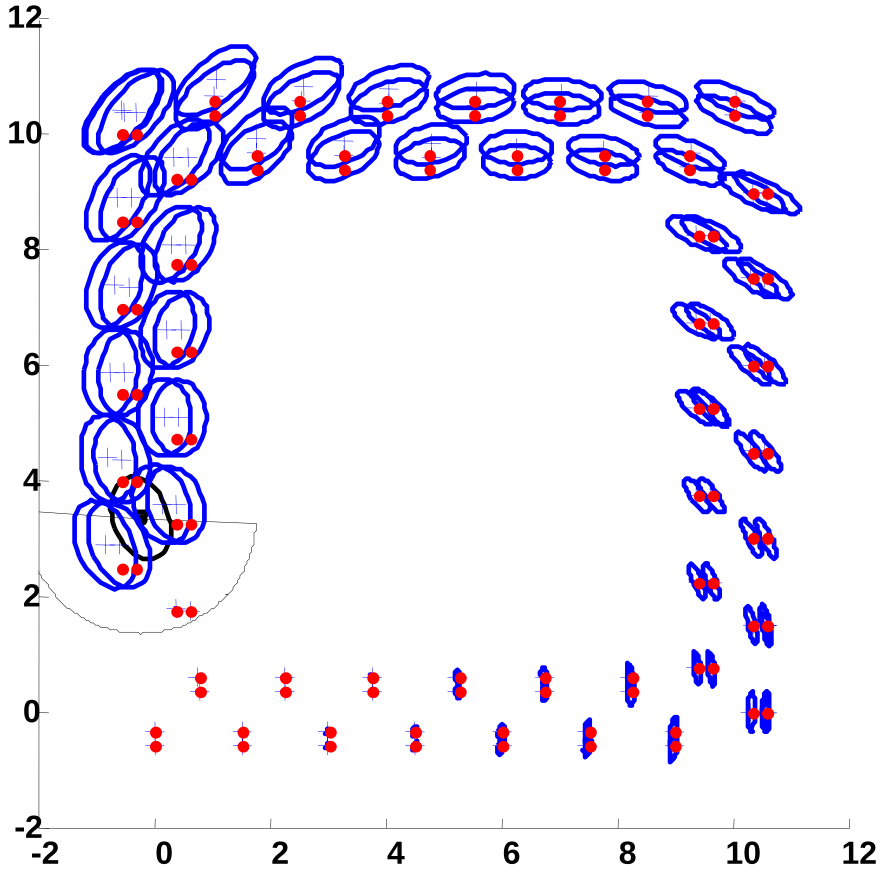


Dead-reckoning drift

Dead-reckoning, moving forward

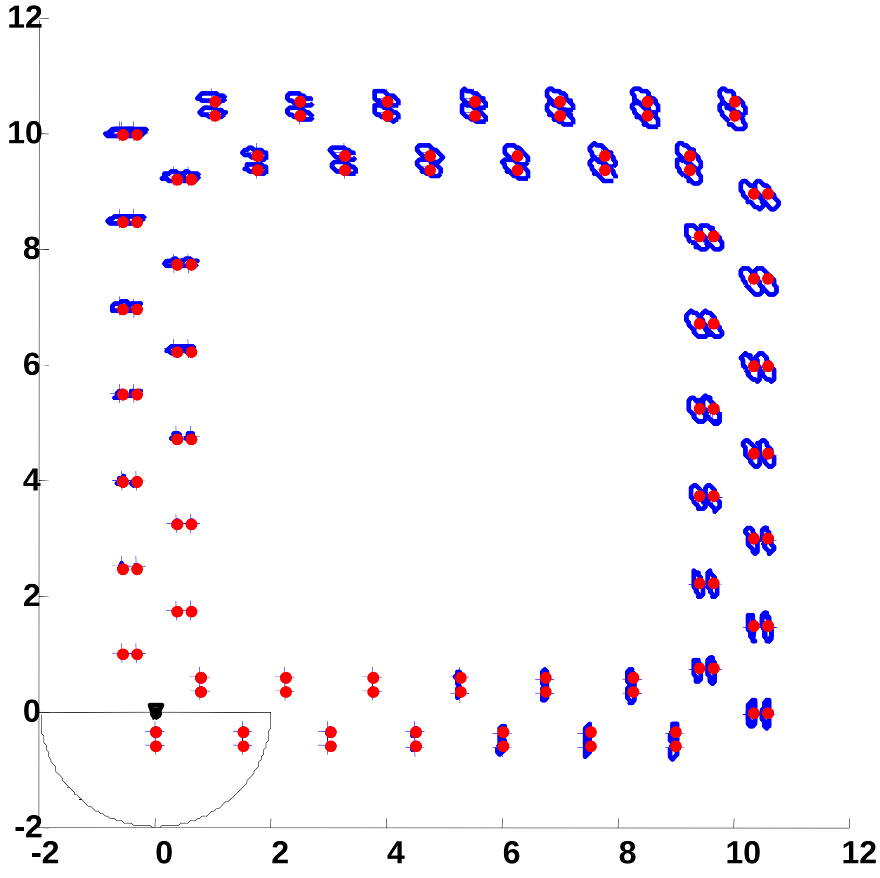


Bad news...



Uncertainty still grows!

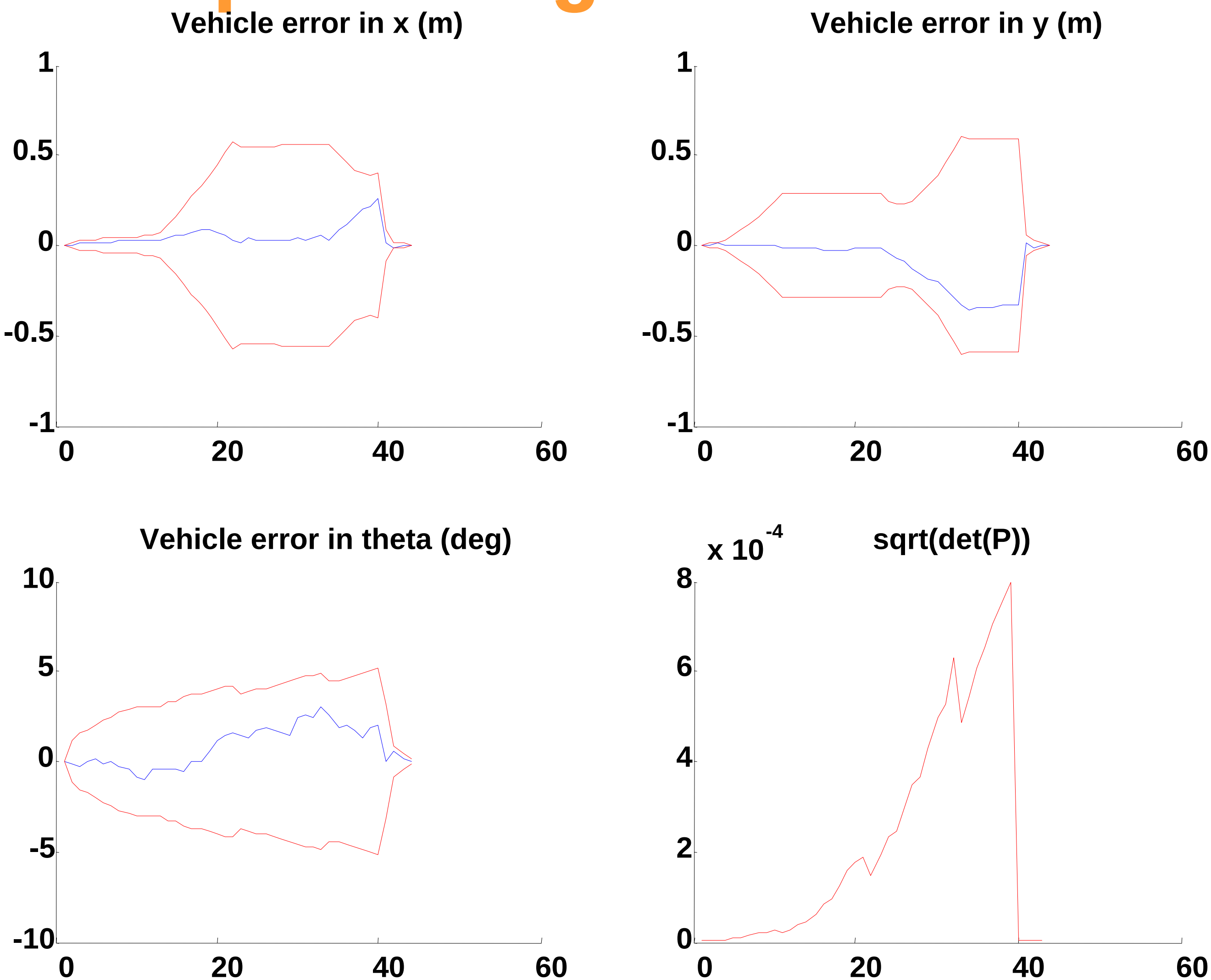
Good news!



Loop closing reduces uncertainty!

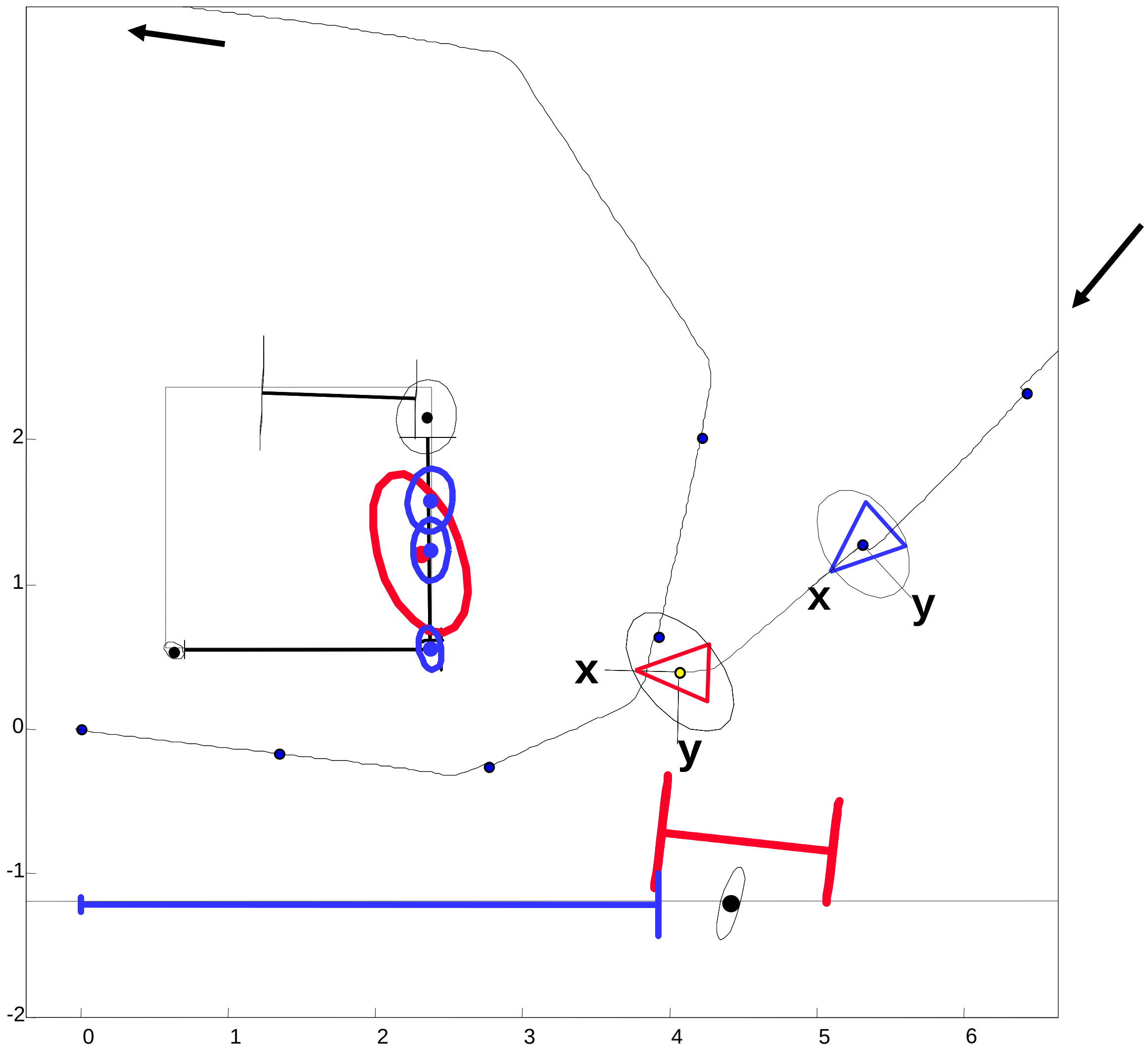
ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Loop closing in EKF-SLAM



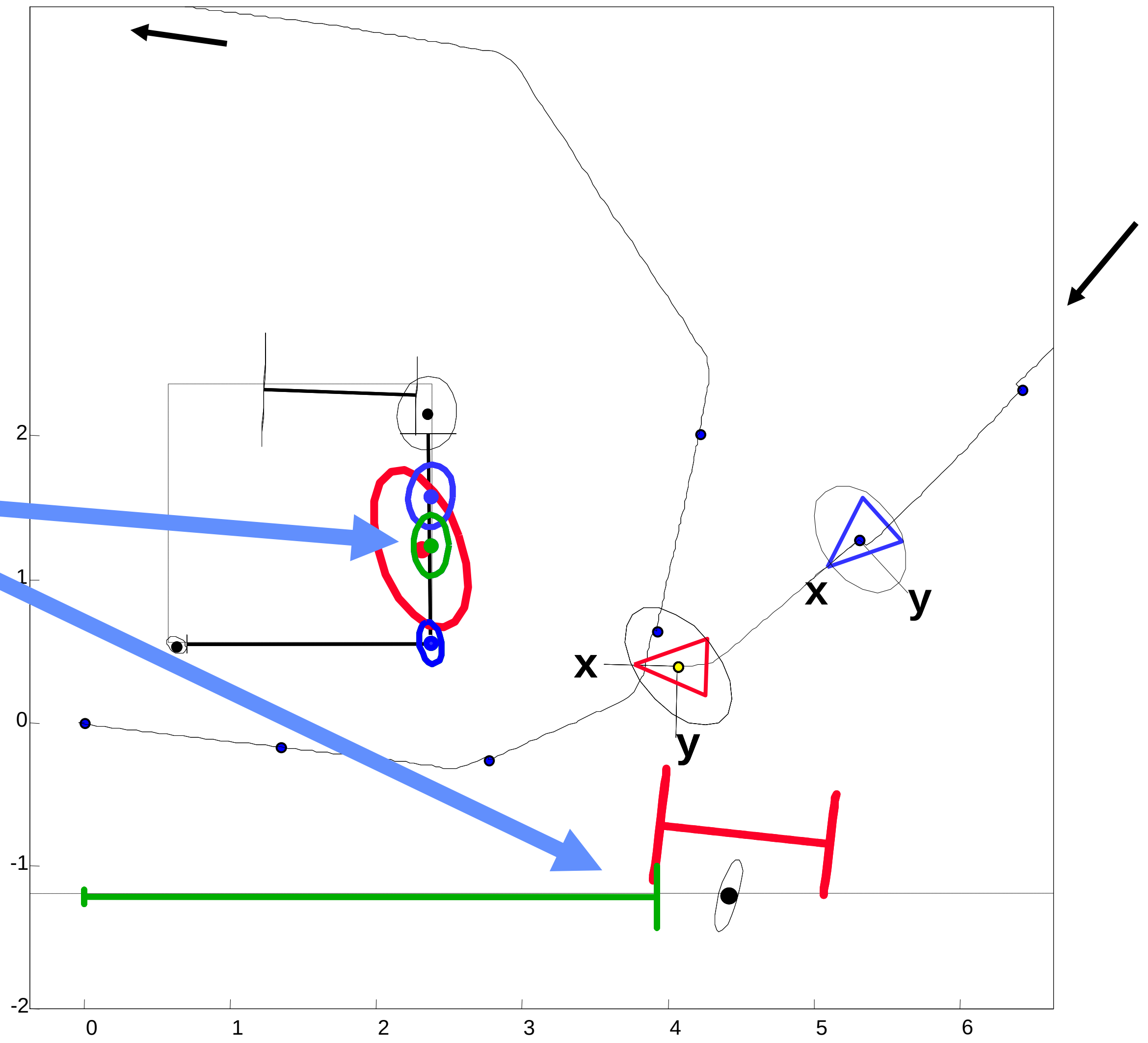
Loop closing reduces uncertainty!

Loop closing



Nearest Neighbor

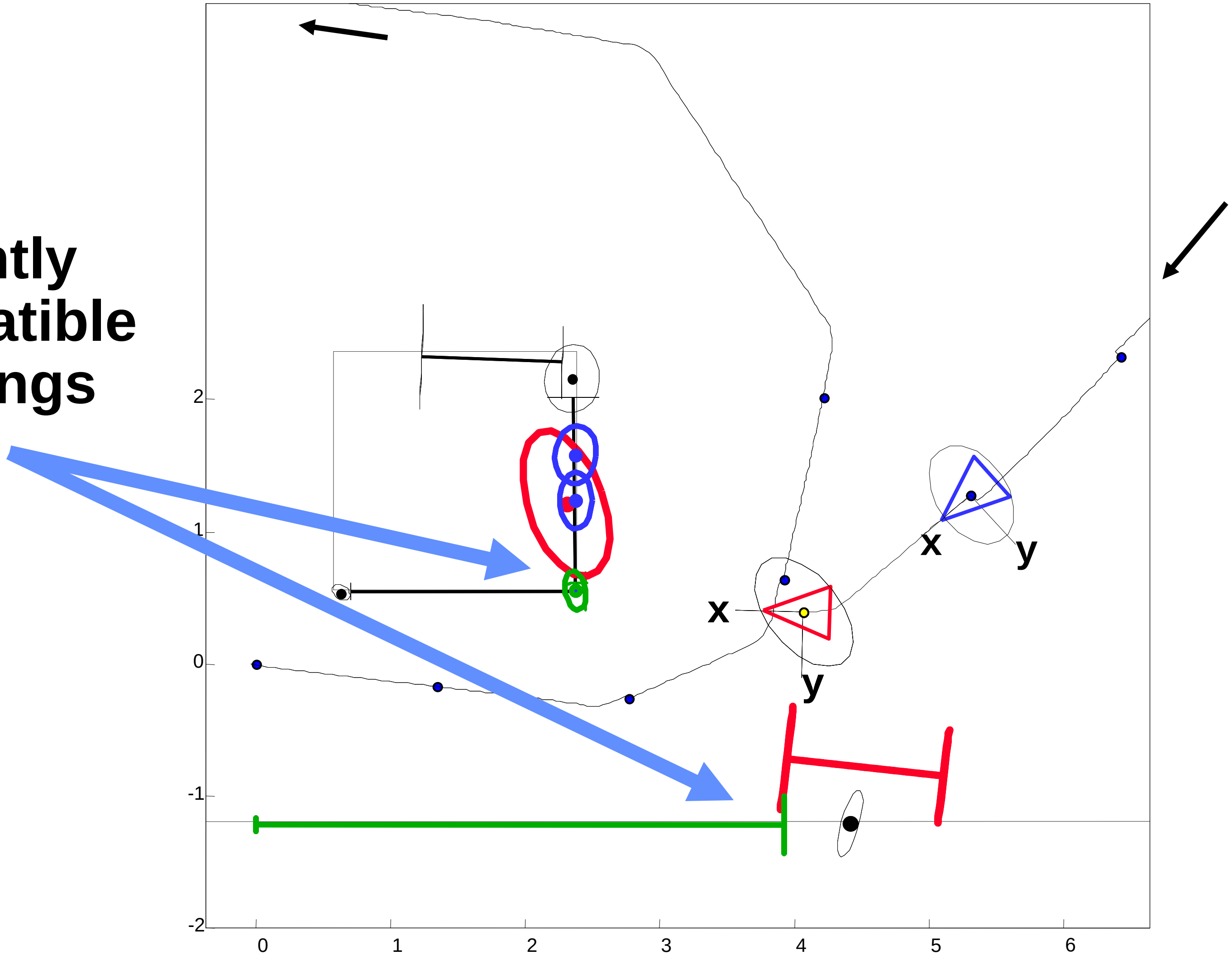
Individually compatible pairings



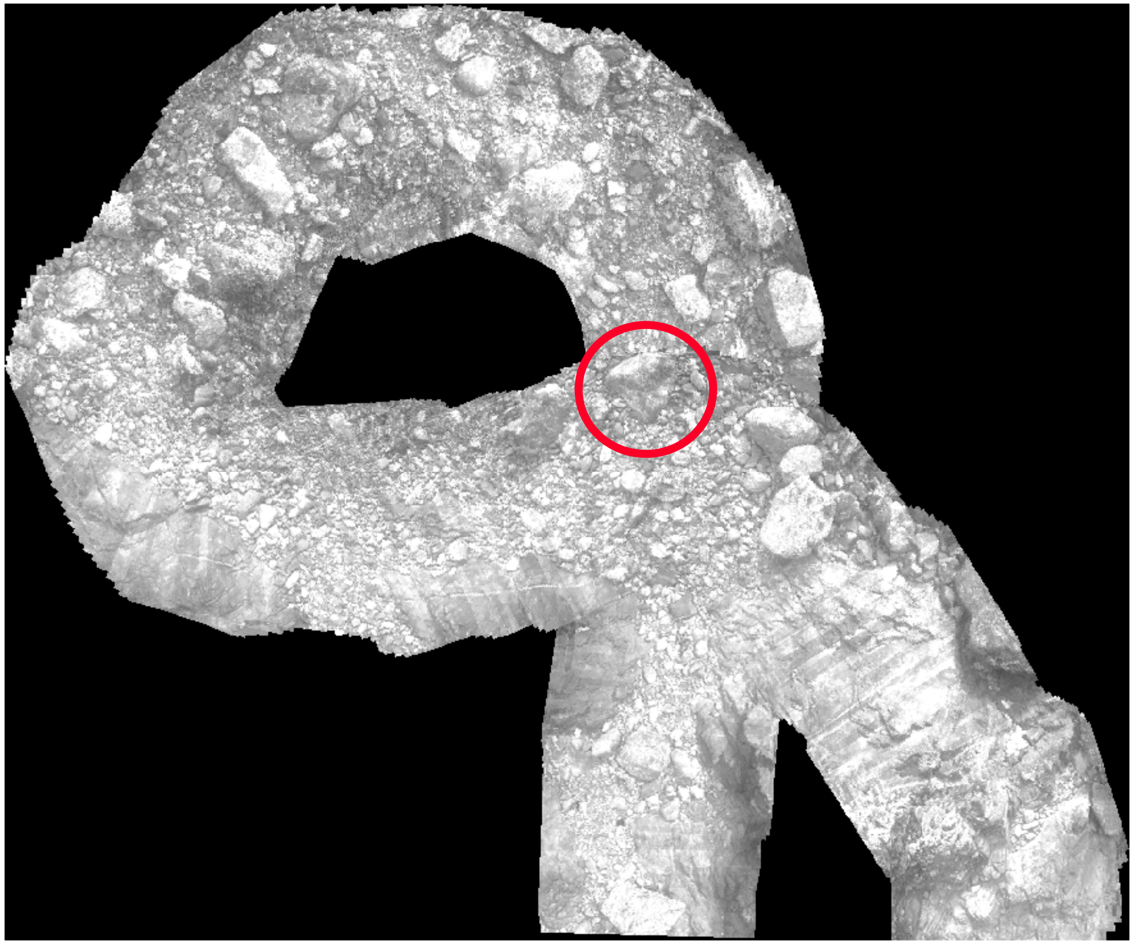
Wrong answer!

Joint Compatibility at work

Jointly compatible pairings

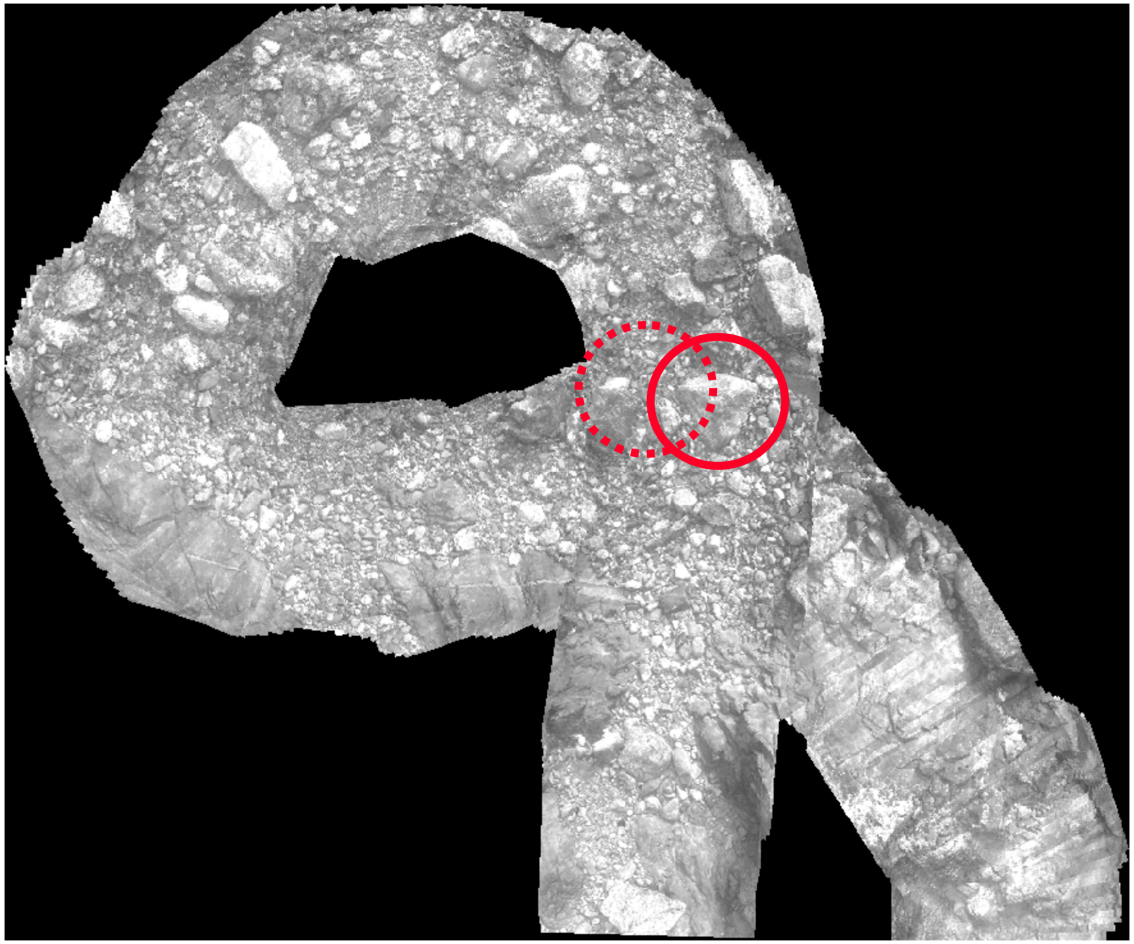


Loop closing in mosaicing use first



Joint work with R. García, University of Girona

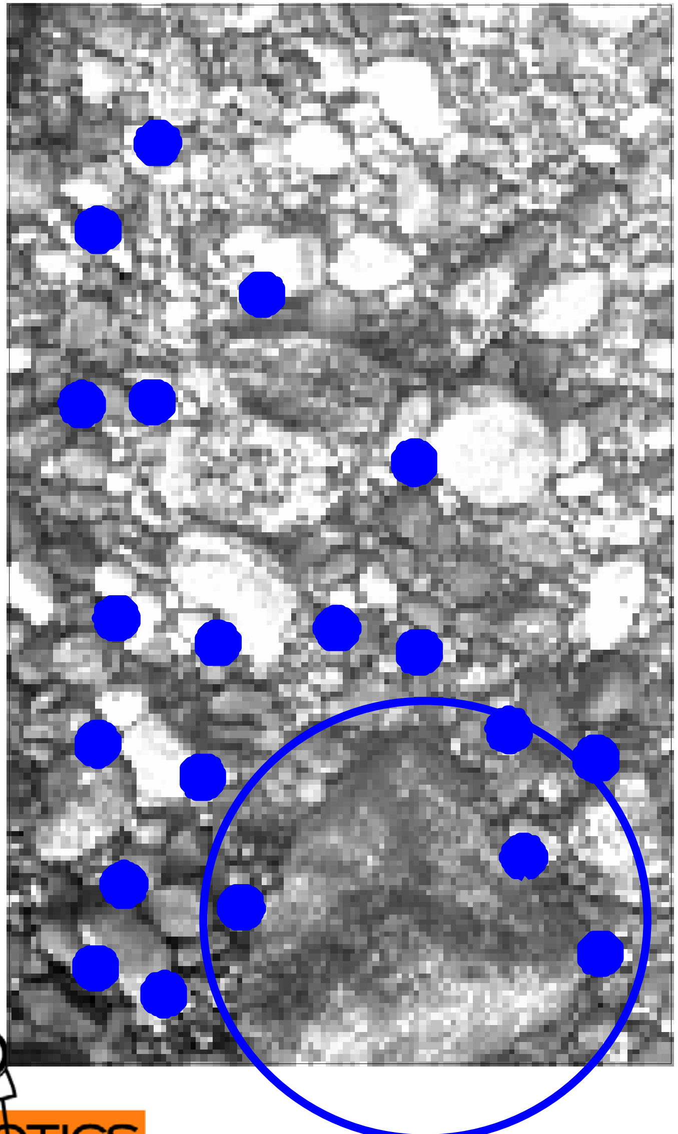
Loop closing in mosaicing: use Last



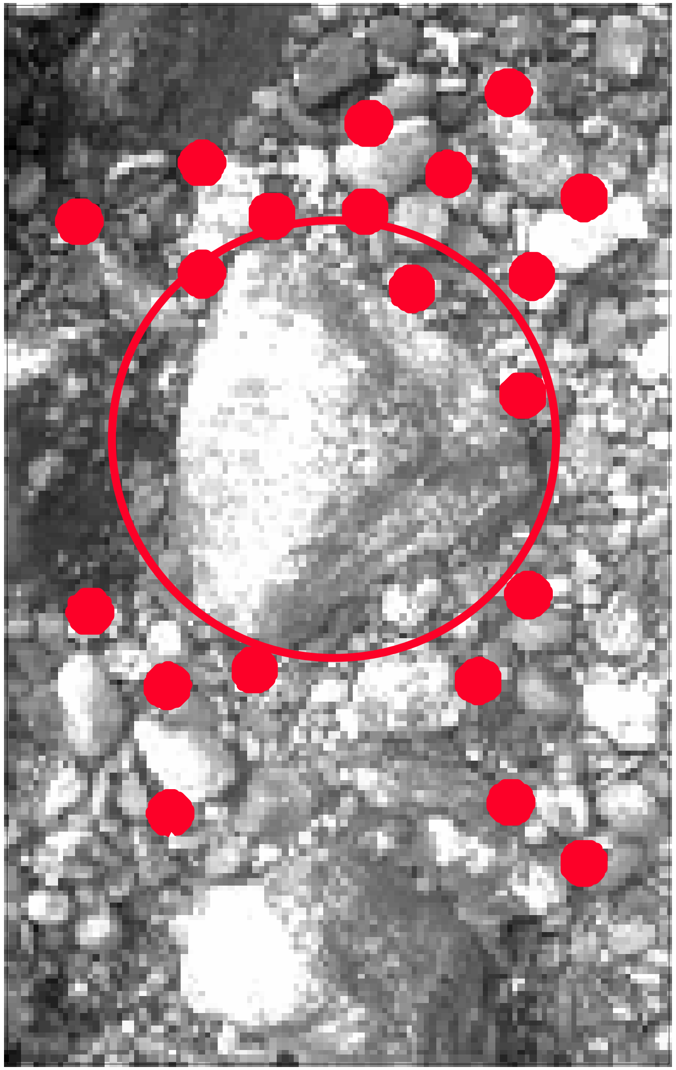
Sequential mosaicing is a form of odometry

