Pure Visual SLAM

José Neira
Universidad de Zaragoza

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

Joint work with:
Andrew Davison, Lina Paz, Pedro Piníes,
Ian Reid, Juan Tardós

Outline

1. Motivation

2. Monocular SLAM
    1. Feature representation
    2. Data association
    3. Hierarchical SLAM algorithm

3. SLAM using only stereo
    1. Feature representation
    2. D&C SLAM algorithm

4. Conclusions

Motivation

SLAM seeks to answer this question:

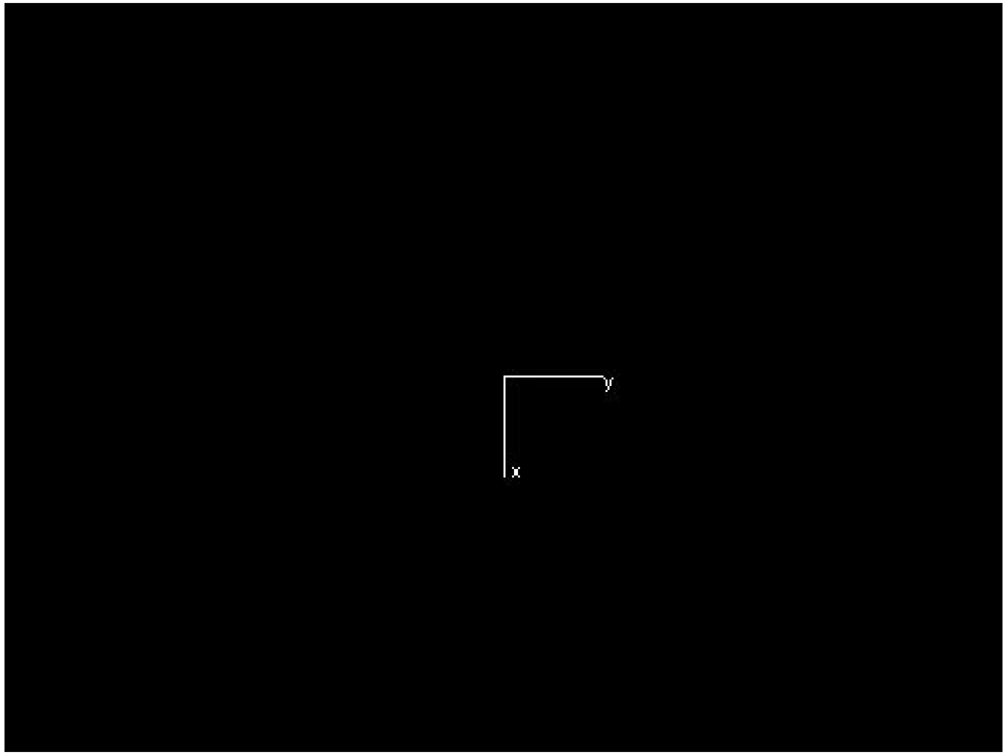
Is it possible to use a vehicle, starting at an

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

build a map of the environment,

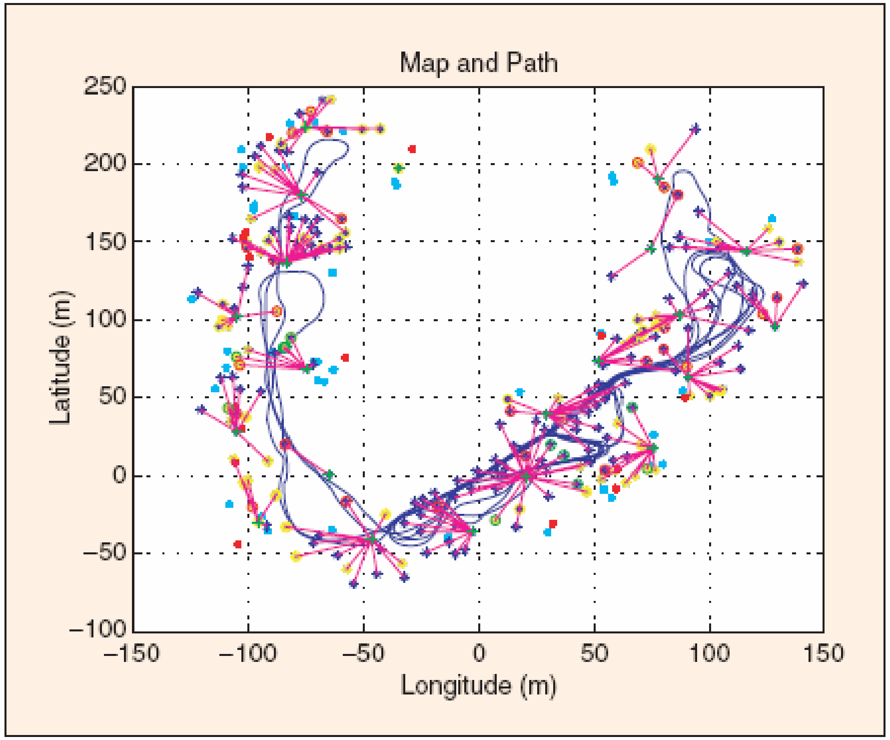
- and **at the same time**

use the map to determine the vehicle location?



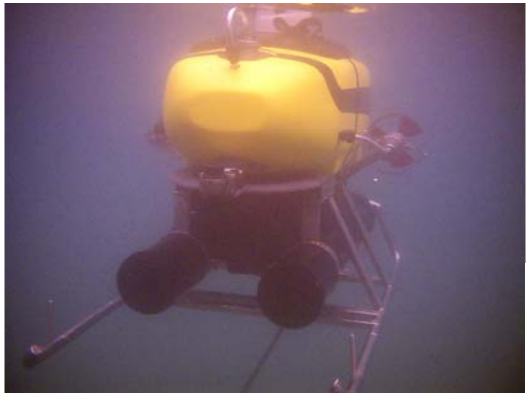
(video: Paul Newman)

Outdoor vehicles



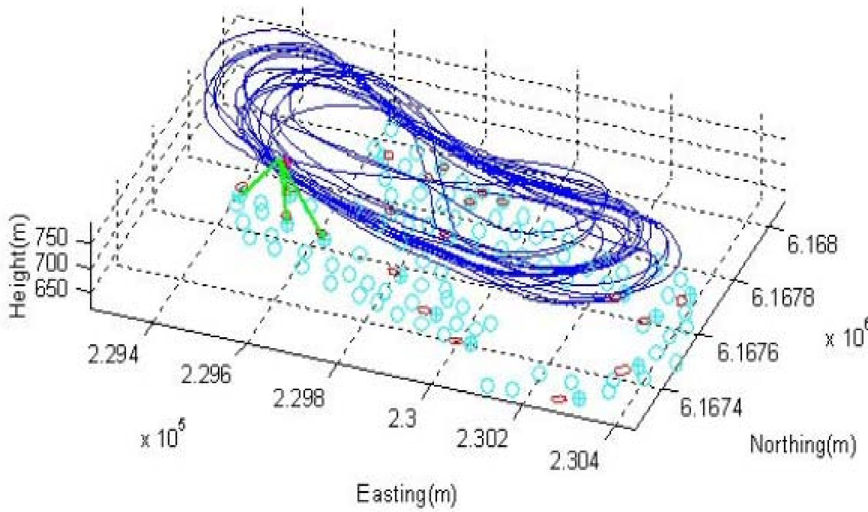
Victoria Park, Univ. Syndey

Underwater, Airborne



Garbi, Univ. Girona, Spain

Brumby, Univ. Syndey



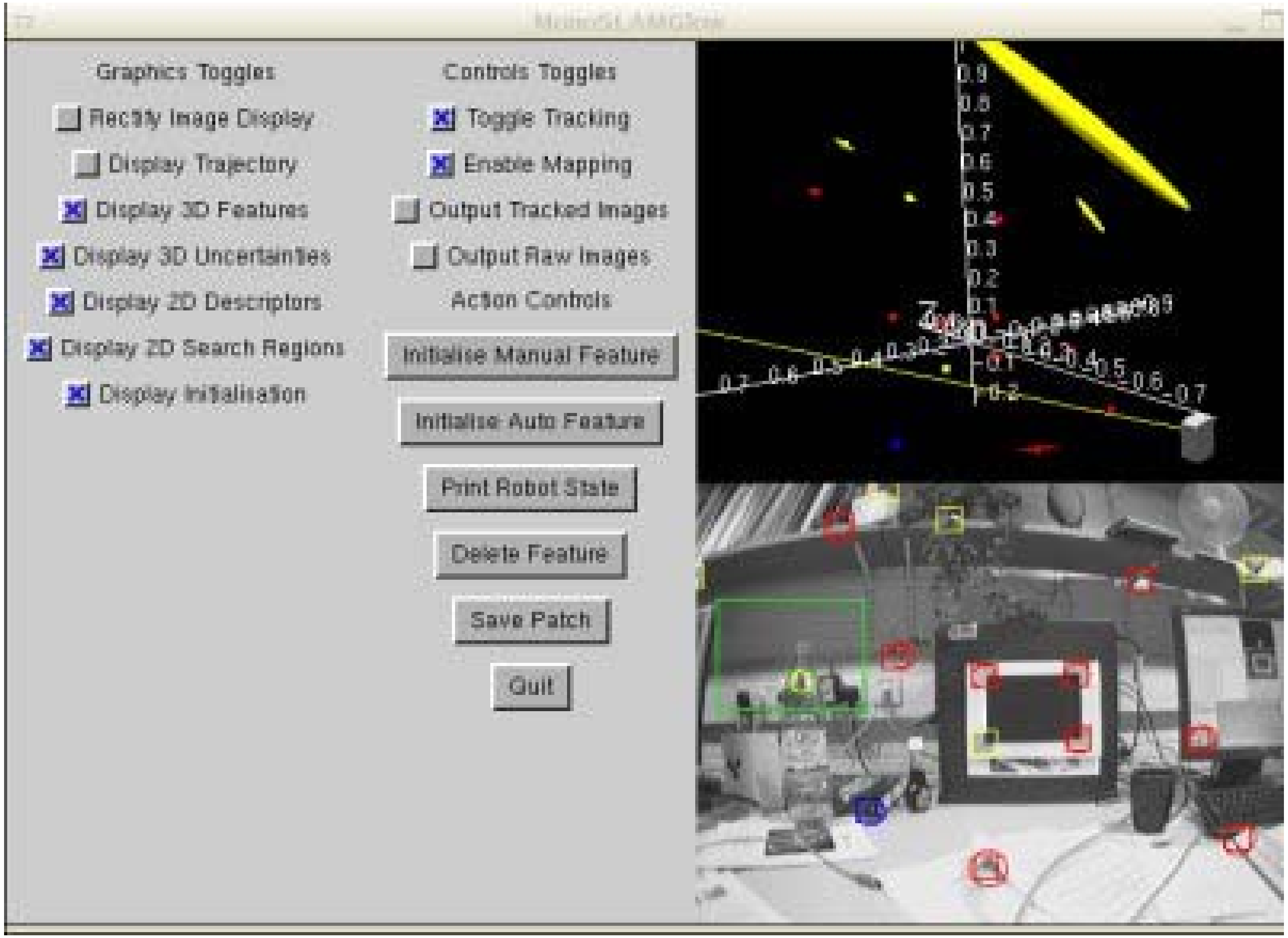
Monocular SLAM

A Unibrain fire-I camera, a laptop and a firewire cable

290 m.

ROBOTICS UNIVERSIDAD DE ZARAGOZA

Monoslam (A. Davison)



Basic EKF SLAM

EKF state vector:

- Camera state
- Features state

Camera state:

- Position
- Orientation
- Linear Velocity
- Angular Velocity

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_c \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \dots \\ \mathbf{y}_n \end{pmatrix} \qquad \mathbf{x}_c = \begin{pmatrix} \mathbf{r}^{BC} \\ \mathbf{\Psi}^{BC} \\ \mathbf{v}^{B} \\ \mathbf{w}^{C} \end{pmatrix}$$

3D features representation

3D points:

- \- Cartesian coordinates

$$\mathbf{y}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$$

Inverse depth points:

\- Camera position the first time the feature was seen

\- Azimuth

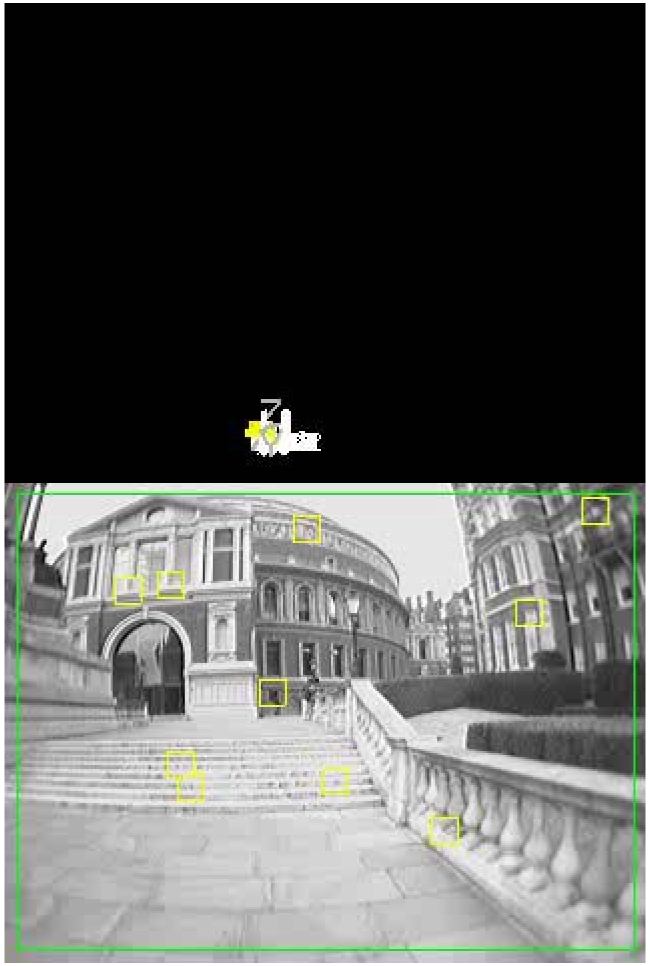
\- Elevation

\- Inverse depth

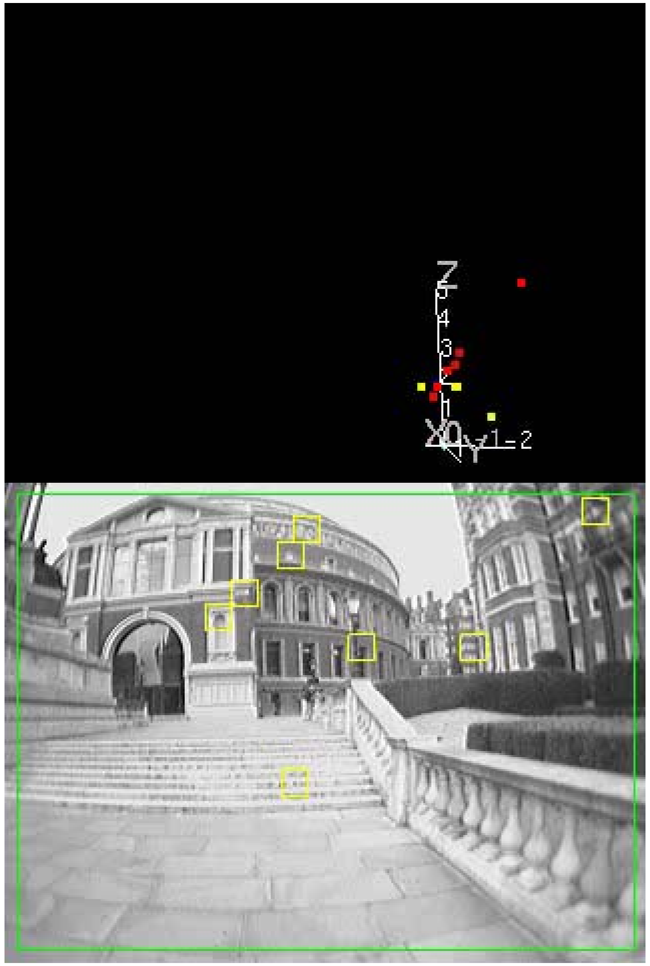
$$\mathbf{y}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ \theta_i \\ \phi_i \\ \rho_i \end{pmatrix}$$

J.M.M. Montiel, J. Civera, A.J. Davison: **Unified inverse depth parametrization for monocular SLAM.** Conditionally accepted, IEEE Transactions on Robotics, 2008.

