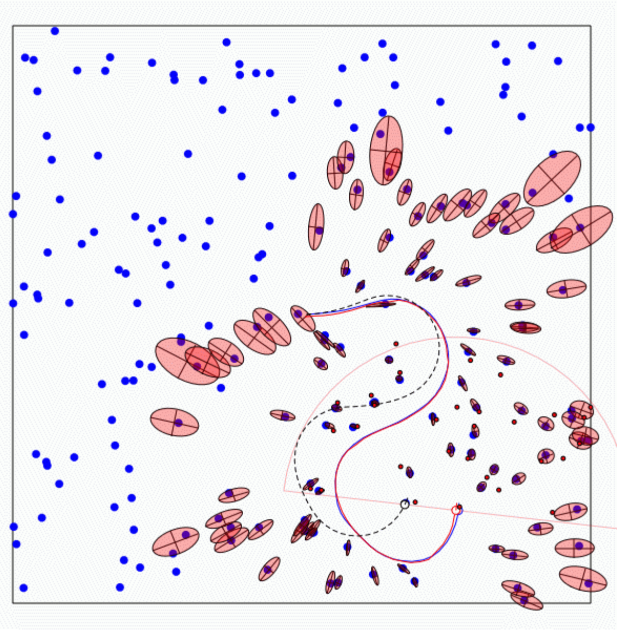
CSE-571
Probabilistic Robotics

SLAM: Simultaneous Localization and Mapping

The SLAM Problem



A robot is exploring an unknown, static environment.

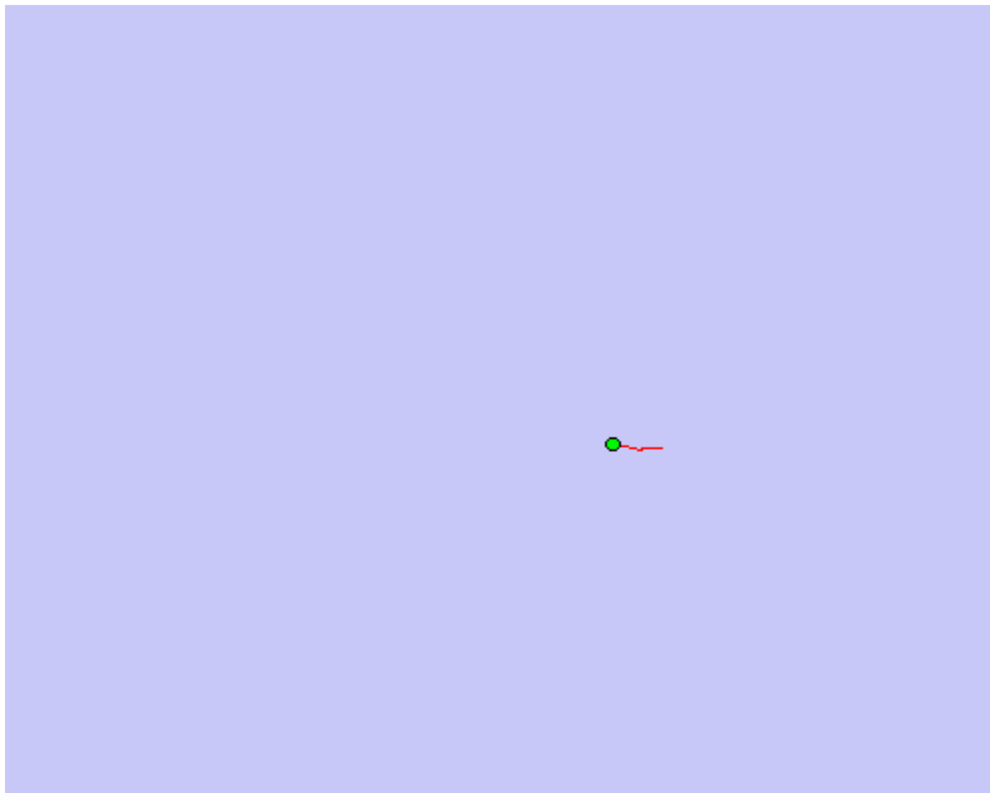
Given:

- The robot's controls
- Observations of nearby features

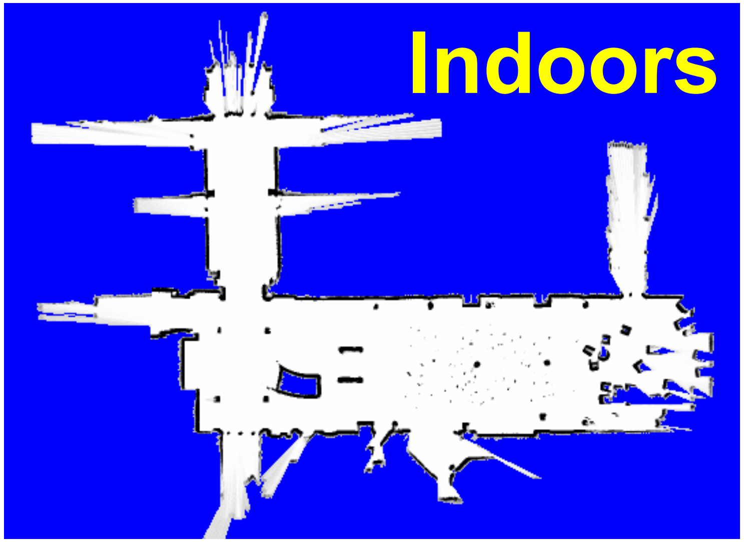
Estimate:

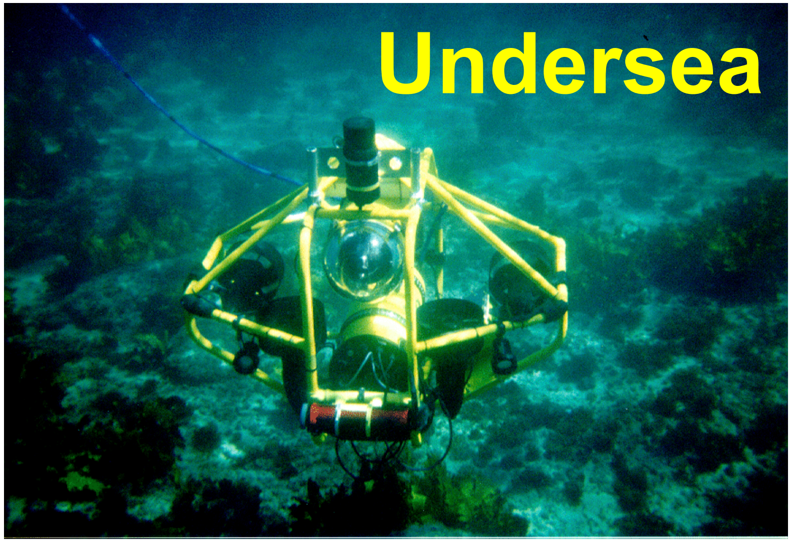
- Map of features
- Path of the robot

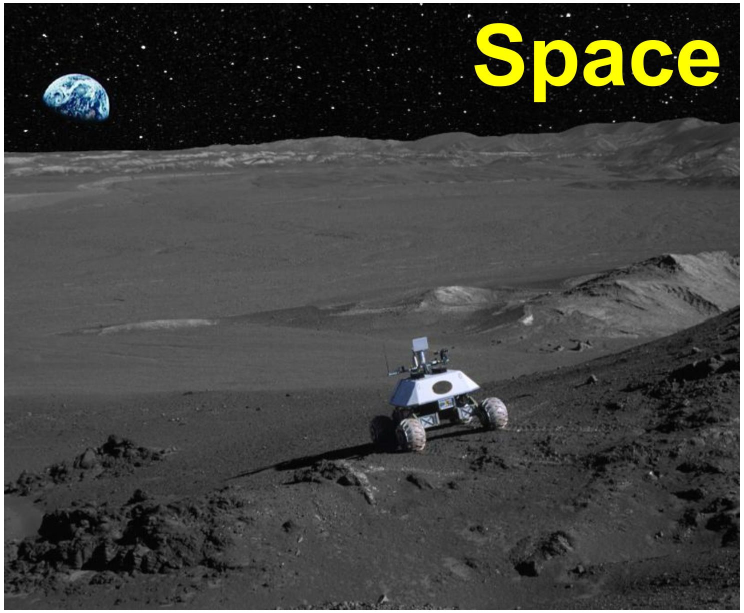
Mapping with Raw Odometry



SLAM Applications

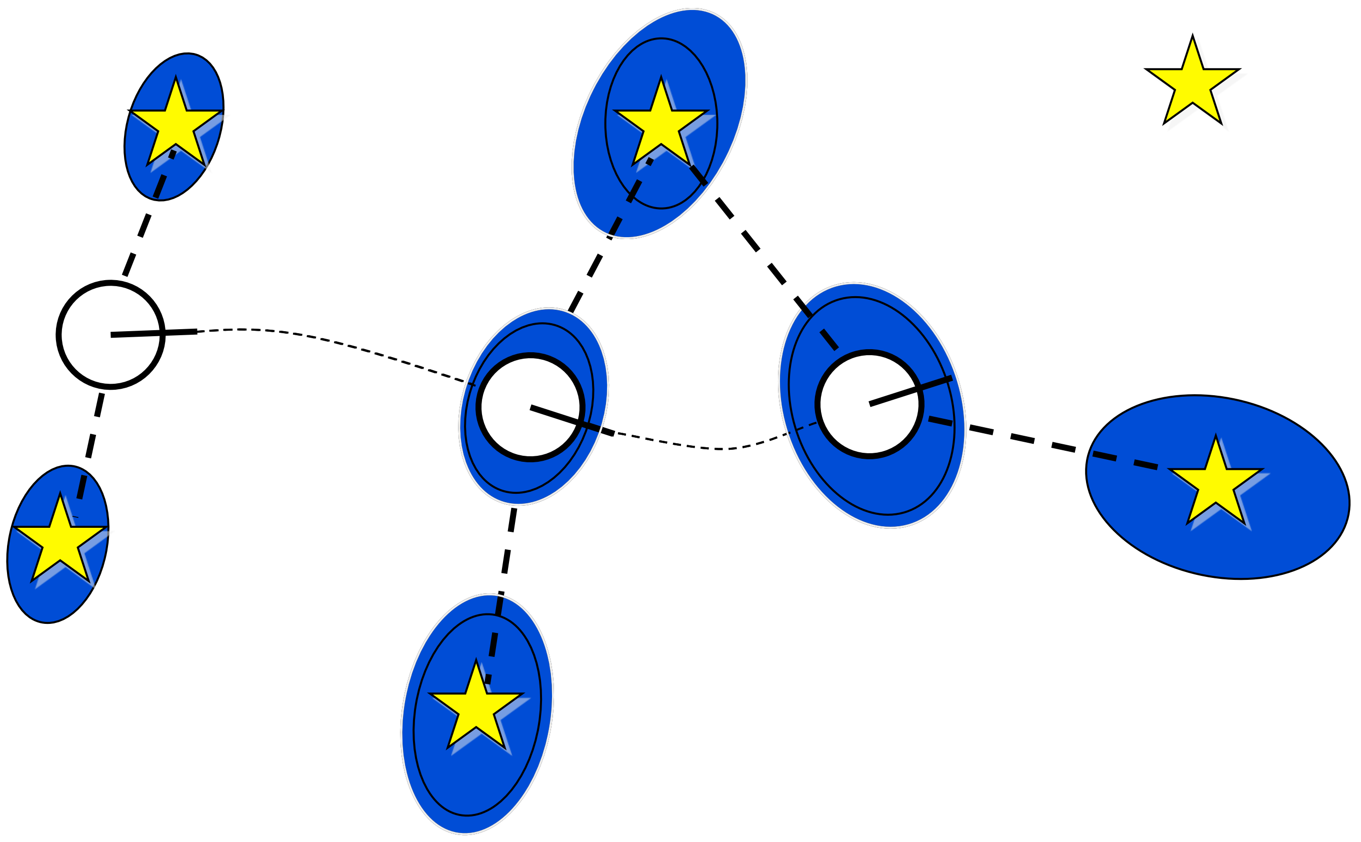






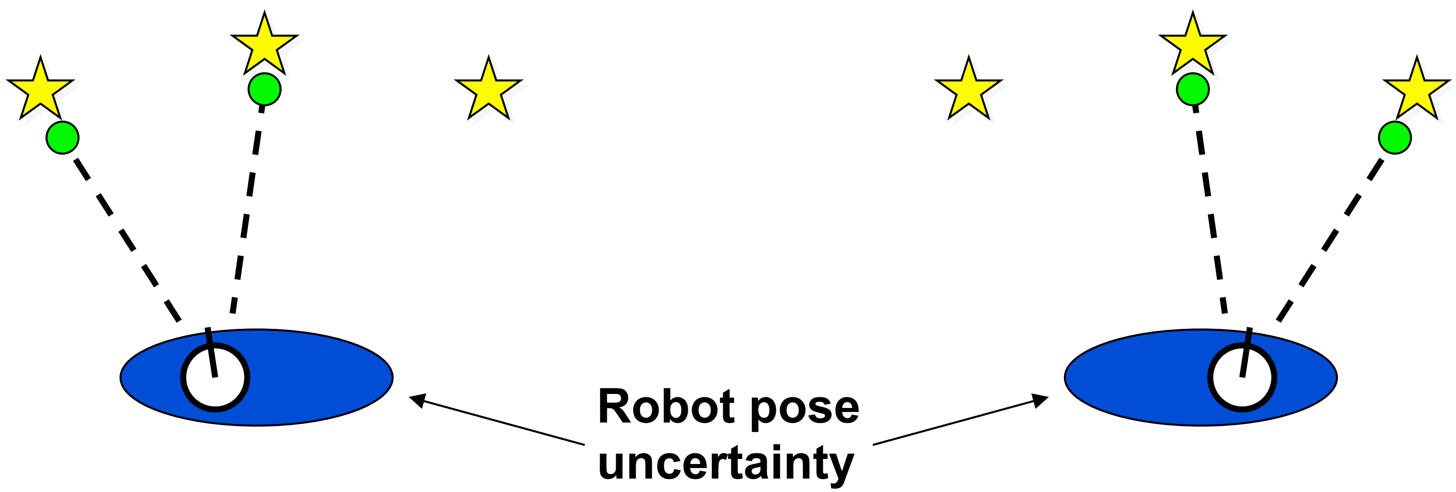
Why is SLAM a hard problem?