The loop closing problem

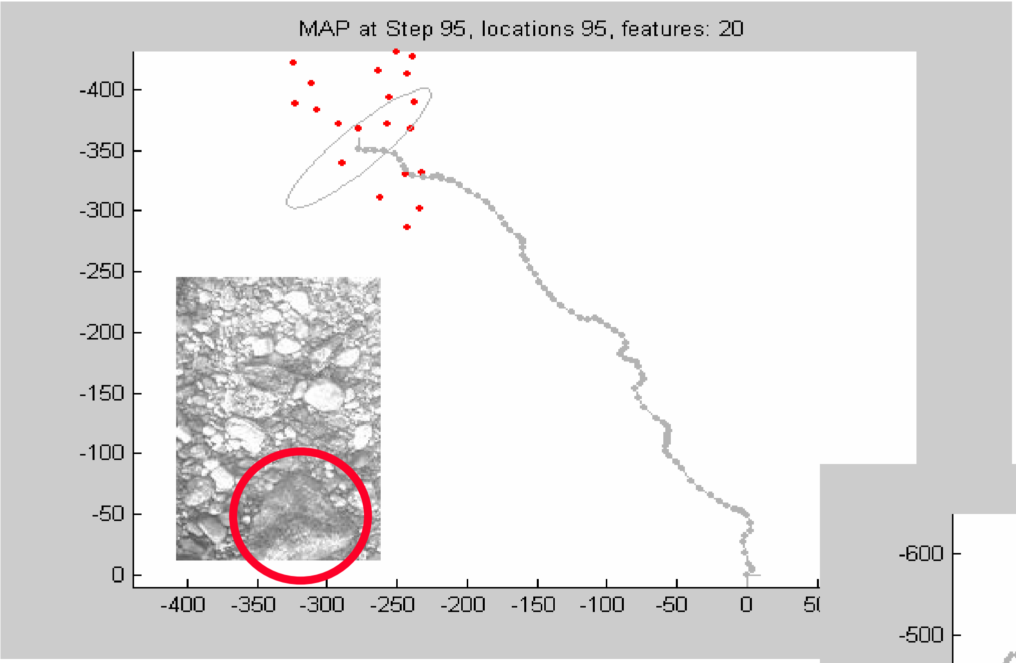
- Loop beginning

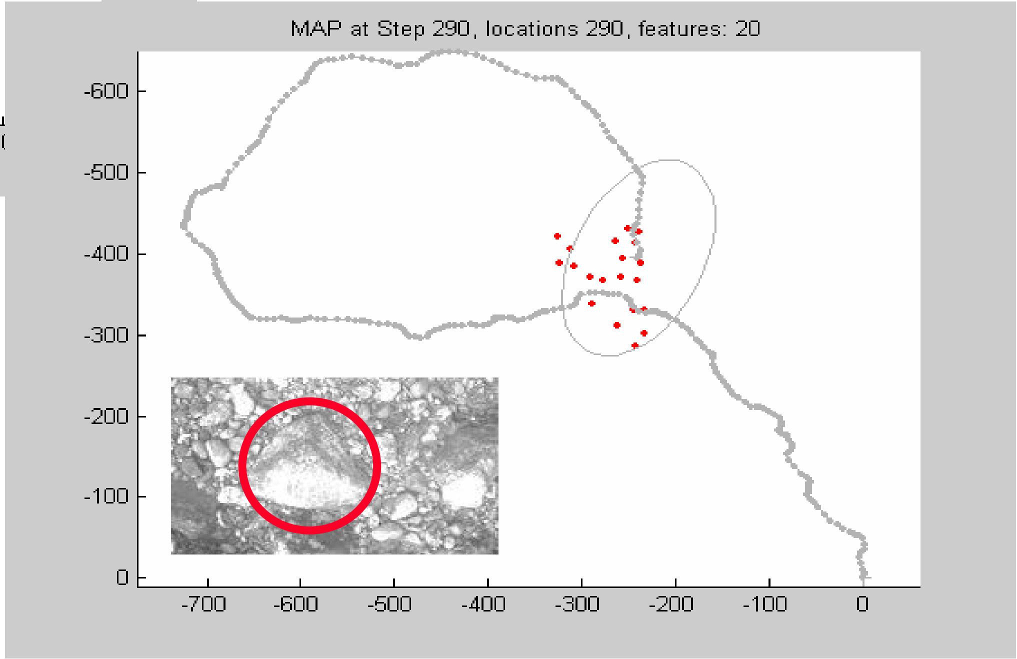


- Loop end



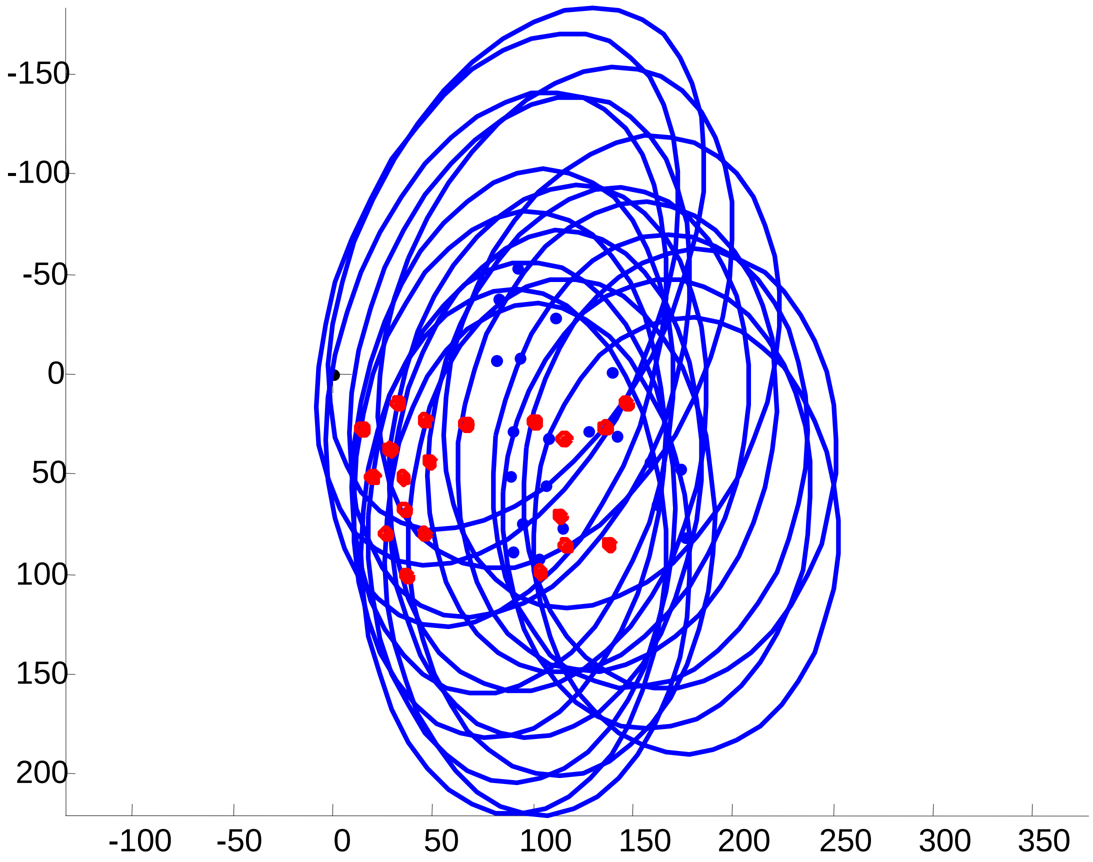
The loop closing problem





ROBOTICS
UNIVERSIDAD DE ZARAGOZA

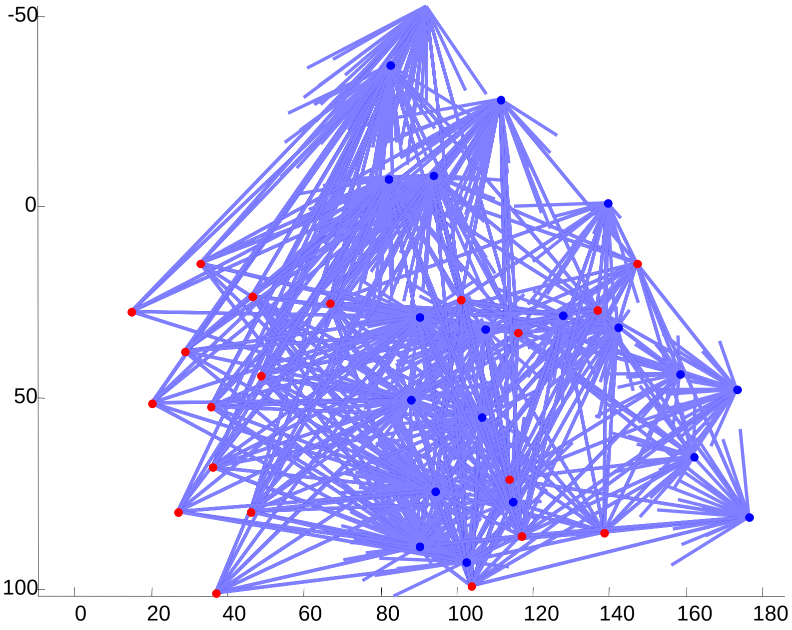
The loop closing problem



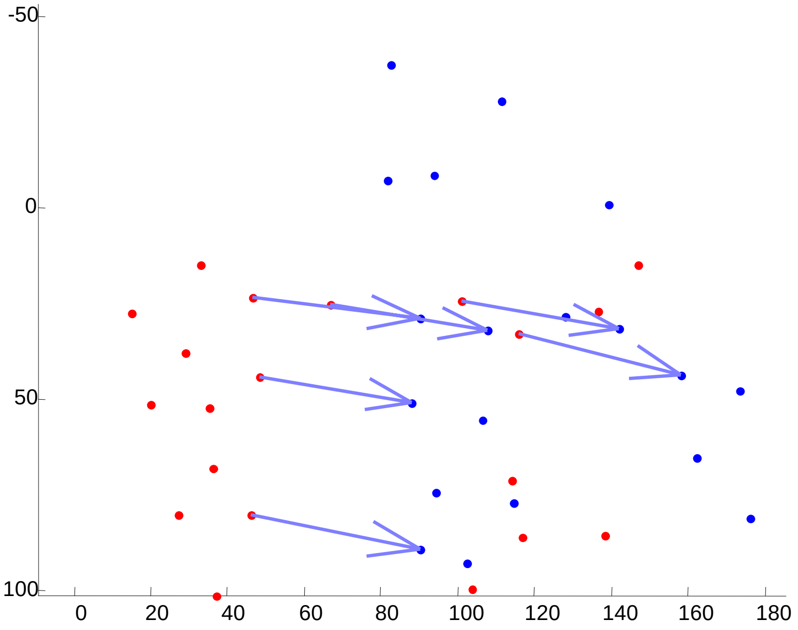
Measurements (red) and predicted features (blue)

The loop closing problem

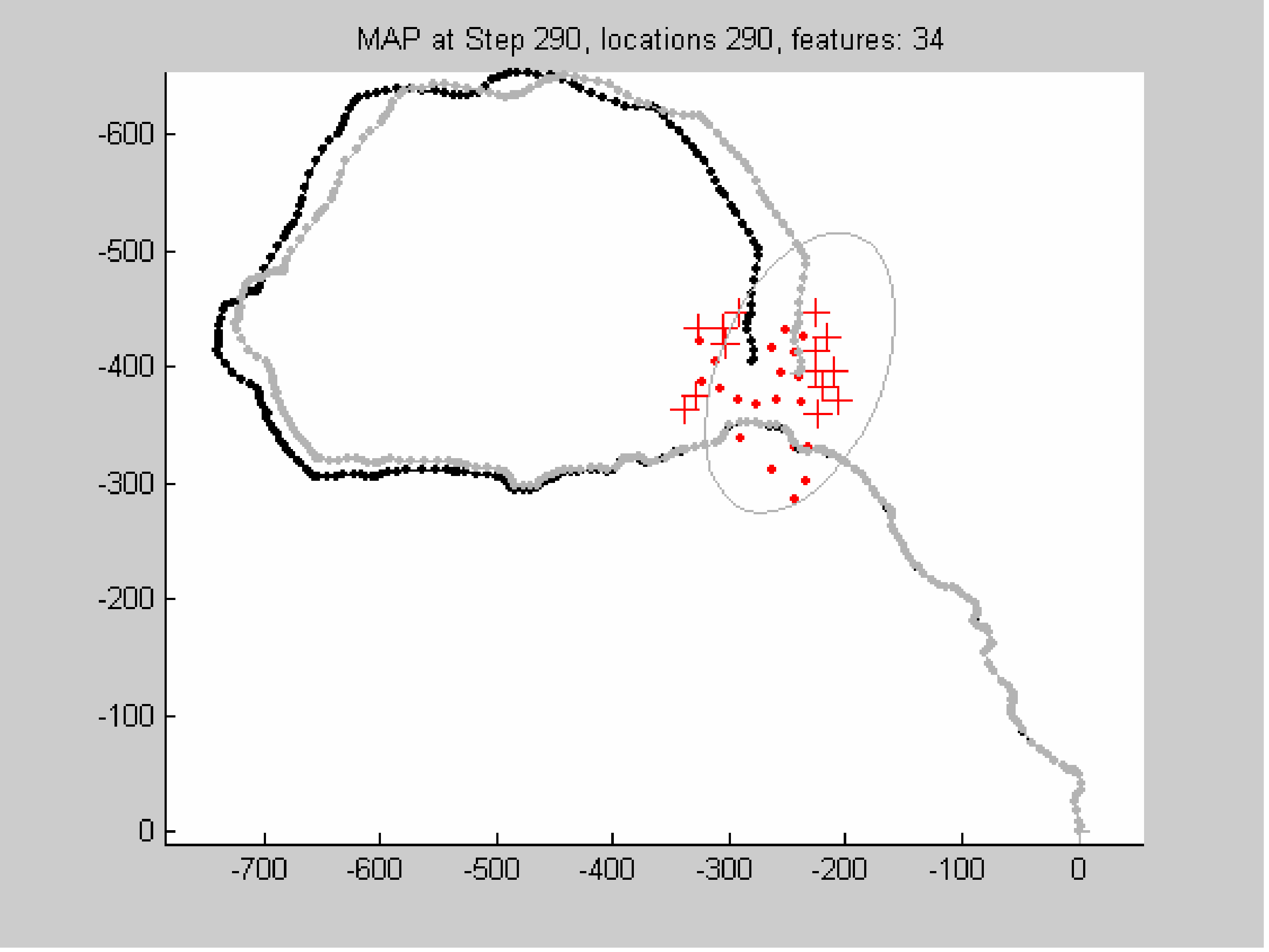
- Individual compatibility



- Joint Compatibility



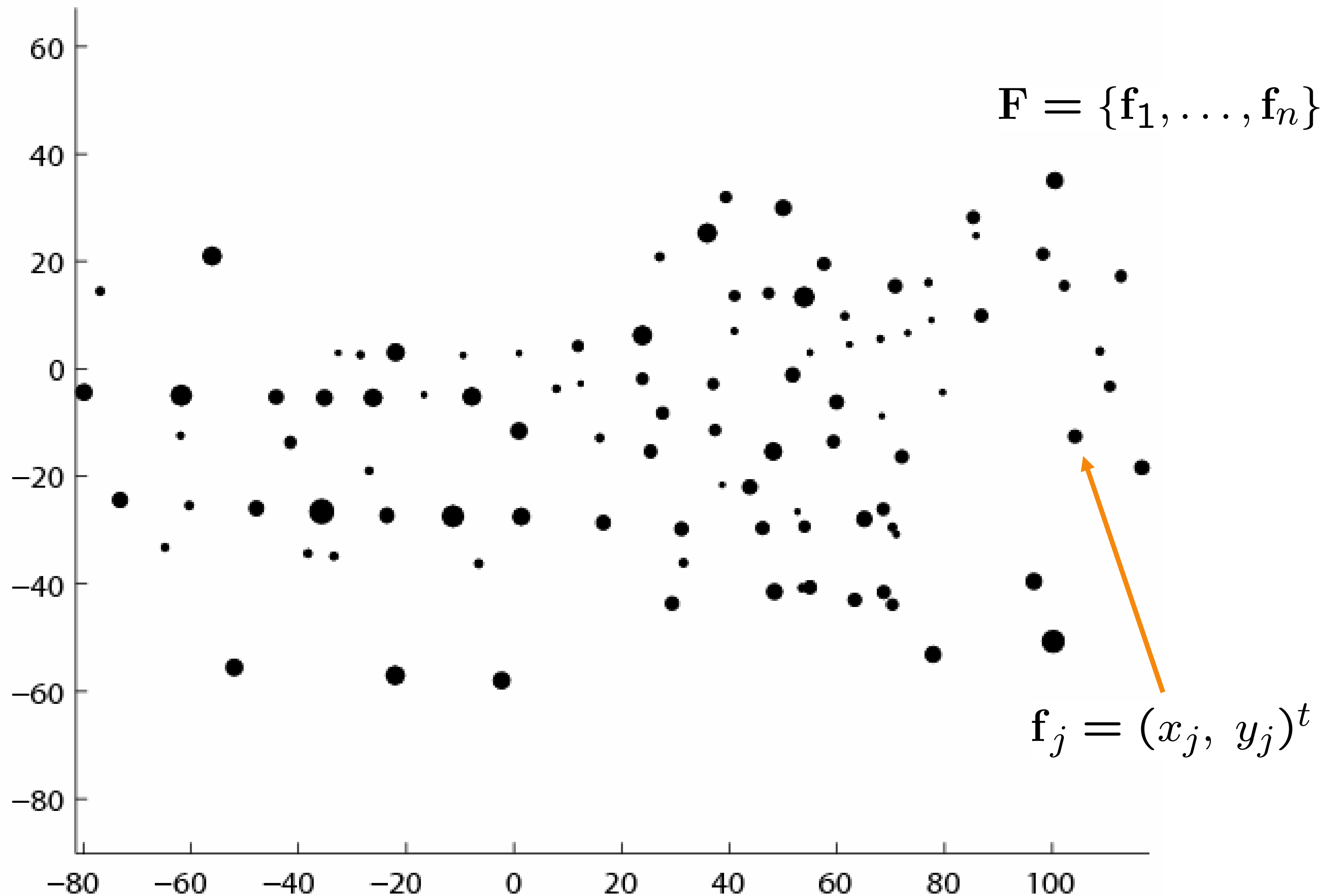
The loop closing problem



5. The Global Localization problem

Global Localization

- Robot placed in a previously mapped environment

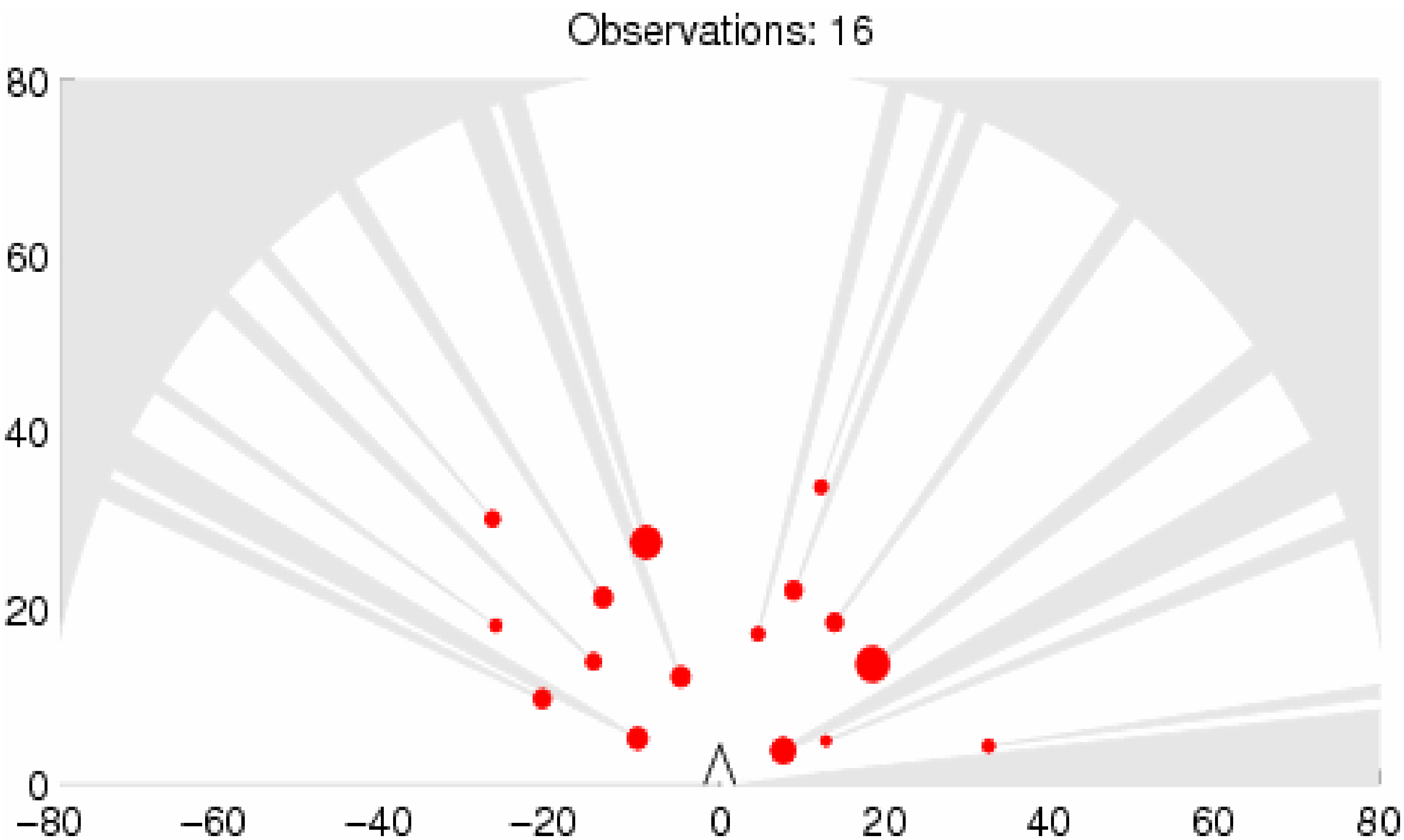


http://www.acfr.usyd.edu.au/homepages/academic/enebot/victoria_park.htm

Problem Definition

- On-board sensor obtains *m* measurements:

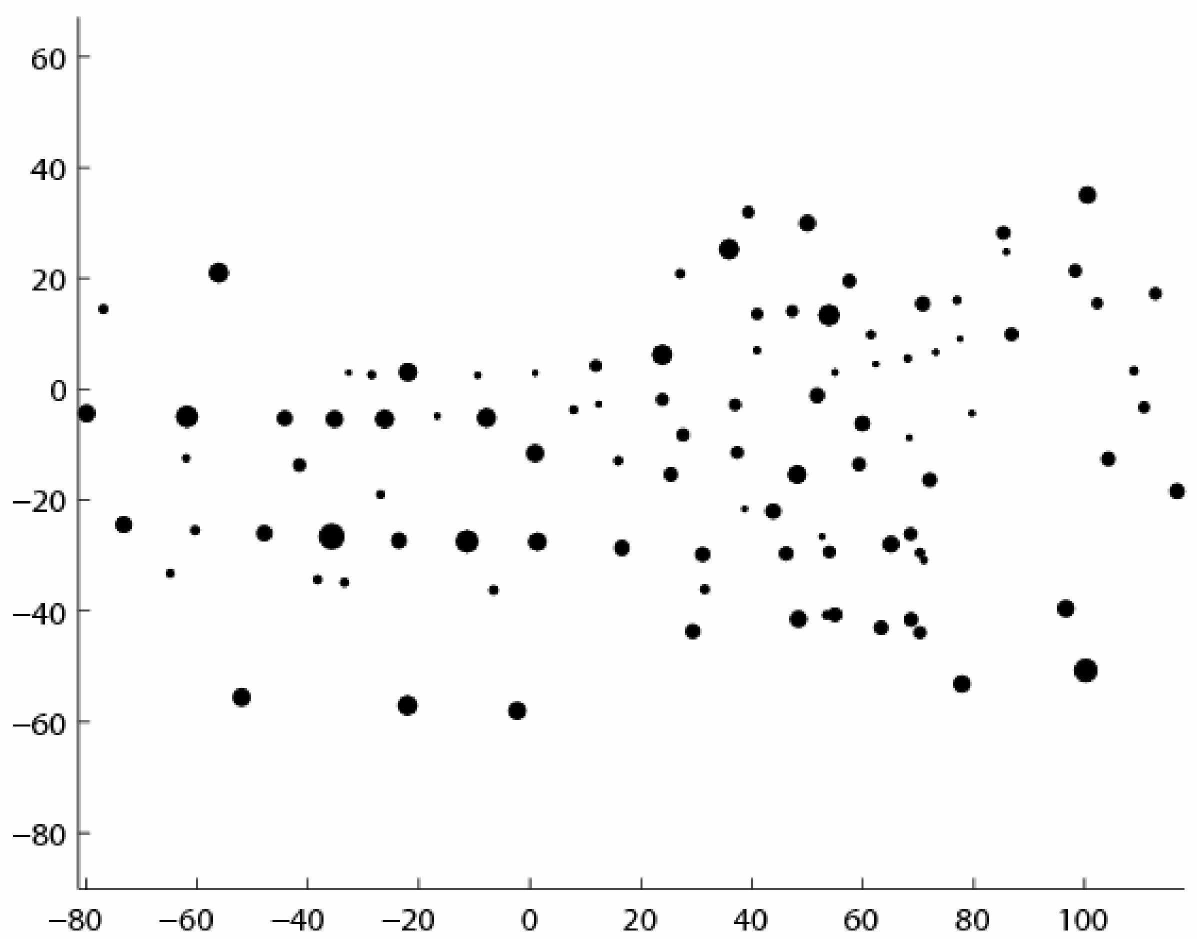
$$\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$$

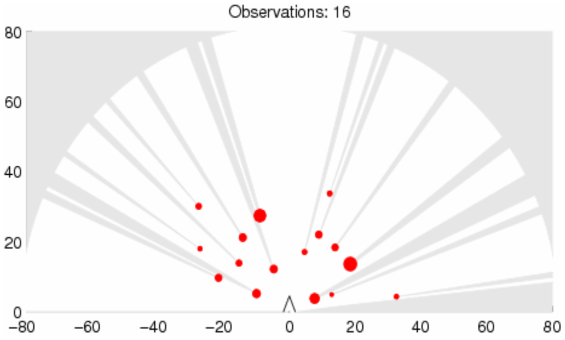


http://www.acfr.usyd.edu.au/homepages/academic/enebot/victoria_park.htm

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

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

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

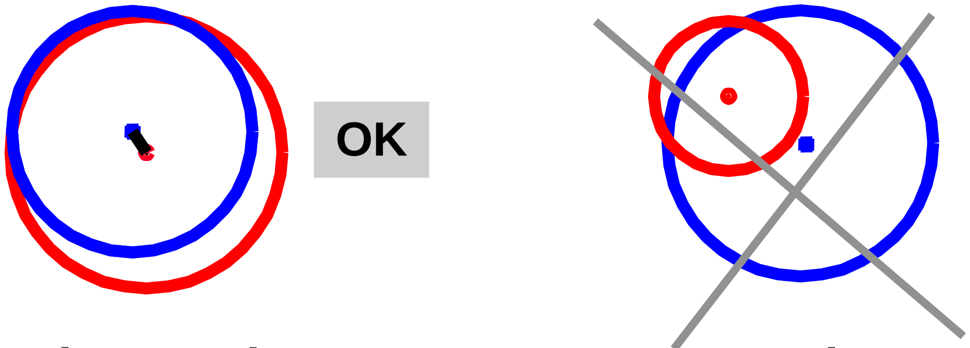
No vehicle location

- **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

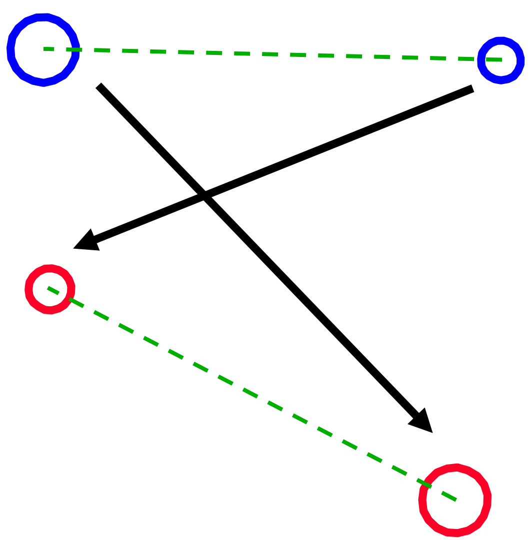
ROBOTICS UNIVERSIDAD DE ZARAGOZA

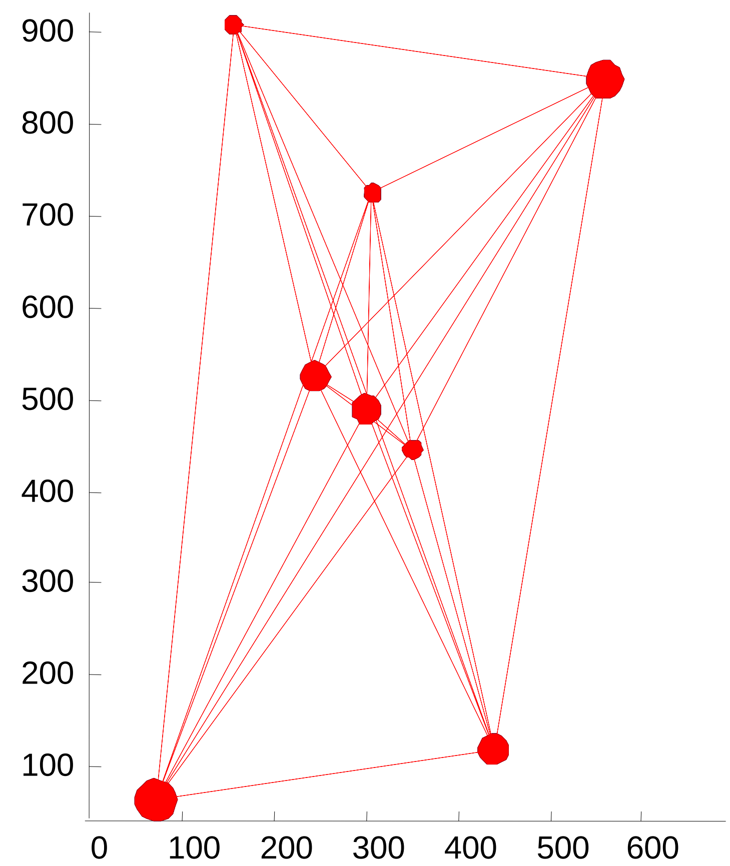
No vehicle location

- **Binary constraints:**

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

distances between points:

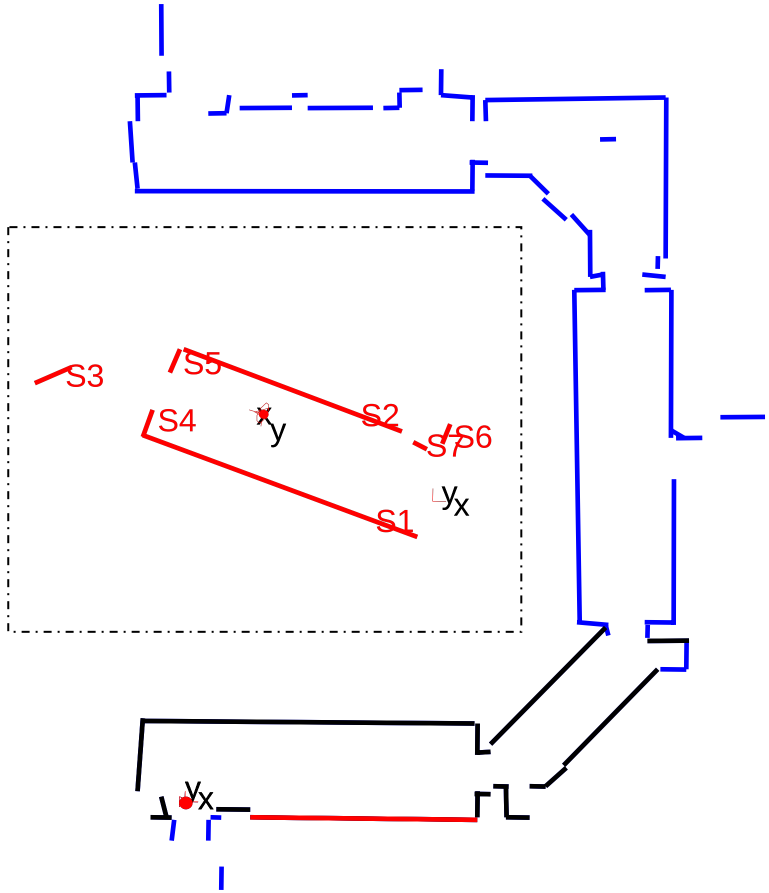


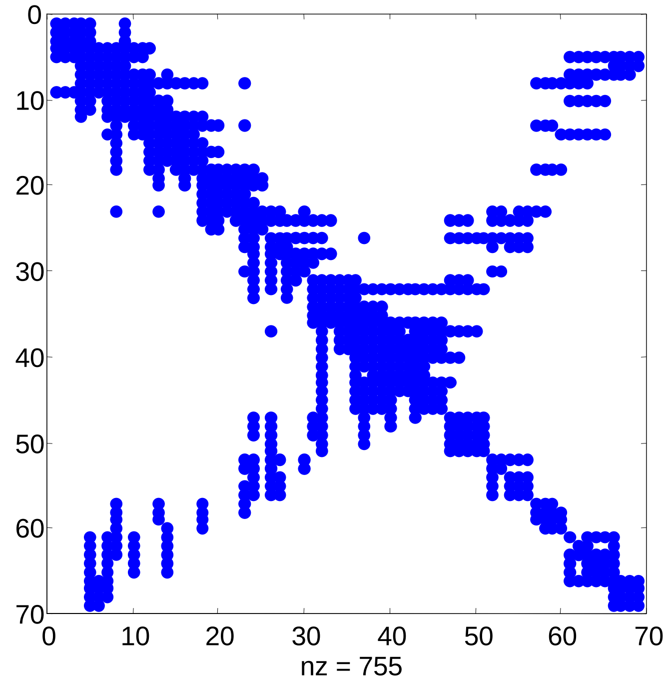


$$\frac{m\,(m-1)}{2}\text{constraints}$$

Locality

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



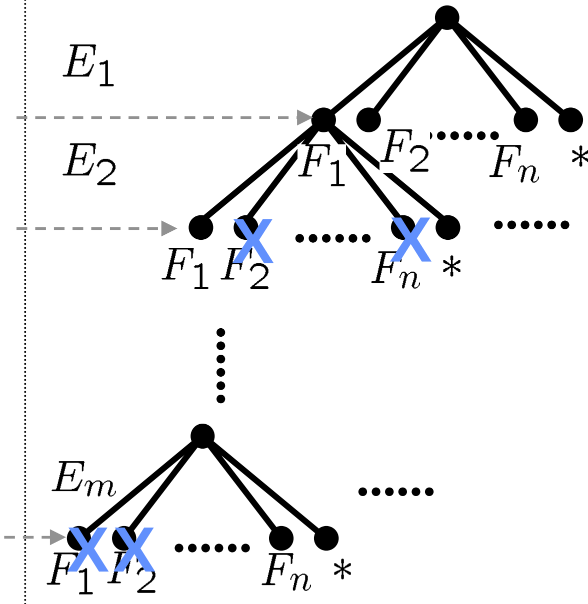


Covisibility matrix

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

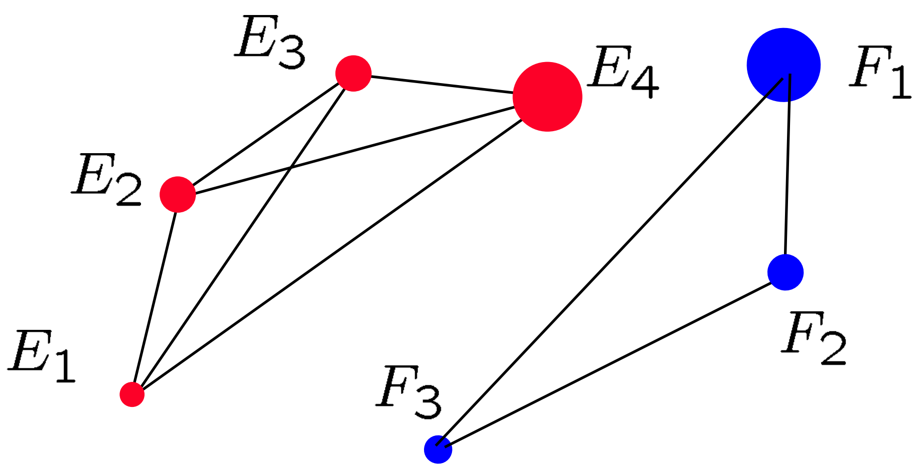
Algorithm 1: Geometric Constraints Branch and Bound (Grimson, 1990)