Data association

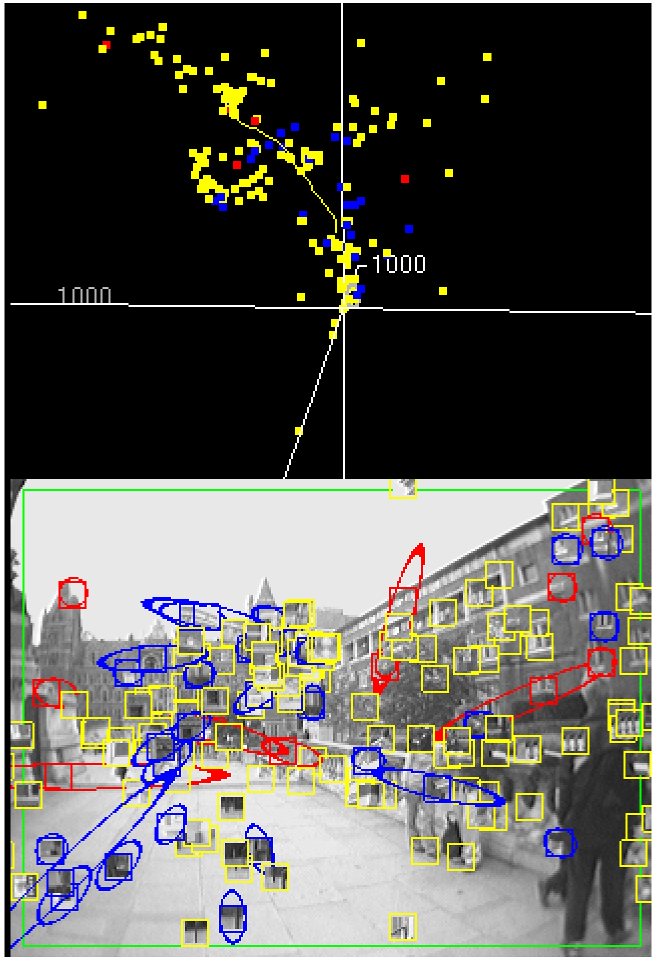


Individual tracks

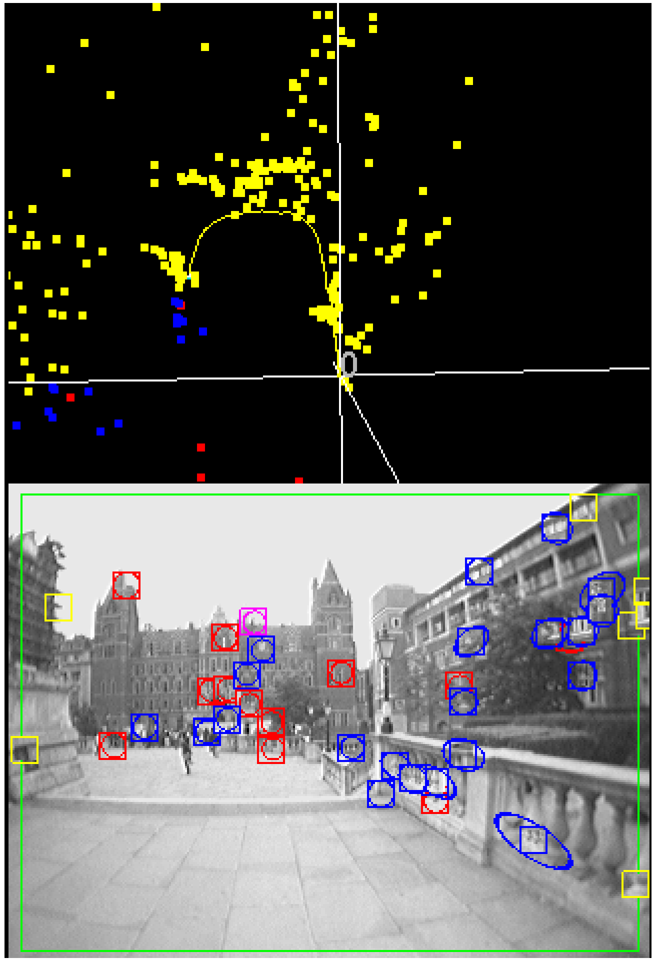


Jointly compatible tracks

Nearest neighbor .vs. Joint Compatibility



Individual Tracks



Jointly Compatible Tracks (cost: 2ms)

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**

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

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

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

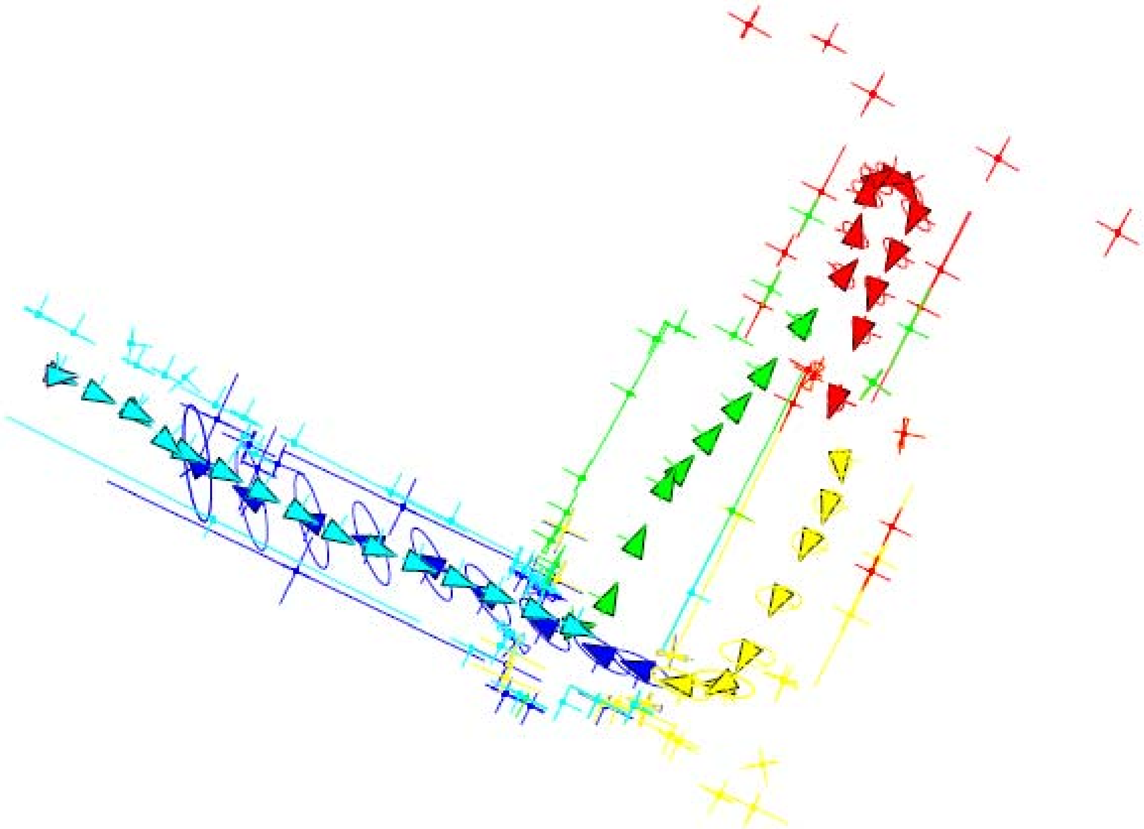
$\mathcal{H}_k = \text{data\_association}(\mathbf{x}_{k|k-1}^B, \mathbf{P}_{k|k-1}^B, \mathbf{z}_k, \mathbf{R}_k)$

$O(n^2)$ $[\mathbf{x}_k^B, \mathbf{P}_k^B] = \text{EKF\_update}(\mathbf{x}_{k|k-1}^B, \mathbf{P}_{k|k-1}^B, \mathbf{z}_k, \mathbf{R}_k, \mathcal{H}_k)$

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

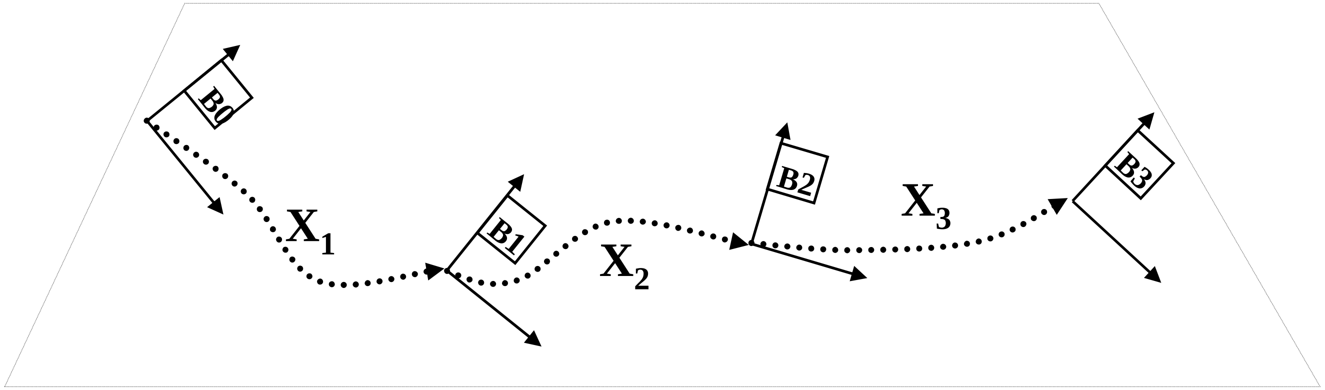
end for

Scalable EKF SLAM: Independent Local Maps

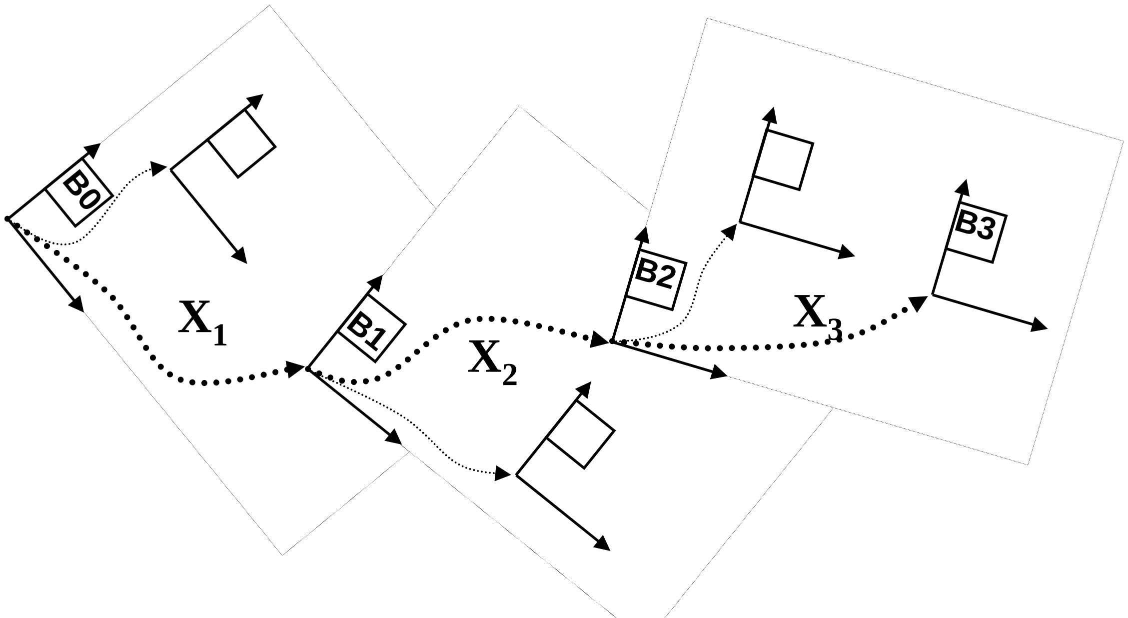


Hierarchical SLAM

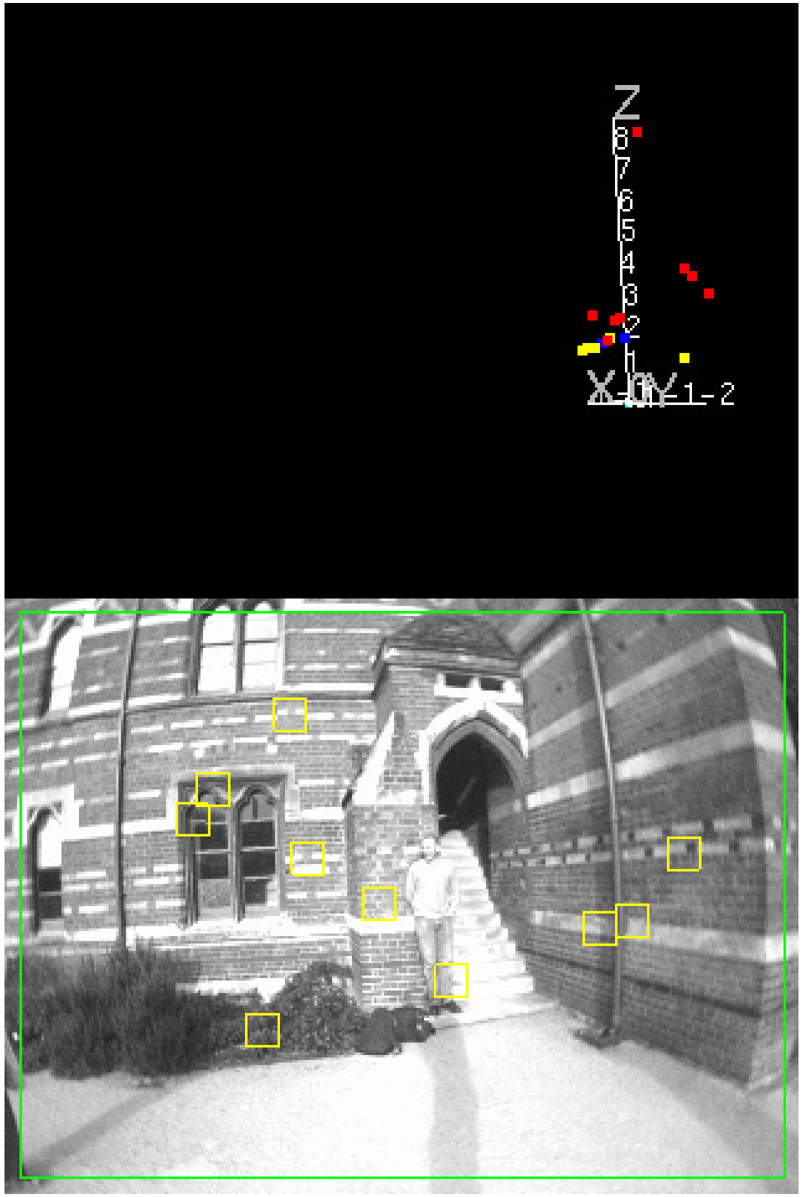
- Global level: adjacency graph and relative stochastic map



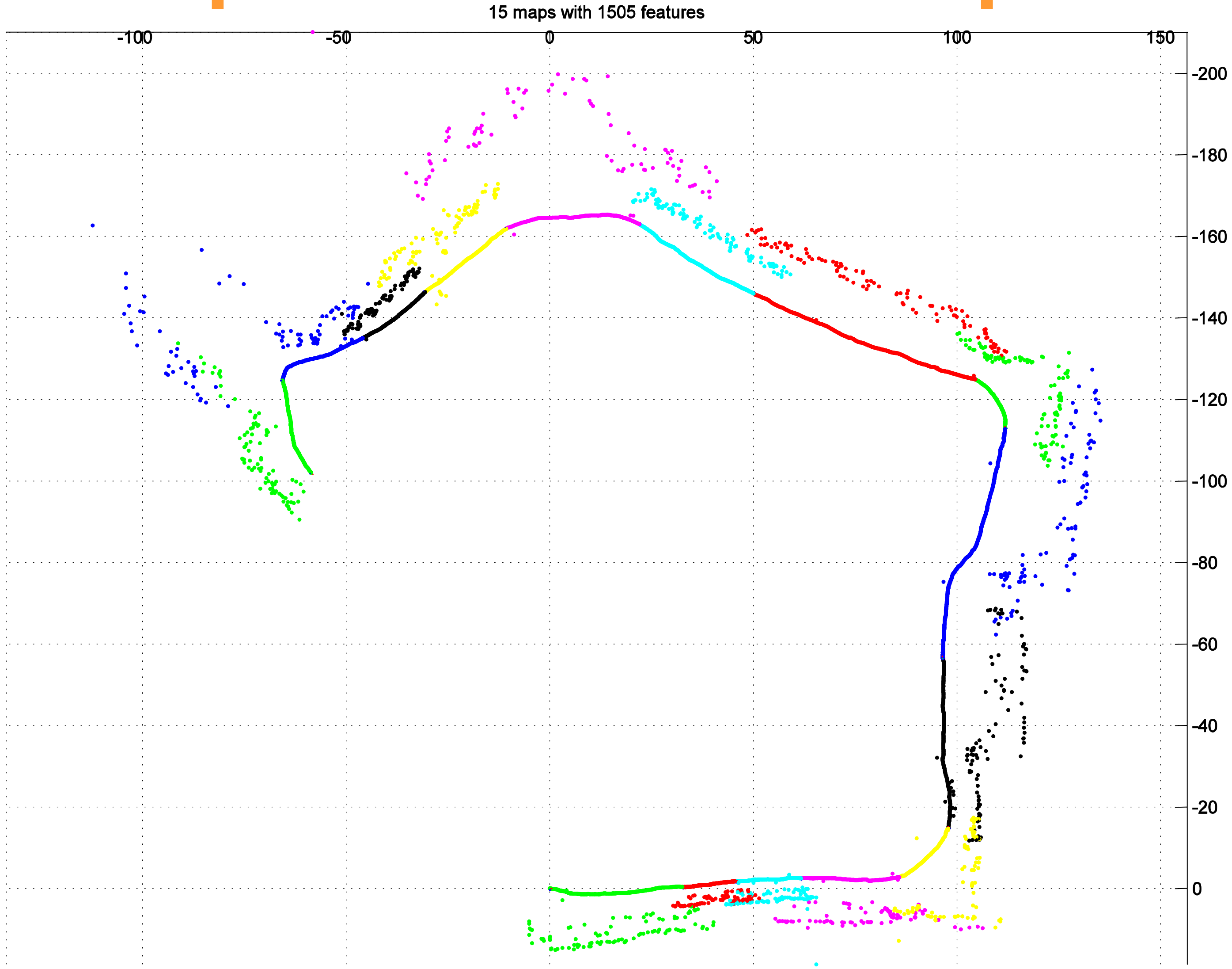
- Local level: statistically independent local maps