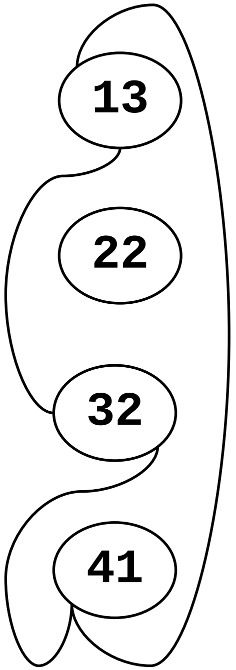
```
procedure GCBB (H, i):
if i > m -- leaf node?
    if pairings(H) > pairings(Best) -- did better?
        estimate_location(H)
        if joint_compatibility(H)
            Best = H
        fi
    fi
else
    for j in {1...n}
        if unary(i, j) ∧ binary(i, j, H)
            GCBB([H j], i + 1)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



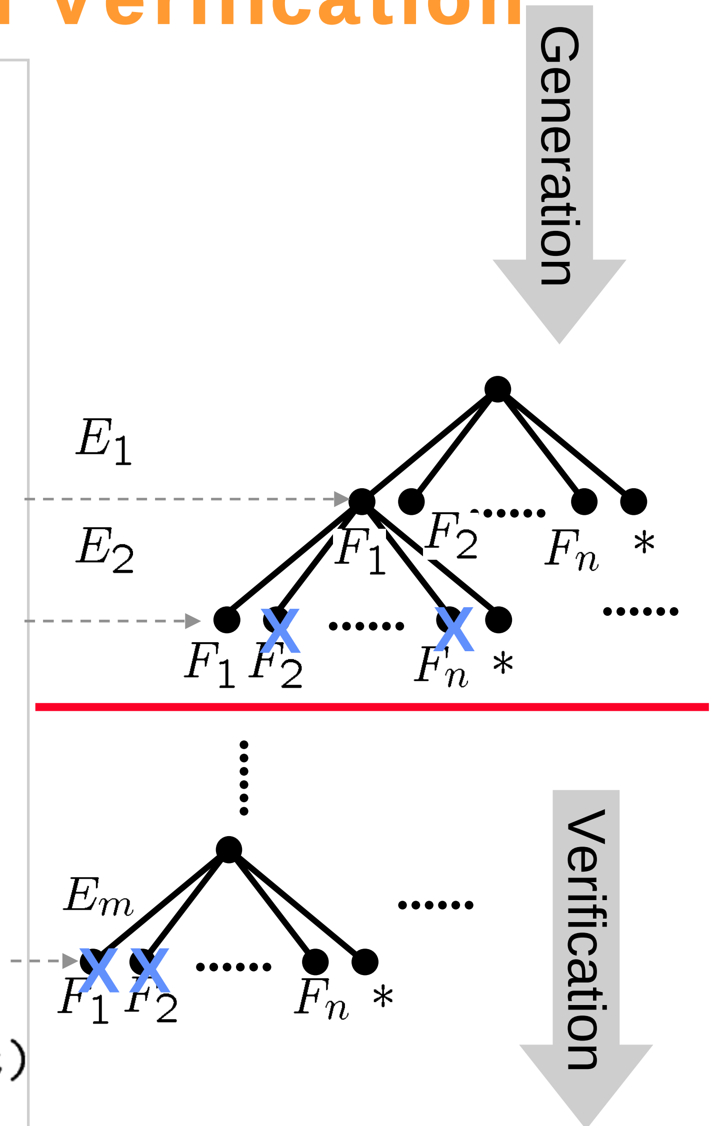


- Carrahan, Pardalos (1990)

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
```

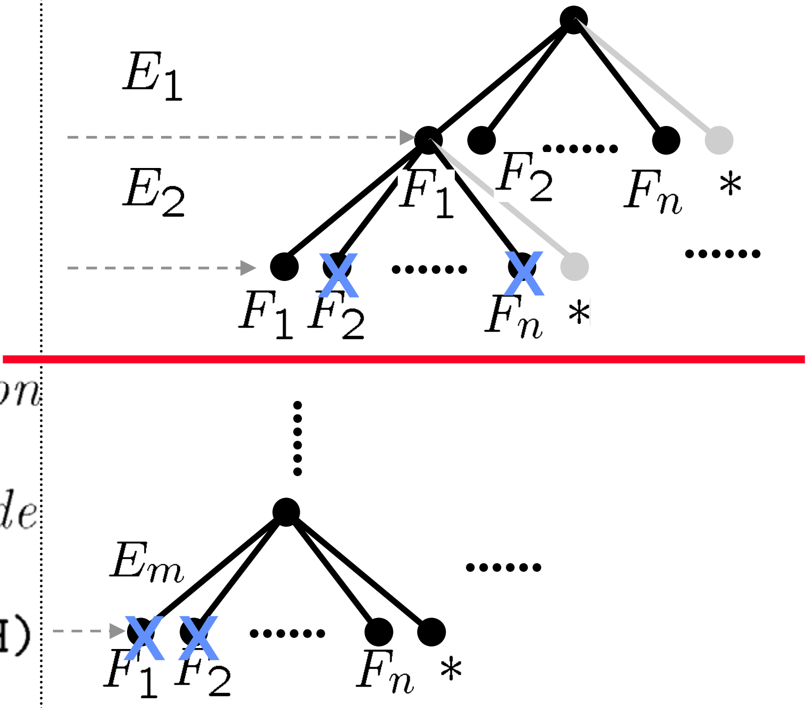


ROBOTICS UNIVERSIDAD DE ZARAGOZA

Algorithm 4: RANSAC

procedure RS (H, i):

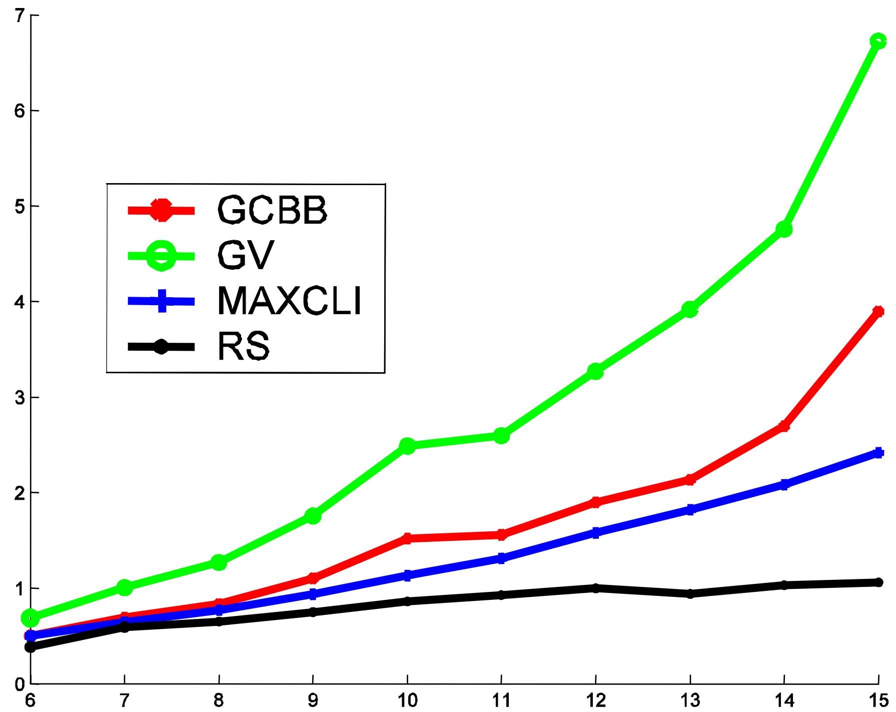
```
if i > m
   if pairings(H) > pairings(Best)
      Best = H
   fi
elseif pairings(H) == 3
   estimate_location_(H)
   if joint_compatibility(H)
      JCBB(H, i) -- hypohtesis verification
   fi
else -- branch and bound without star node
   for j in {1...n}
      if unary(i, j) ∧ binary(i, j, H)
 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

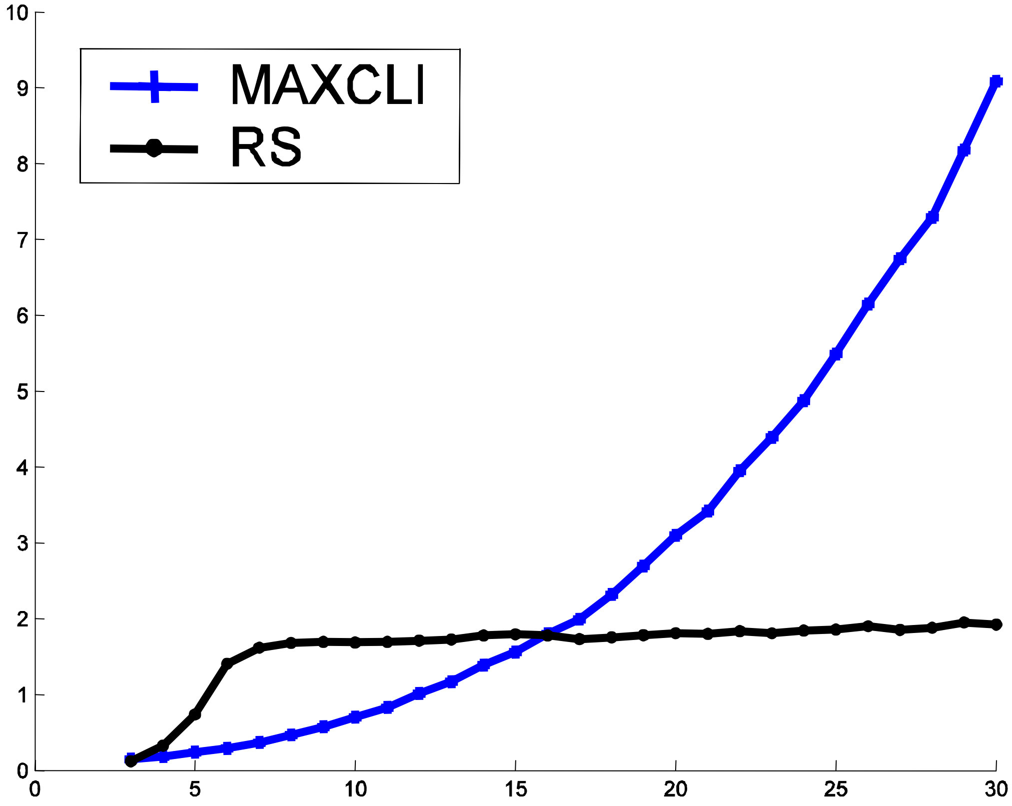


ROBOTICS
UNIVERSIDAD DE ZARAGOZA

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

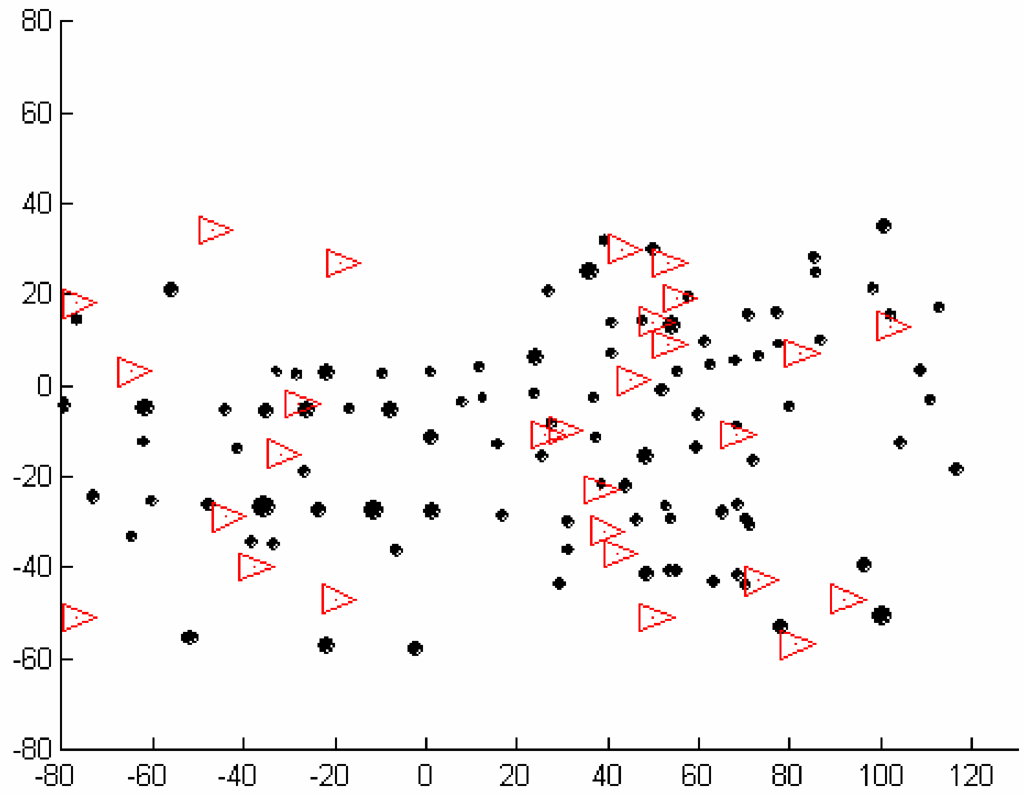
In configuration space

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:

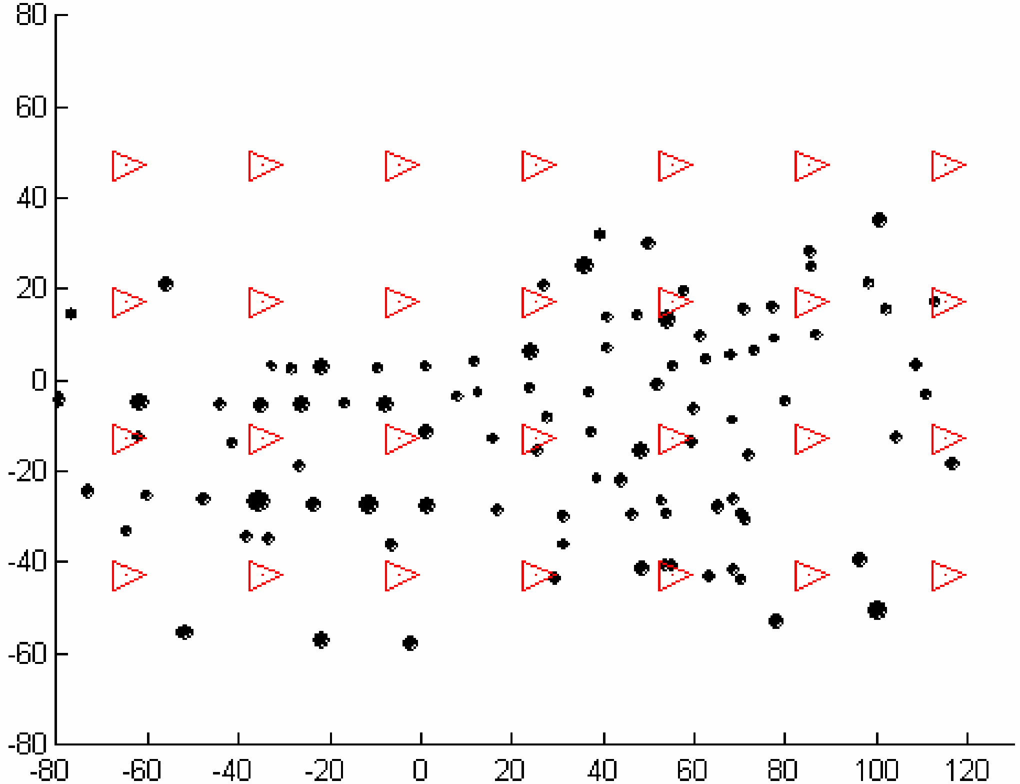
$votes = 0$
for each hypothesis $\mathbf{x} \in \mathbf{X}$ **do**
  for each measurement $\mathbf{z}_i \in \mathbf{Z}$ **do**
    $\mathbf{F}_i = \texttt{predict\_features}(\mathbf{x}, \mathbf{z}_i)$
    if $\texttt{any\_compatible\_feature}(\mathbf{F}_i, \mathbf{F})$ **then**
      $votes(\mathbf{x}) = votes(\mathbf{x}) + 1$
    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:

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

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

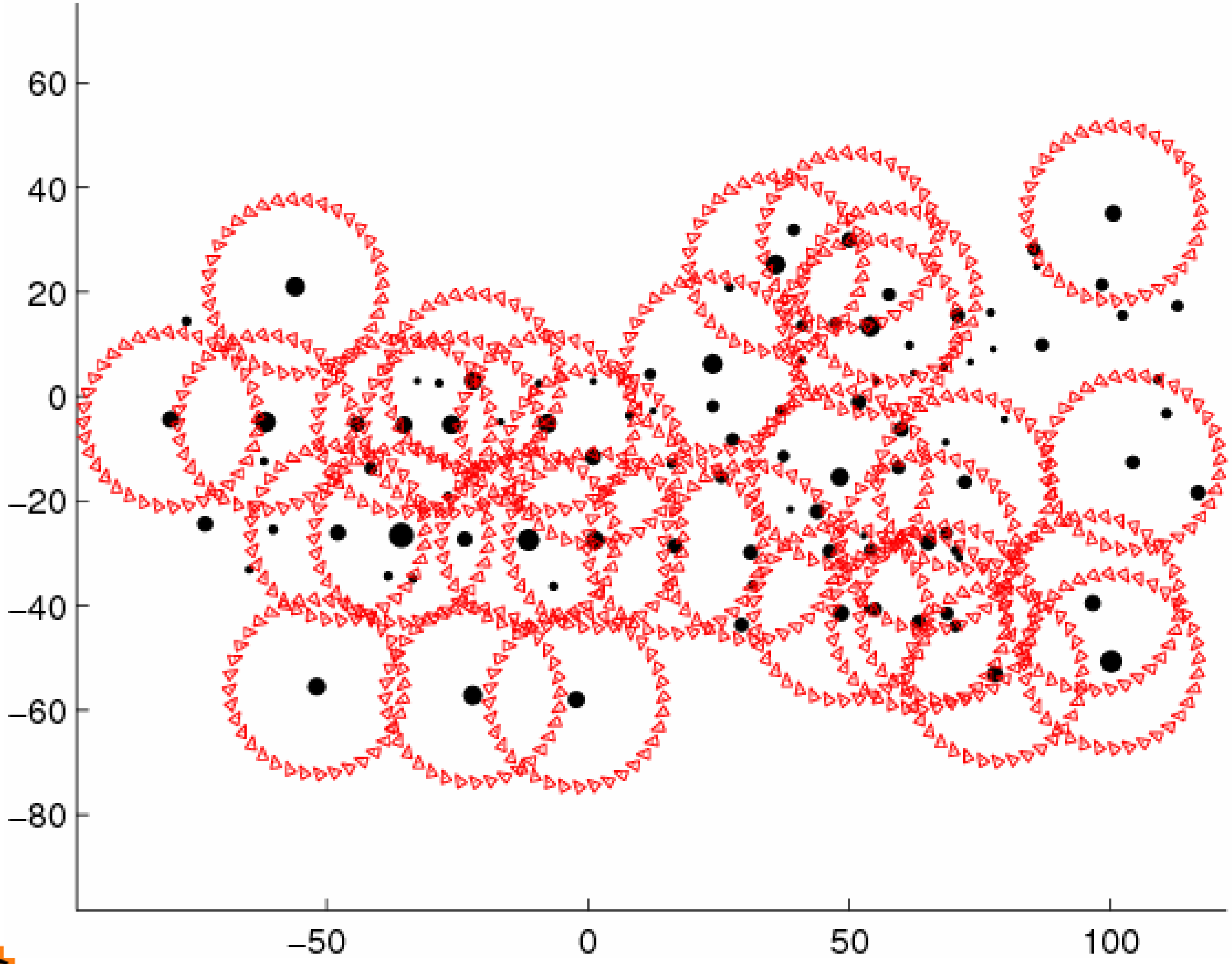


- Markov Localization

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Voting in configuration space

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



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Alternative 2: pairing-driven

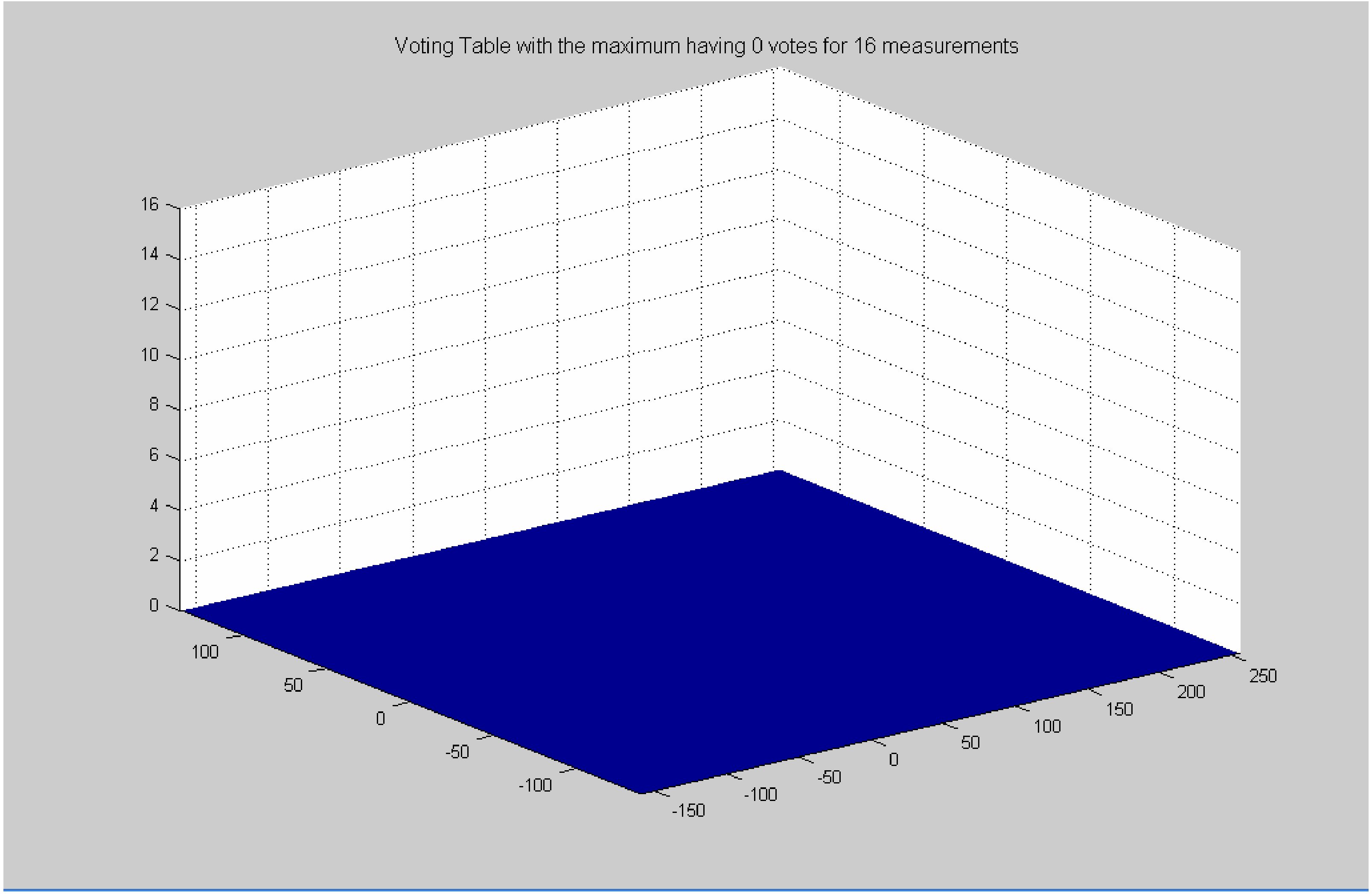
- Consider each **measurement-feature pairing** in turn

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

Results

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



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

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.

UNIVERSIDAD DE ZARAGOZA

Conclusions

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

6. Appendix

Details of the EKF SLAM algorithm

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:

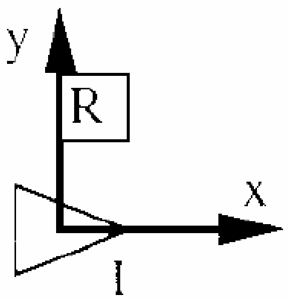
$$\mathbf{x}_L^B = \begin{bmatrix} \rho_2 \\ \theta_2 \end{bmatrix}$$

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

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

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

Vehicle motion in 2D



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

$$\mathbf{x}^A_B = \begin{bmatrix} x_1 \\ y_1 \\ \phi_1 \end{bmatrix} \qquad \mathbf{x}^B_C = \begin{bmatrix} x_2 \\ y_2 \\ \phi_2 \end{bmatrix}$$

Composition:

$$\mathbf{x}^A_C = \mathbf{x}^A_B \oplus \mathbf{x}^B_C = \begin{bmatrix} x_1 + x_2 \cos\phi_1 - y_2 \sin\phi_1 \\ y_1 + x_2 \sin\phi_1 + y_2 \cos\phi_1 \\ \phi_1 + \phi_2 \end{bmatrix}$$

Inversion:

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

Odometry in 2D

Jacobians:

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

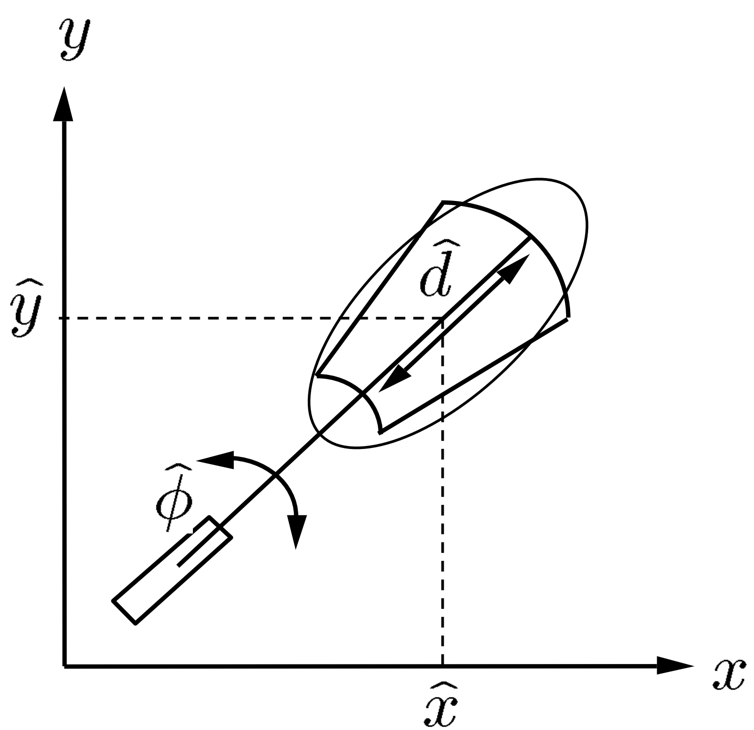
$$J_{2\oplus}\{\mathbf{x}_B^A, \mathbf{x}_C^B\} = \left.\frac{\partial\left(\mathbf{x}_B^A \oplus \mathbf{x}_C^B\right)}{\partial \mathbf{x}_C^B}\right|_{(\hat{\mathbf{x}}_B^A, \hat{\mathbf{x}}_C^B)} = \begin{bmatrix} \cos\phi_1 & -\sin\phi_1 & 0 \\ \sin\phi_1 & \cos\phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Sensor measurements

- In polar coordinates:



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

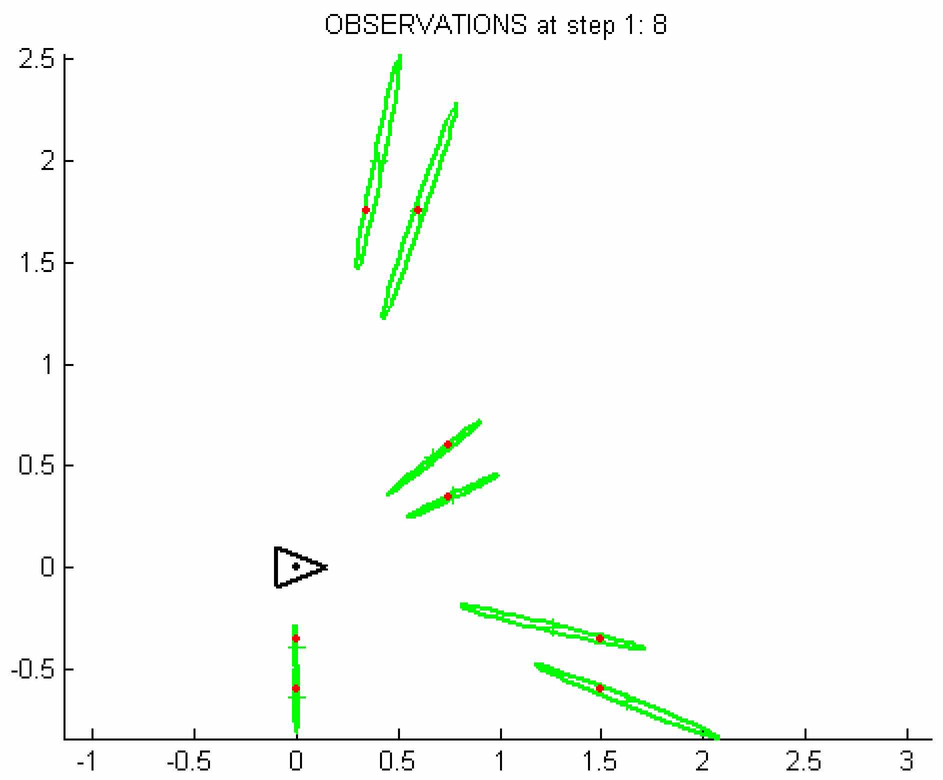
- In cartesian coordinates:

$$\hat{x} = \hat{d}\cos\hat{\phi}$$
$$\hat{y} = \hat{d}\sin\hat{\phi}$$

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

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

The basic EKF SLAM Algorithms



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):

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

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:

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

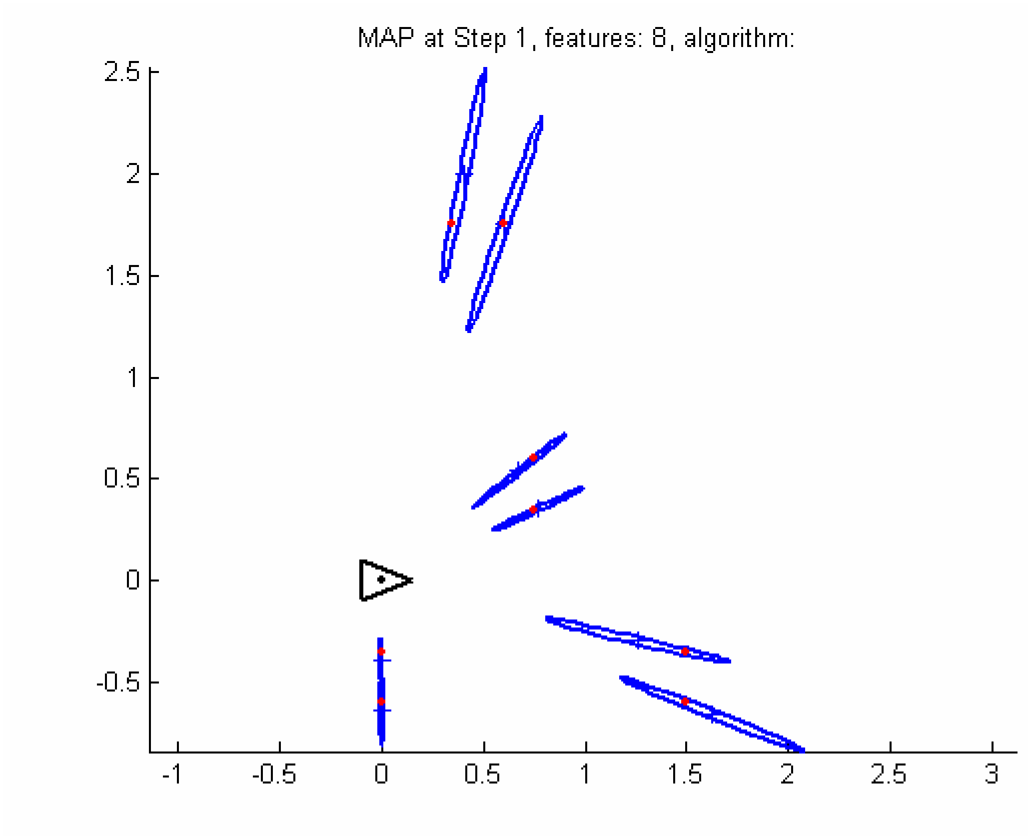
EKF-SLAM: add new features

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

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

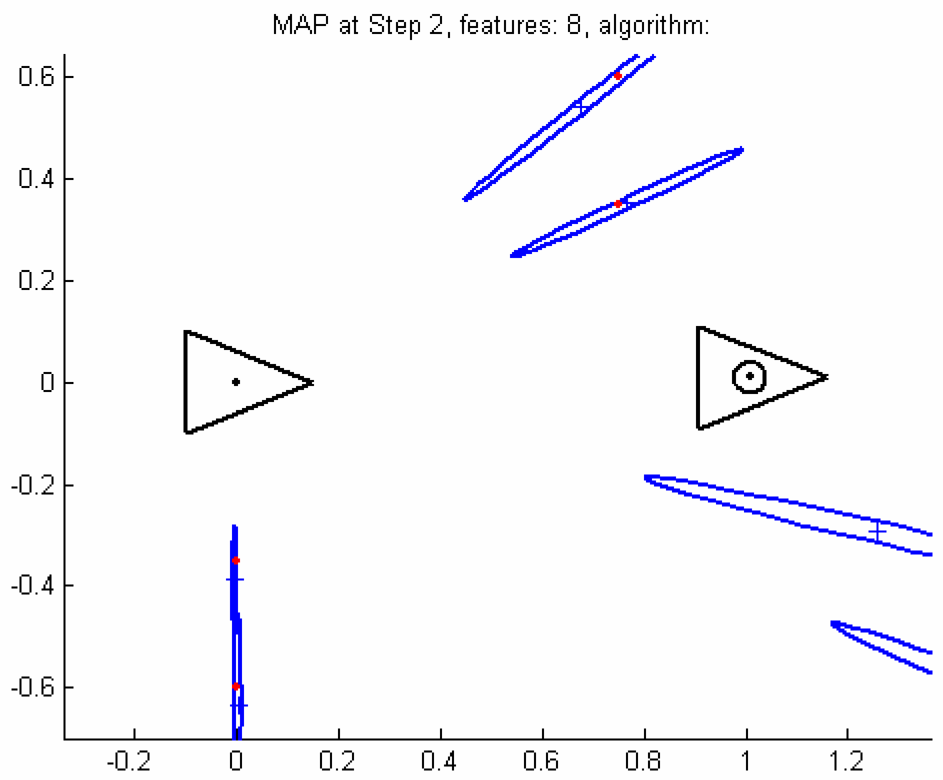
ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: add new features



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

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:

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

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

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

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

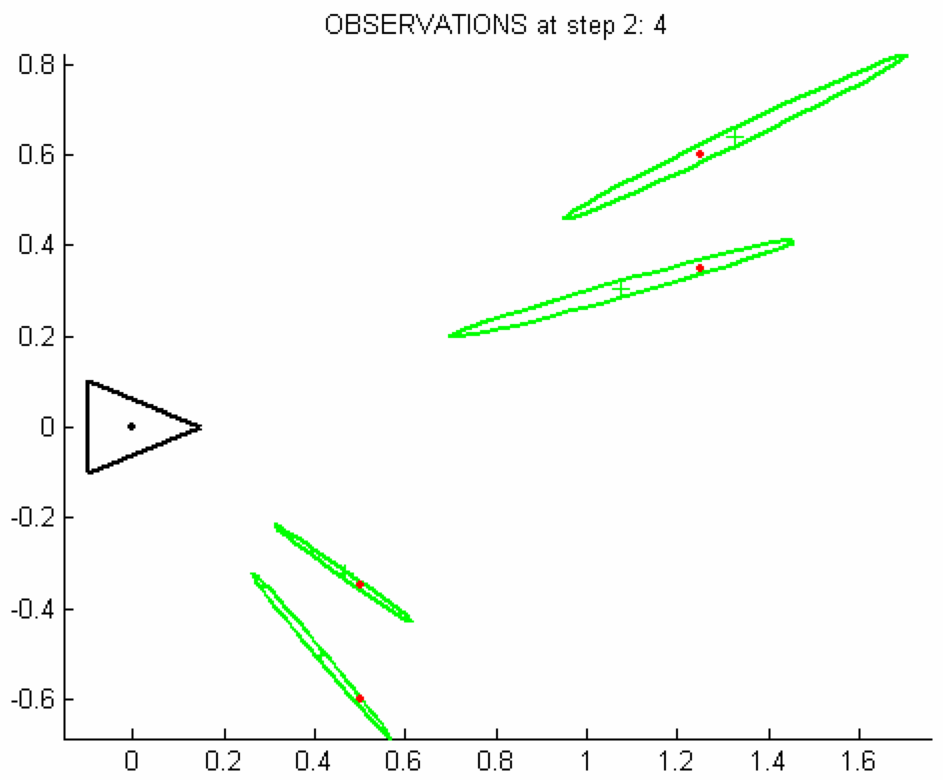
EKF-SLAM: compute robot motion

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

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: Observations



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: 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) :

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

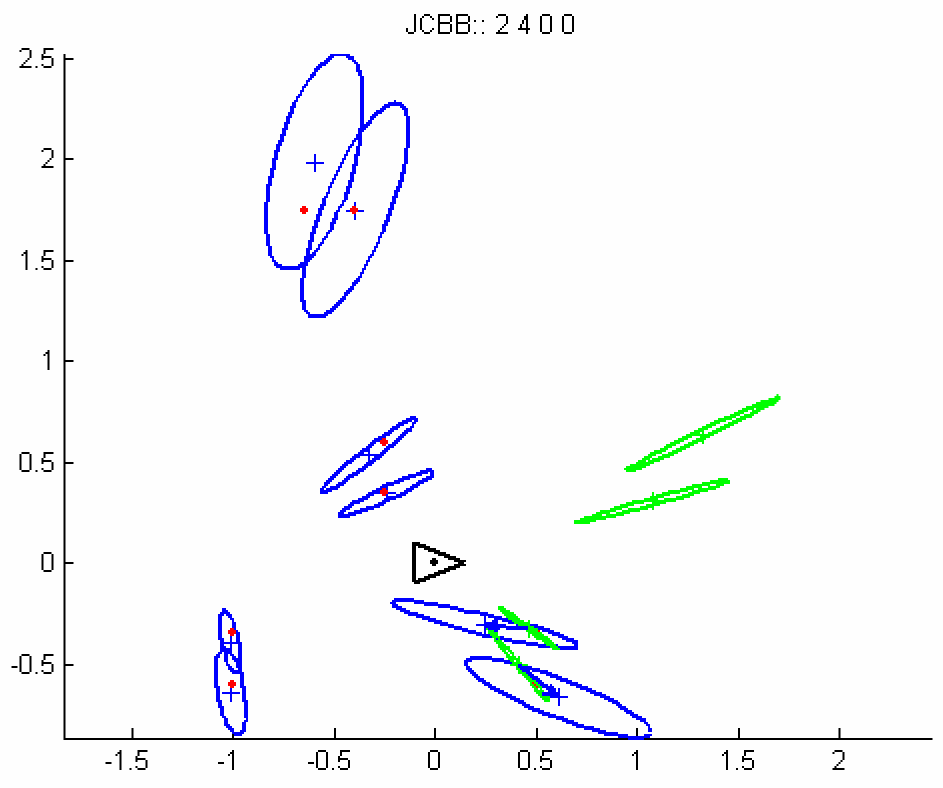
Measurement equation:

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

Sensor model (white noise):

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

EKF-SLAM: Data association



EKF-SLAM: Observations

Linearization:

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

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

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)

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:

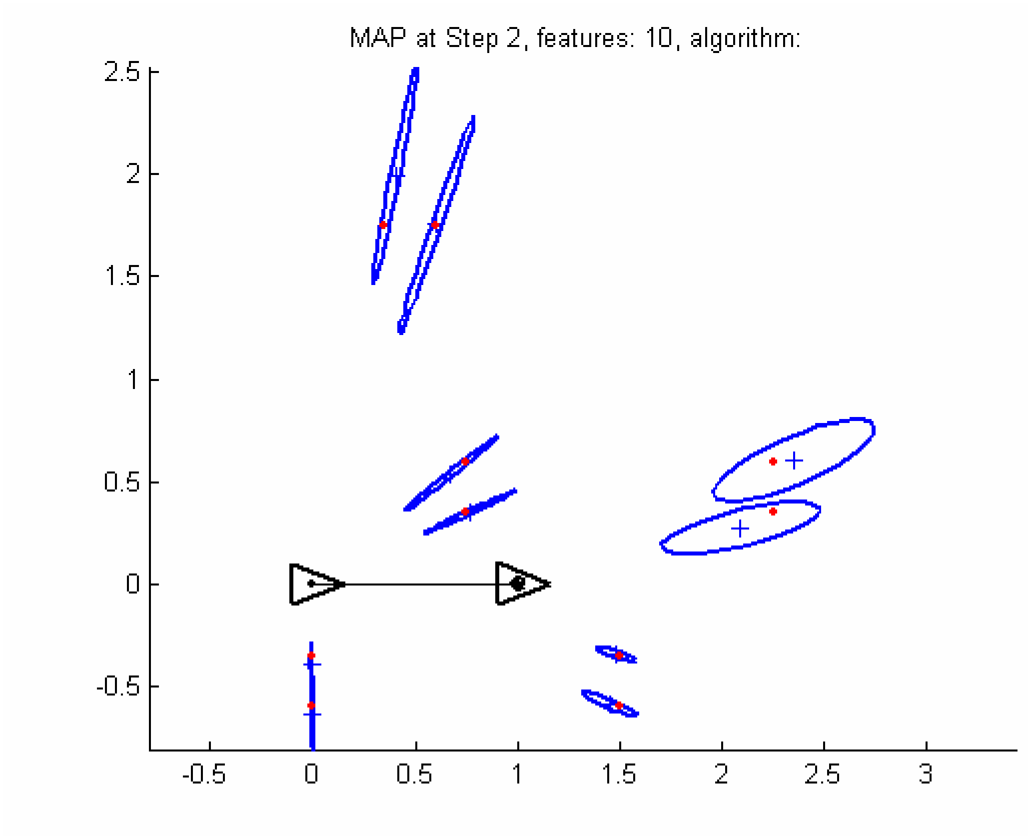
$$\mathbf{P}_k^B = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}^B$$

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



SLAM en entornos grandes. SLAM multivehículo

José Neira
Universidad de Zaragoza

jneira@unizar.es
http://www.cps.unizar.es/~jneira/

INDEX

1. Limitations of the basic EKF SLAM algorithm
    1. The linearization problem
    2. The scaling problem
2. Alternatives
    1. Independent local maps
    2. Map joining
    3. Multivehicle SLAM
    4. Hierarchical SLAM
3. Conclusion

1. Limitations of the basic EKF SLAM algorithm

The scaling problem

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

The prediction step

EKF SLAM prediction

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

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

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

EKF SLAM prediction

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

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

EKF prediction is *O(n)*

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Adding new features

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:

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

EKF SLAM: add new features

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

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

Adding new features is *O(n)*

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

The update step

EKF SLAM: map update

m **observations:**

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

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

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

Filter gain: $\mathbf{K}_k = \mathbf{P}^B_{k|k-1}\mathbf{H}^T_k(\mathbf{H}_k\mathbf{P}^B_{k|k-1}\mathbf{H}^T_k + \mathbf{R}_k)^{-1}$

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

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

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*

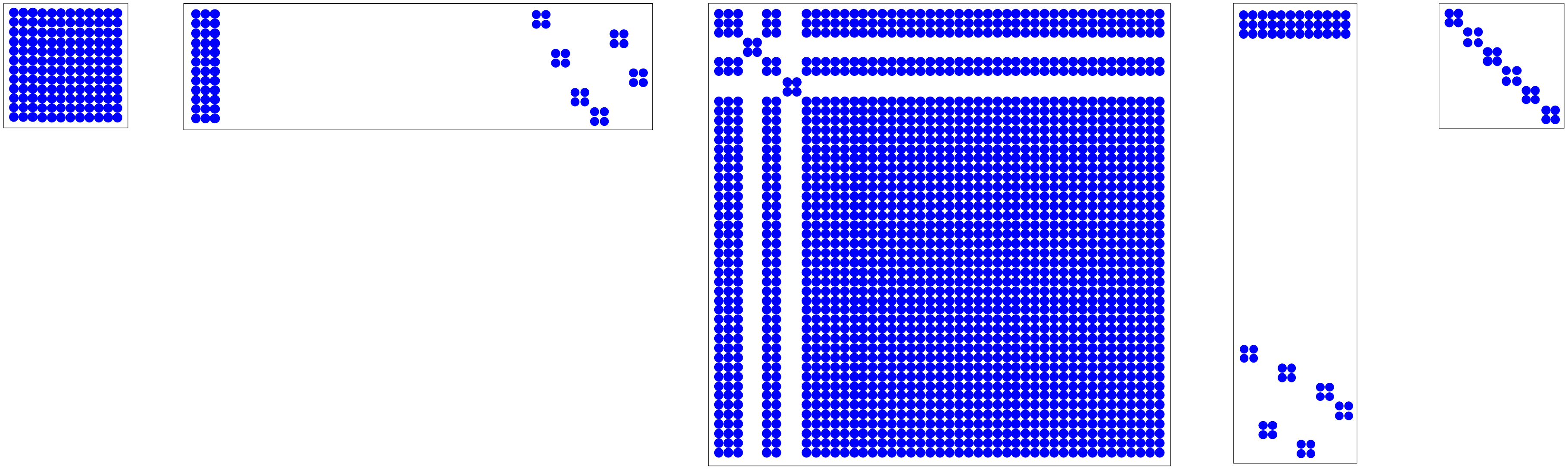
$O(rn^2)$ operations?

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

The innovation matrix

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

rxr *rxt* *nxn* *nxr* *rxr*

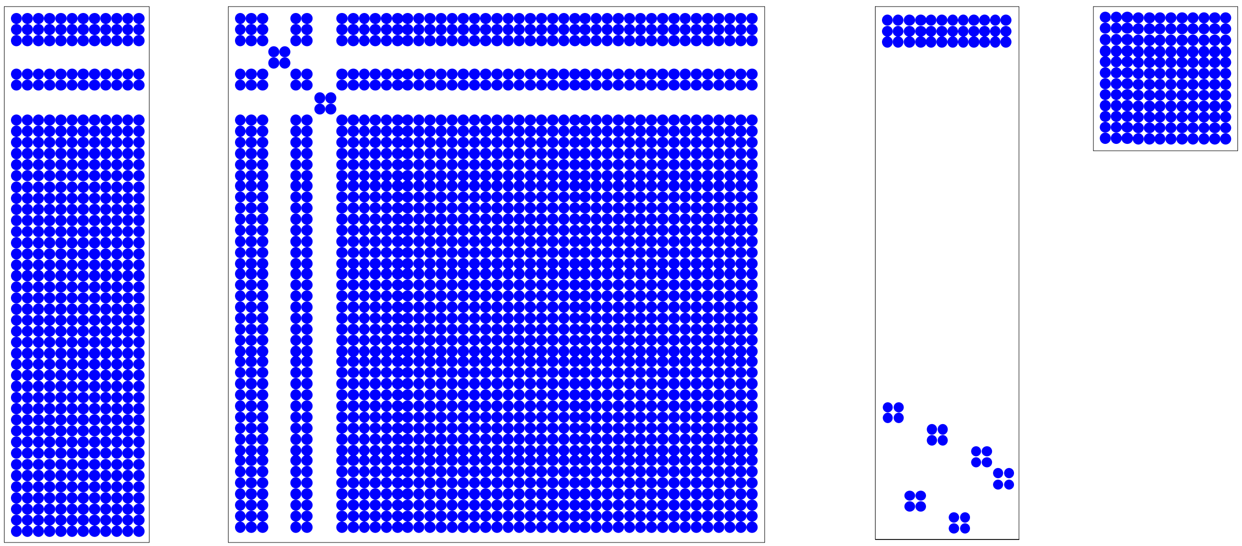


$O(rn) = O(n)$ operations

The Kalman gain matrix

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

nxr *nxn* *txr* *rxr*



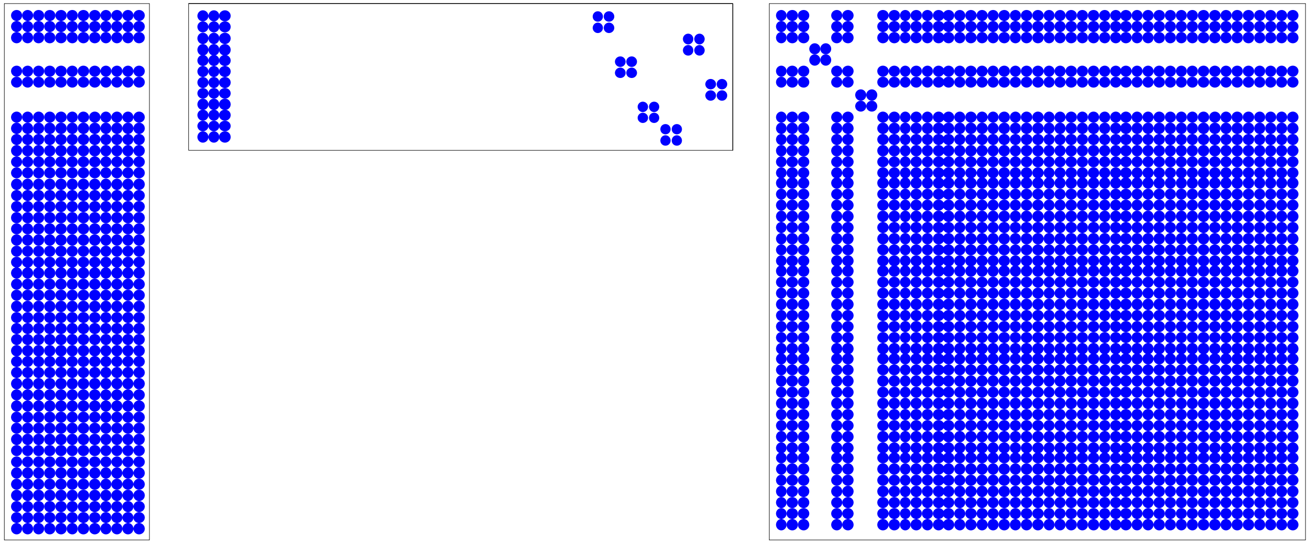
$O(r^2 n) = O(n)$ operations

The covariance matrix

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

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

$\ldots\ldots \qquad \ldots\ldots$



nxr *rxt* *nxn*

$O(rn^2) = O(n^2)$ operations

Efforts to reduce complexity

- Decoupled Stochastic Mapping (Leonard and Feder, 2000) (Jensfelt 2001) ***O(1)***
- Local Mapping Algorithm (Chong and Kleeman 1999) ***O(1)***
- Suboptimal SLAM (Guivant and Nebot 2001) ***O(n)***
- Sparse Weight Filter (Julier 2001) ***O(n)***
- Sparse Extended Information Filter (Thrun et al 2003) ***O(1)***

Approximate, or pessimistic solutions

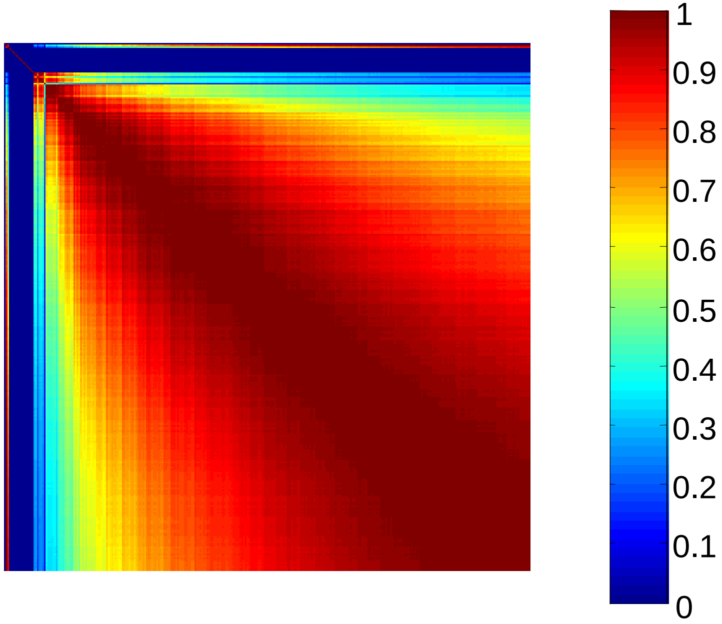
- Postponement (Davidson 1998, Knight, Davidson and Reed 2001)
- Compressed Filter (Guivant and Nebot 2001)
- Constrained Local Submap Filter (Williams 2001)
- Map Joining (Tardós et. al, 2002)

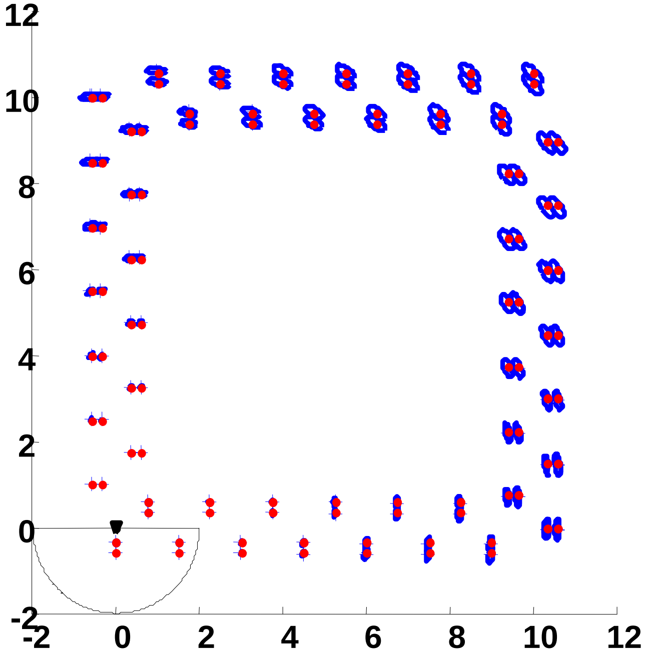
Exact solutions that delay global map updating, and strongly reduce cost.

But still *O*(n^2)

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Are correlations necessary?

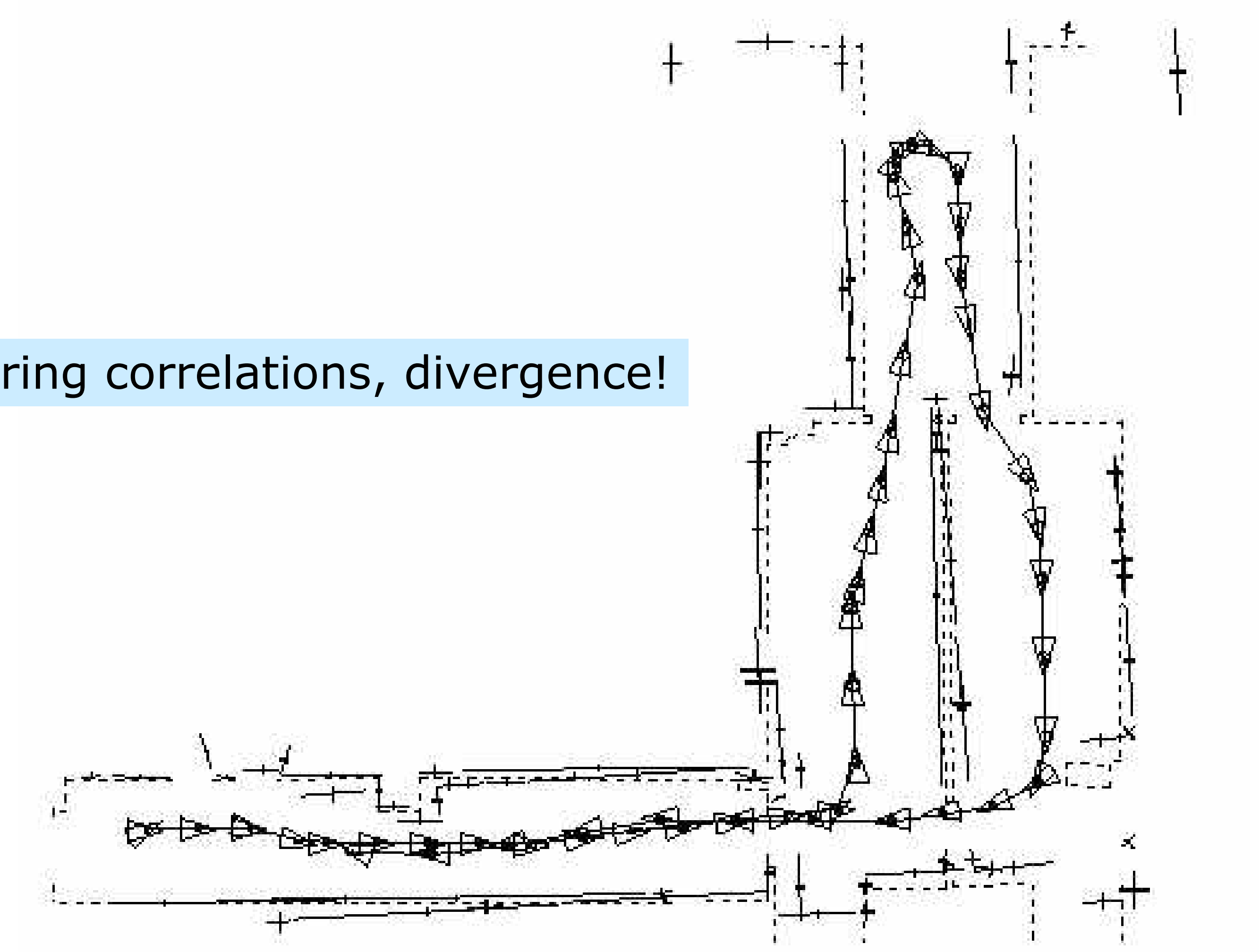