Keble College, Oxford

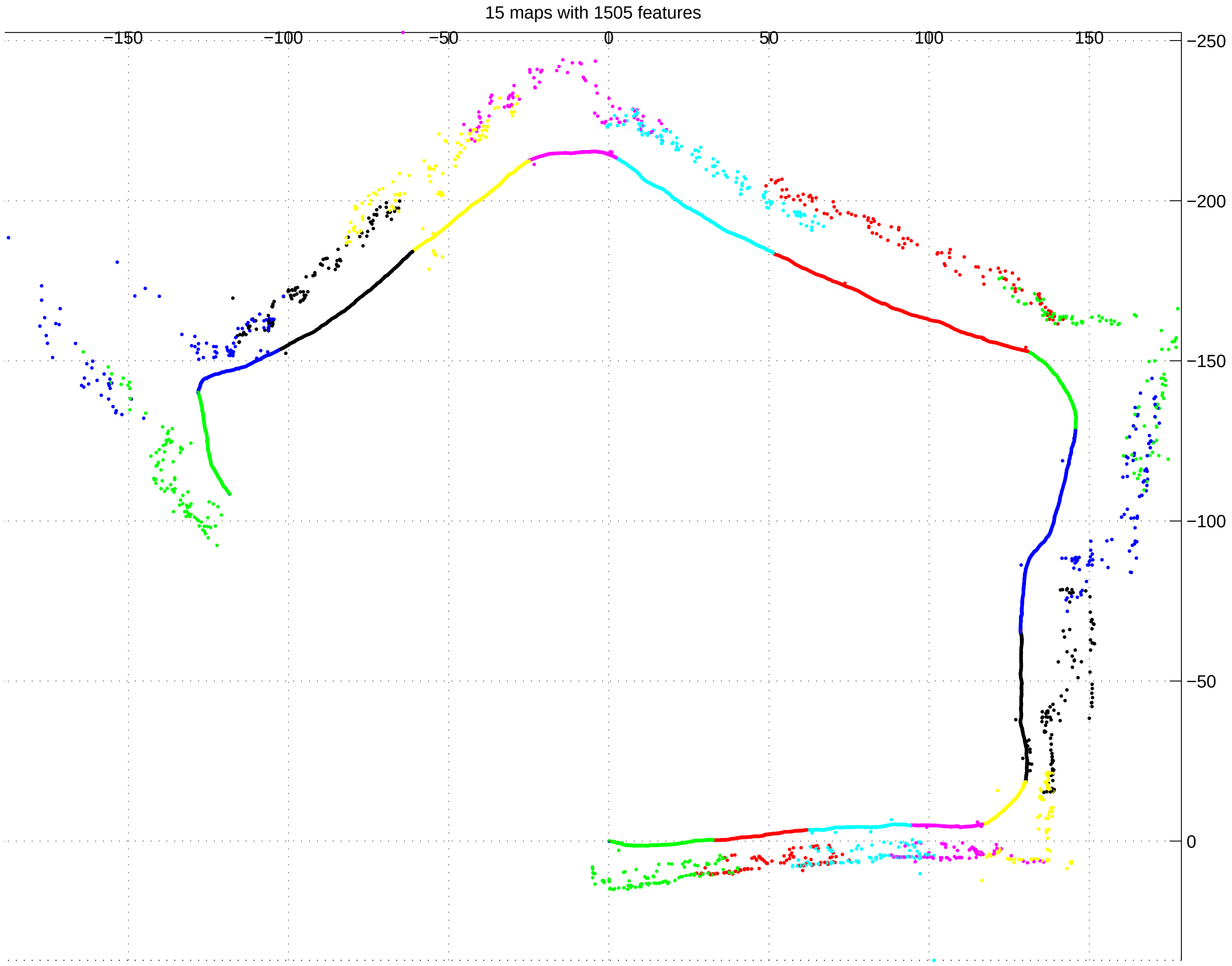


Sequence of local maps



The scale is arbitrary (not observable)

With scale compensation



Map Matching

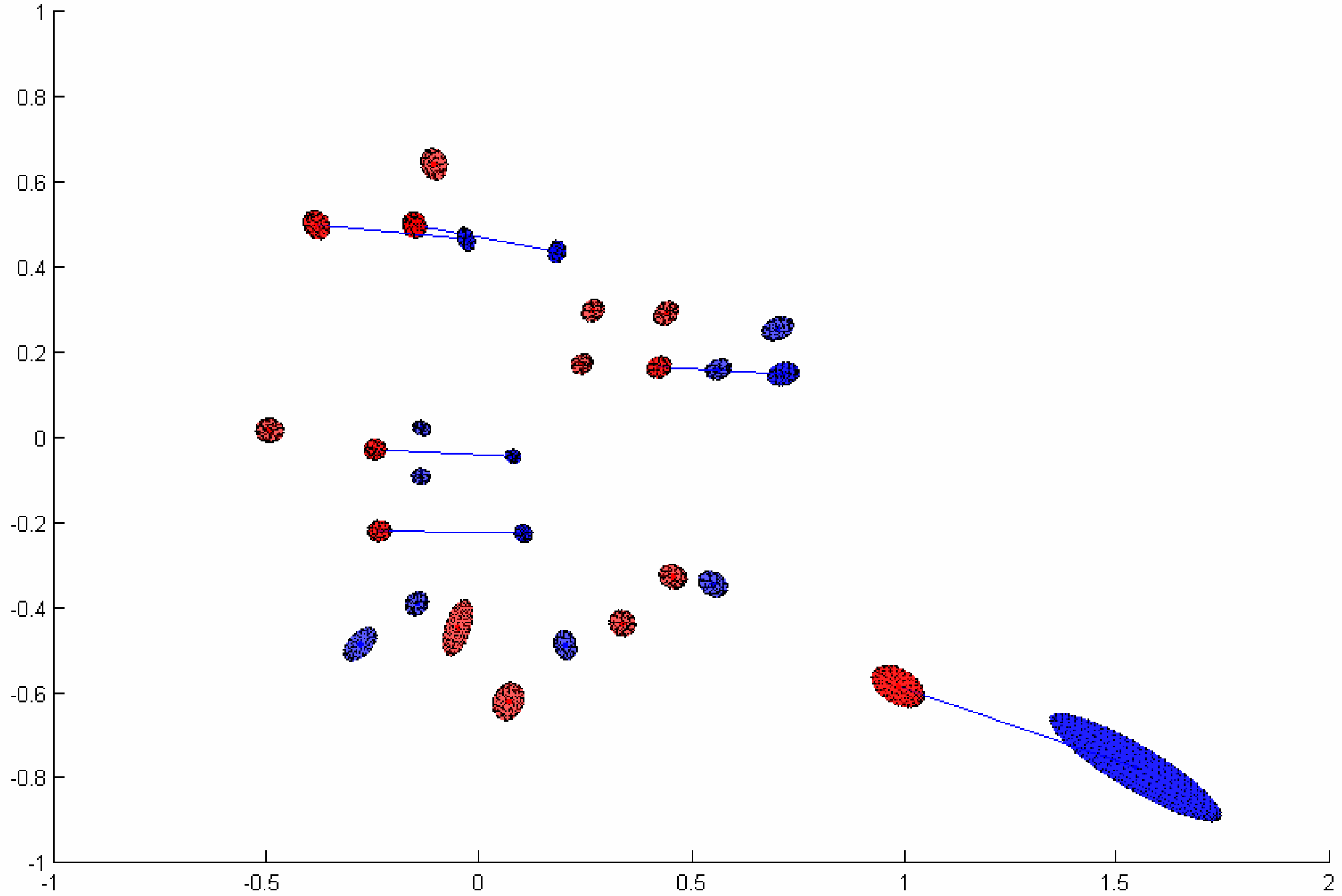
Unary Constraints

Cloud 1

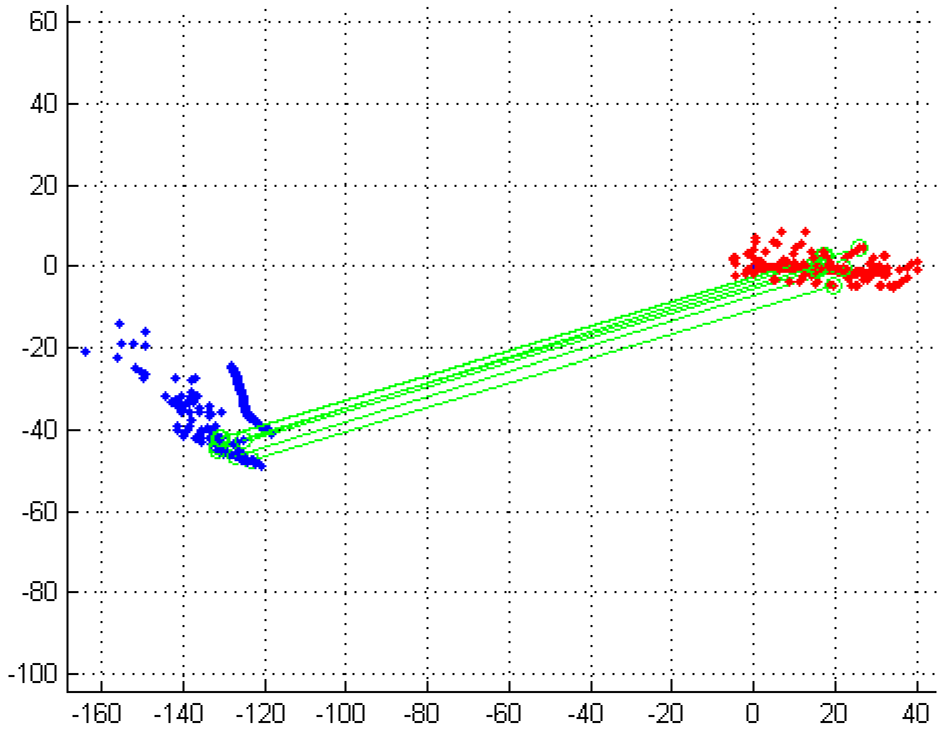
Cloud 2

Map Matching

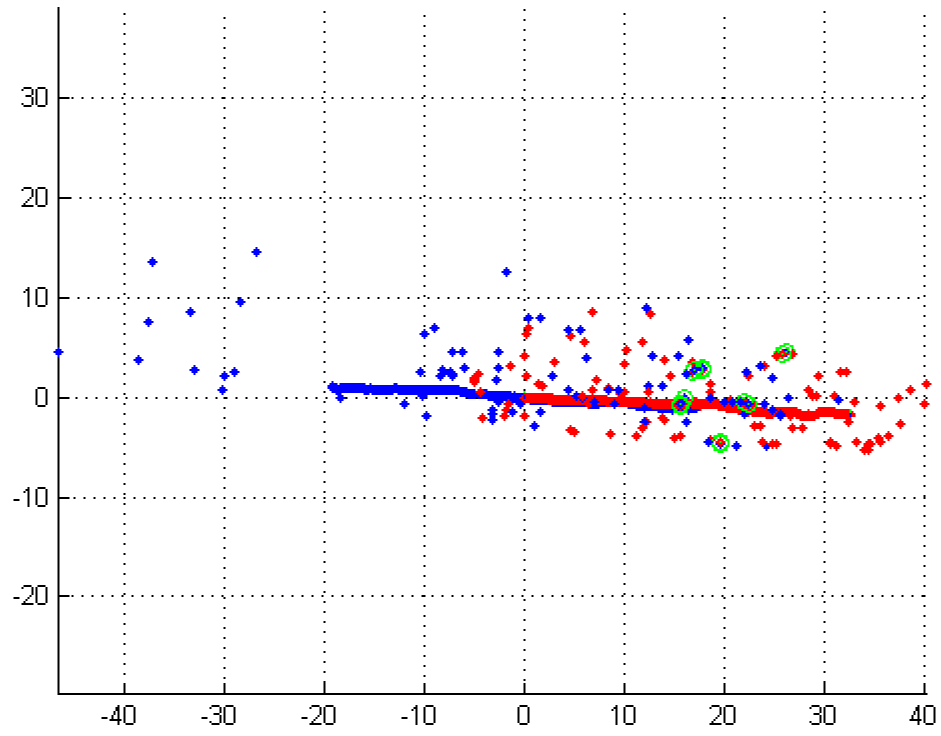
Binary Constraints



Map-to-Map Matching



Matchings found



Aligned Submaps

Nonlinear constrained optimization

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

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

$$\mathbf{h}(\mathbf{x}) = \mathbf{0}$$

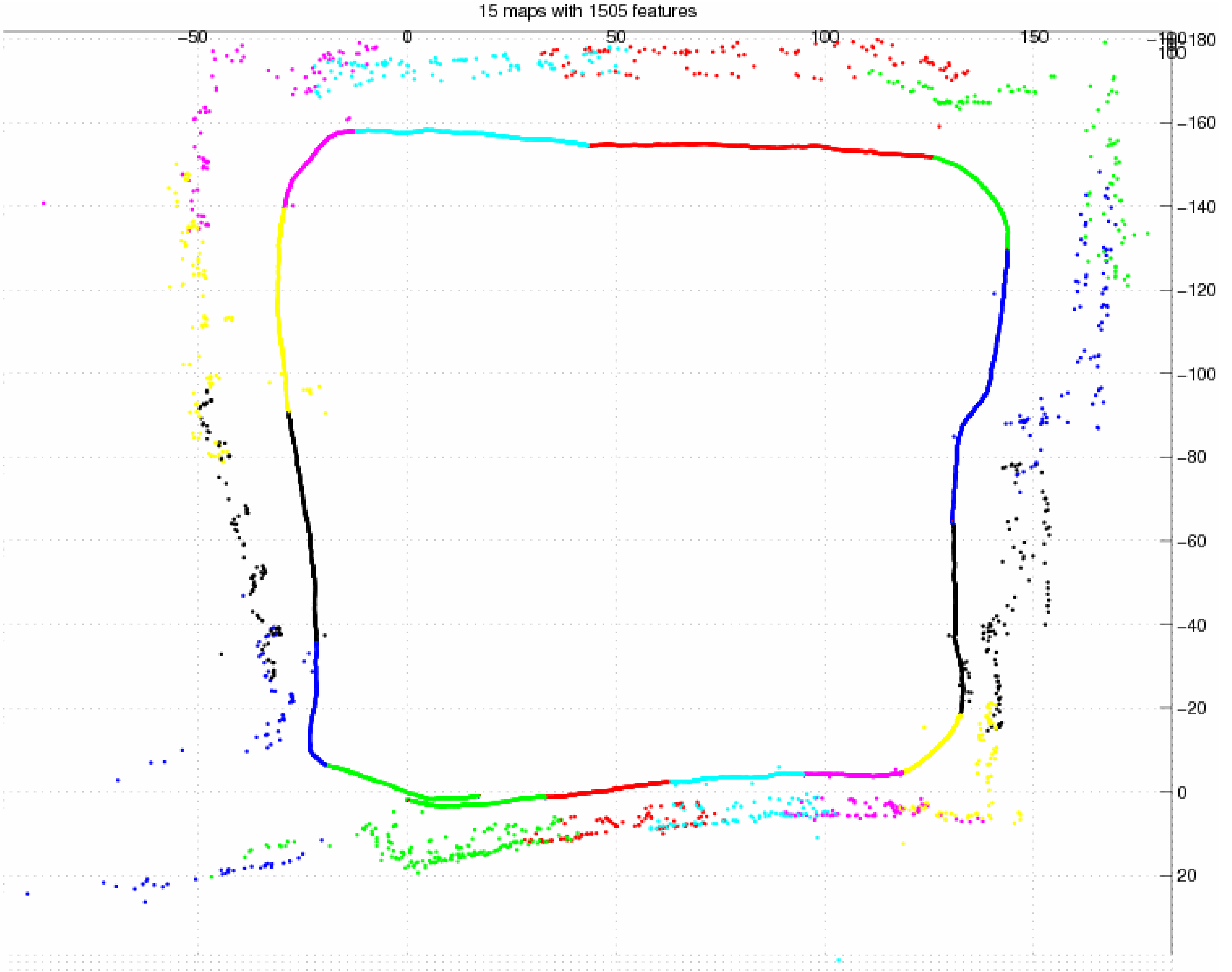
- Sequential Quadratic Programming (SQP) :

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

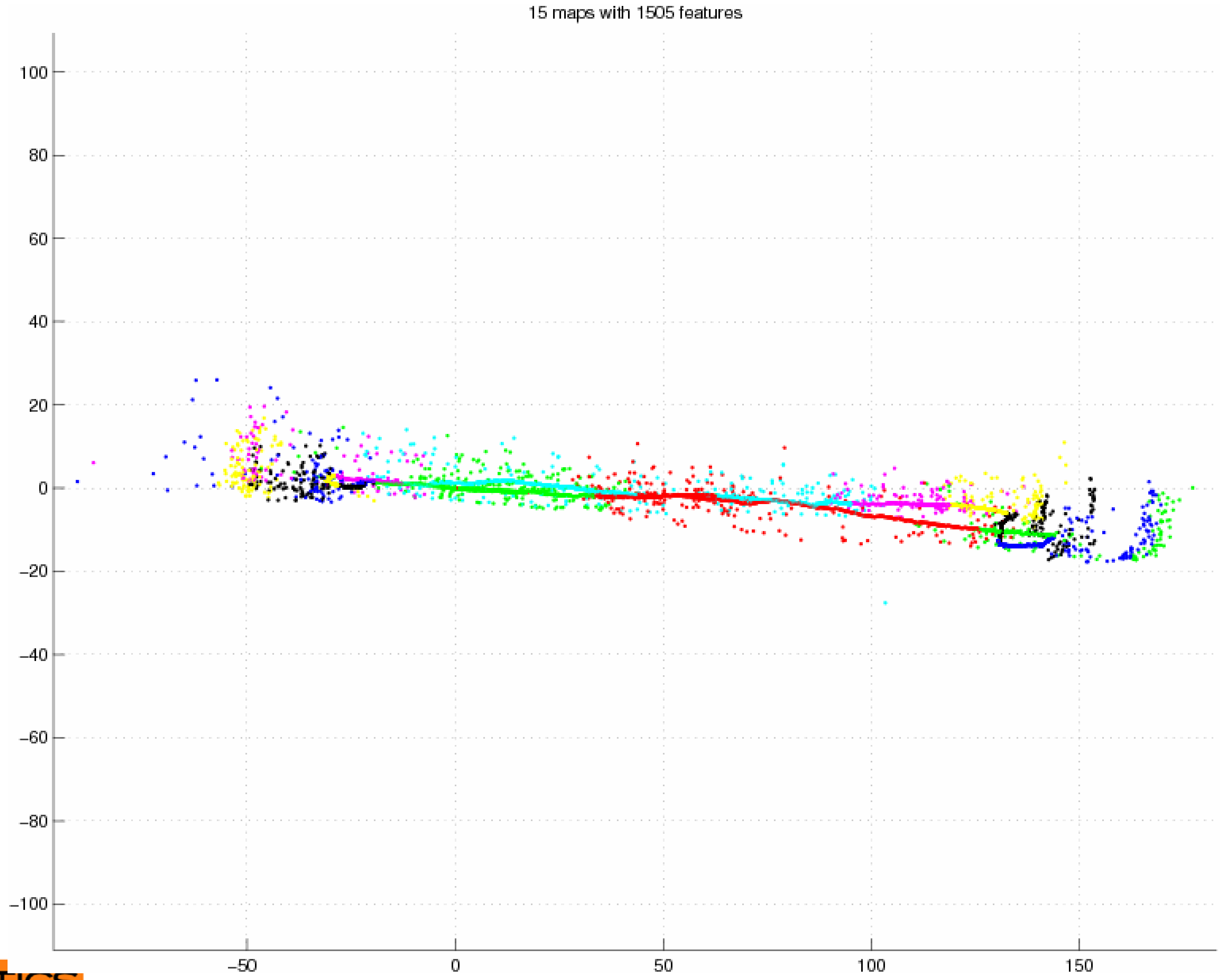
$$\begin{aligned} \mathbf{P}_i &= \mathbf{P}_0 - \mathbf{P}_0\mathbf{H}_i^T \left(\mathbf{H}_i\mathbf{P}_0\mathbf{H}_i^T\right)^{-1} \mathbf{H}_i\mathbf{P}_0 \\ \hat{\mathbf{x}}_{i+1} &= \hat{\mathbf{x}}_i - \mathbf{P}_i\mathbf{P}_0^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_0) - \mathbf{P}_0\mathbf{H}_i^T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

  » Iterate until convergence

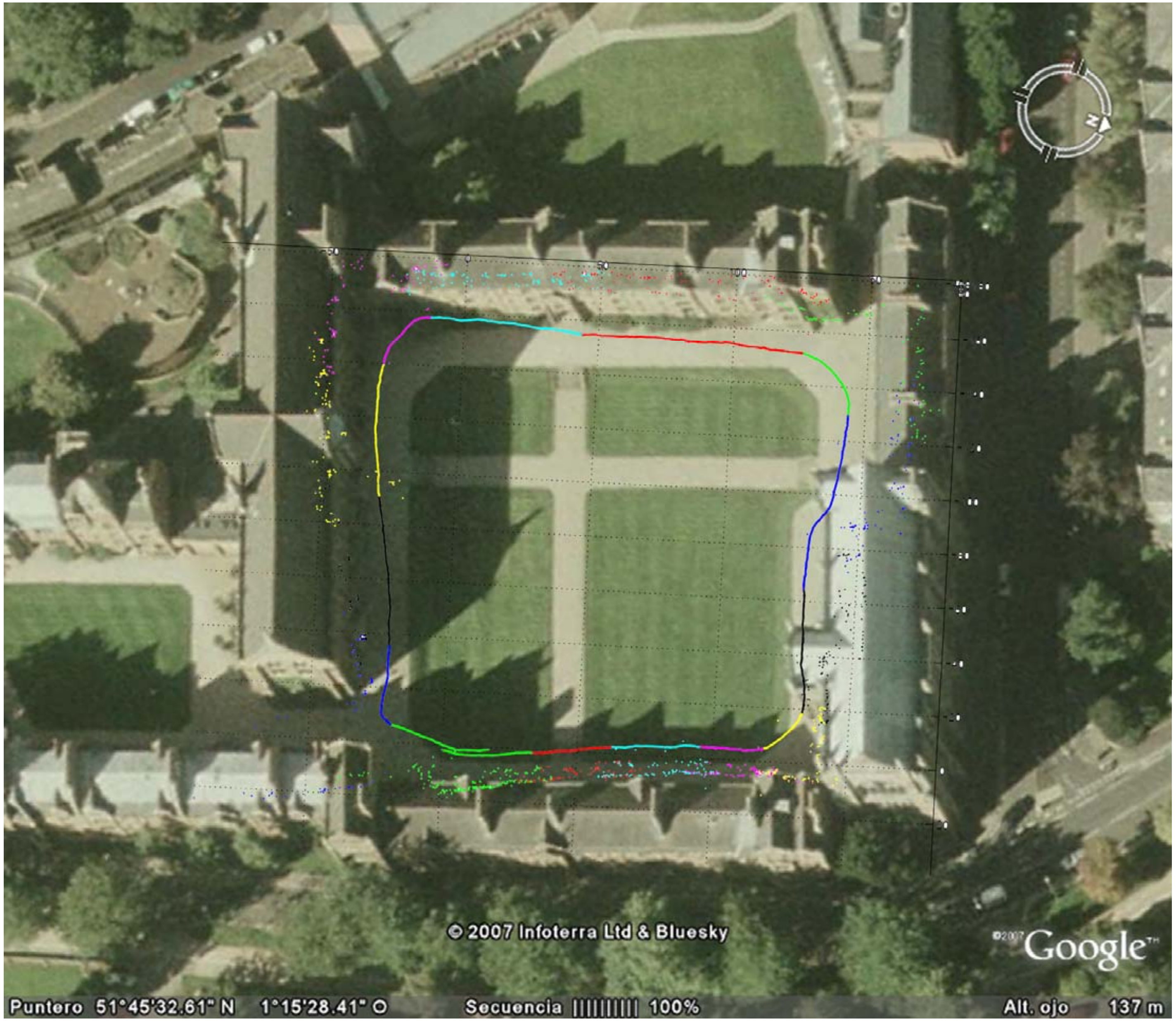
Loop closing



Loop closing (lateral view)



Keble College, Oxford (290m)



Results

- Local Map building in real-time @30Hz
  - 60 features per map using inverse depth
  - Bigger maps if converted to (x,y,z)
- Joint Compatibility search adds only 2ms in the worst case
- Map-to-map matching in 1s (in Matlab)
  - With a new algorithm based on graph theory
- Loop optimization takes 800ms (6 iterations)
- **The scale drifts along the map**