```
sigmas = sqrt(diag(Cov))' ;
Corr=diag(1./sigmas)*Cov*diag(1./sigmas) ;
```

The importance of Correlations

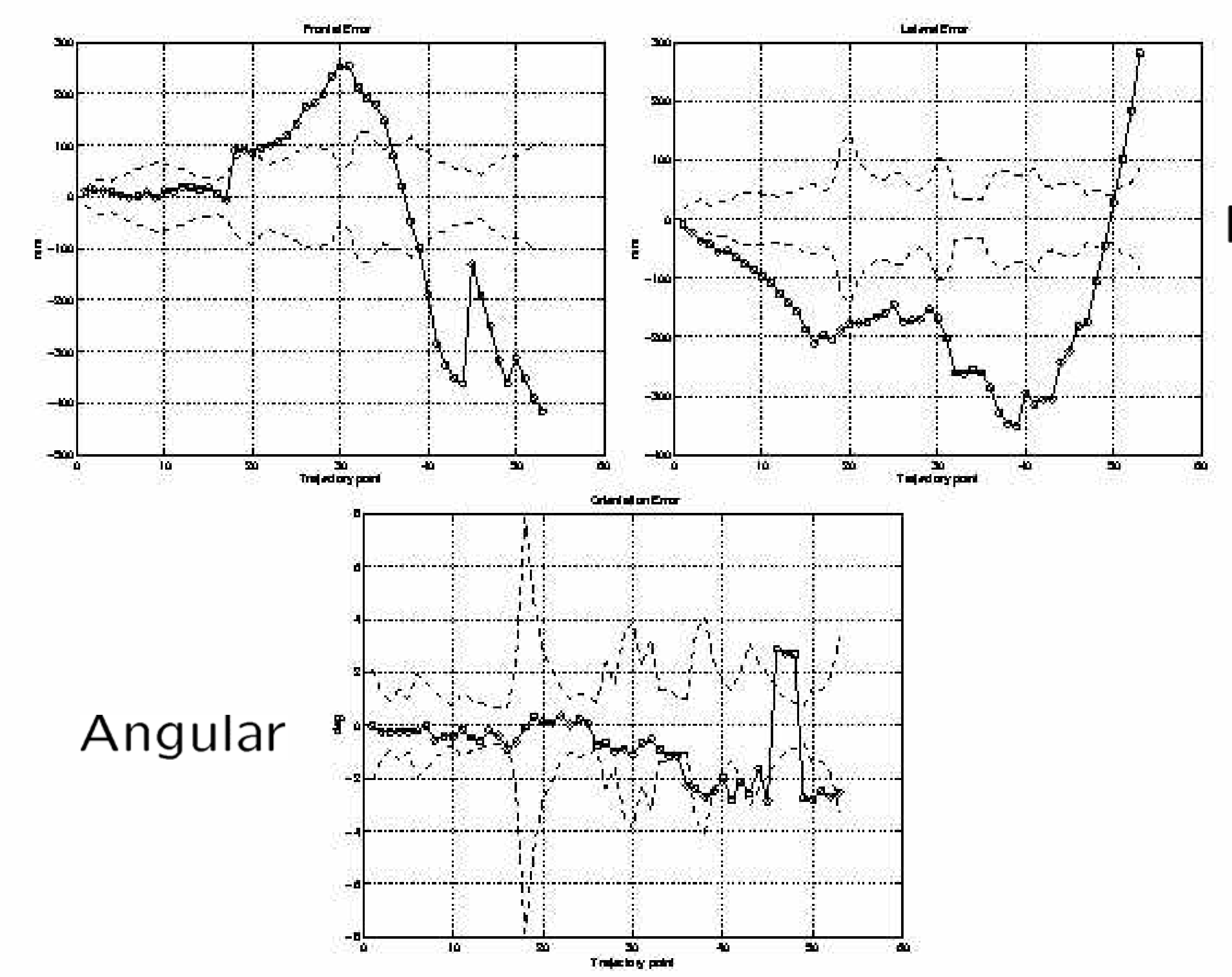
Ignoring correlations, divergence!



The importance of Correlations

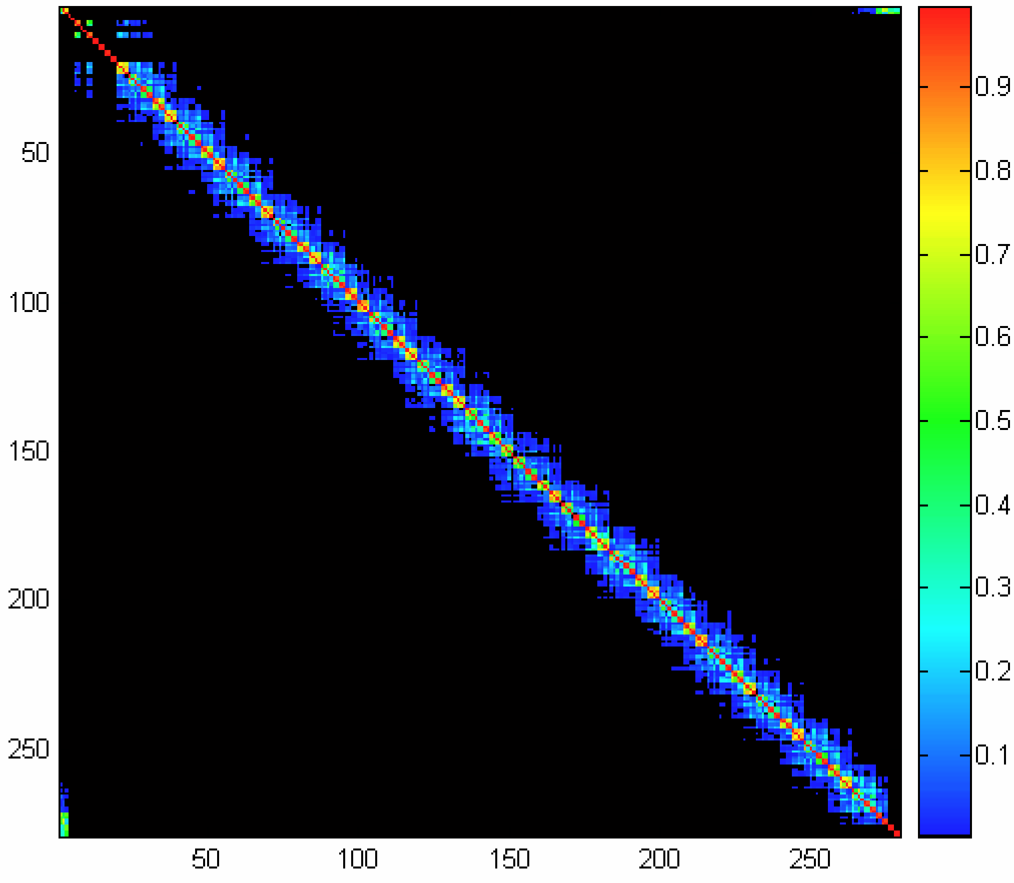
Frontal

Lateral



Angular

Inverse correlations



This observation is the basis of SEIFs

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

The linearization problem

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

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

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

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

The Consistency Problem

True map value	EKF-SLAM estimation
$\mathbf{x}_k^W$	$\hat{\mathbf{x}}_k^W$
	$\mathbf{P}_k^W$

- An estimator is **consistent** if:

$$E\left[\mathbf{x}_k^W - \hat{\mathbf{x}}_k^W\right] = \mathbf{0}$$

Unbiased

$$E\left[\left(\mathbf{x}_k^W - \hat{\mathbf{x}}_k^W\right)\left(\mathbf{x}_k^W - \hat{\mathbf{x}}_k^W\right)^T\right] = \mathbf{P}_k^W$$

The Mean Square Error matches the filter computed Covariance

  - Pessimistic covariance is OK (but not too pessimistic)
  - Optimistic covariance = Inconsistency = Filter divergence

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*)

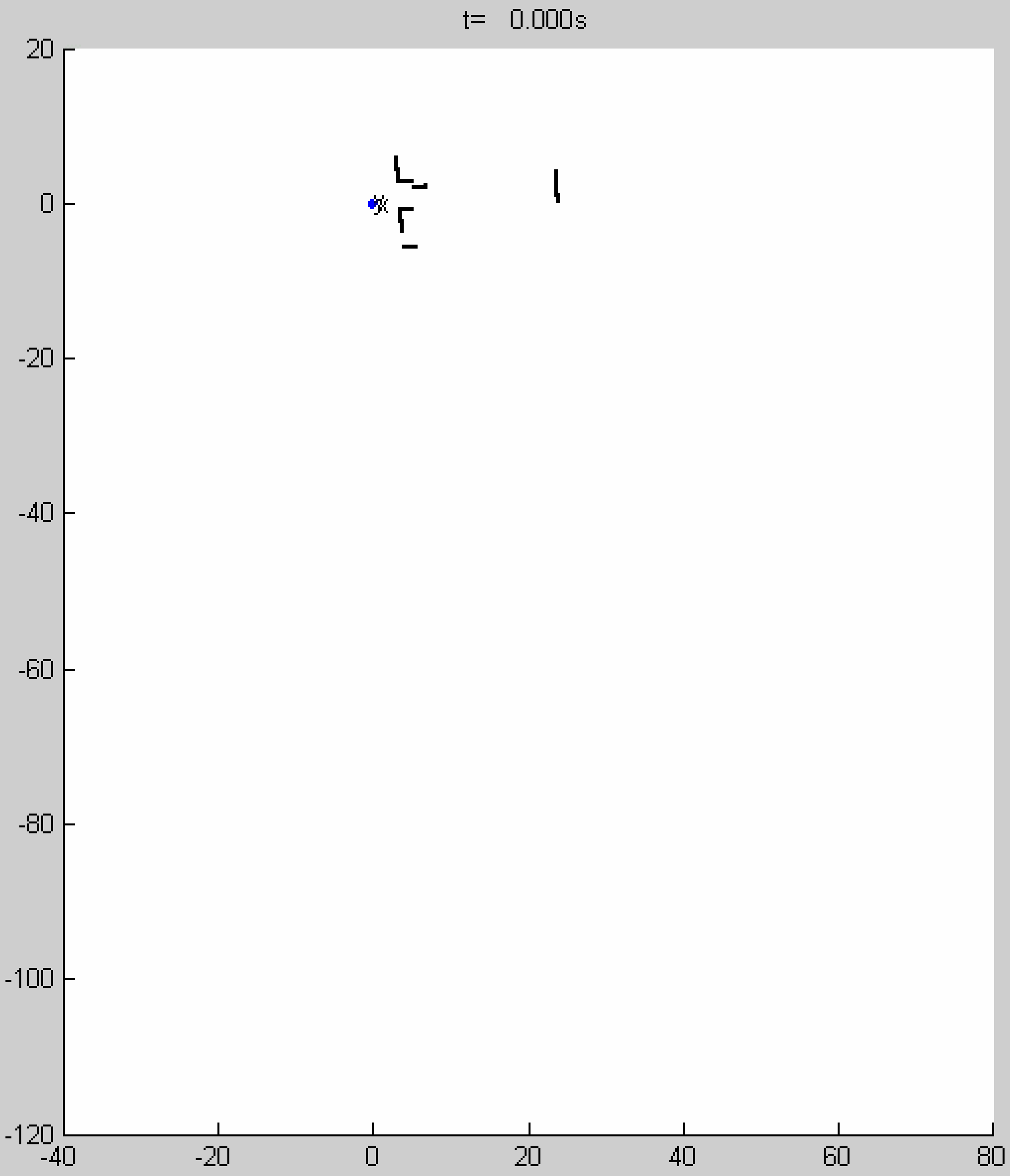
$$D_{ij}^2 = \left(\mathbf{z}_i - \mathbf{h}_j(\hat{\mathbf{x}}_k^W)\right)^T \left(\mathbf{H}_j \mathbf{P}_k^W \mathbf{H}_j^T + \mathbf{R}_i\right)^{-1} \left(\mathbf{z}_i - \mathbf{h}_j(\hat{\mathbf{x}}_k^W)\right)$$

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

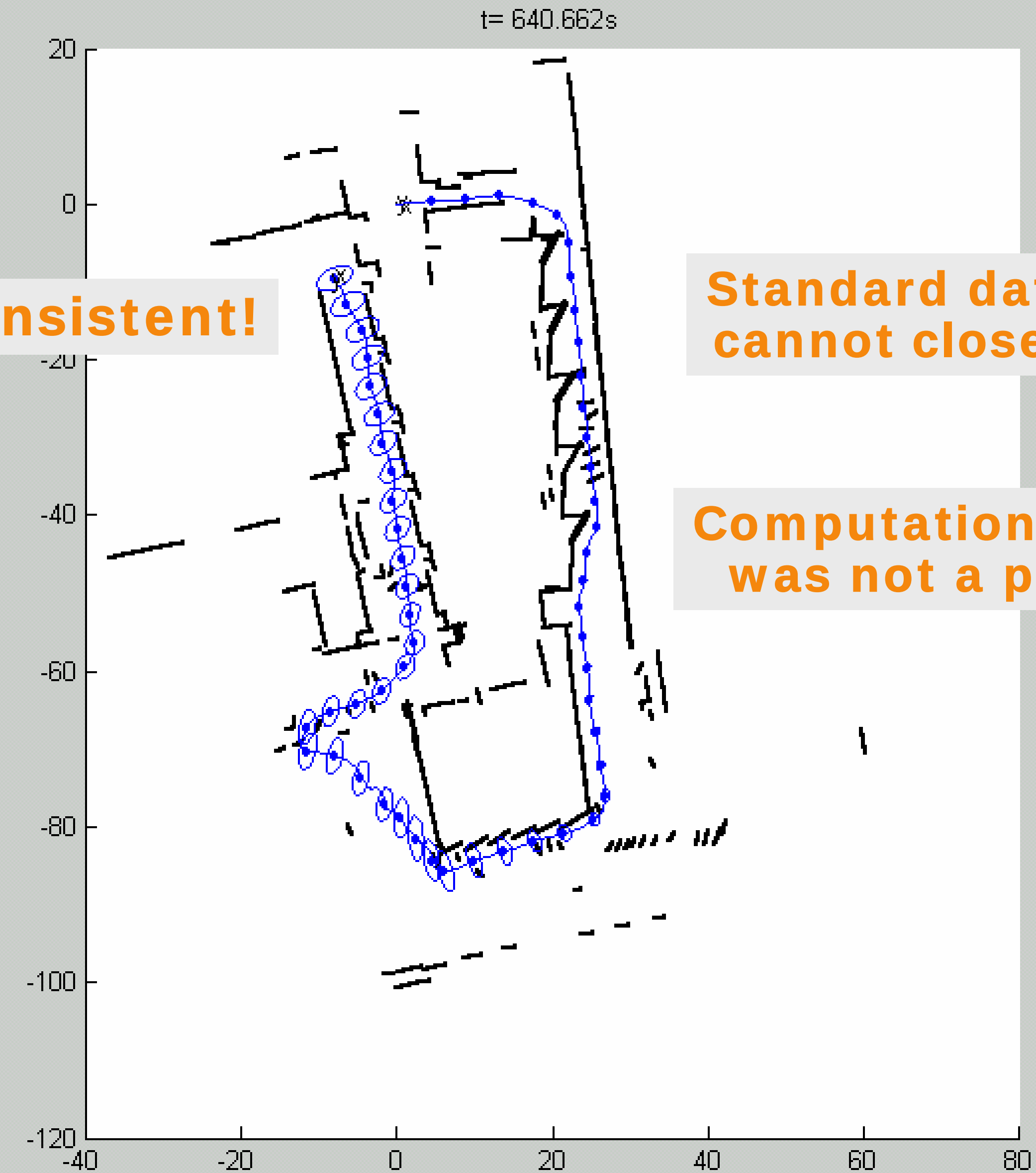
Critical when closing big loops

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

EKF-SLAM: Real Example



EKF-SLAM: Real Example



Inconsistent!

Standard data association cannot close a 250m loop

Computational complexity was not a problem here!

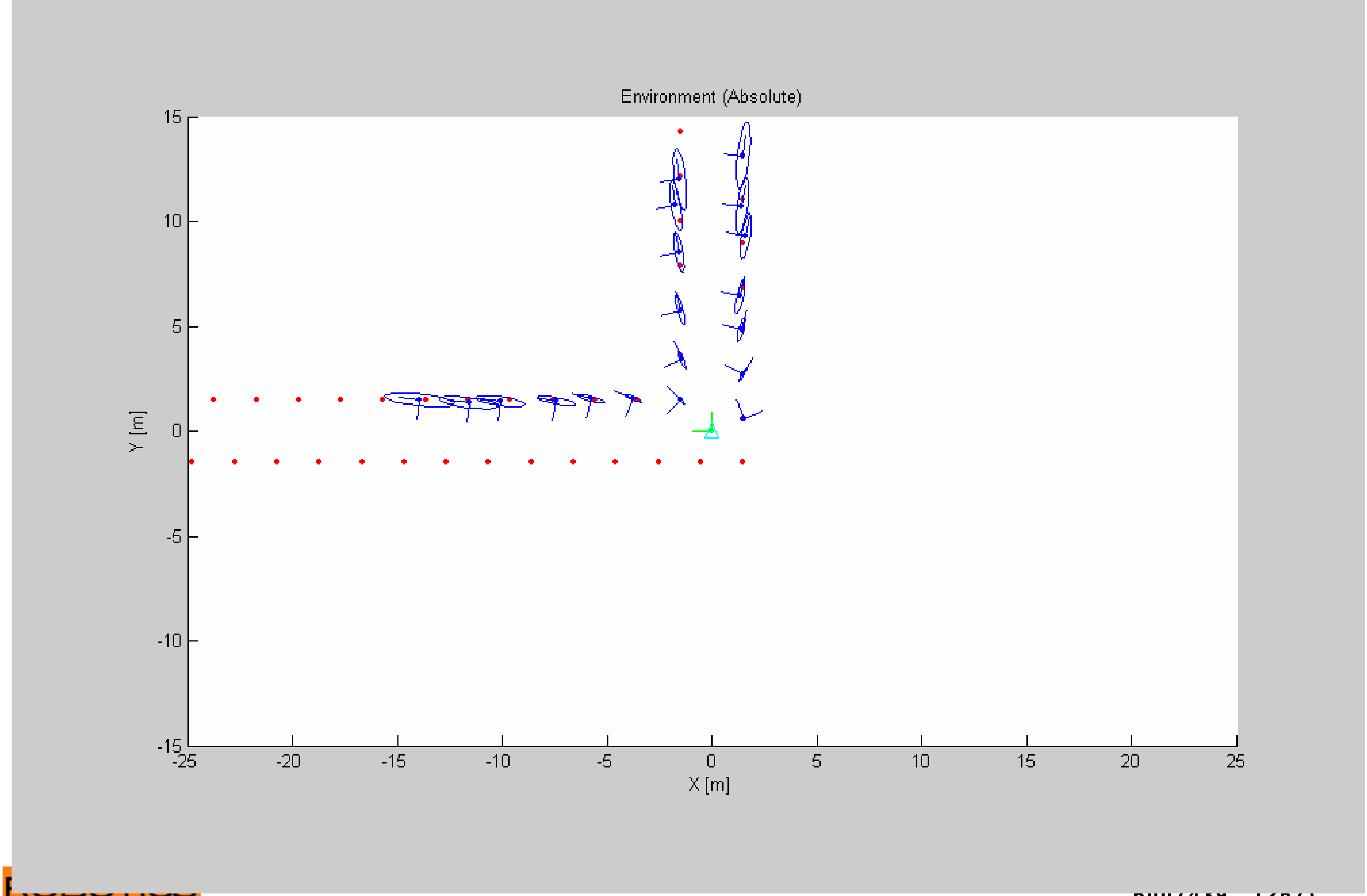
EKF-SLAM: Simulation

- 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)

EKF-SLAM: Simulat

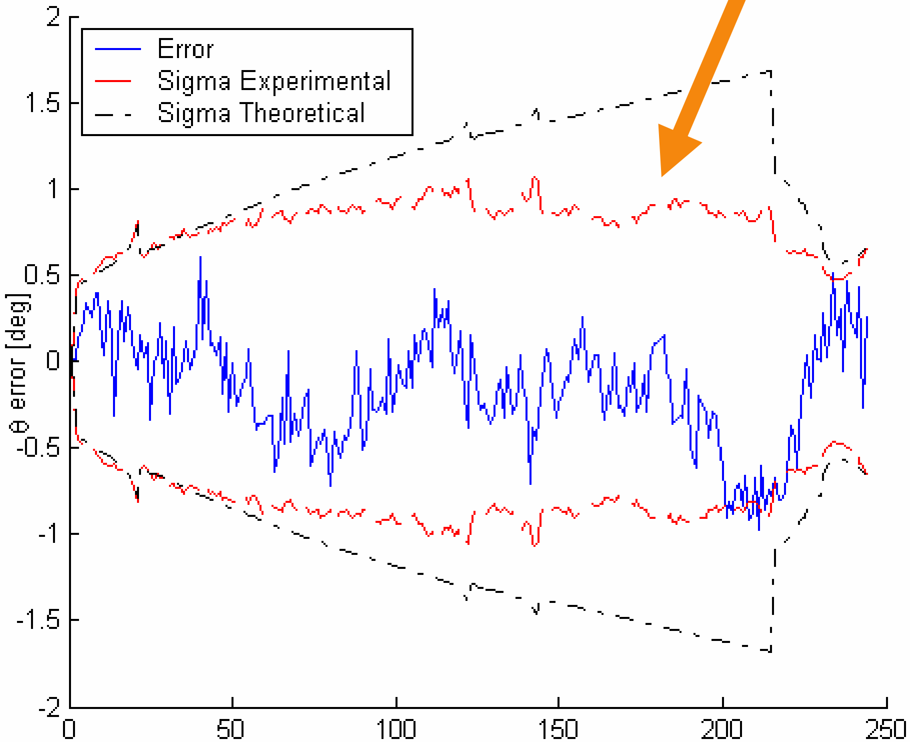
Perfect data association and noise model

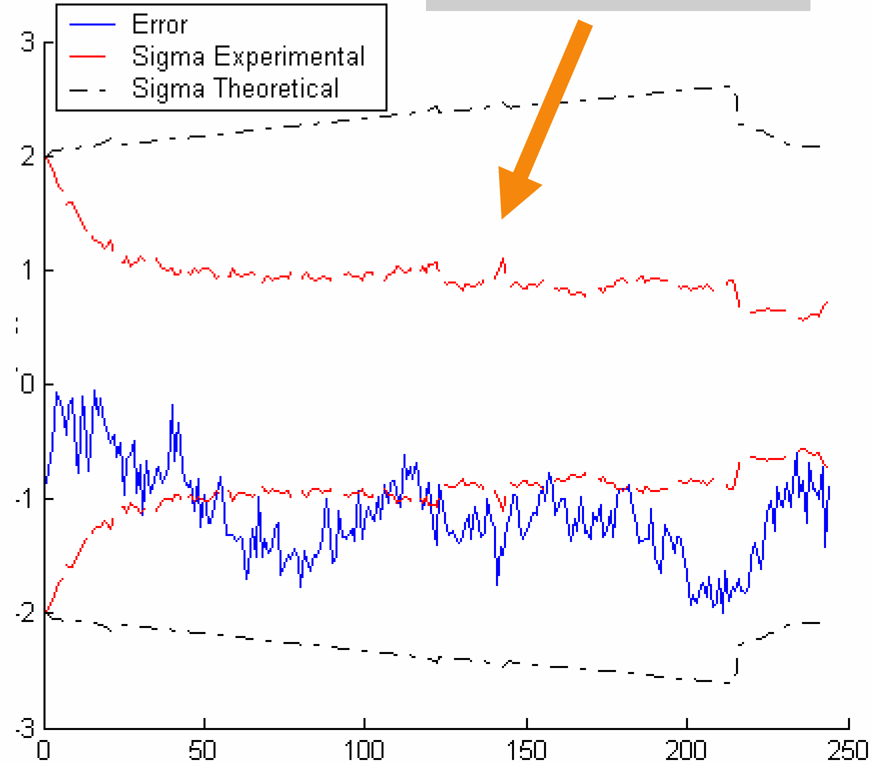


EKF-SLAM: Covariance

Optimistic

Violates lower bound





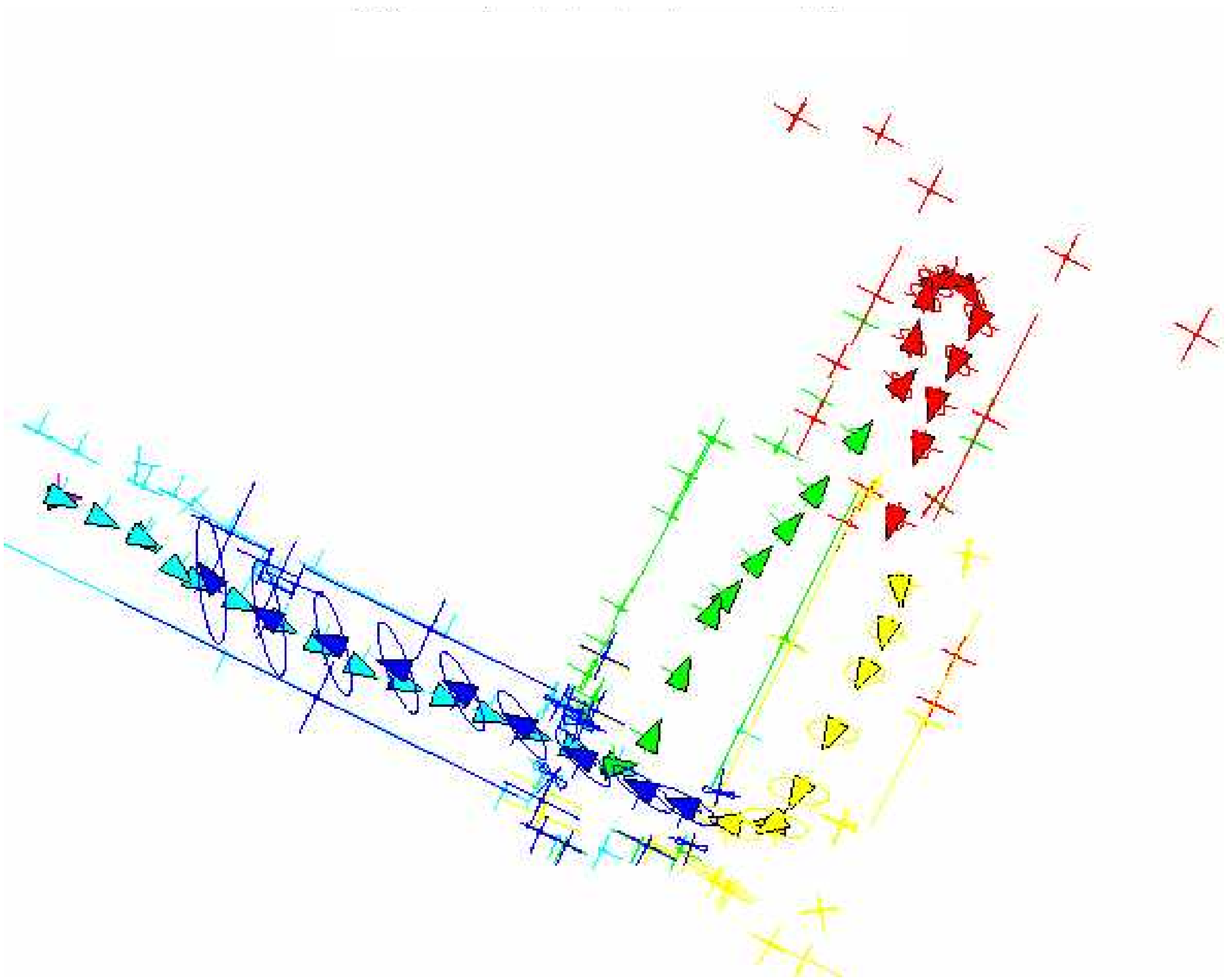
Initial uncertainty = 0

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

Overcoming these problems: Local maps

Local mapping



Local map building

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

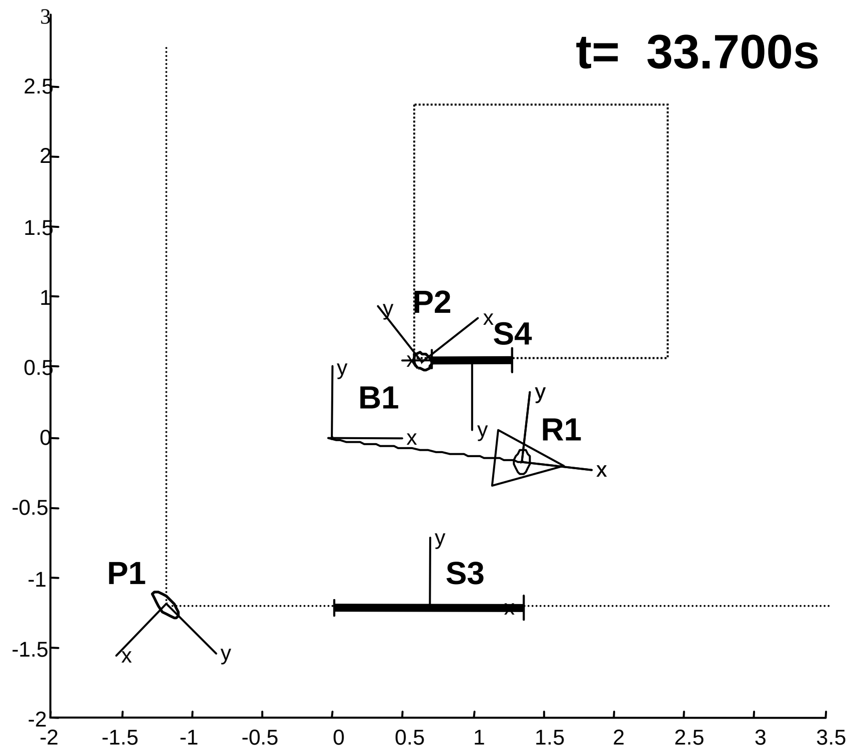
$$\hat{\mathbf{x}}^{B}_{R_0} = \mathbf{0}$$
$$\mathbf{P}^{B}_{R_0} = \mathbf{0}$$

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

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

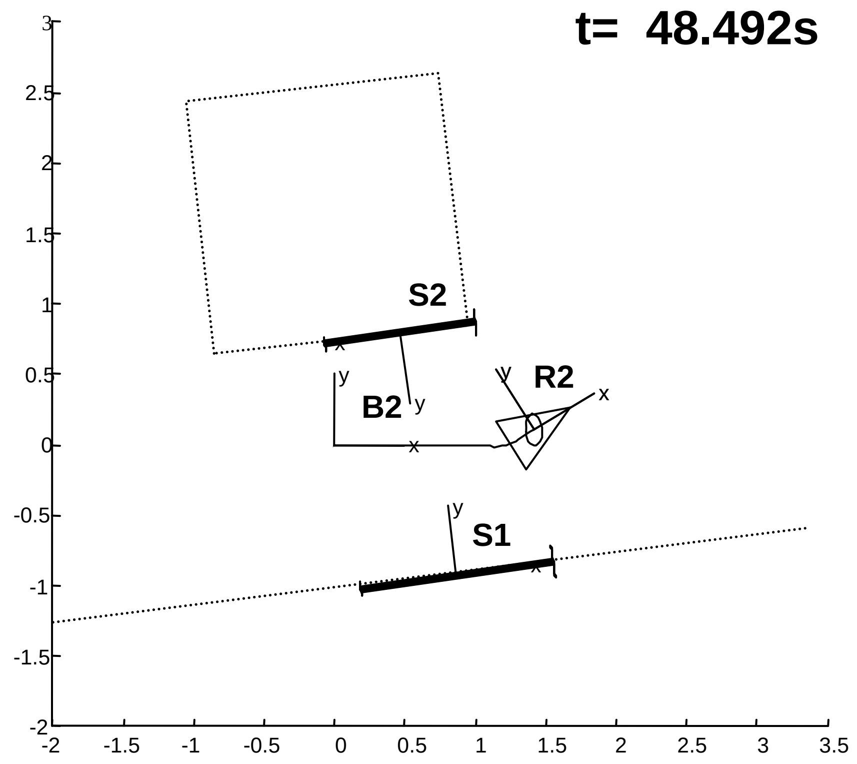
- No information is shared:

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

Maps are uncorrelated

- Common reference:

$$B_2 = R_1$$



Independent Local Maps

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

- Build independent local maps (Tardós 02)
  - Constant time complexity
- Map joining (Tardós 02)
  - Improves consistency and precision
- Map matching (Neira 03)
  - Correct environment topology
- Hierarchical SLAM (Estrada 05)
  - Scalable map representation
  - Good precision after loop closing
  - Fast convergence
- Similar approaches: CLSF (Williams 02), NCFM (Bailey 02), ATLAS (Bosse 03), CTS (Newman 03)

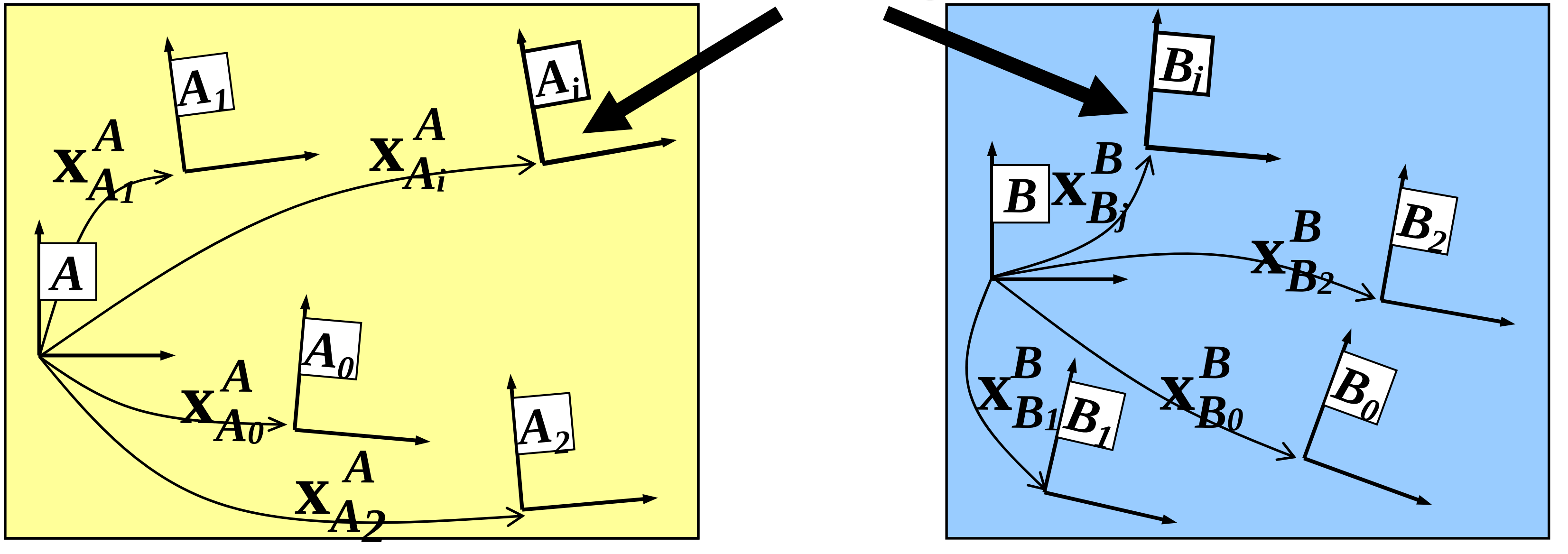
ROBOTICS UNIVERSIDAD DE ZARAGOZA

Map Joining

Map Joining