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

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

SLAM using only stereo

- Experimental setup

A bumblebee, a laptop and a firewire cable

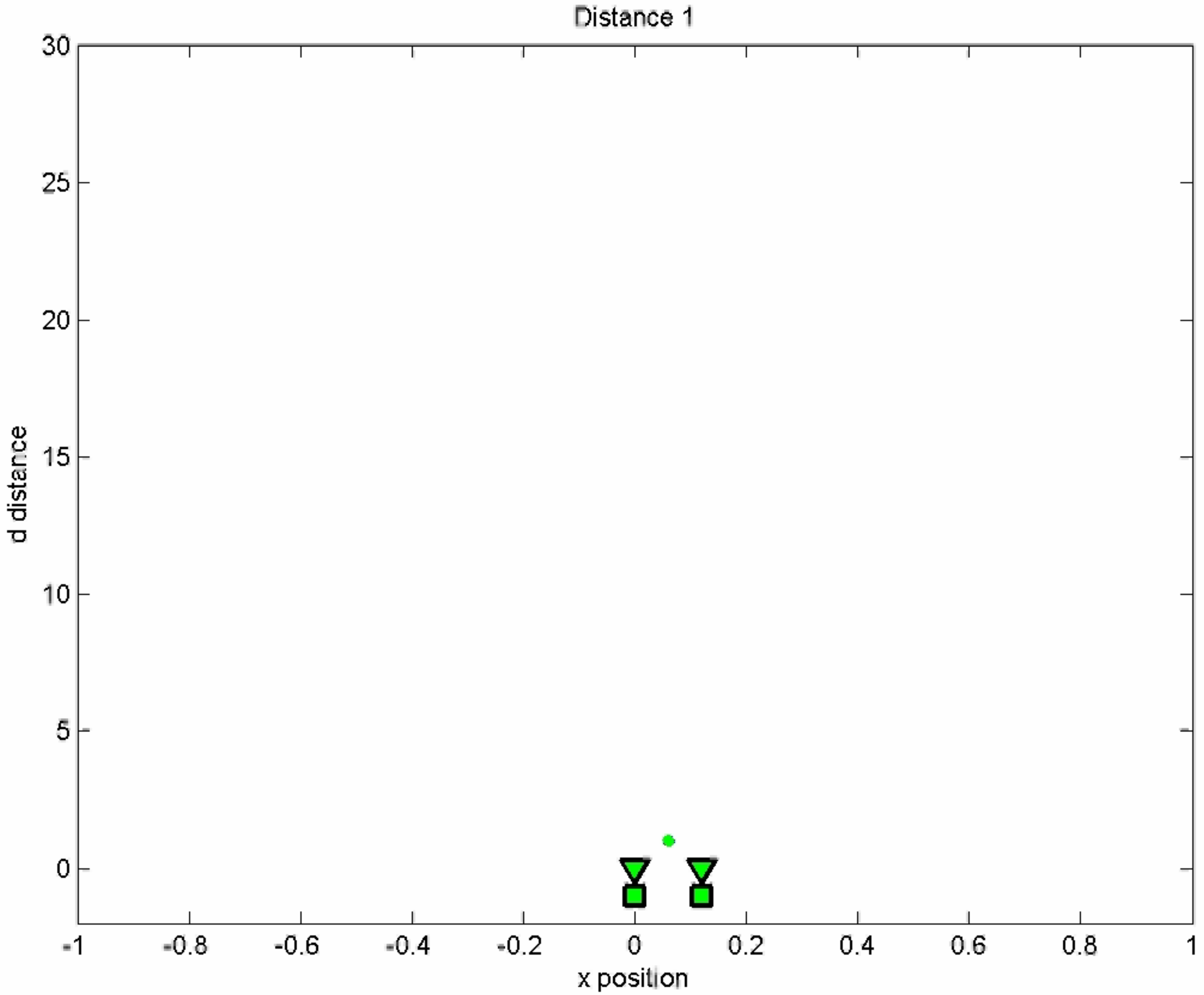
Pure Stereo SLAM

180m indoor loop, CPS, Zaragoza

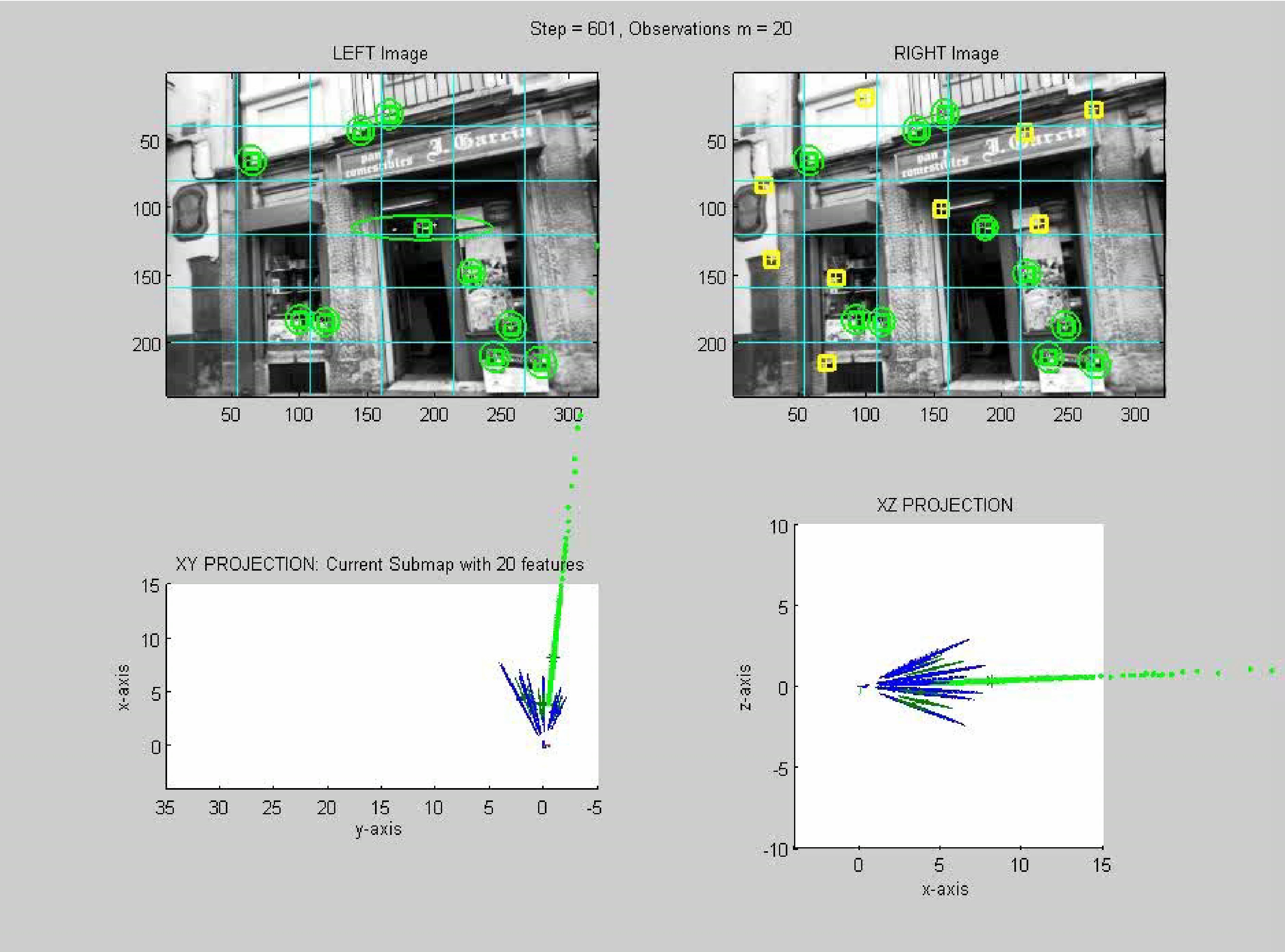
Pure Stereo SLAM

150m outdoor loop, public square, Zaragoza

Depth .vs. Inverse Depth

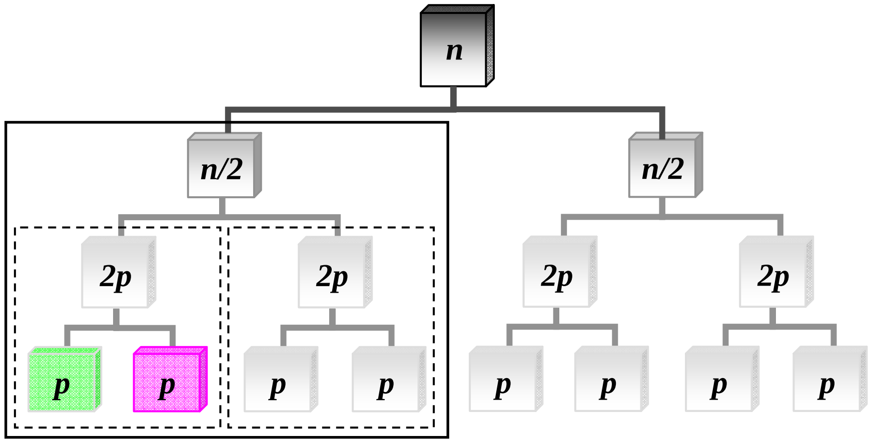


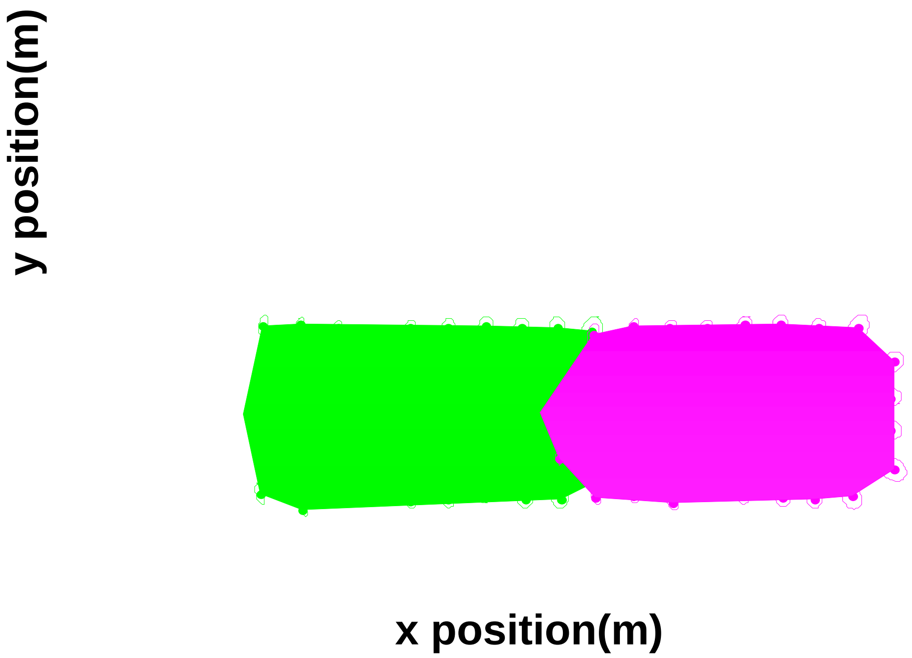
Basic EKF SLAM



Divide & Conquer SLAM

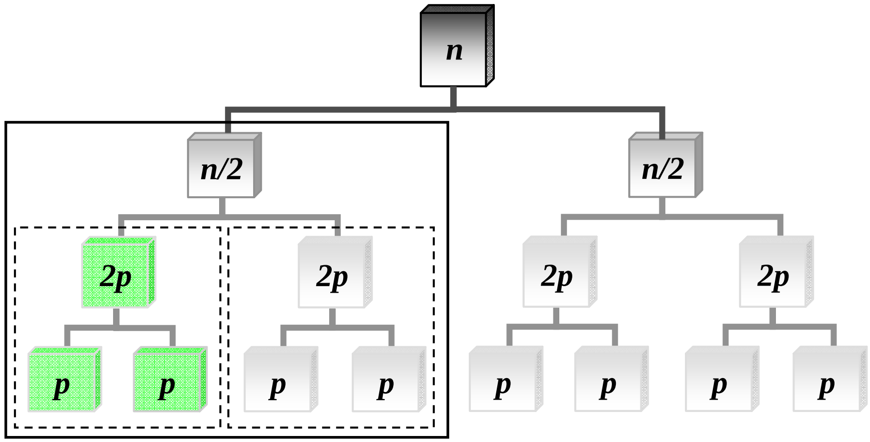
Number of Maps : 2

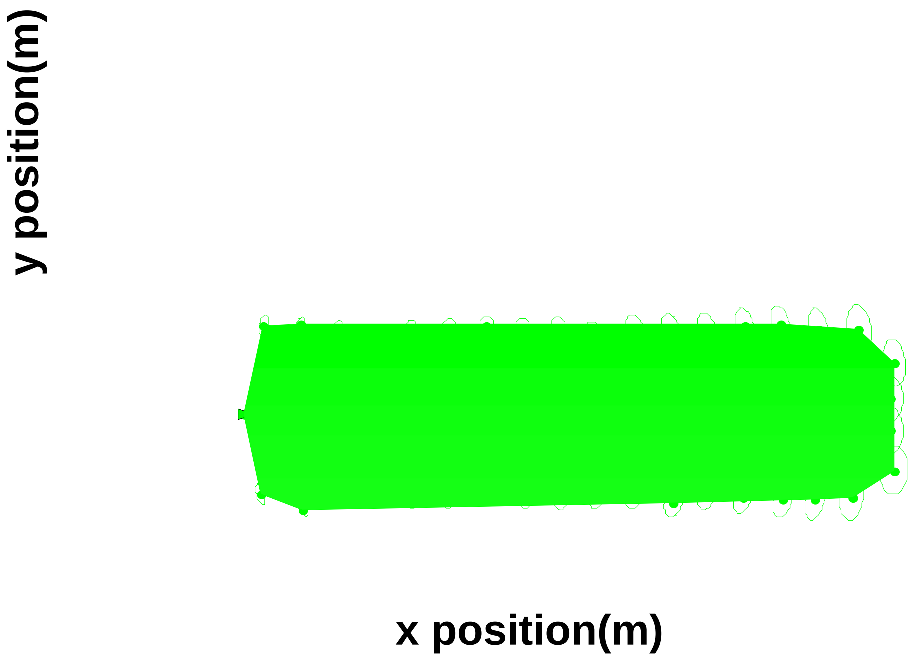




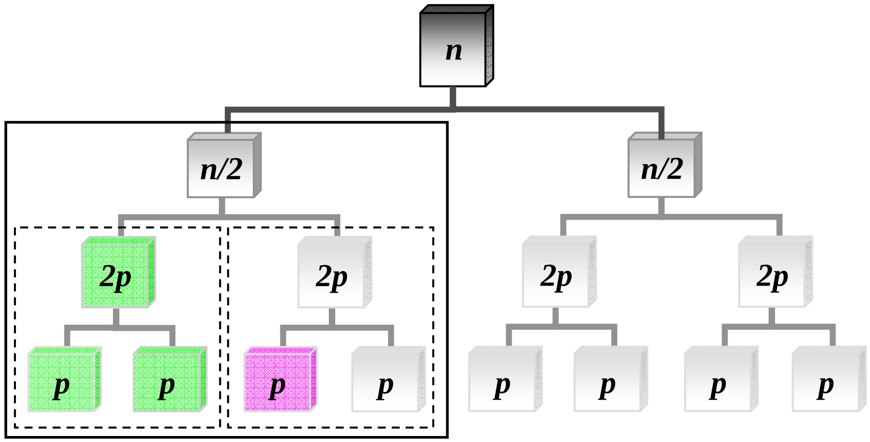
Divide & Conquer SLAM

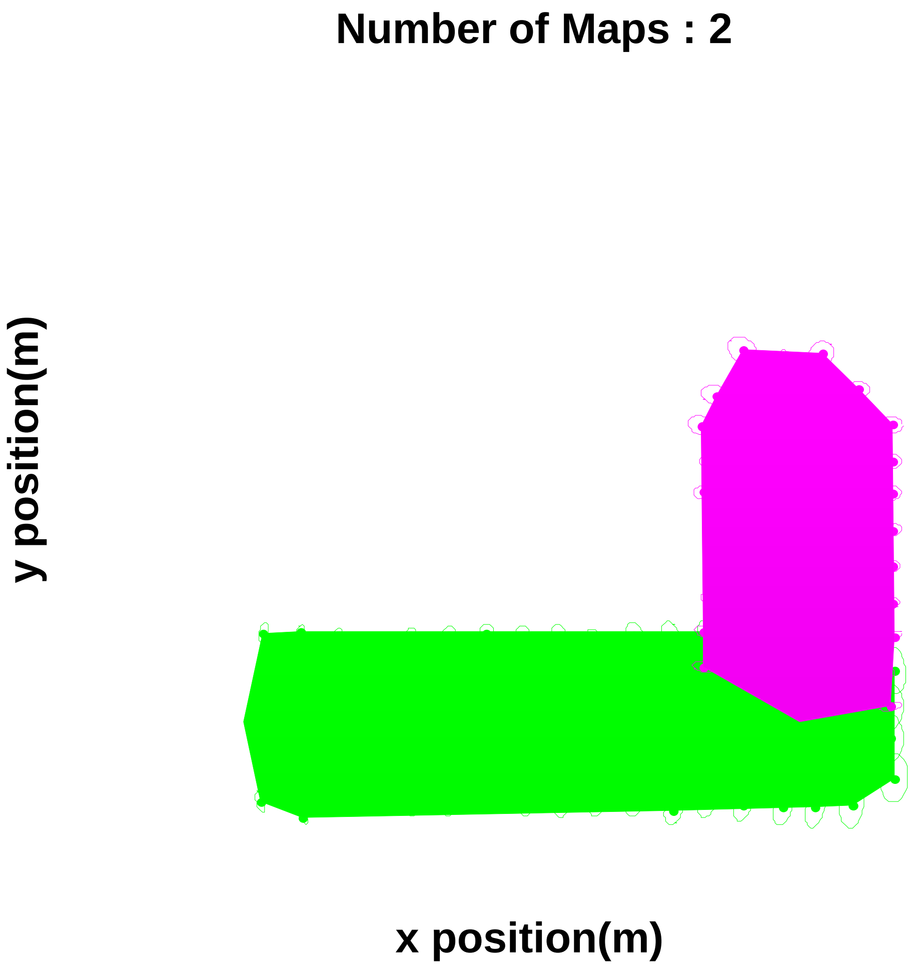
Number of Maps : 1



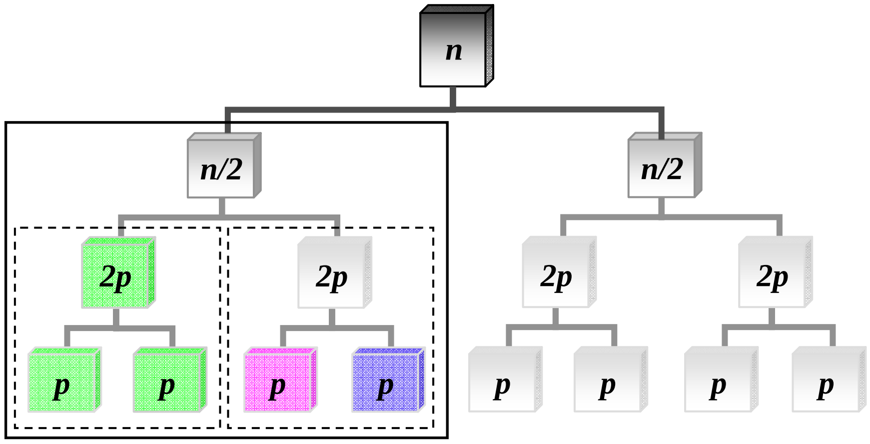