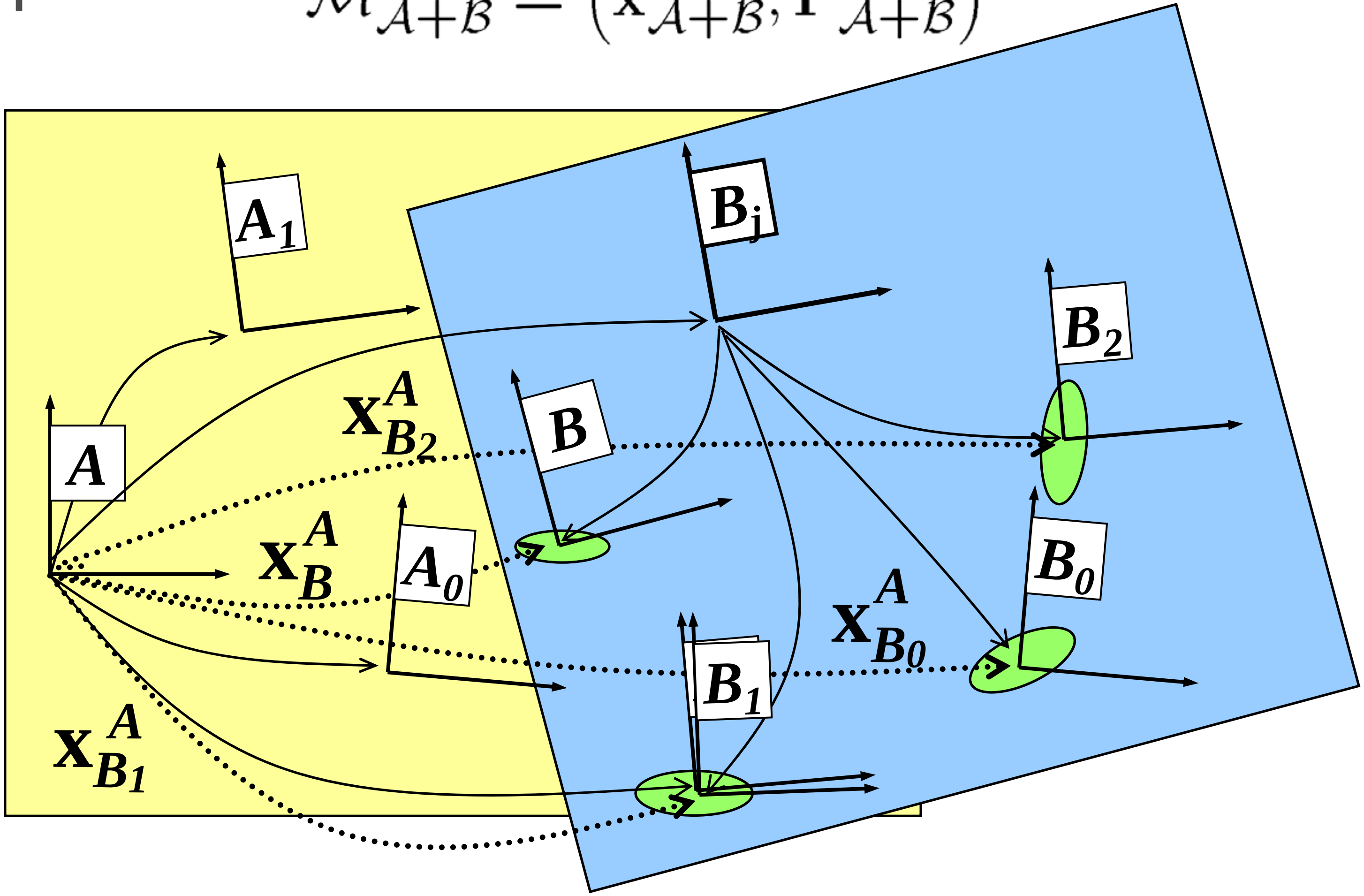
- Given:
  1. Two statistically independent stochastic maps
  2. A common reference

$\mathcal{M}_{\mathcal{A}}^{A} = \left(\hat{\mathbf{x}}_{\mathcal{A}}^{A}, \mathbf{P}_{\mathcal{A}}^{A}\right)$ $\quad A_i = B_j \quad$ $\mathcal{M}_{\mathcal{B}}^{B} = \left(\hat{\mathbf{x}}_{\mathcal{B}}^{B}, \mathbf{P}_{\mathcal{B}}^{B}\right)$

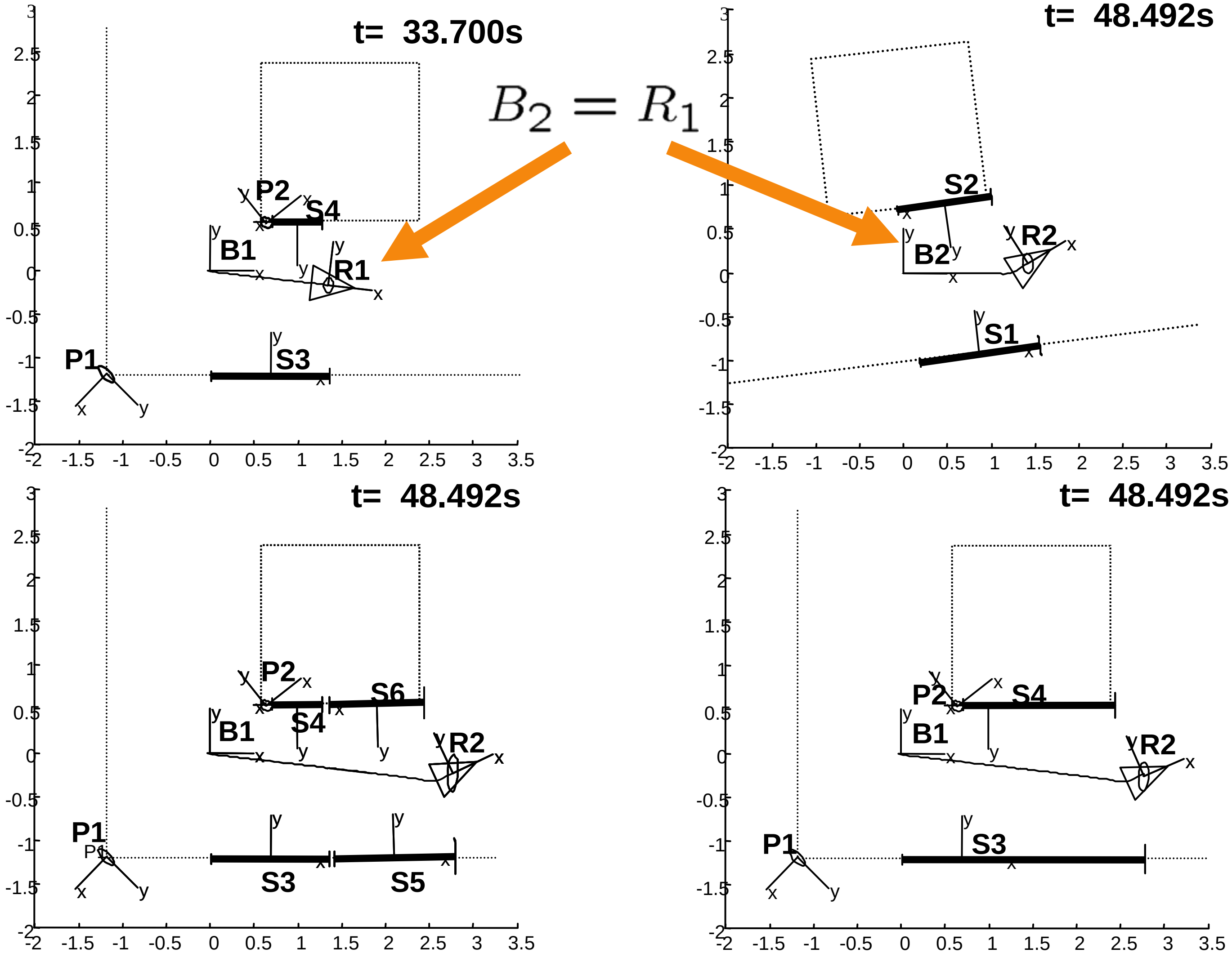


Map Joining

- Conveys the information of the two maps into a single **fully consistent** stochastic map: $\mathcal{M}^{A}_{\mathcal{A}+\mathcal{B}} = \left(\hat{\mathbf{x}}^{A}_{\mathcal{A}+\mathcal{B}}, \mathbf{P}^{A}_{\mathcal{A}+\mathcal{B}}\right)$



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

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Map Joining

- New state vector: $\hat{\mathbf{x}}^{A}_{\mathcal{A}+\mathcal{B}} = \begin{bmatrix} \hat{\mathbf{x}}^{A}_{\mathcal{A}} \\ \hat{\mathbf{x}}^{A}_{\mathcal{B}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}^{A}_{\mathcal{A}} \\ \hat{\mathbf{x}}^{A}_{A_i} \oplus \hat{\mathbf{x}}^{B_j}_{B_0} \\ \vdots \\ \hat{\mathbf{x}}^{A}_{A_i} \oplus \hat{\mathbf{x}}^{B_j}_{B_m} \end{bmatrix}$

- New covariance matrix:

$$\begin{aligned}
\mathbf{P}^{A}_{\mathcal{A}+\mathcal{B}} &= \mathbf{J}^{\mathcal{A}+\mathcal{B}}_{\mathcal{A}} \mathbf{P}^{A}_{\mathcal{A}} \mathbf{J}^{\mathcal{A}+\mathcal{B}^T}_{\mathcal{A}} + \mathbf{J}^{\mathcal{A}+\mathcal{B}}_{\mathcal{B}} \mathbf{P}^{B_j}_{\mathcal{B}} \mathbf{J}^{\mathcal{A}+\mathcal{B}^T}_{\mathcal{B}} \\
&= \begin{bmatrix} \mathbf{P}^{A}_{\mathcal{A}} & \mathbf{P}^{A}_{\mathcal{A}} \mathbf{J}_1^T \\ \mathbf{J}_1 \mathbf{P}^{A}_{\mathcal{A}} & \mathbf{J}_1 \mathbf{P}^{A}_{\mathcal{A}} \mathbf{J}_1^T \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_2 \mathbf{P}^{B_j}_{\mathcal{B}} \mathbf{J}_2^T \end{bmatrix}
\end{aligned}$$

$$\mathbf{J}^{\mathcal{A}+\mathcal{B}}_{\mathcal{A}} = \frac{\partial \hat{\mathbf{x}}^{A}_{\mathcal{A}+\mathcal{B}}}{\partial \hat{\mathbf{x}}^{A}_{\mathcal{A}}} = \begin{bmatrix} \mathbf{I} \\ \mathbf{J}_1 \end{bmatrix} \qquad \mathbf{J}_1 = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{J}_{1\oplus}\left\{\hat{\mathbf{x}}^{A}_{A_i}, \hat{\mathbf{x}}^{B_j}_{B_0}\right\} & \cdots & \mathbf{0} \\ \vdots & & \vdots & & \vdots \\ \mathbf{0} & \cdots & \mathbf{J}_{1\oplus}\left\{\hat{\mathbf{x}}^{A}_{A_i}, \hat{\mathbf{x}}^{B_j}_{B_m}\right\} & \cdots & \mathbf{0} \end{bmatrix}$$

$$\mathbf{J}^{\mathcal{A}+\mathcal{B}}_{\mathcal{B}} = \frac{\partial \hat{\mathbf{x}}^{A}_{\mathcal{A}+\mathcal{B}}}{\partial \hat{\mathbf{x}}^{B_j}_{\mathcal{B}}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{J}_2 \end{bmatrix} \qquad \mathbf{J}_2 = \begin{bmatrix} \mathbf{J}_{2\oplus}\left\{\hat{\mathbf{x}}^{A}_{A_i}, \hat{\mathbf{x}}^{B_j}_{B_0}\right\} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{J}_{2\oplus}\left\{\hat{\mathbf{x}}^{A}_{A_i}, \hat{\mathbf{x}}^{B_j}_{B_m}\right\} \end{bmatrix}$$

Map joining is $O(n_1 n_2)$ on the maps

Matching and Fusion

- Matching function:

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

- Joint matching function for the hypothesis:

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

- Joint innovation test:

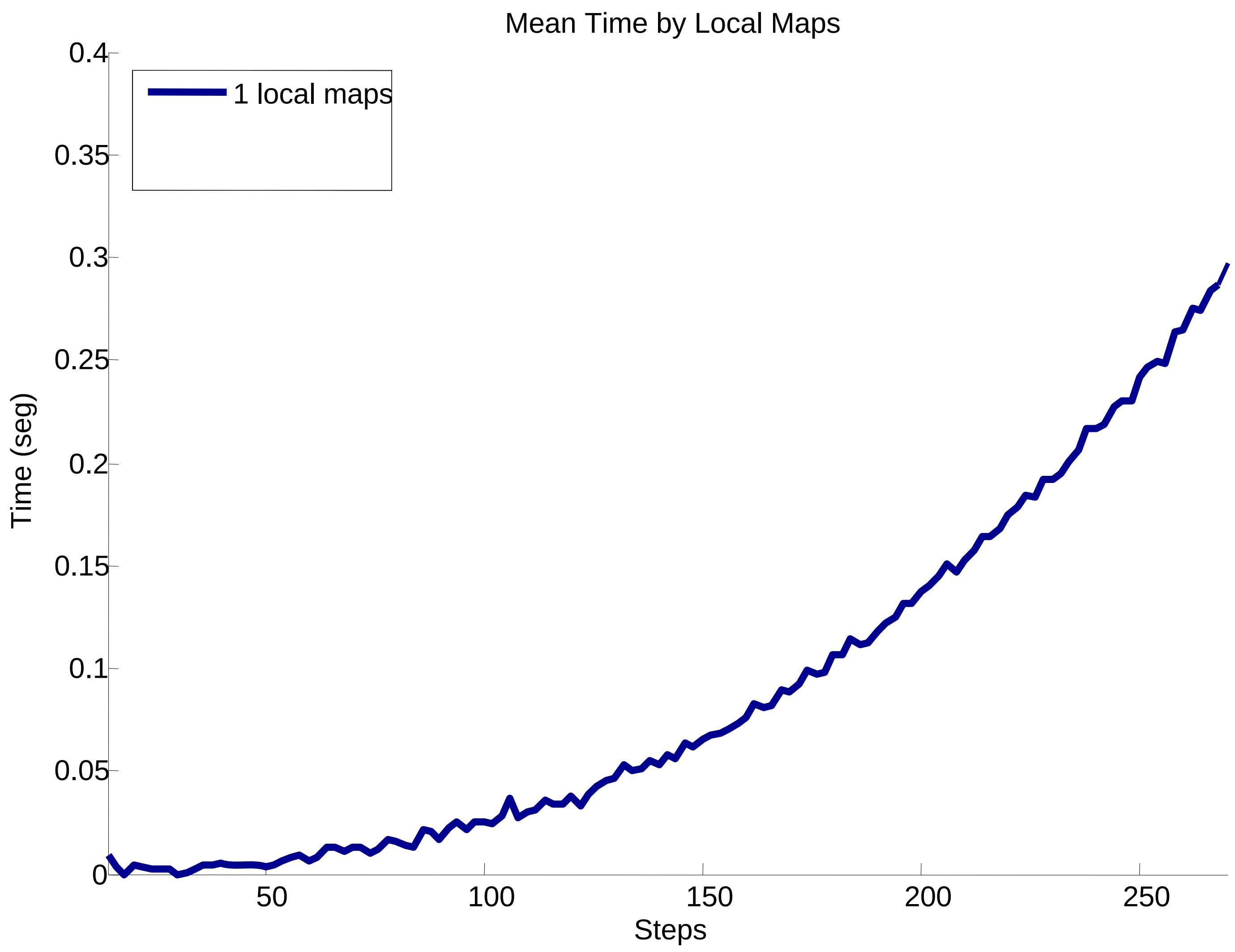
$$D^2_{\mathcal{H}} = \mathbf{h}^T_{\mathcal{H}} \left(\mathbf{H}_{\mathcal{H}} \mathbf{P} \mathbf{H}^T_{\mathcal{H}} \right)^{-1} \mathbf{h}_{\mathcal{H}} < \chi^2_{d,\alpha}$$

- Map update using EKF:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} - \mathbf{K}_k \mathbf{h}_{\mathcal{H}} \\ \mathbf{P}_k &= \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_{\mathcal{H}} \right) \mathbf{P}_{k-1} \\ \mathbf{K}_k &= \mathbf{P}_{k-1} \mathbf{H}^T_{\mathcal{H}} \left(\mathbf{H}_{\mathcal{H}} \mathbf{P}_{k-1} \mathbf{H}^T_{\mathcal{H}} \right)^{-1} \end{aligned}$$

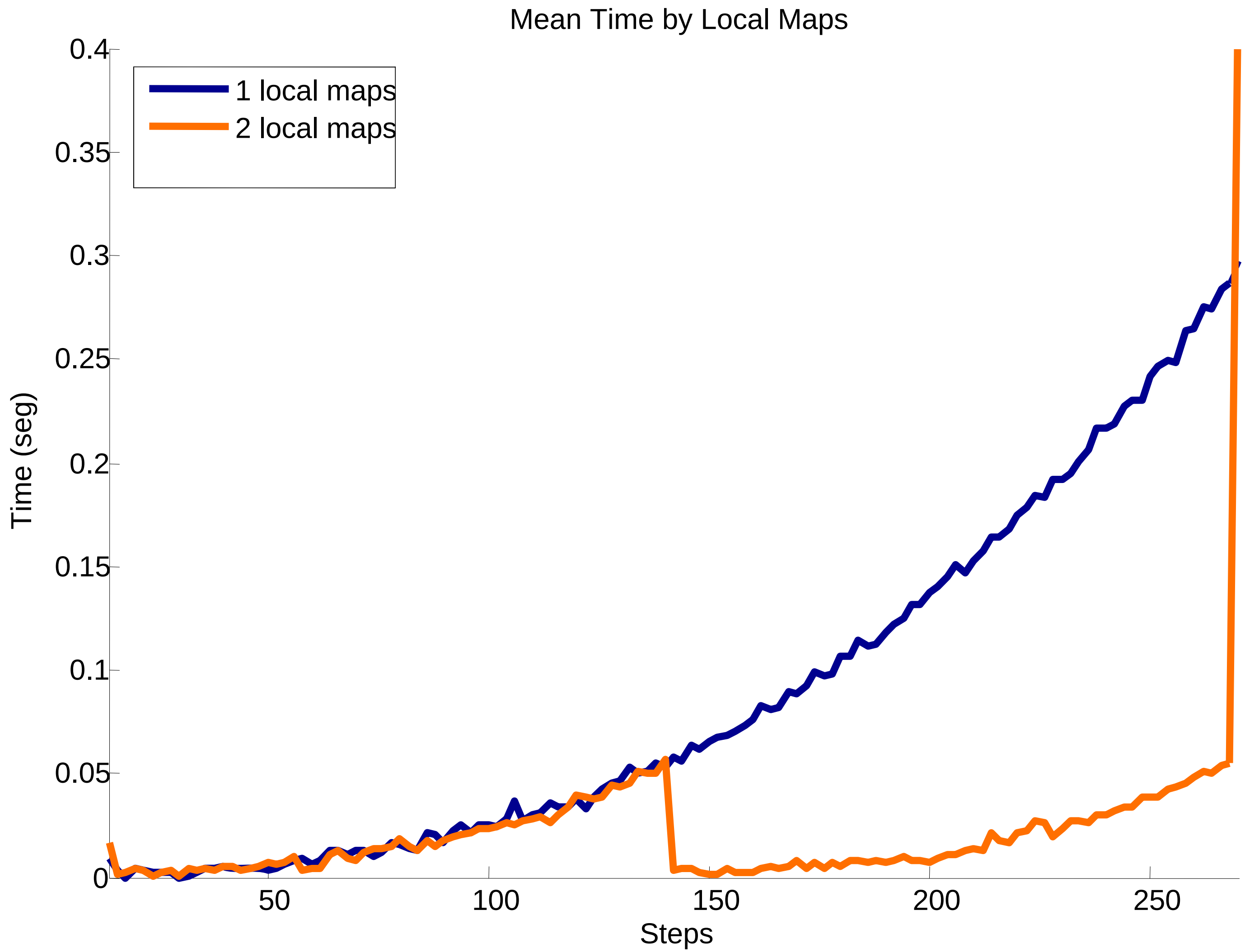
Matching and fusion is $O((n_1+n_2)^2)$

EKF updates



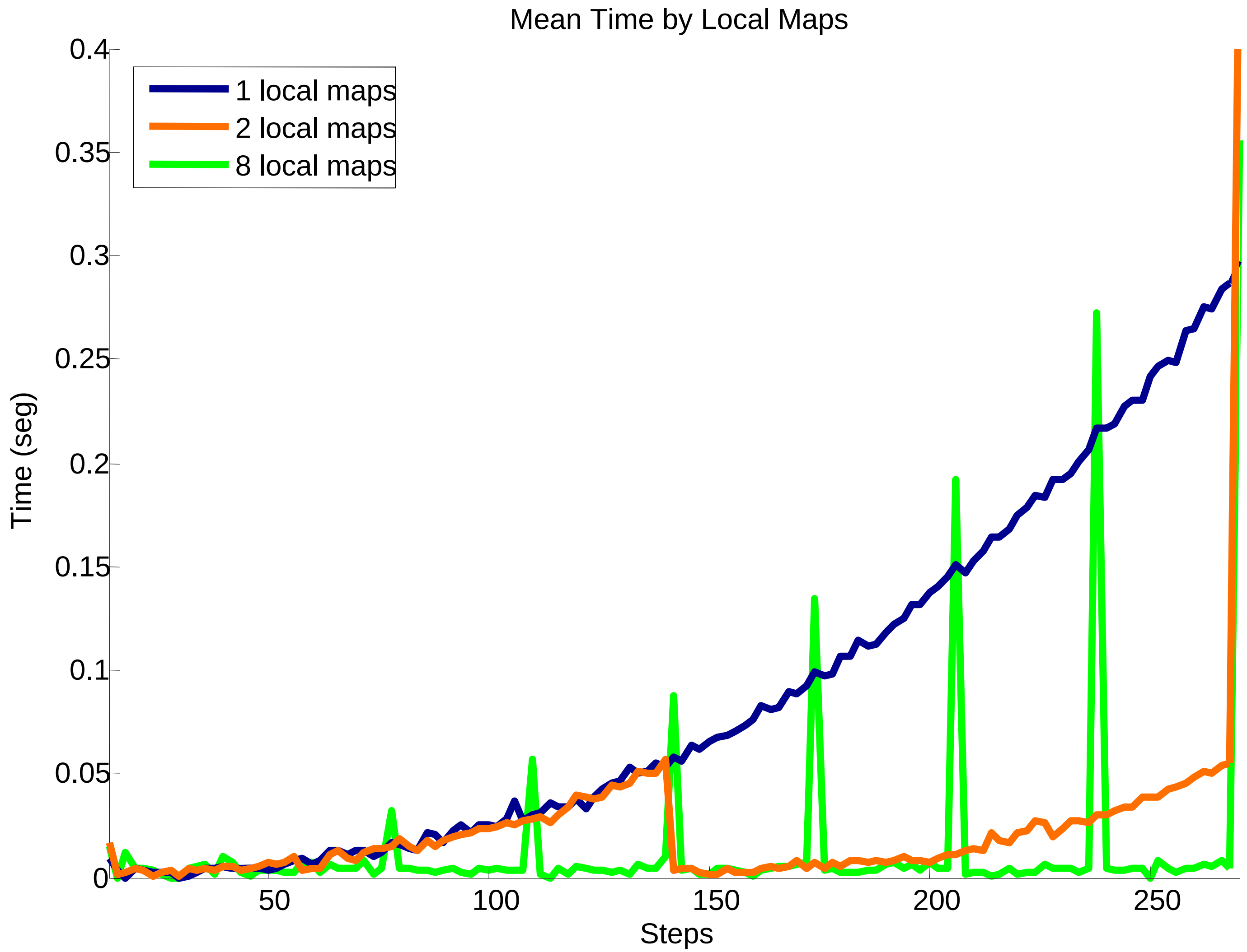
Mean Time by Local Maps
1 local maps
Time (seg)
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
50
100
150
200
250
Steps

Map Joining



Mean Time by Local Maps
1 local maps
2 local maps
Time (seg)
Steps
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
50
100
150
200
250

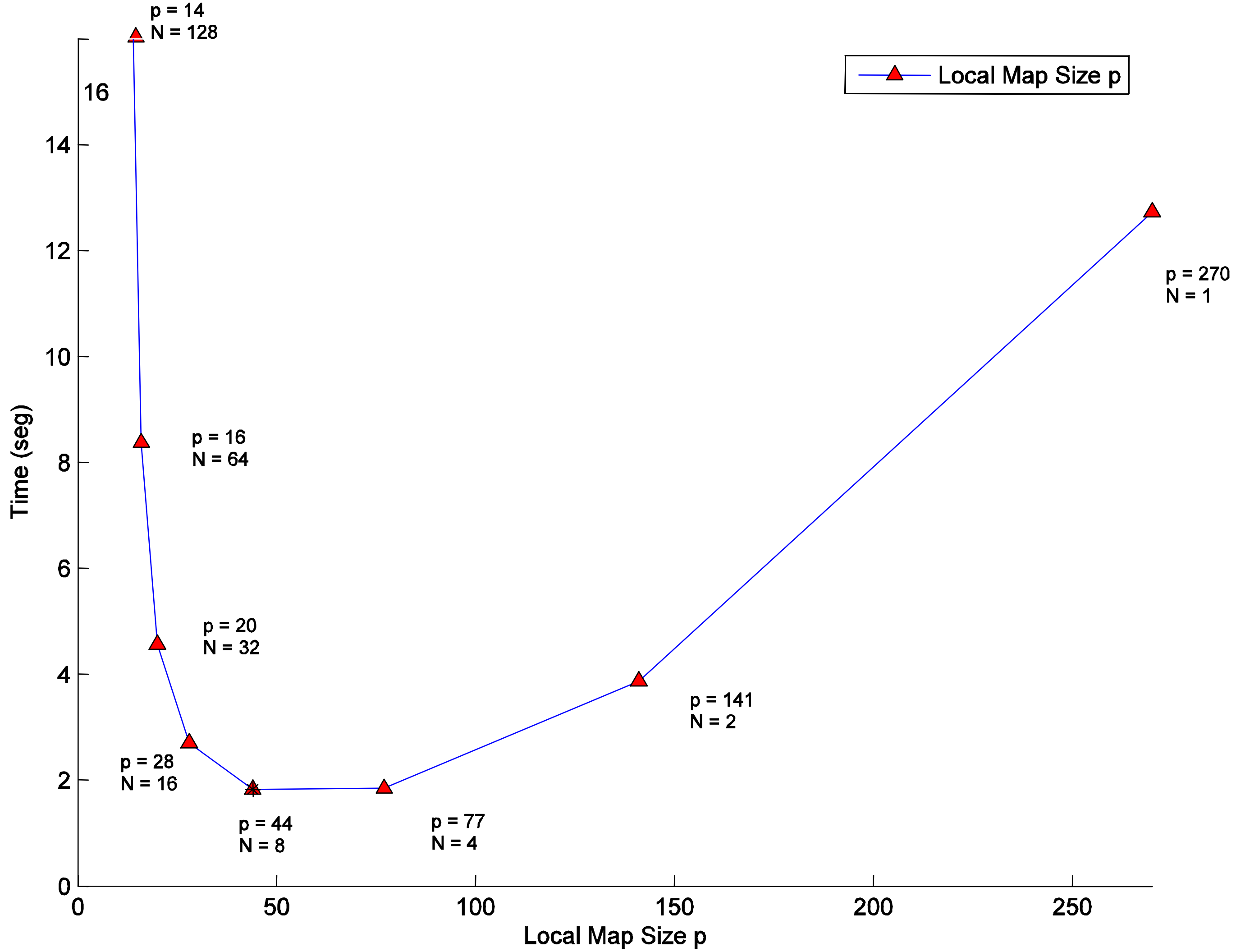
Map Joining



Map joining is $O(n^2)$

Local map size

- Size matters!

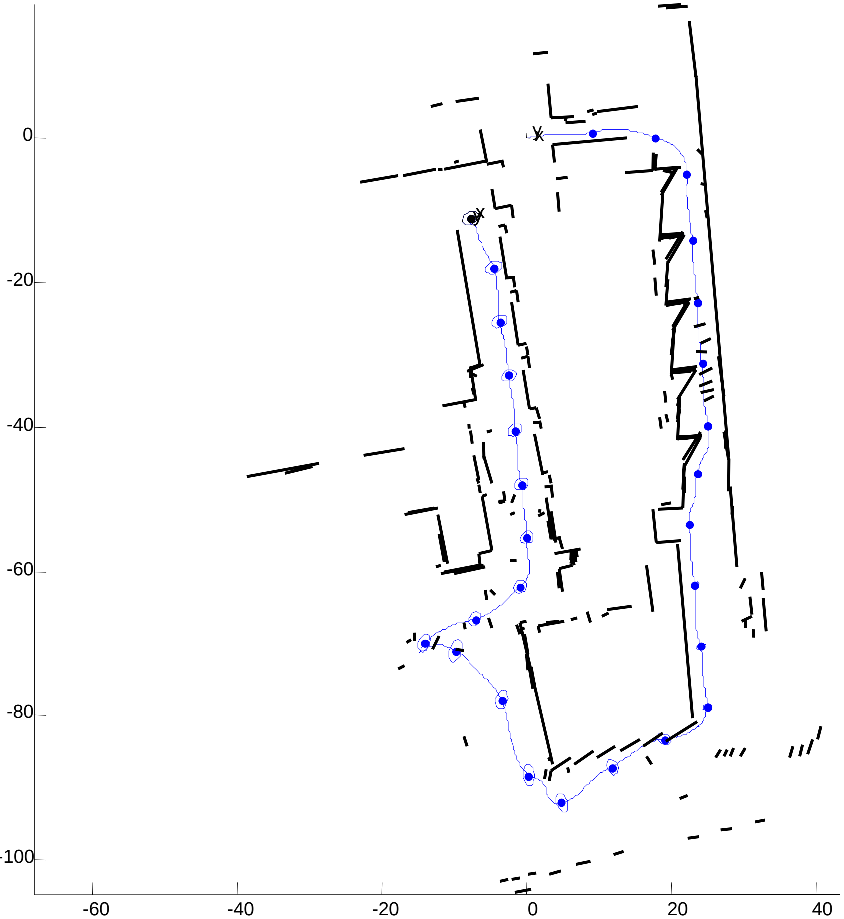


Computational cost for different local map sizes

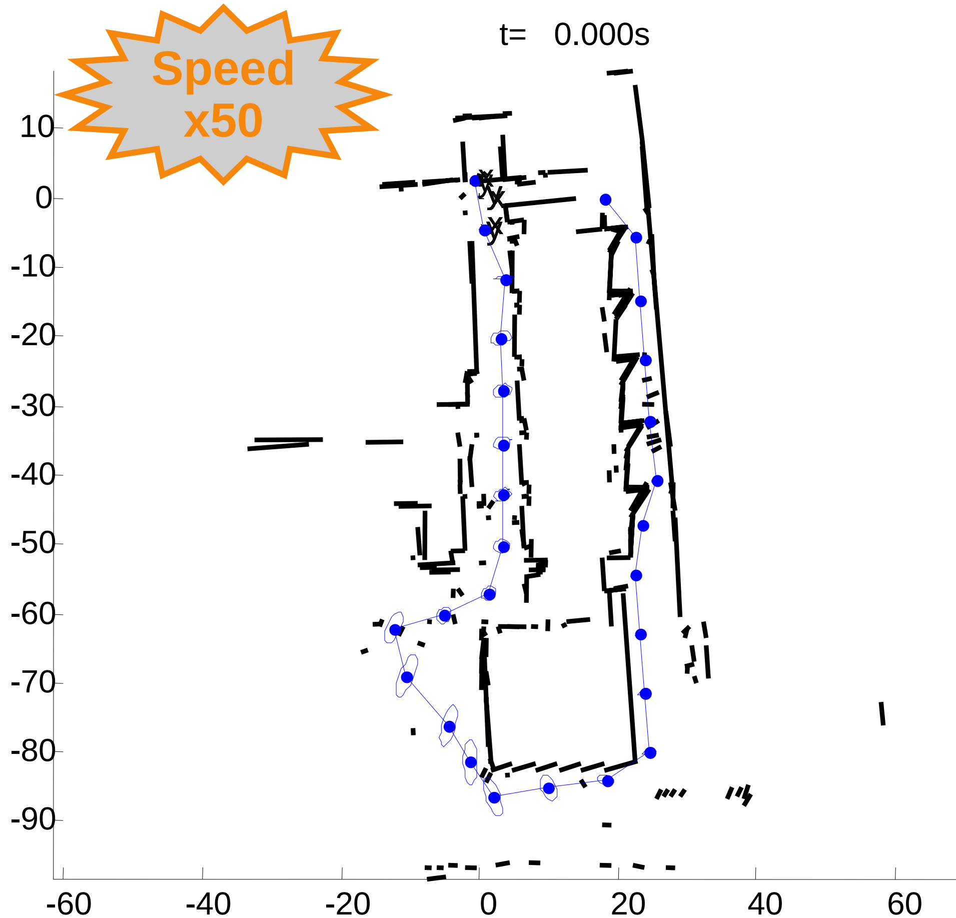
ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Map Joining closes the loop!

- One full SLAM run



- Map joining of 28 local maps



Local maps bound linearization error effects

Local map size

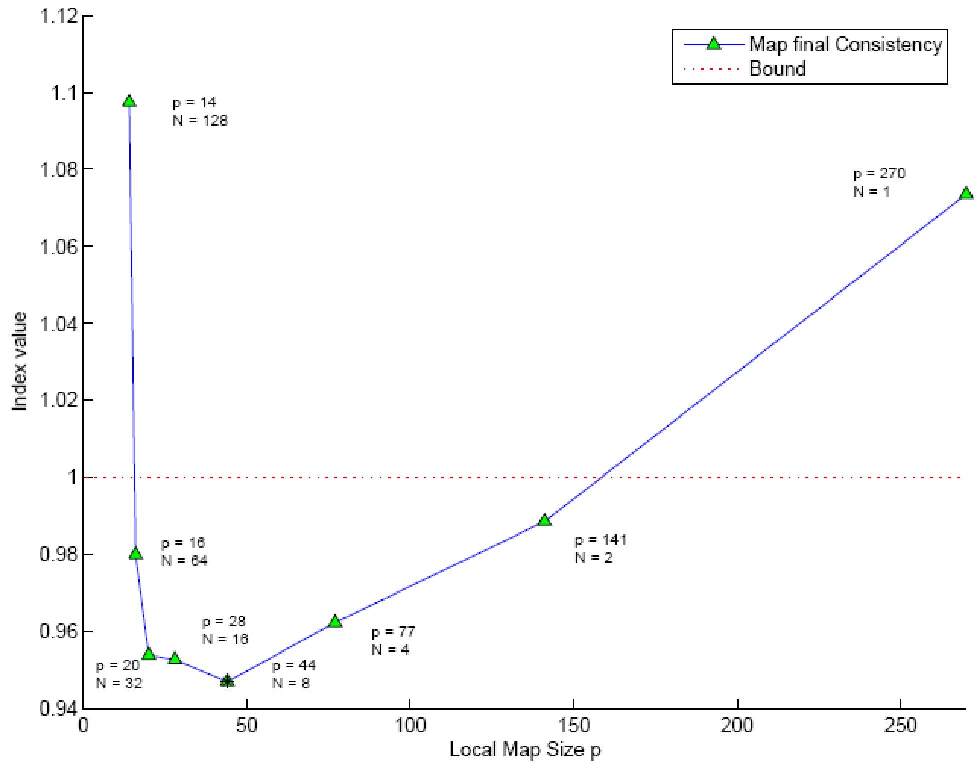
- Monte Carlo runs

$$D^2 = (\mathbf{x} - \hat{\mathbf{x}})^T \; \mathbf{P}^{-1} \; (\mathbf{x} - \hat{\mathbf{x}})$$

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

- Consistency index

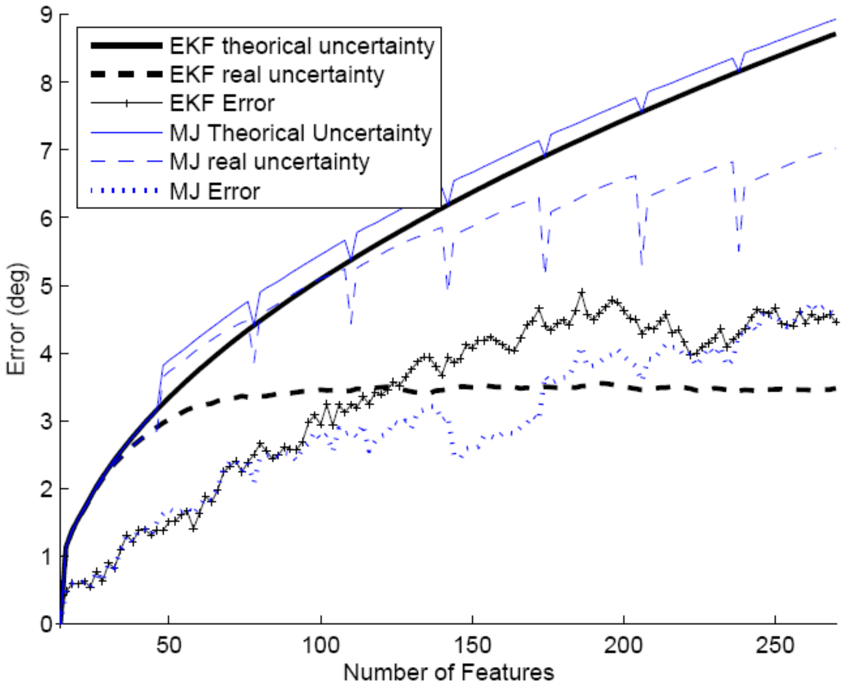
$$CI = \frac{D^2}{\chi^2_{r,1-\alpha}}$$



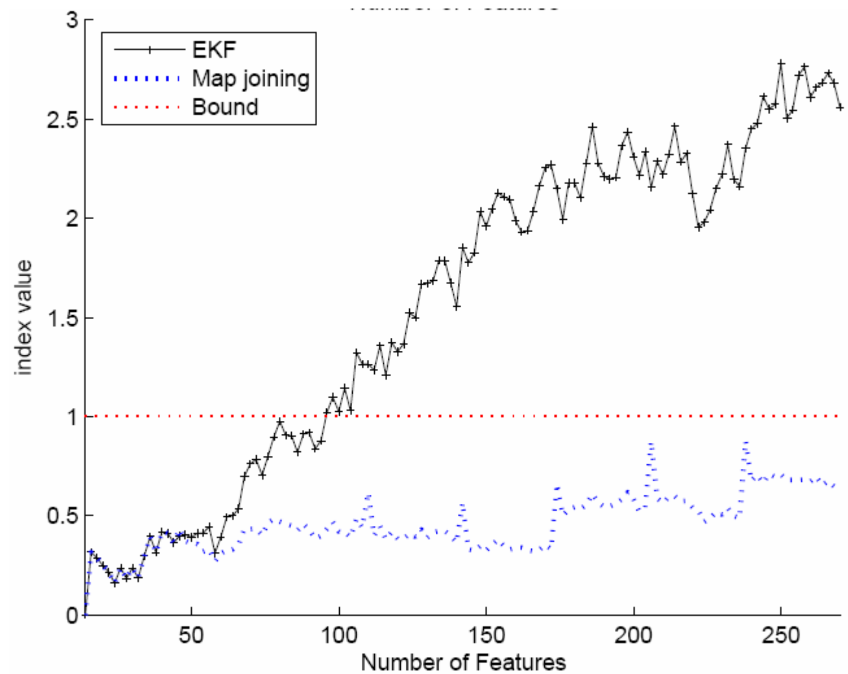
Consistency index for different map sizes

Local map size

- Mean vehicle orientation error for full EKF and Map Joining 8 local maps

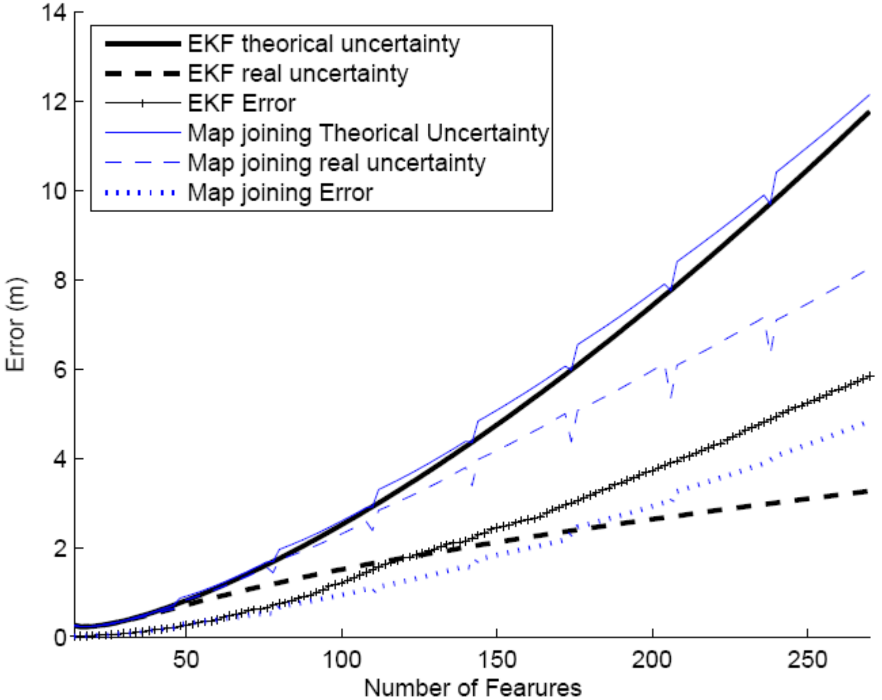


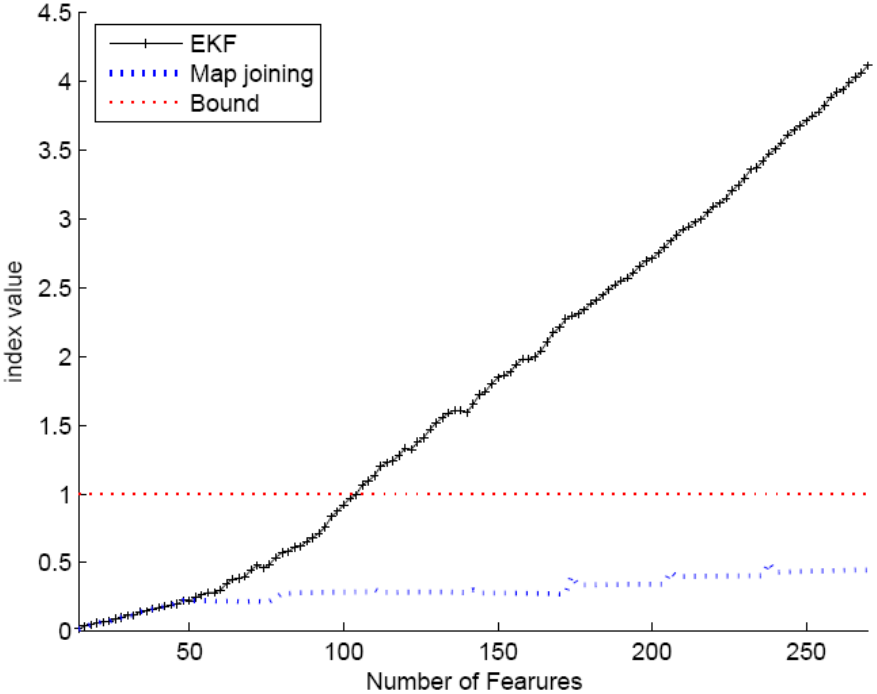
Vehicle orientation error



Vehicle orientation consistency index

Local map size



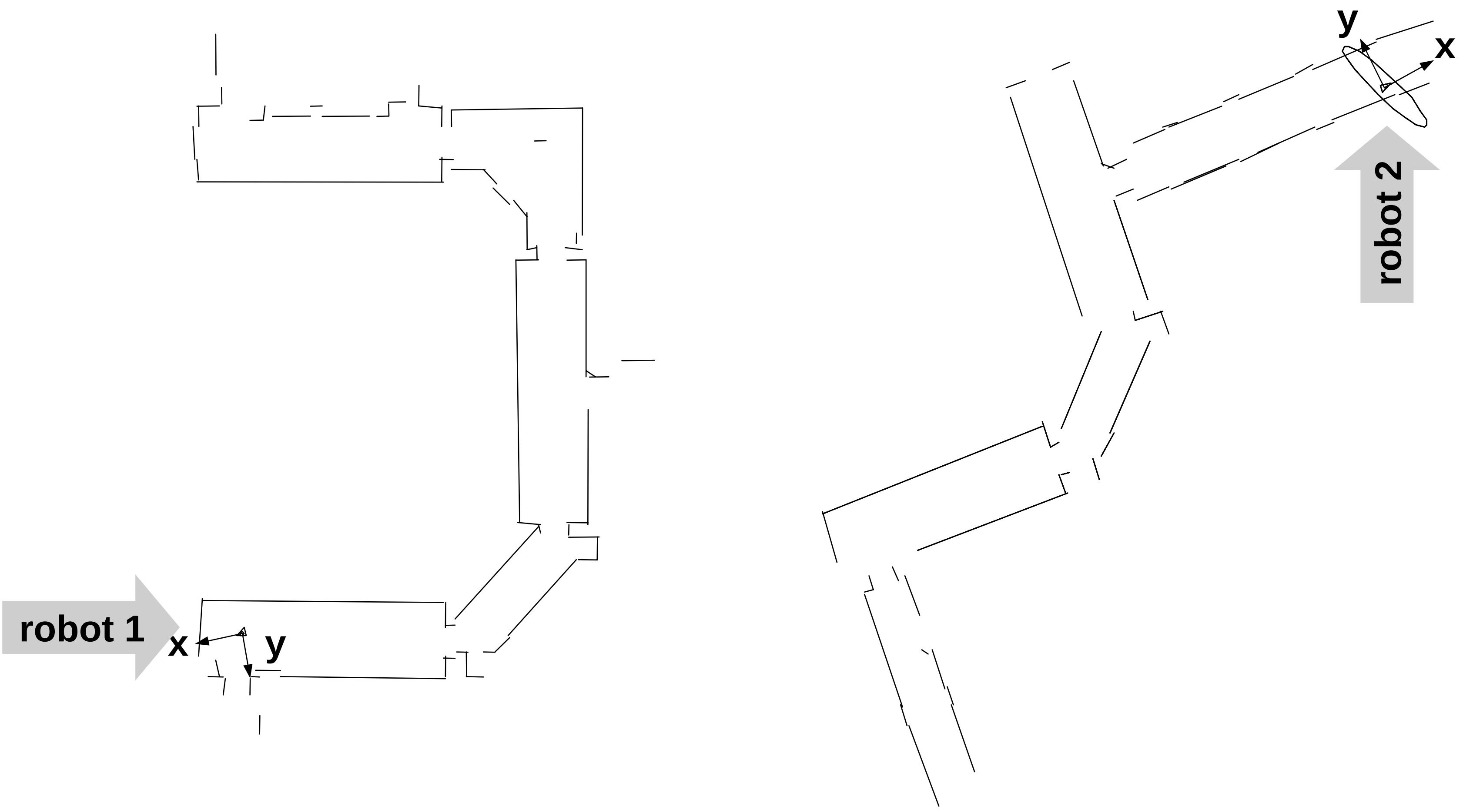


Mean feature position error

Mean feature consistency index

Multivehicle SLAM

Multirobot map building

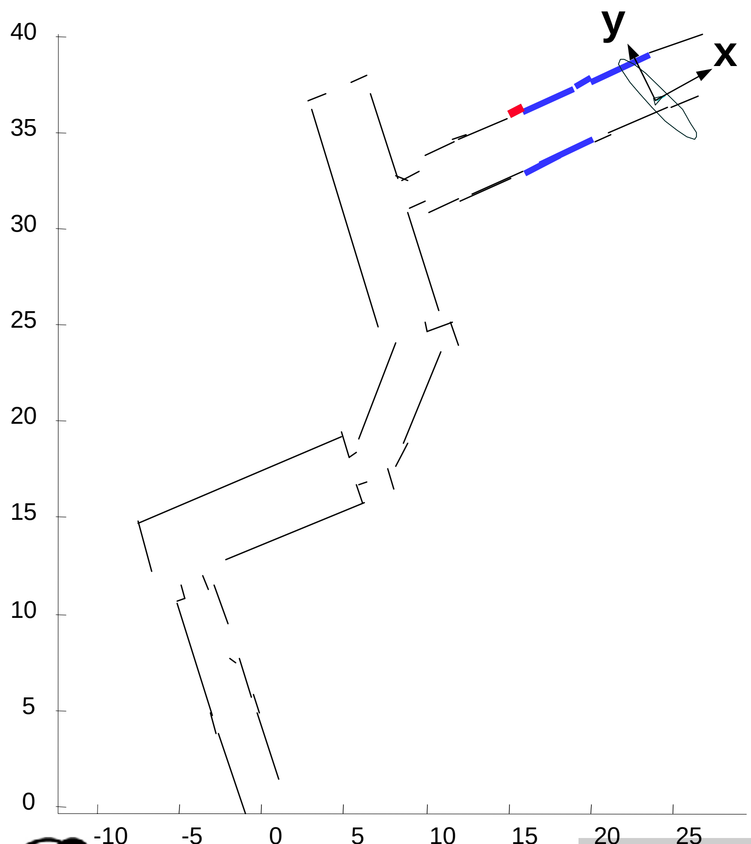


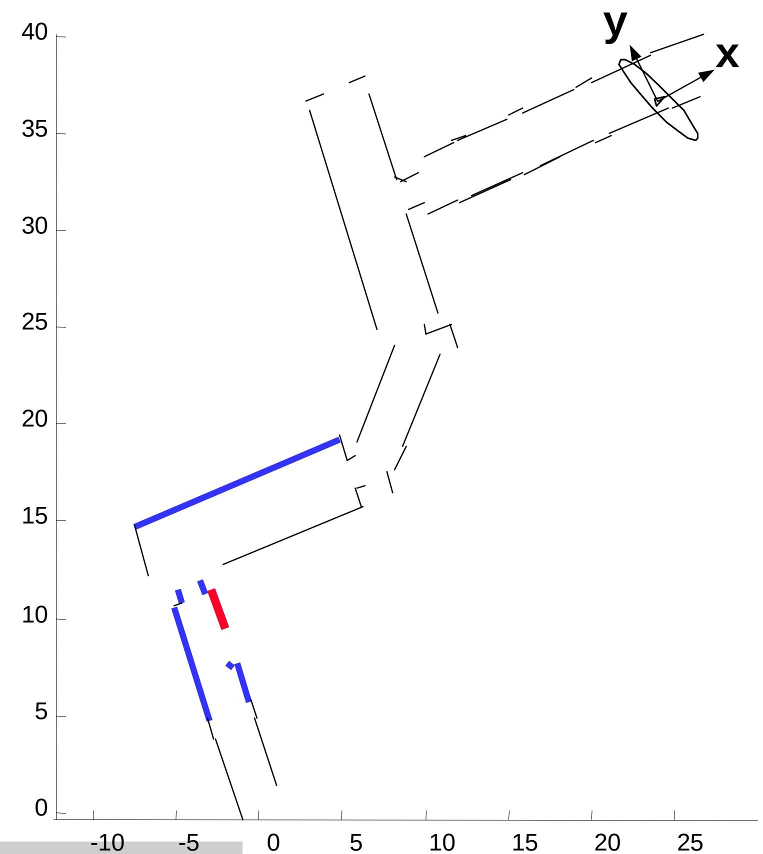
Match, join and fuse to one full stochastic map

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Multirobot map building

- Randomly select a **feature** in one map
- Try to associate its **covisible features** in the other map



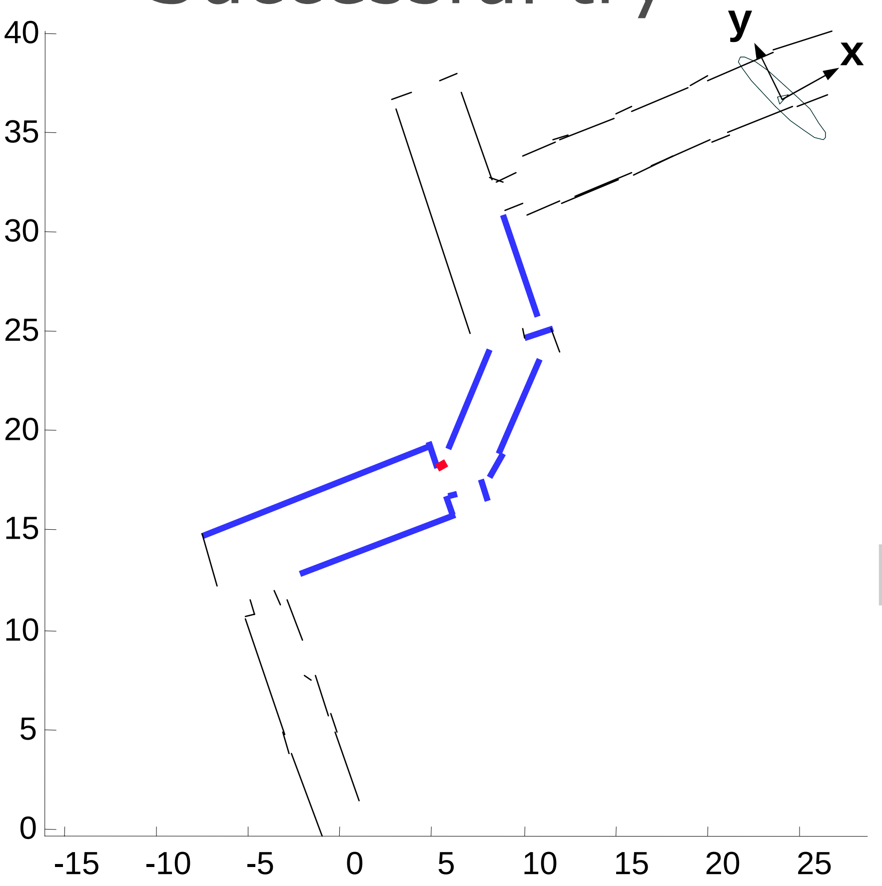


Unsuccessful tries

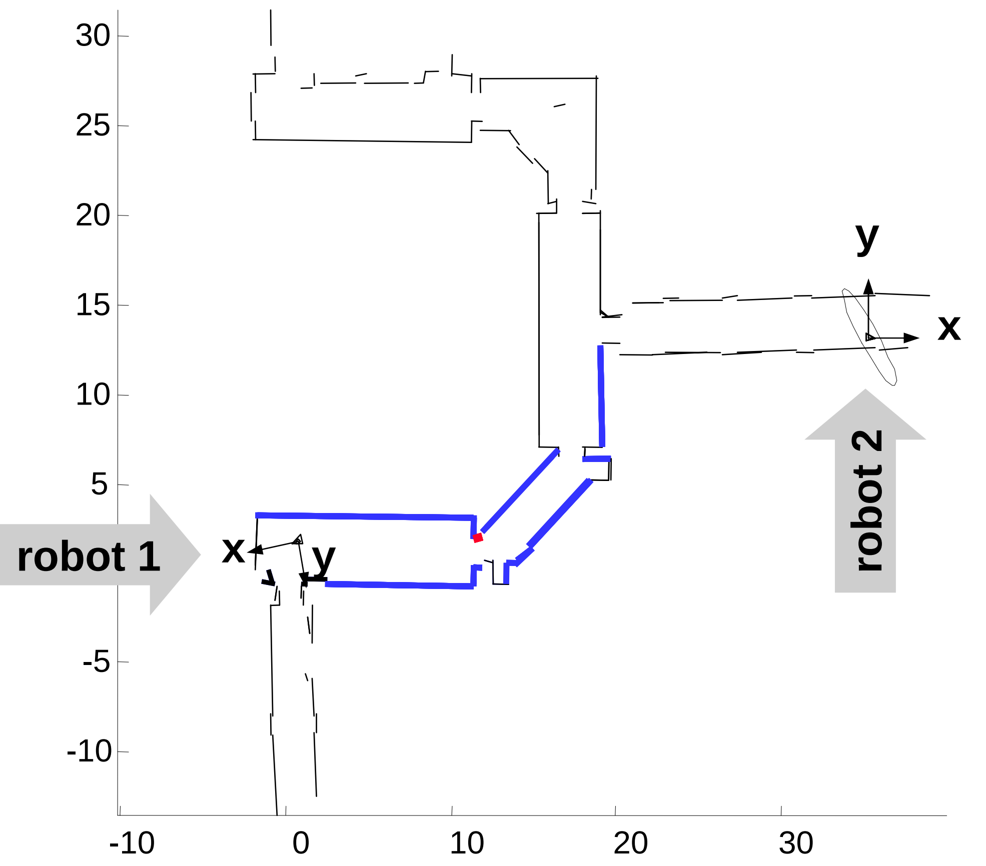
ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Multirobot map building

- Successful try

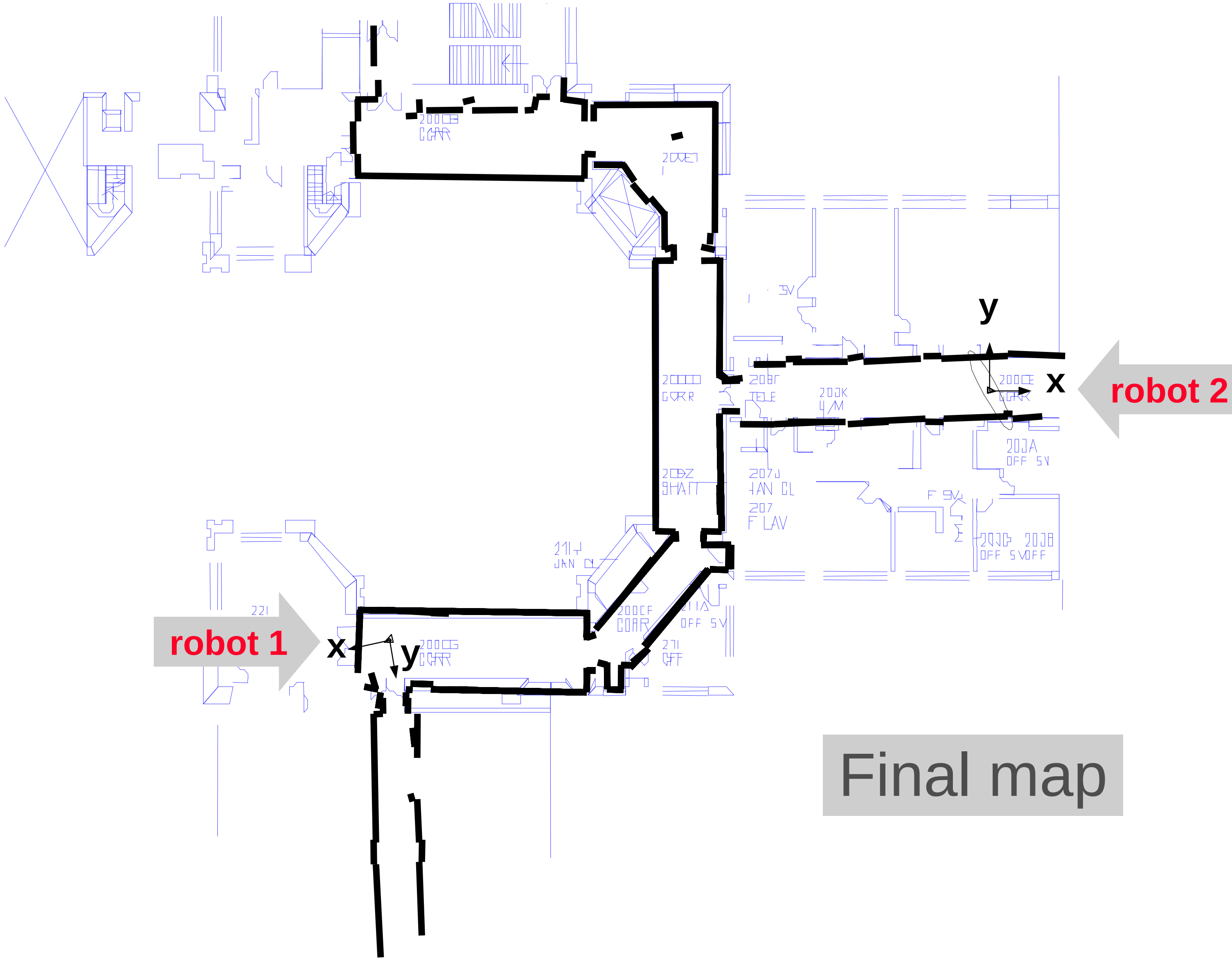


- Final map



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

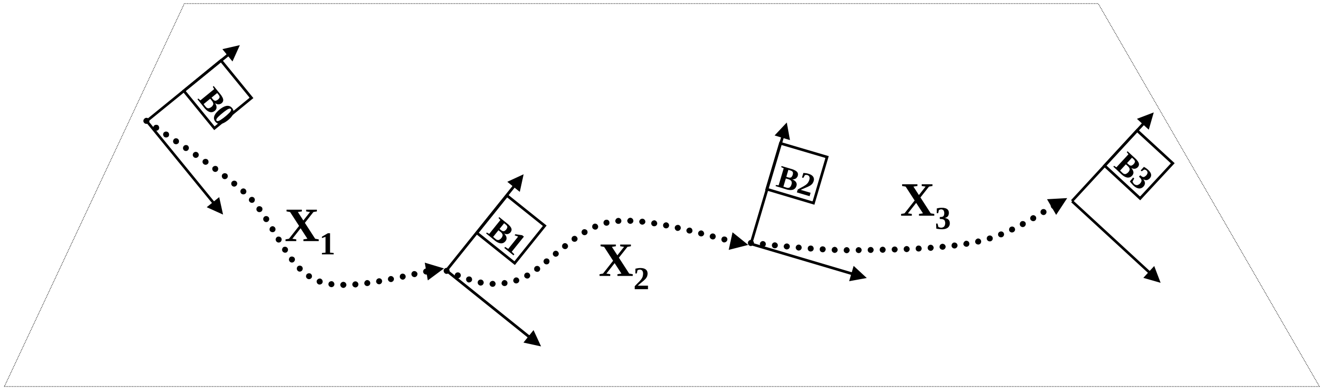
Multivehicle SLAM



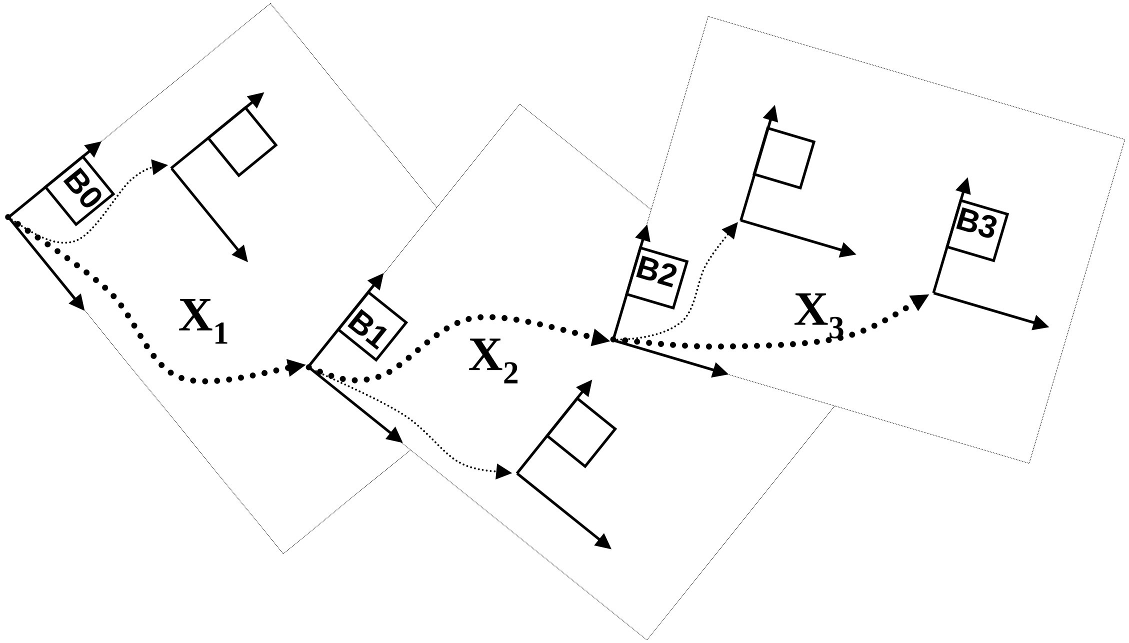
Hierarchical SLAM

Hierarchical SLAM

- Global level: adjacency graph and relative stochastic map



- Local level: statistically independent local maps



Hierarchical SLAM

- Local maps:

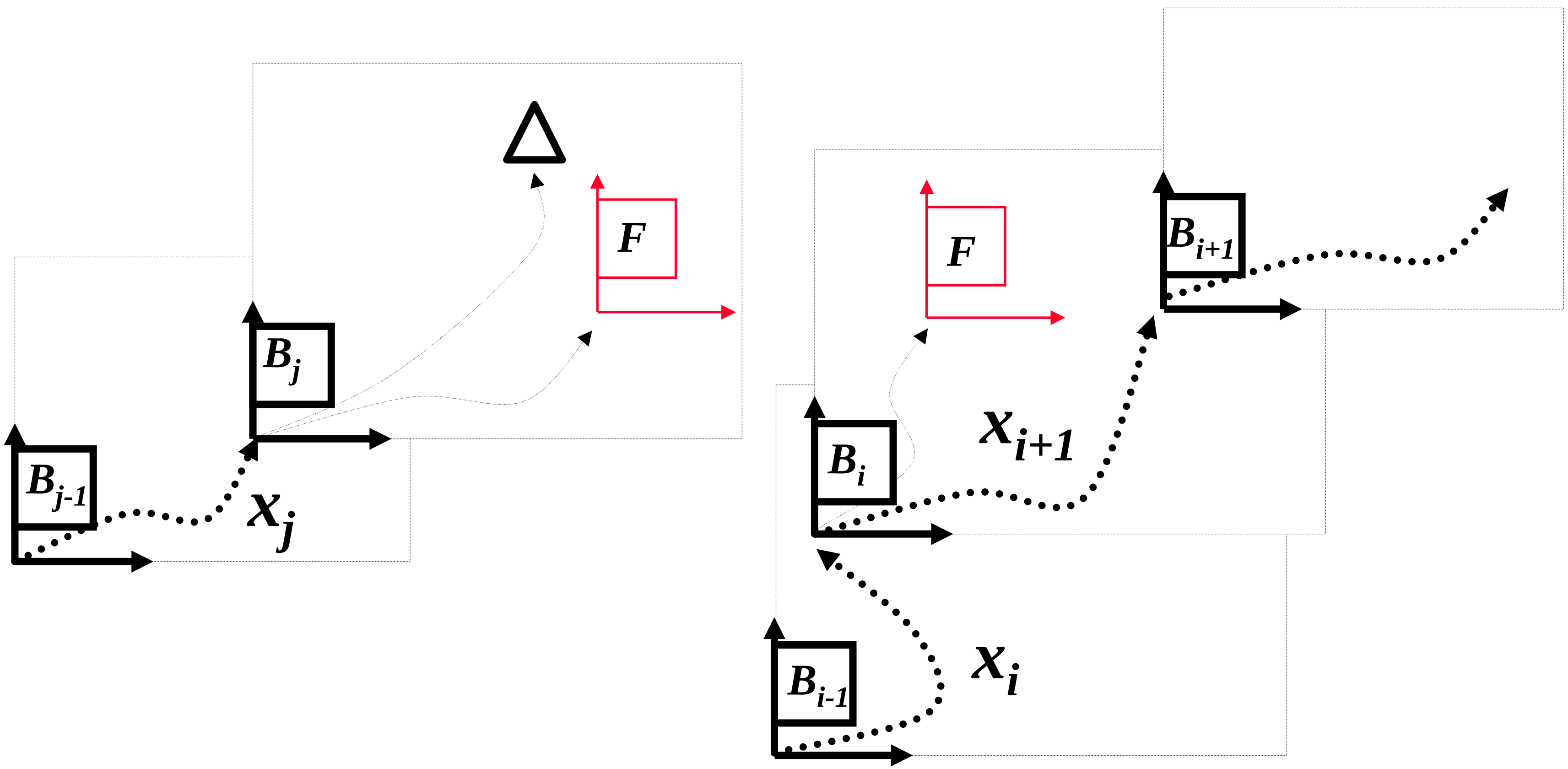
$$\hat{\mathbf{x}}_{\mathcal{F}}^{B} = \begin{bmatrix} \hat{\mathbf{x}}_{R}^{B} \\ \vdots \\ \hat{\mathbf{x}}_{F_n}^{B} \end{bmatrix} ; \; \mathbf{P}_{\mathcal{F}} = \begin{bmatrix} \mathbf{P}_{R}^{B} & \cdots & \mathbf{P}_{RF_n}^{B} \\ \vdots & \ddots & \vdots \\ \mathbf{P}_{F_nR}^{B} & \cdots & \mathbf{P}_{F_nF_n}^{B}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

- Global relative map:

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