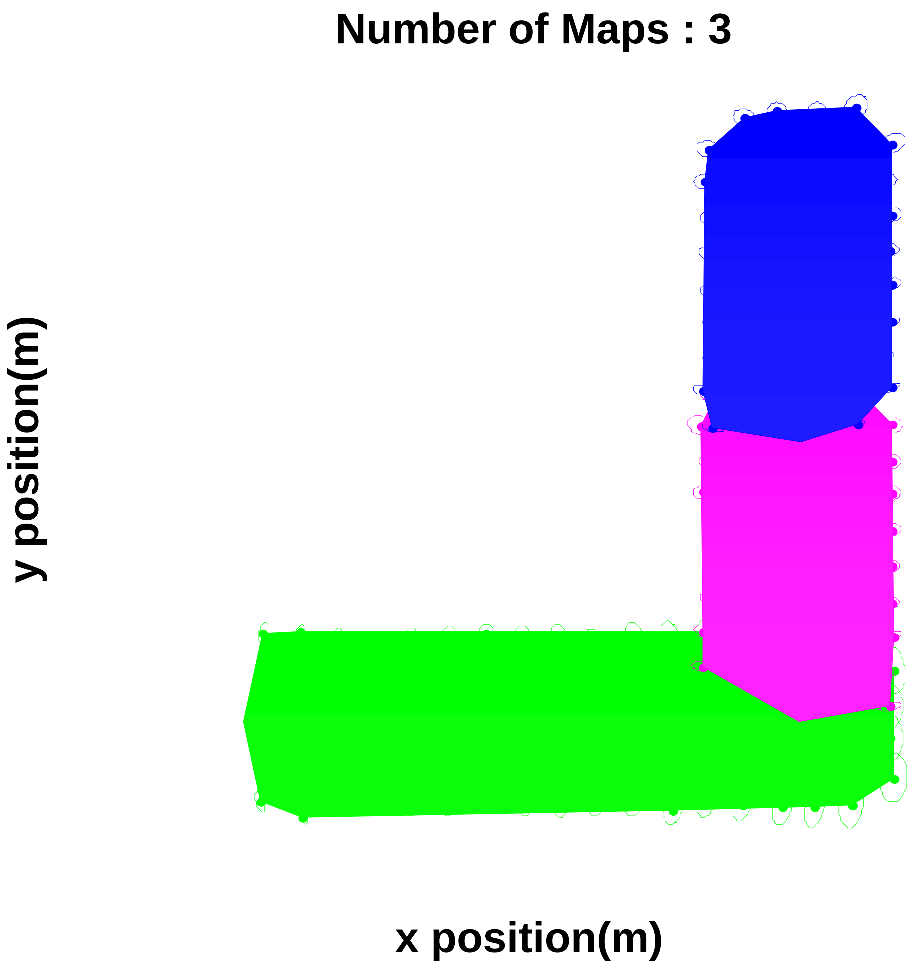
Divide & Conquer SLAM



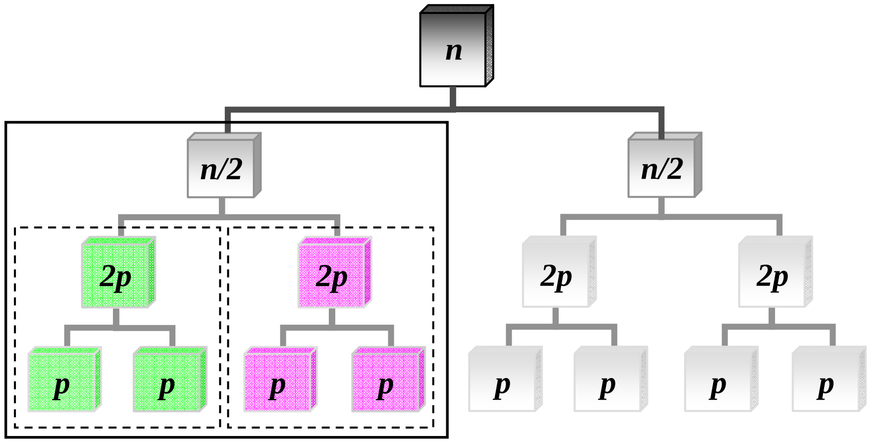


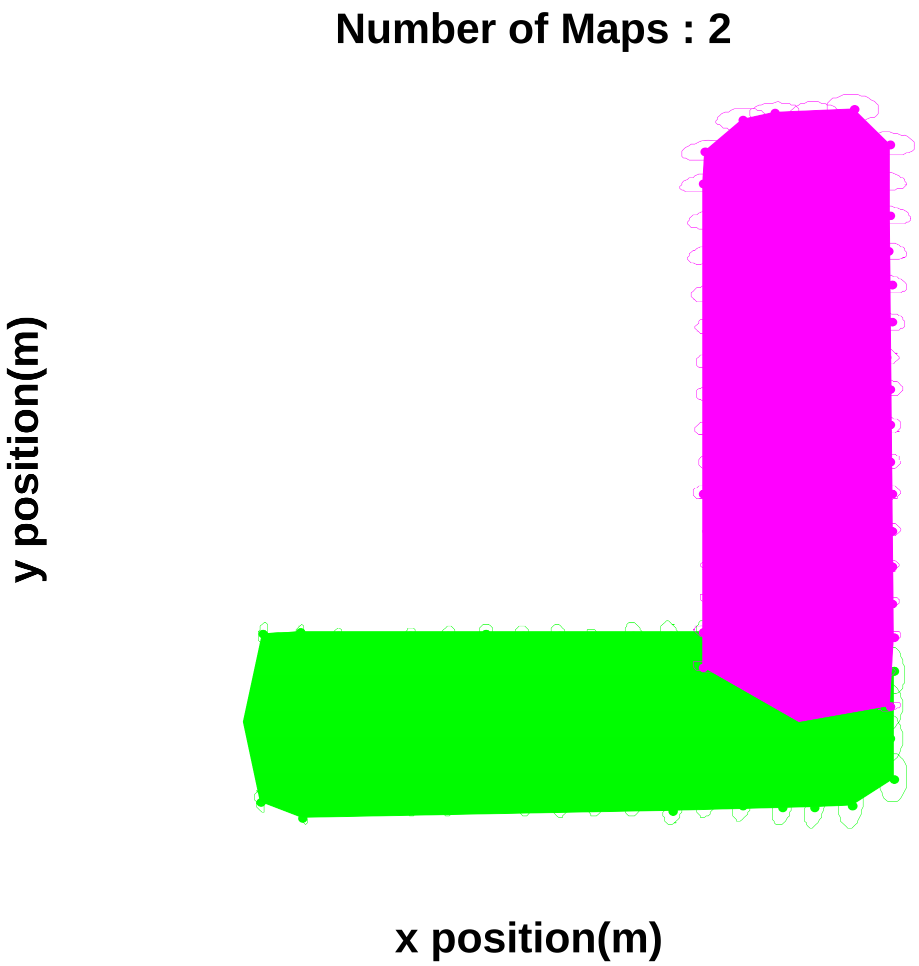
Divide & Conquer SLAM



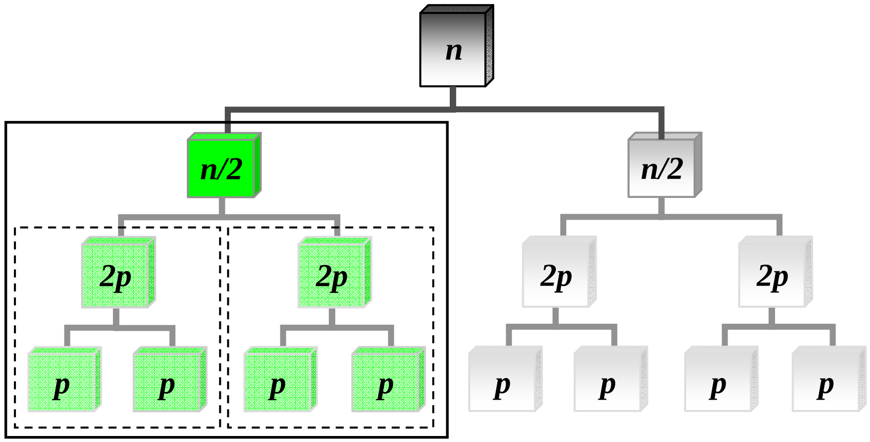


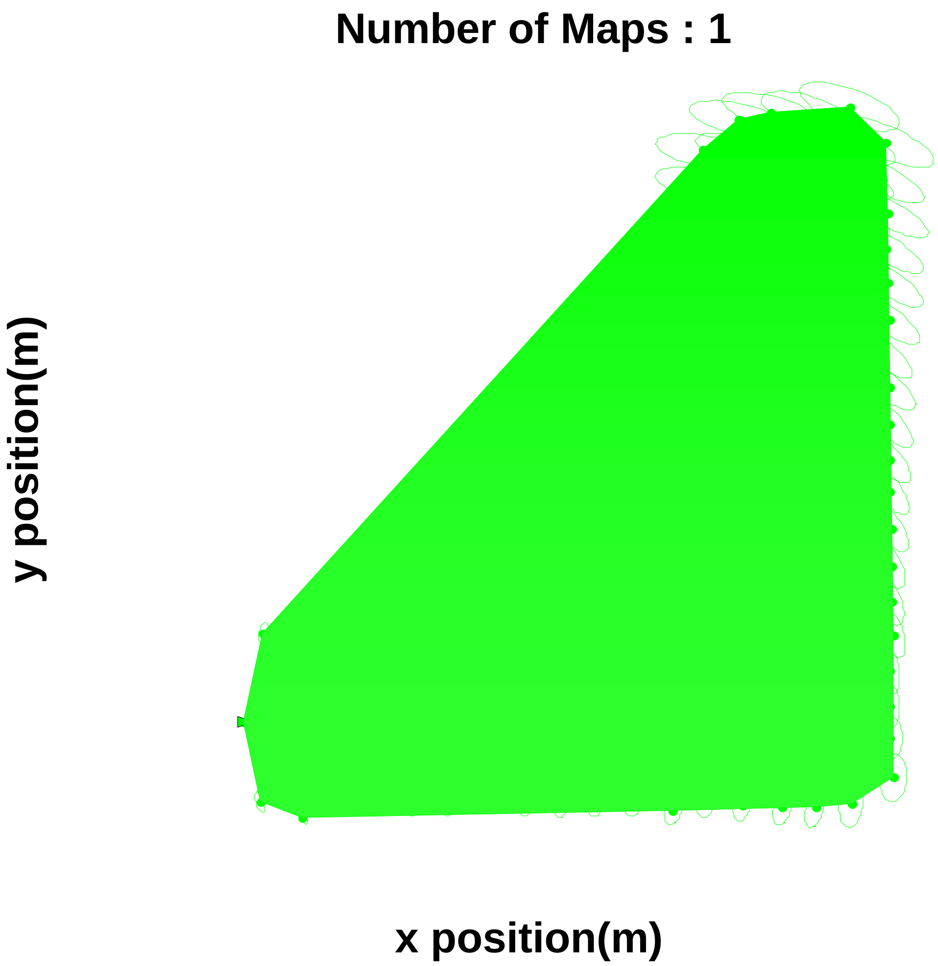
Divide & Conquer SLAM



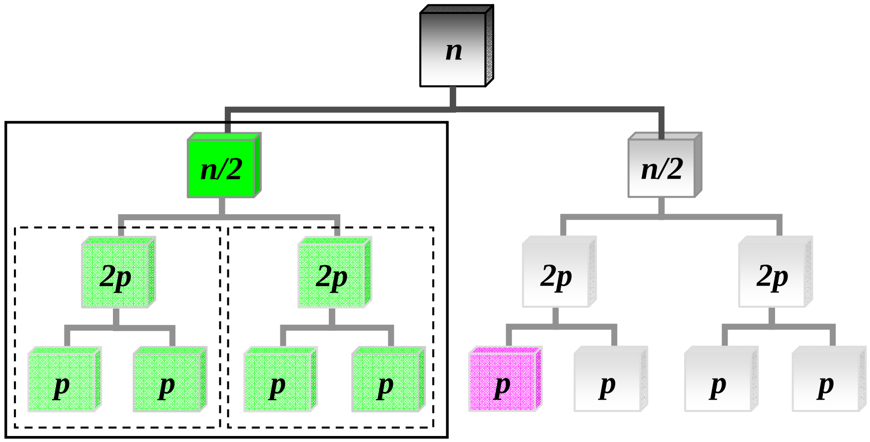


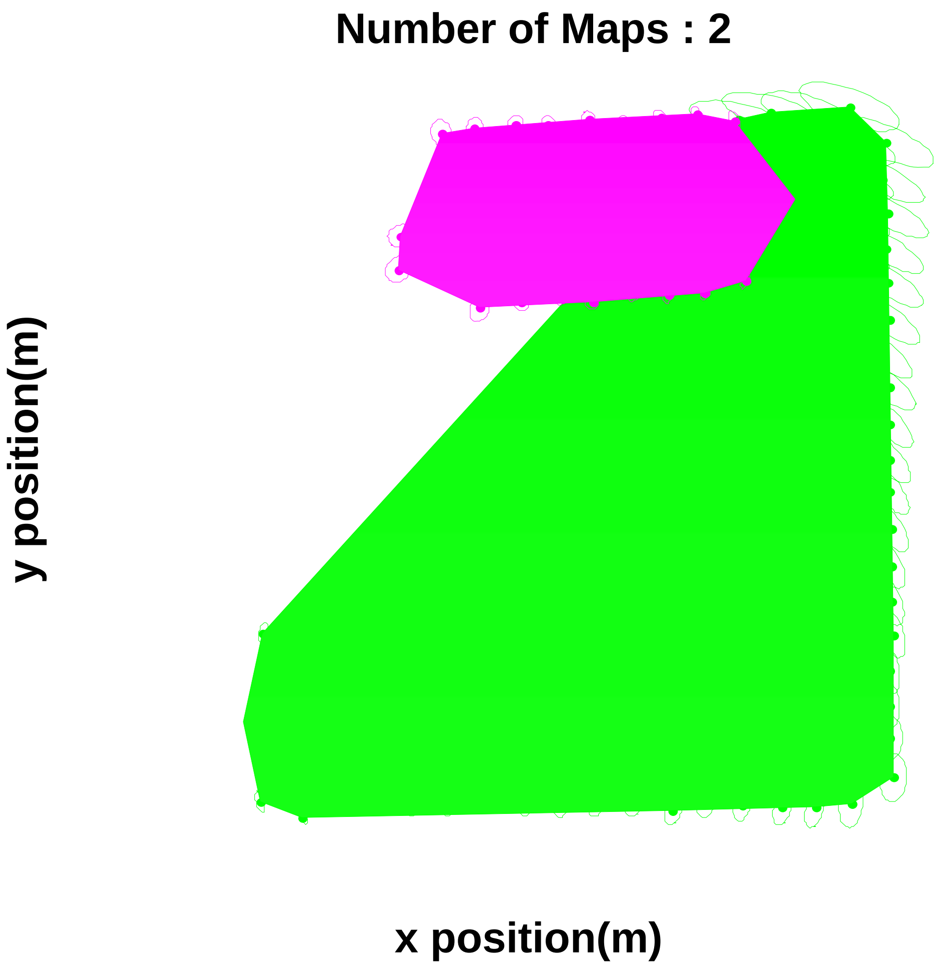
Divide & Conquer SLAM



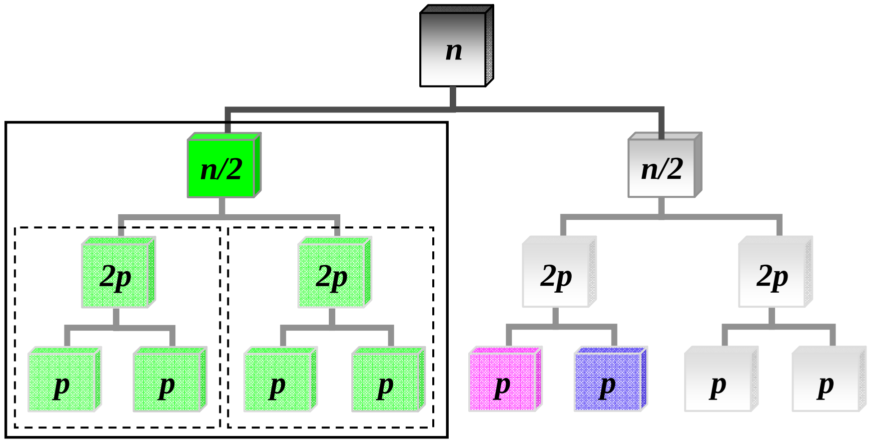


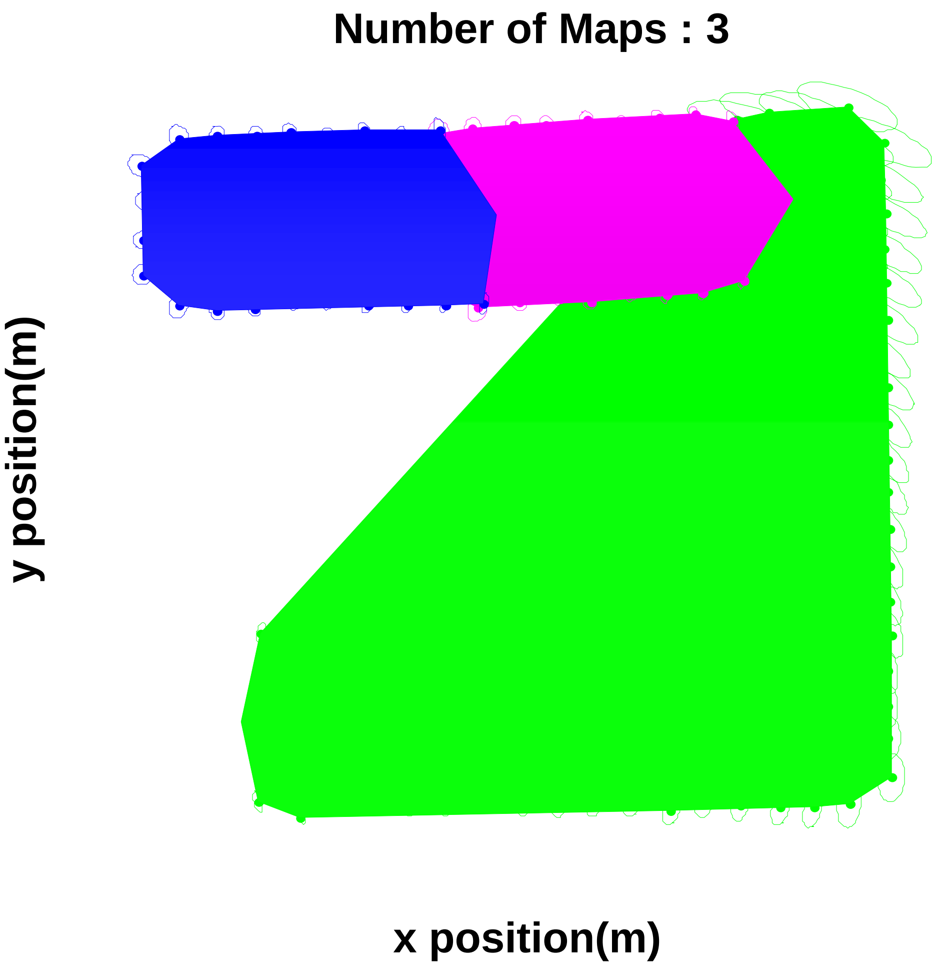
Divide & Conquer SLAM



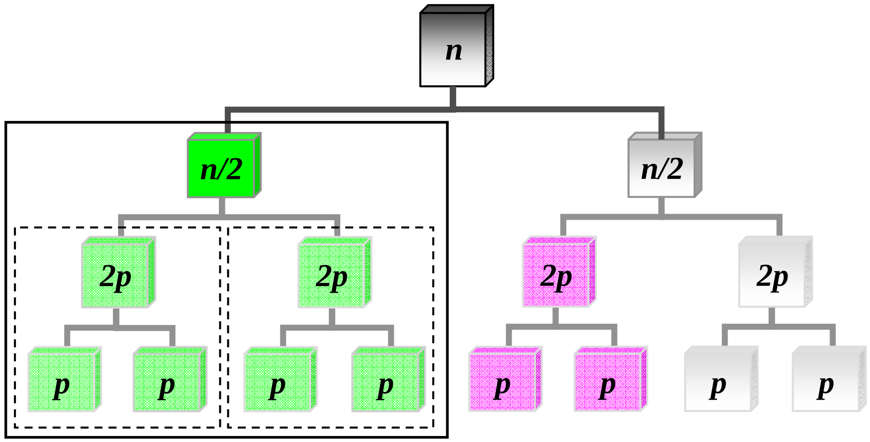


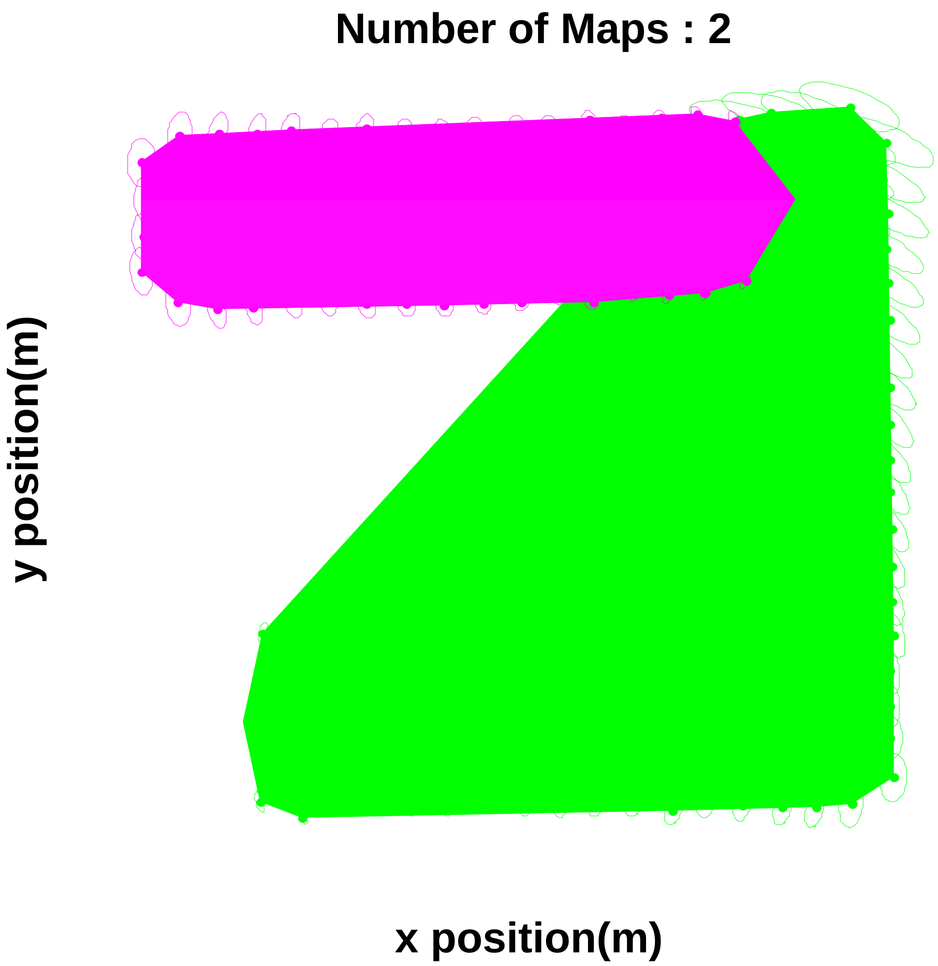
Divide & Conquer SLAM



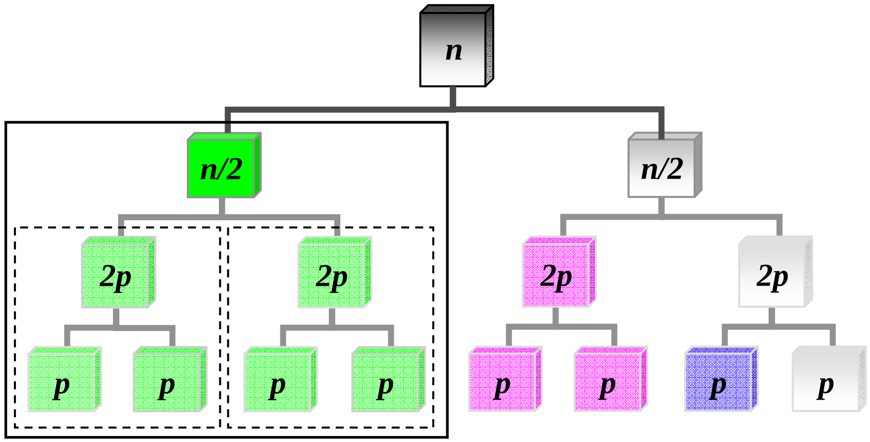


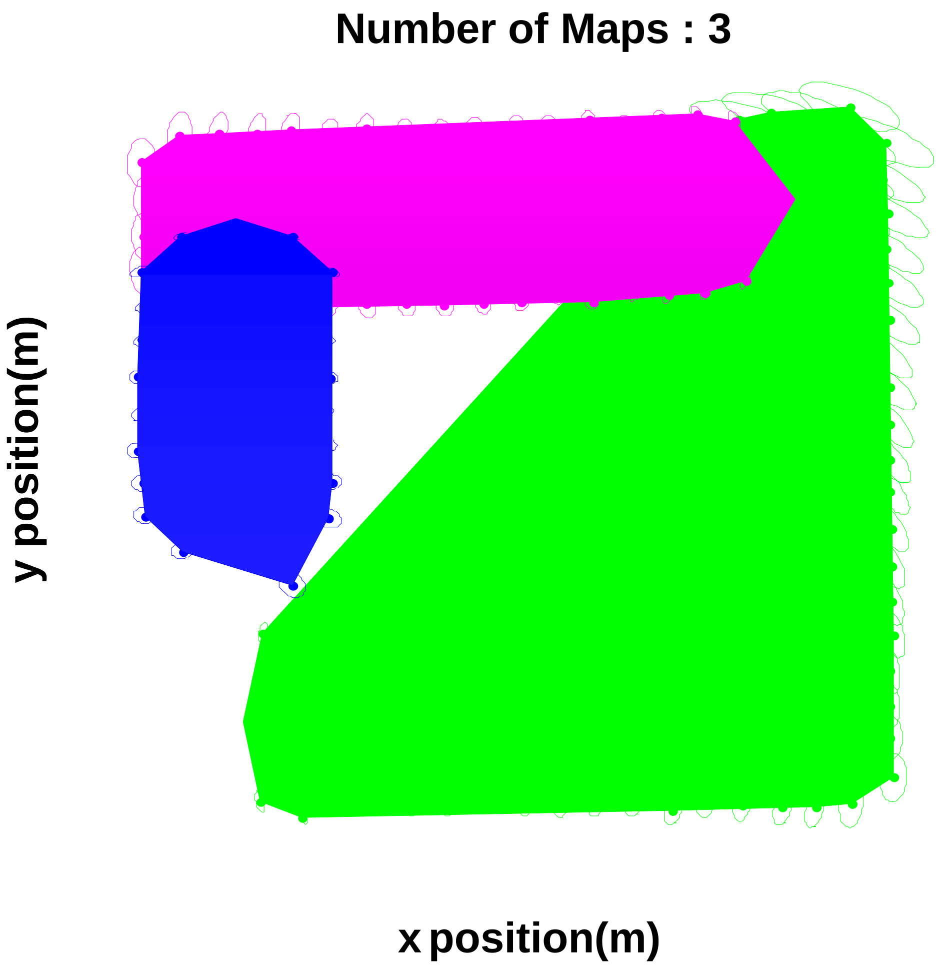
Divide & Conquer SLAM



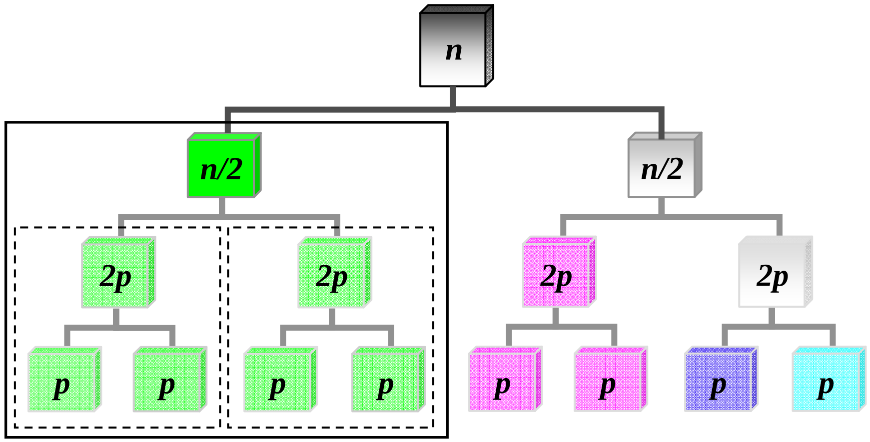


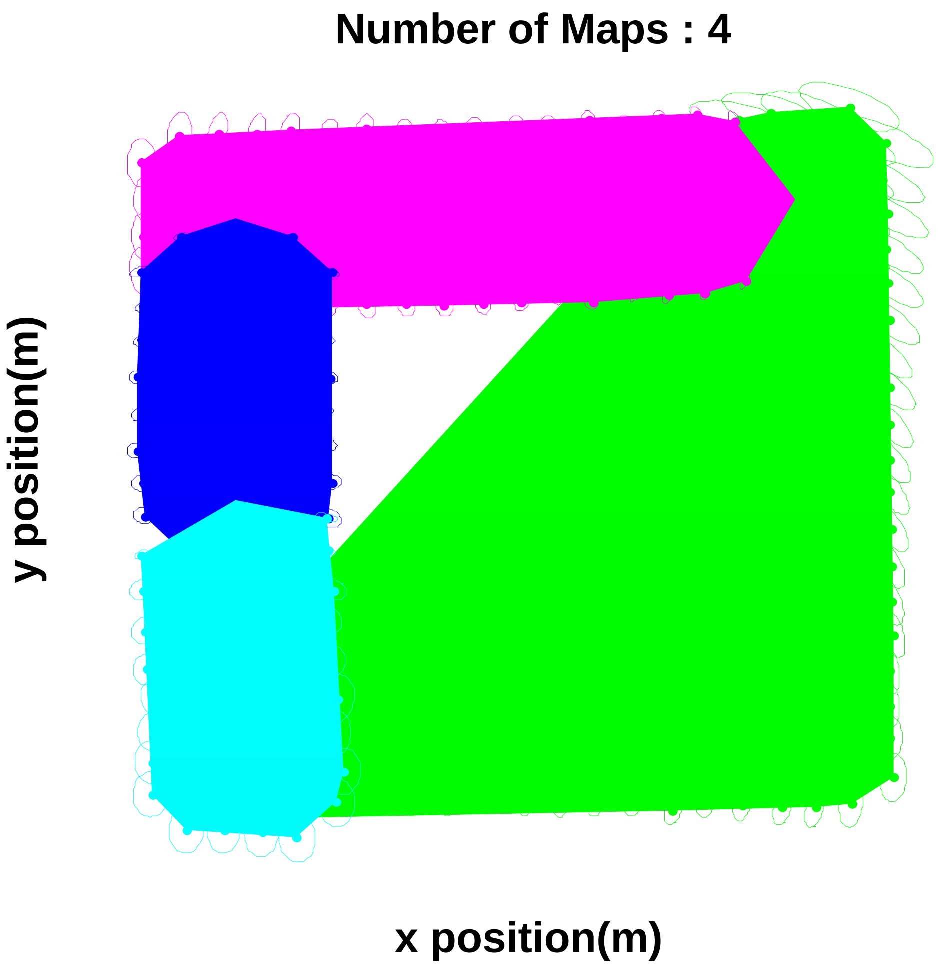
Divide & Conquer SLAM





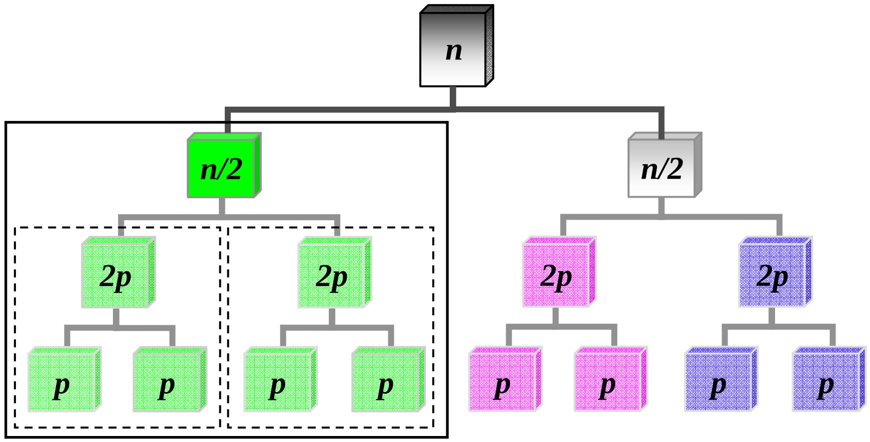
Divide & Conquer SLAM

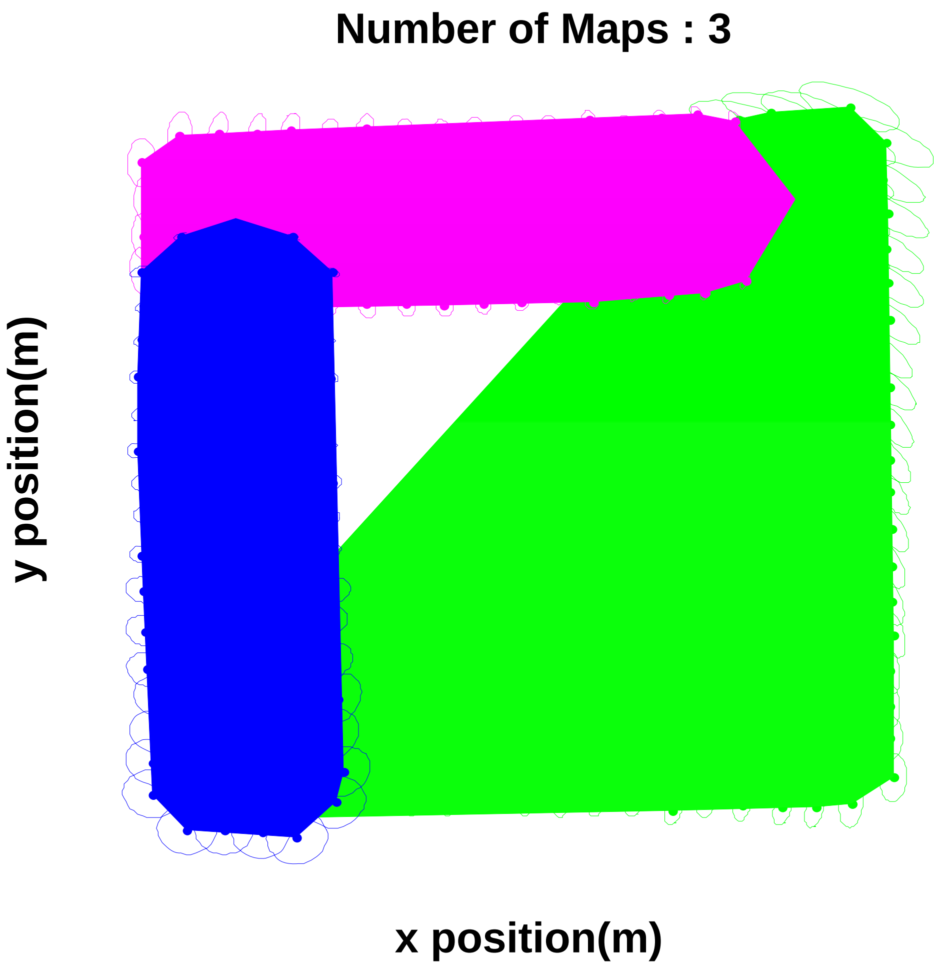




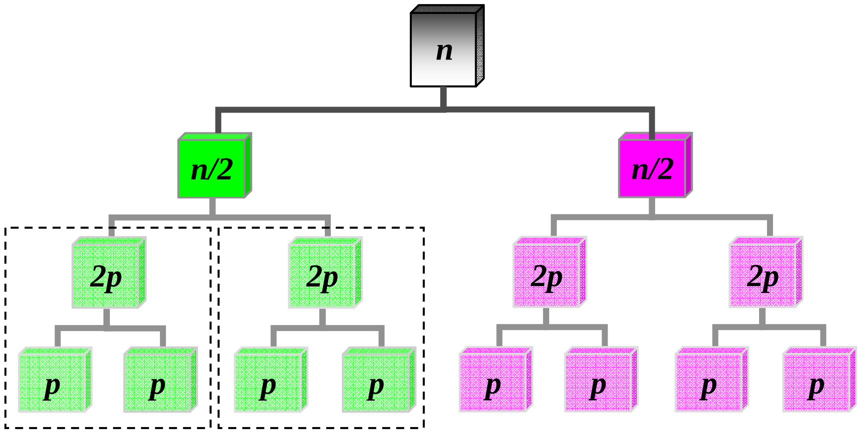
ROBOTICS
UNIVERSIDAD DE ZARAGOZA

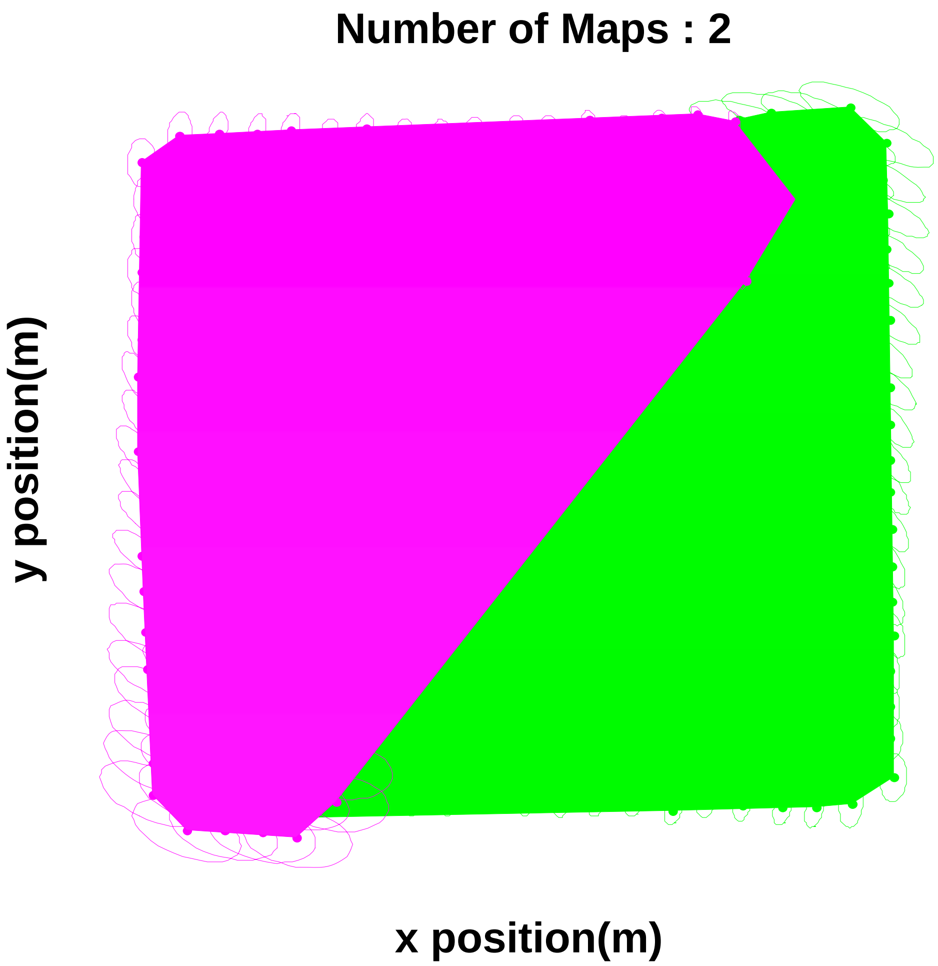
Divide & Conquer SLAM



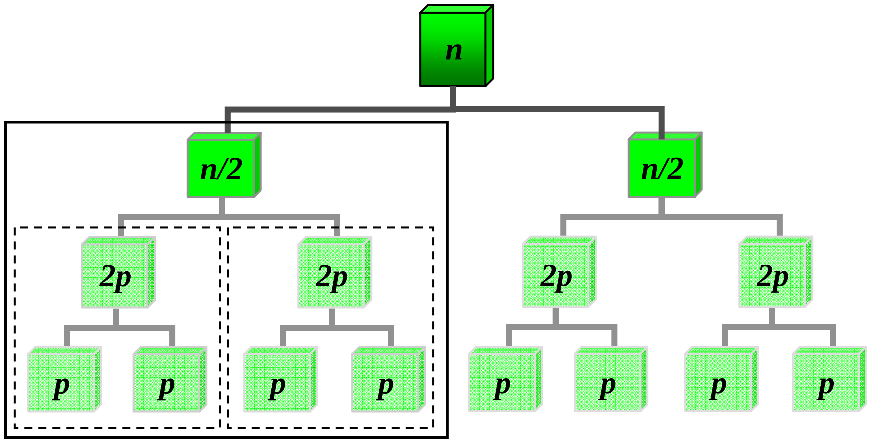


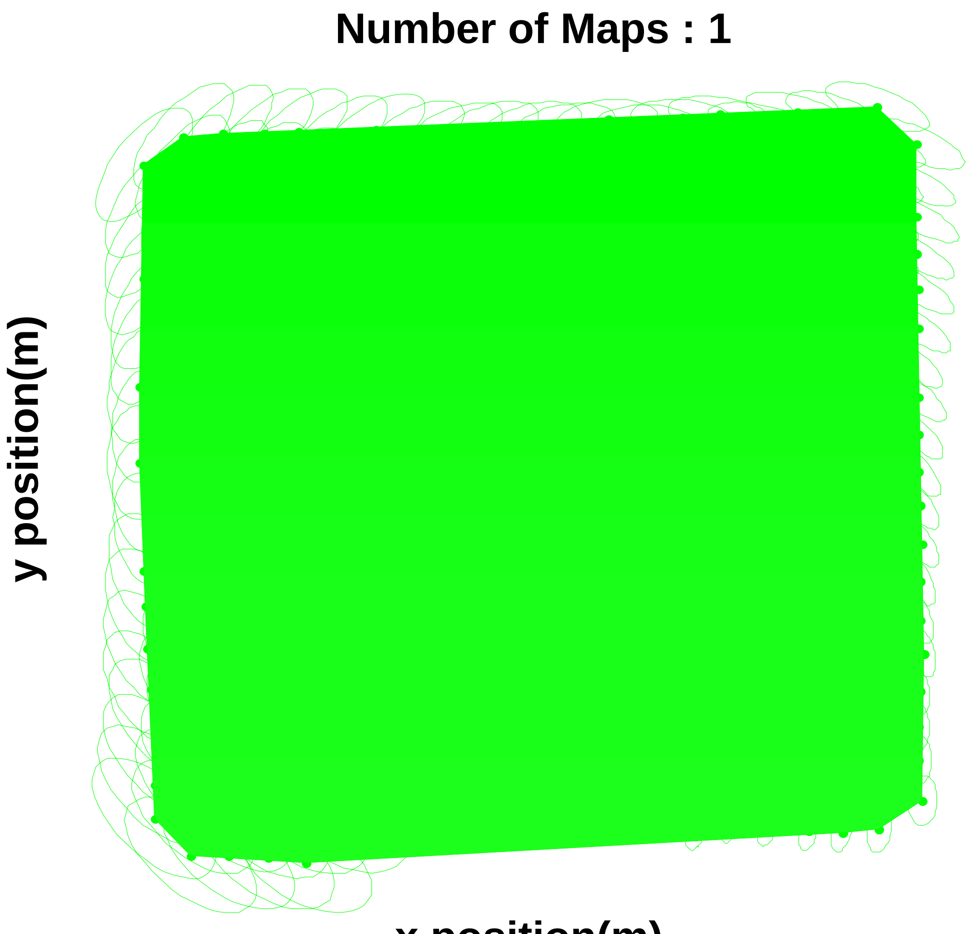
Divide & Conquer SLAM





Divide & Conquer SLAM





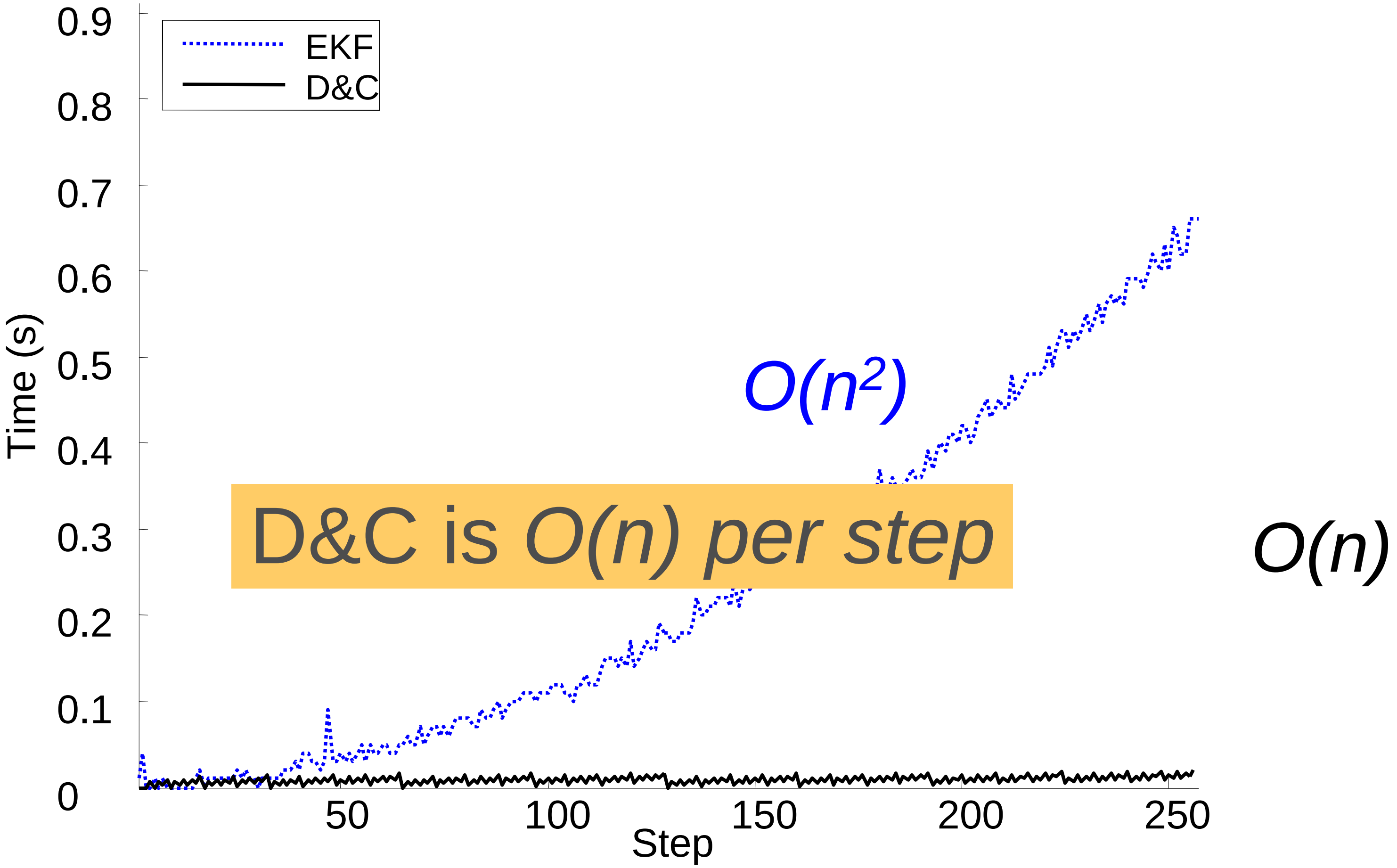
L.M. Paz, P. Jensfelt, J.D. Tardós and J. Neira. **EKF SLAM updates in O(n) with Divide and Conquer SLAM** 2007 IEEE Int. Conf. Robotics and Automation, April 10-14, Rome, Italy

ROBOTICS UNIVERSIDAD DE ZARAGOZA

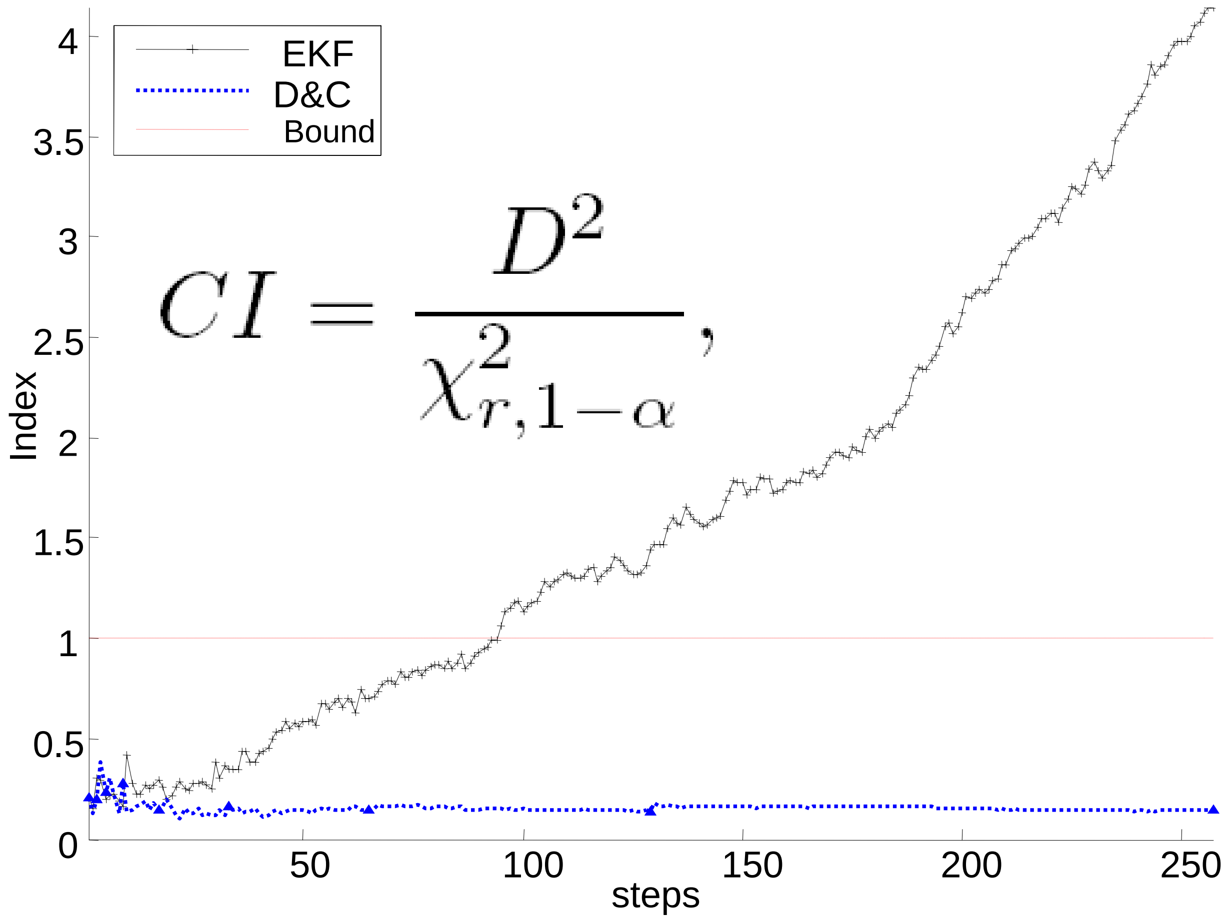
Loop Trajectory

L. Paz, J. Neira and J.D. Tardós **Divide and Conquer: EKF SLAM in O(n)**. Conditionally accepted, IEEE Transactions on Robotics, 2008.