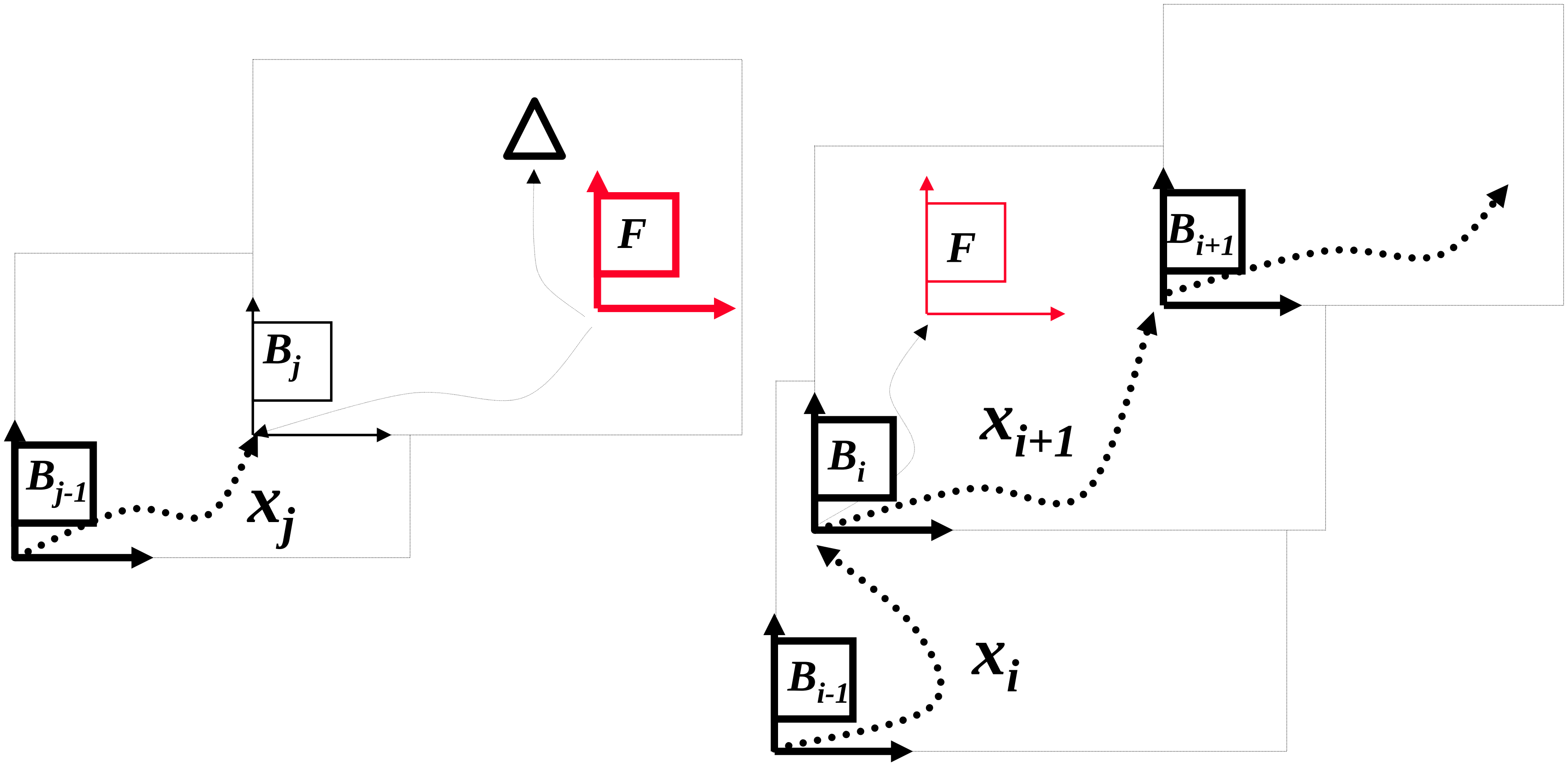
Block diagonal

Loop closing



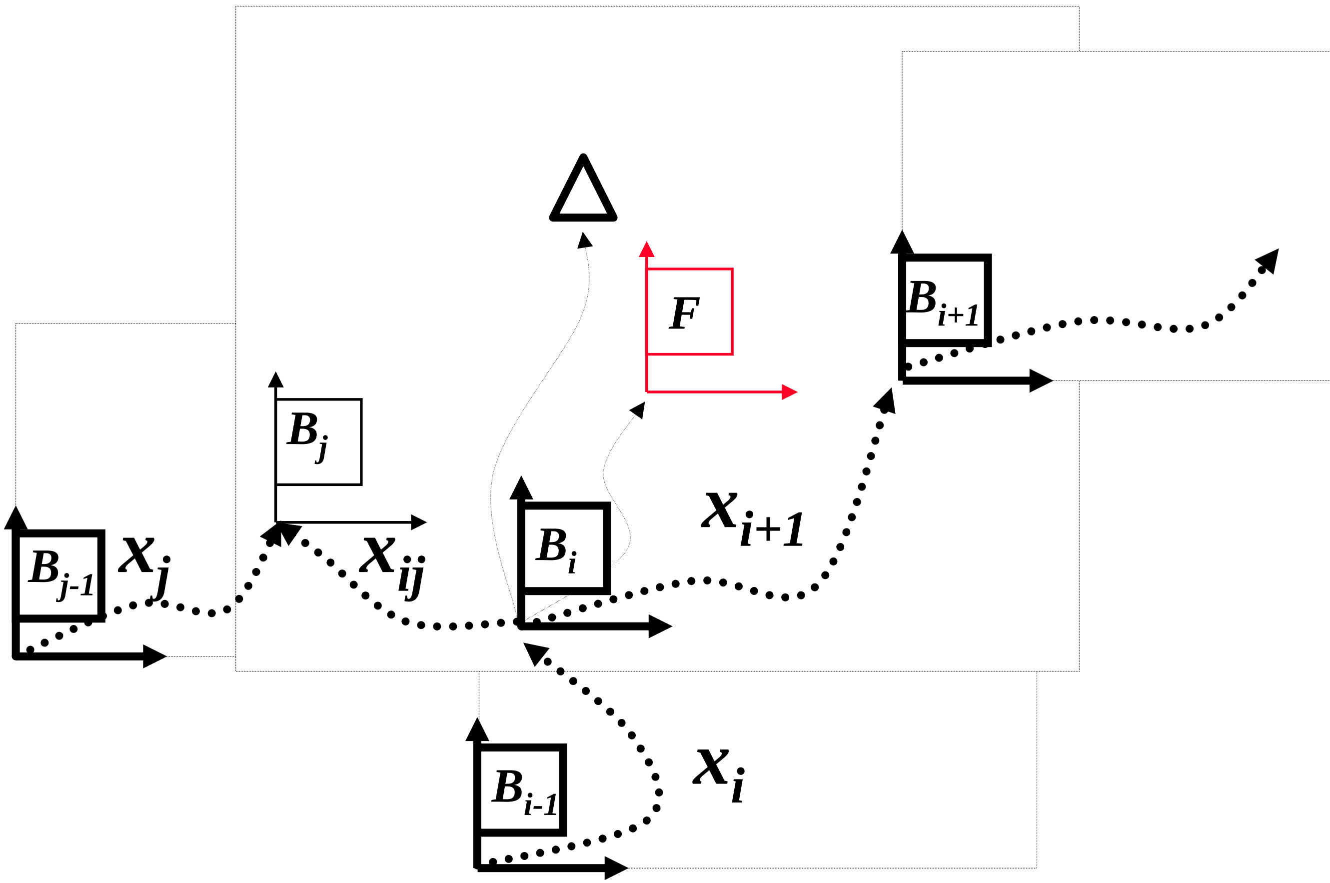
Find a common reference between maps *i* and *j*

Loop closing



Change base reference of map *i* to *F*.

Loop closing



Join maps *i* and *j*.

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Hierarchical SLAM

- Global relative map before loop closing:

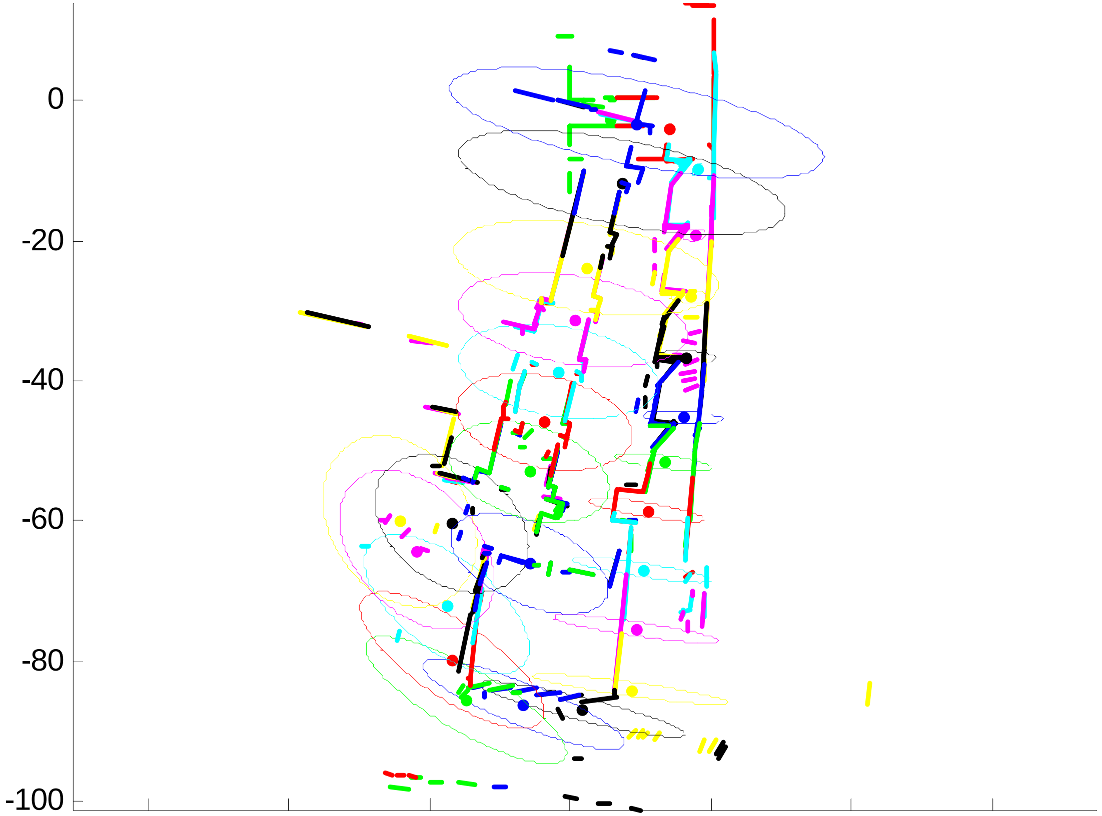
$$\hat{\mathbf{x}} = \begin{bmatrix} \vdots \\ \hat{\mathbf{x}}_{i+1} \\ \vdots \\ \hat{\mathbf{x}}_j \end{bmatrix} ; \ \mathbf{P} = \begin{bmatrix} . & 0 & 0 & 0 \\ 0 & \mathbf{P}_{i+1} & 0 & 0 \\ 0 & 0 & . & 0 \\ 0 & 0 & 0 & \mathbf{P}_j \end{bmatrix}$$

- After loop closing:

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

ROBOTICS UNIVERSIDAD DE ZARAGOZA

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) :

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

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

» Iterate until convergence

ROBOTICS UNIVERSIDAD DE ZARAGOZA

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:

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

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

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Solution 2 (for EKF fans)

- We could impose the loop constraints using an imprecise measurement function:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{w} = \mathbf{0}$$

- With Covariance:

$$\mathbf{P}_z = Cov(\mathbf{w}) = \begin{bmatrix} \mathbf{P}_{z_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{z_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P}_{z_l} \end{bmatrix}$$

- Estimated errors in closing the loops:

$$\mathbf{h}(\hat{\mathbf{x}}) = \begin{bmatrix} \mathbf{h}_1(\hat{\mathbf{x}}) \\ \mathbf{h}_2(\hat{\mathbf{x}}) \\ \vdots \\ \mathbf{h}_l(\hat{\mathbf{x}}) \end{bmatrix}$$

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Iterated Extended Kalman Filter

- Jacobian of the measurement function:

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

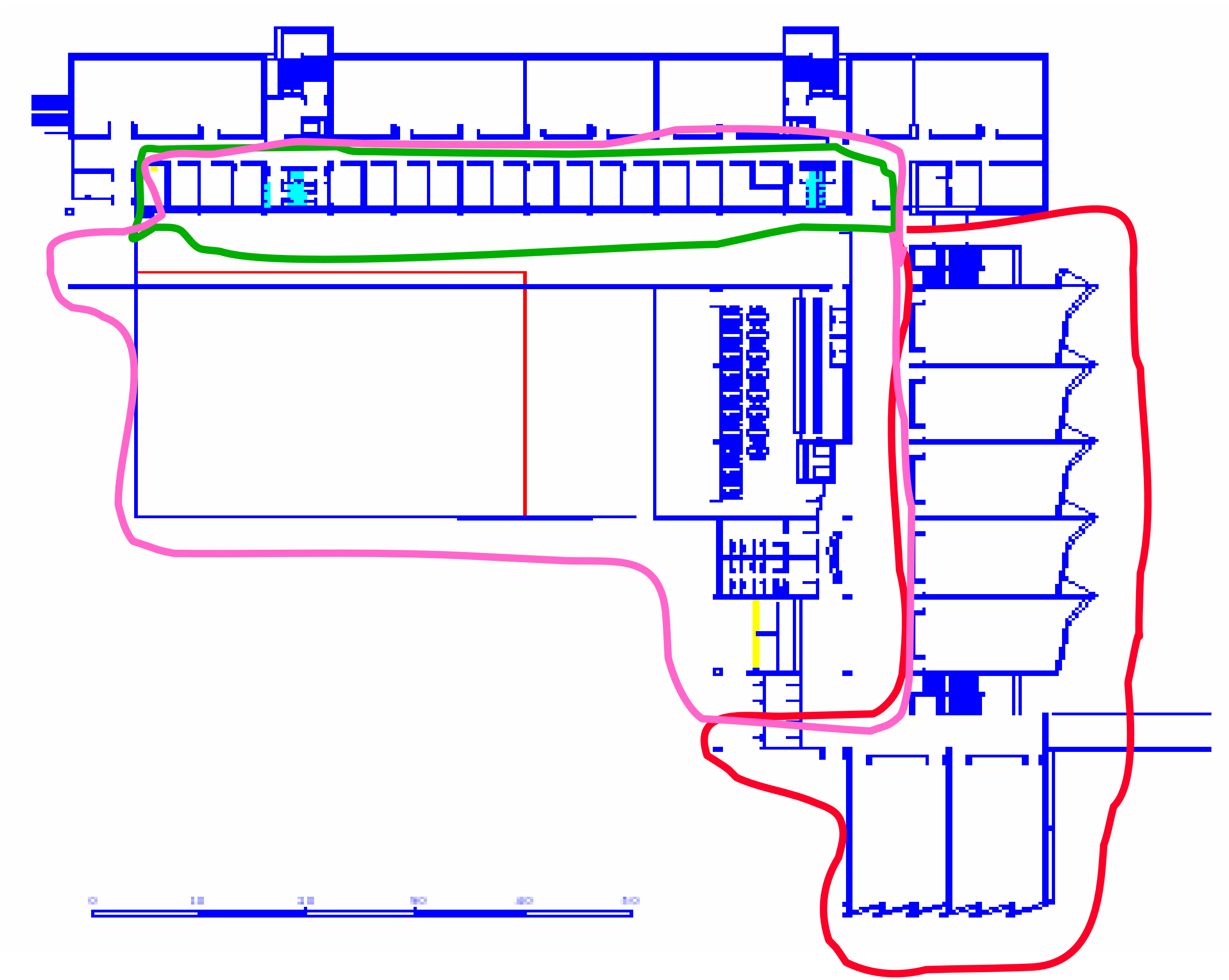
- Iterated EKF equations:

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

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

Multivehicle SLAM

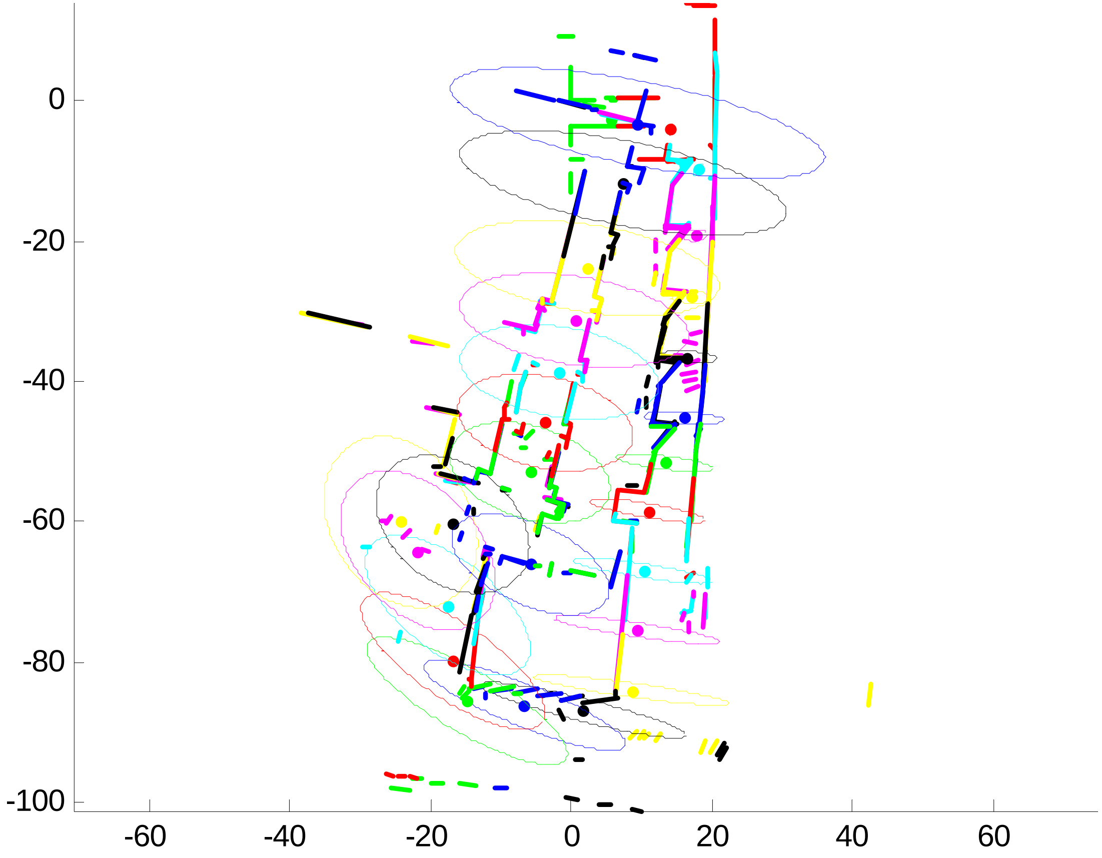
- Experiment



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

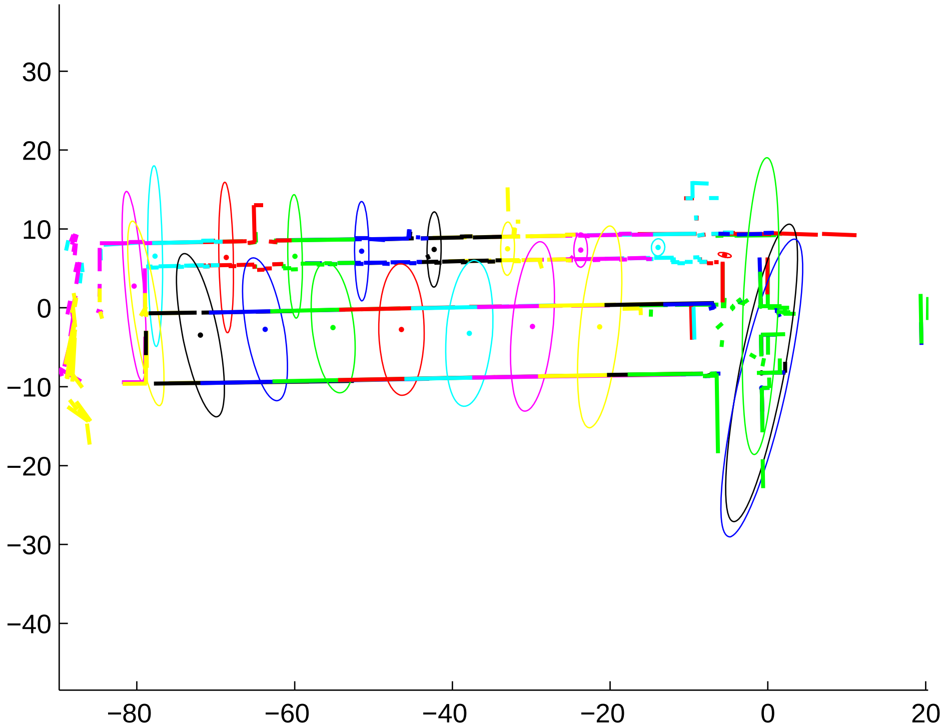
Experiments

- Local maps, first robot



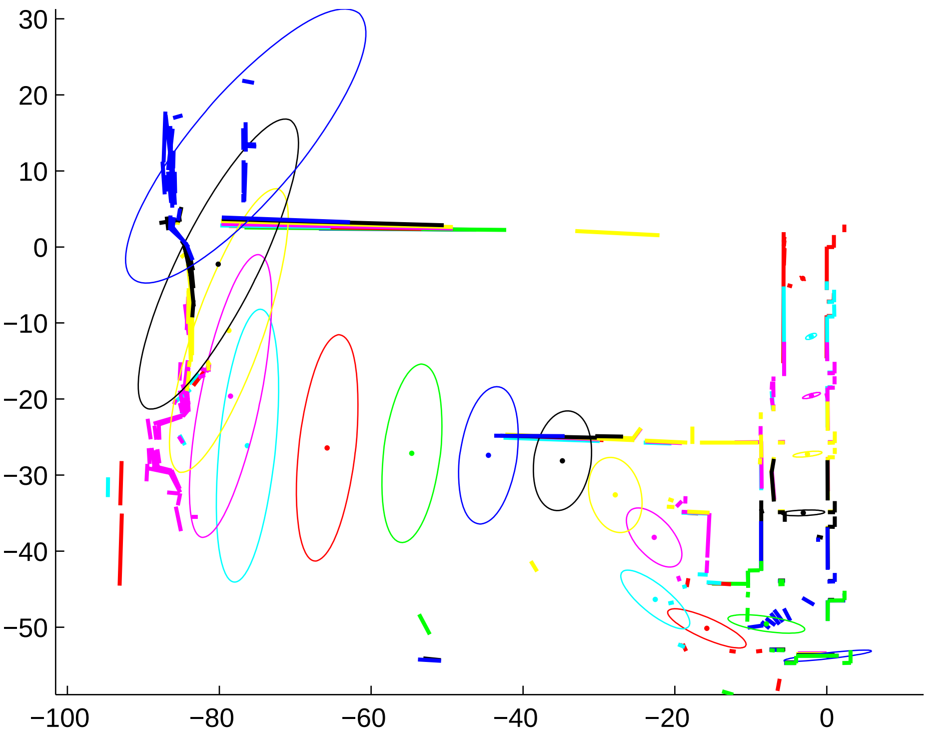
Experiments

- Local maps, second robot



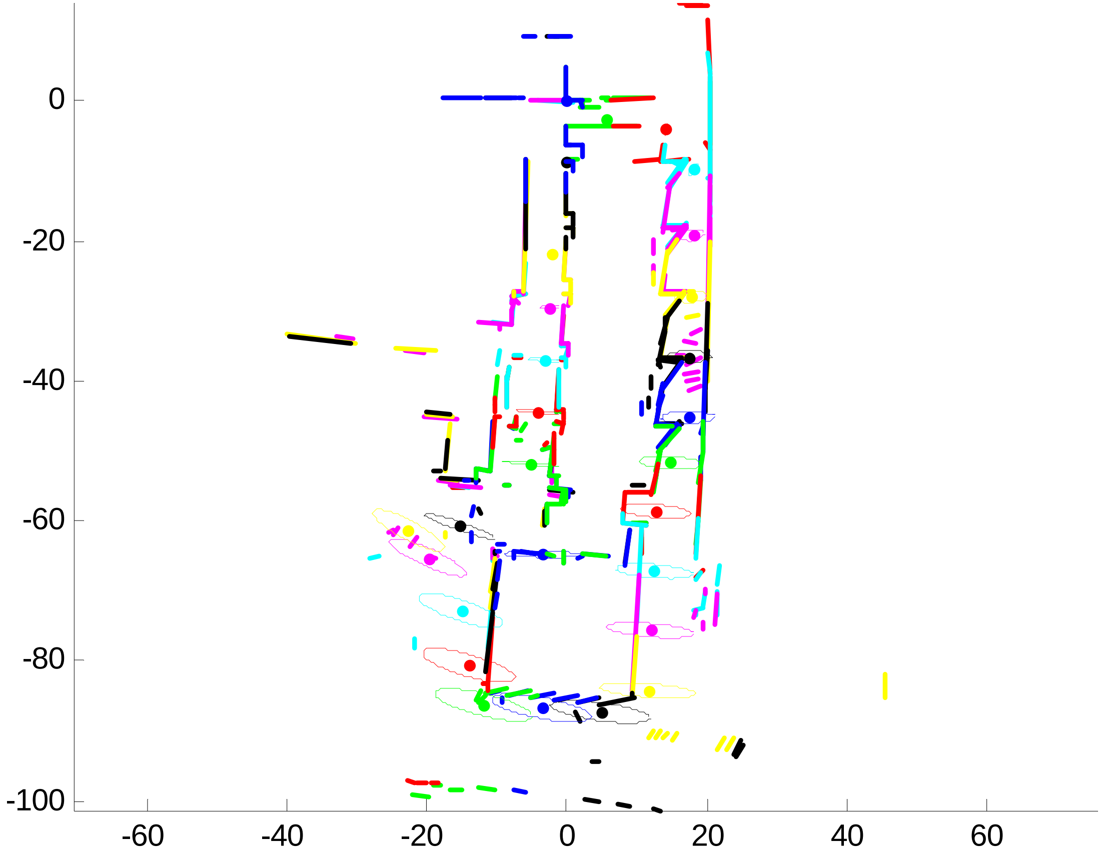
Experiments

- Local maps, third robot



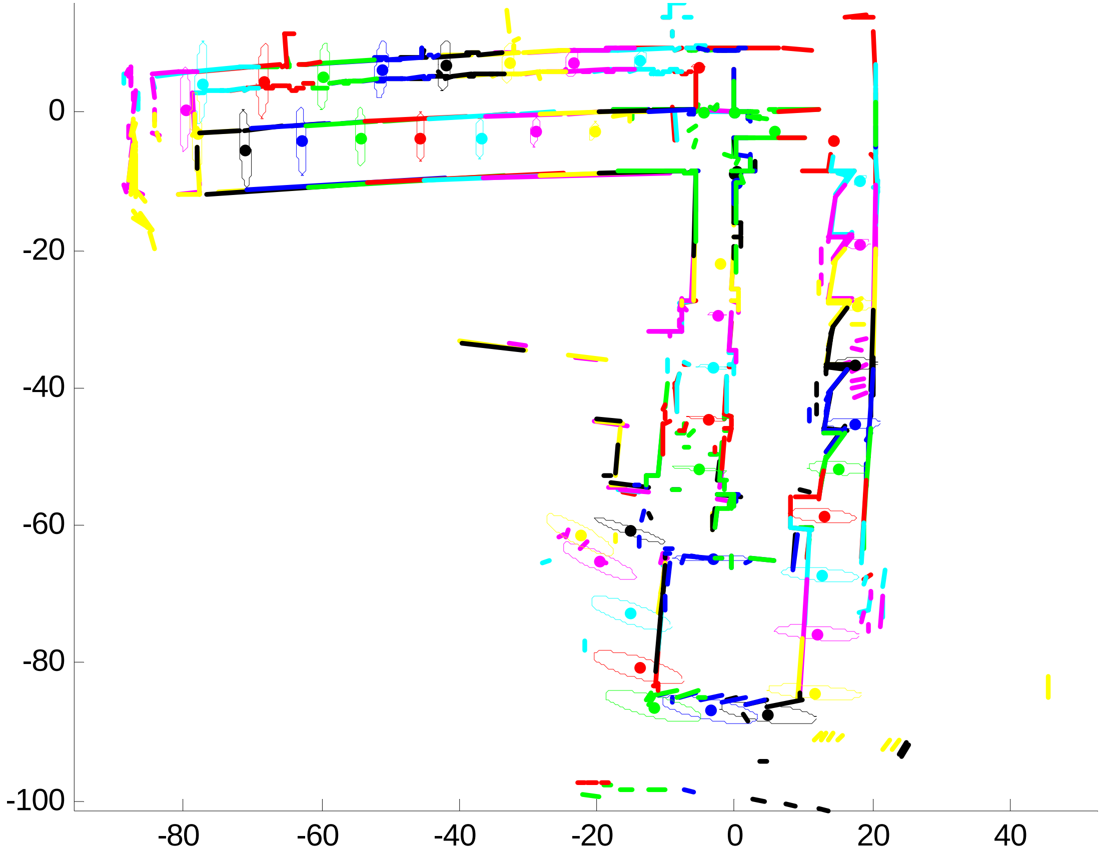
Hierarchical SLAM

- Imposing loop constraint:



Hierarchical SLAM

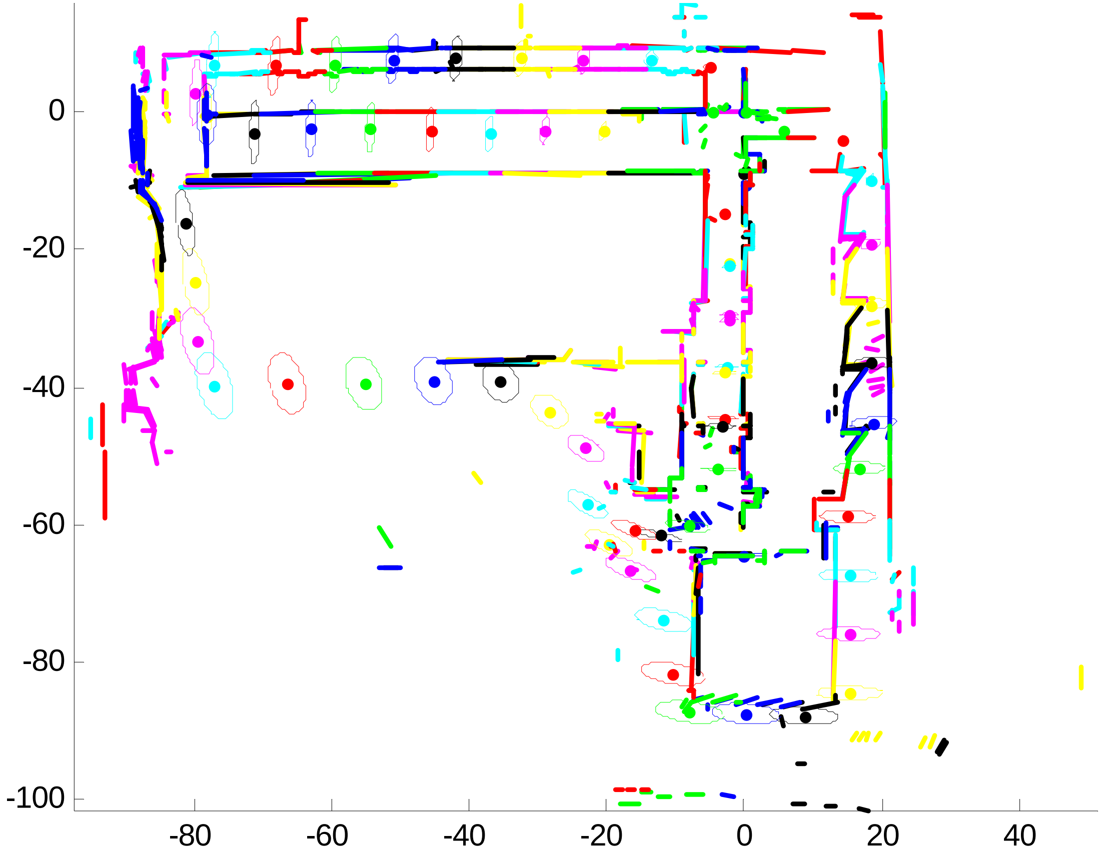
- Second robot



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Hierarchical SLAM

- Third robot



ROBOTICS
UNIVERSIDAD DE ZARAGOZA

Results in less than 1s

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

Conclusions

- EKF-SLAM is only consistent for:
  - The linear case (1D robot)
  - Small scale maps (< 100m)
- Inconsistency only becomes evident if:
  - Ground truth is available
  - Trying to close a big loop (> 100m)
- The consistency problem appears before the computational complexity problem !
- Hierarchical SLAM has a **robust, stable and local parametrization** that allows to efficiently maintain loop consistency for large loops

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