Amortized cost per step



Improved Consistency over EKF SLAM



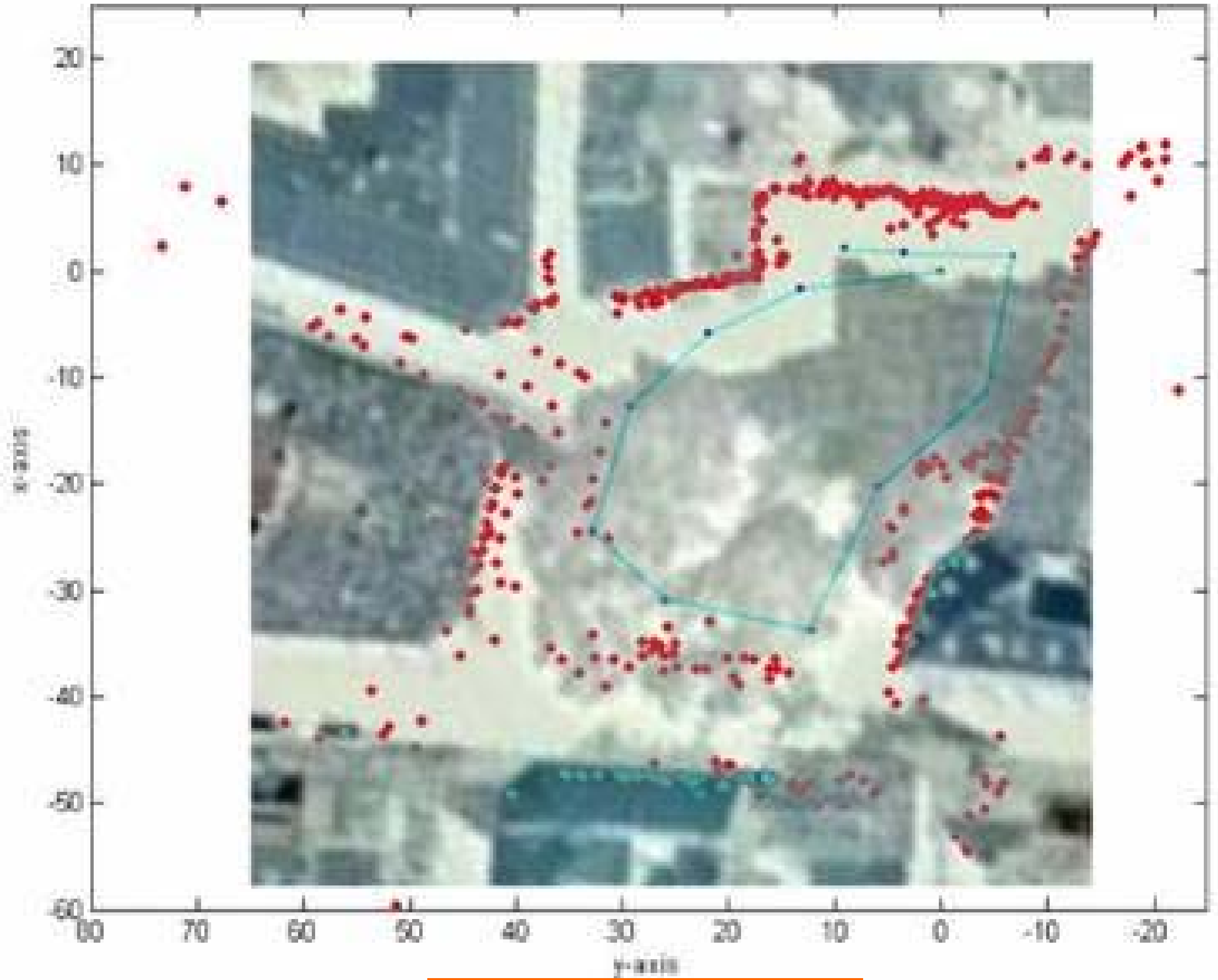
$$CI = \frac{D^2}{\chi^2_{r,1-\alpha}},$$

D&C SLAM is always more consistent

6DOF SLAM with stereo

L. Paz, P. Piníes, J. Neira and J.D. Tardós **Large Scale 6DOF SLAM with Stereo-in-Hand.** Conditionally accepted, IEE Transactions on Robotics, 2008.

6Dof Stereo SLAM, outdoors

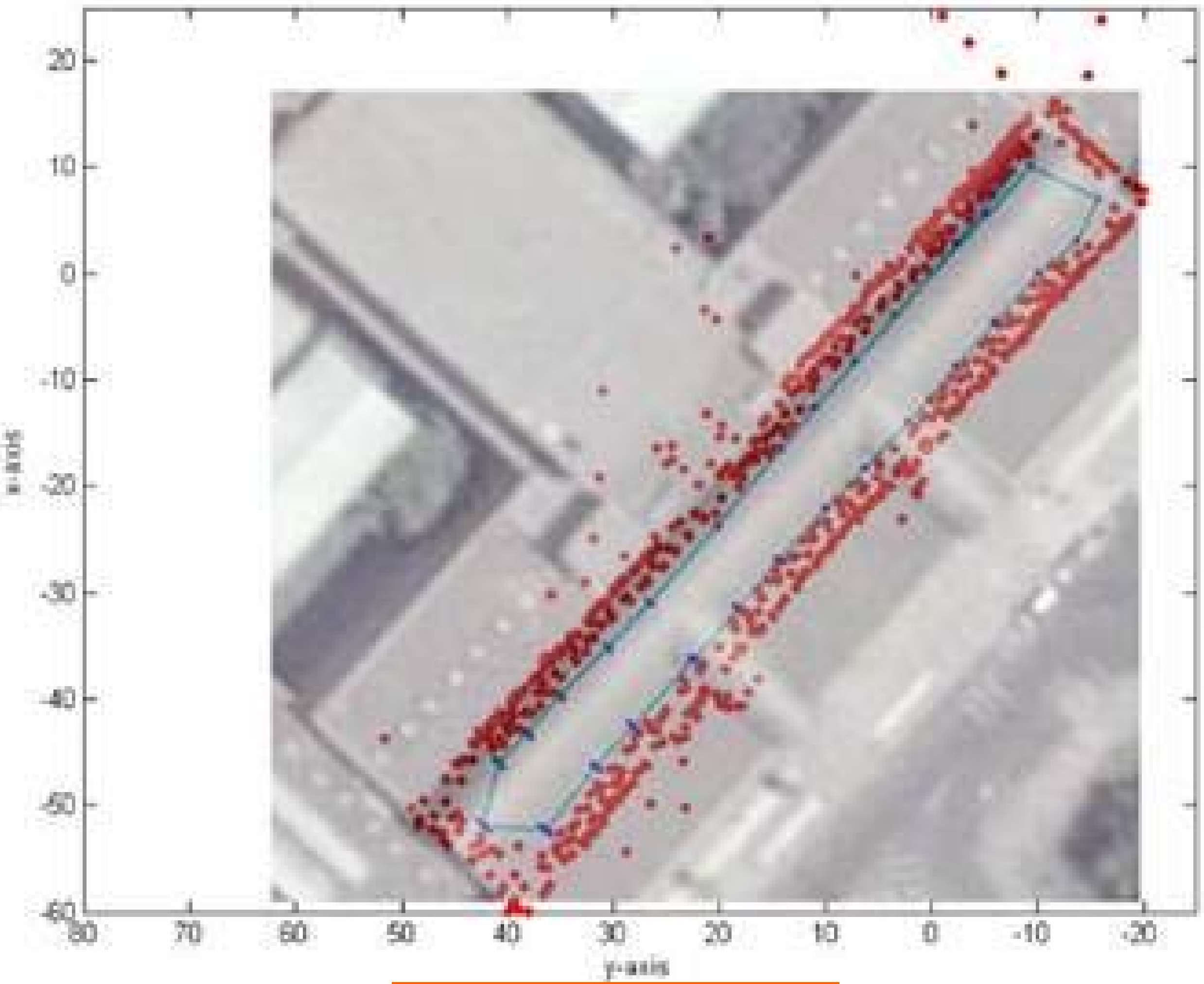


150 m loop

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

6DOF SLAM with stereo

6Dof Stereo SLAM, indoors



180 m loop

Conclusions

- Pure visual SLAM
  - 3D points and inverse depth points
  - Conditionally independent local maps
  - Near real-time execution
  - Loops of hundreds of meters
  - Stereo: no scale drift
- Working on
  - Larger loops
  - Robust place recognition for loop closing