SLAM: robot path and map are both **unknown**



Robot path error correlates errors in the map

Why is SLAM a hard problem?



- In the real world, the mapping between observations and landmarks is unknown
- Picking wrong data associations can have catastrophic consequences
- Pose error correlates data associations

SLAM:
Simultaneous Localization and Mapping

- Full SLAM: Estimates entire path and map!

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

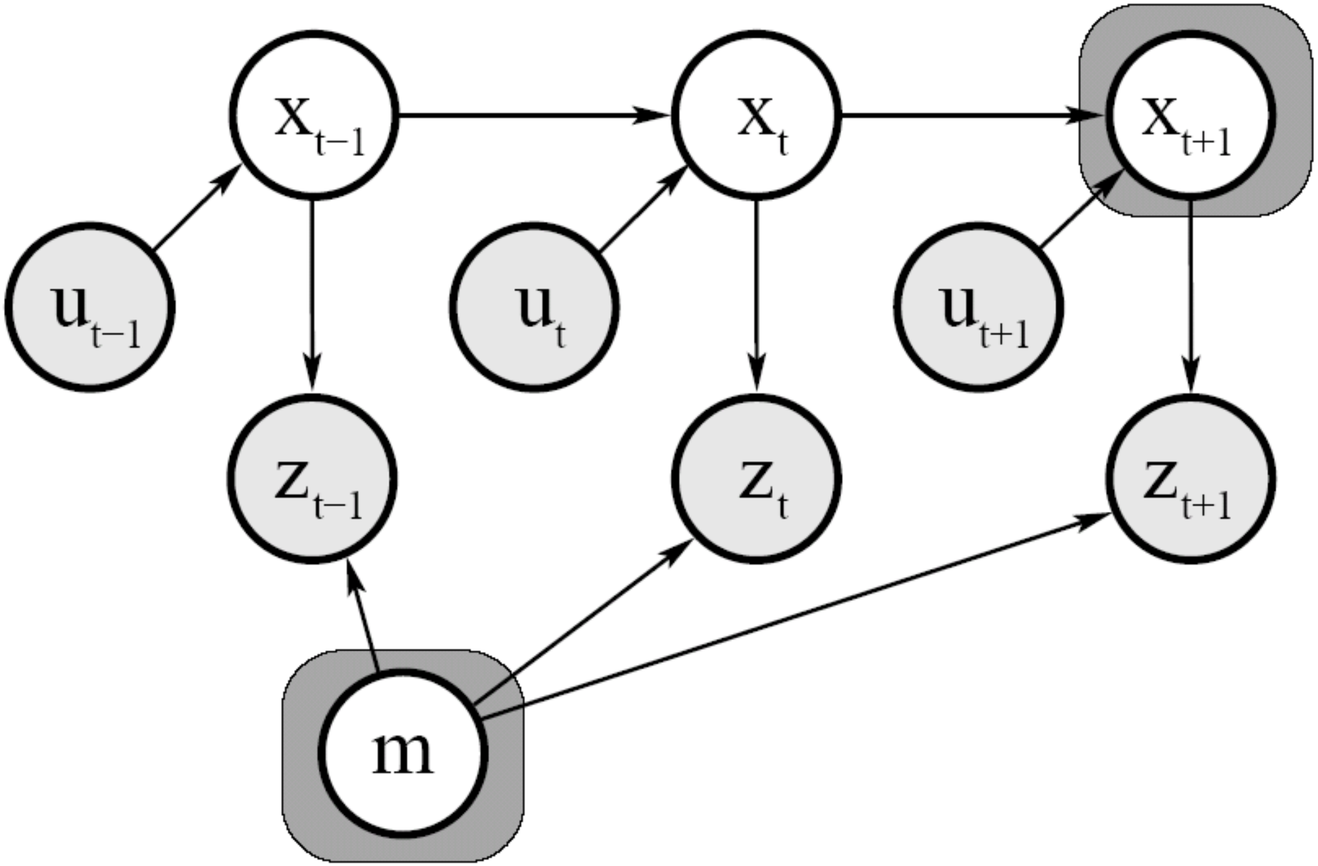
- Online SLAM:

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

Integrations typically done one at a time

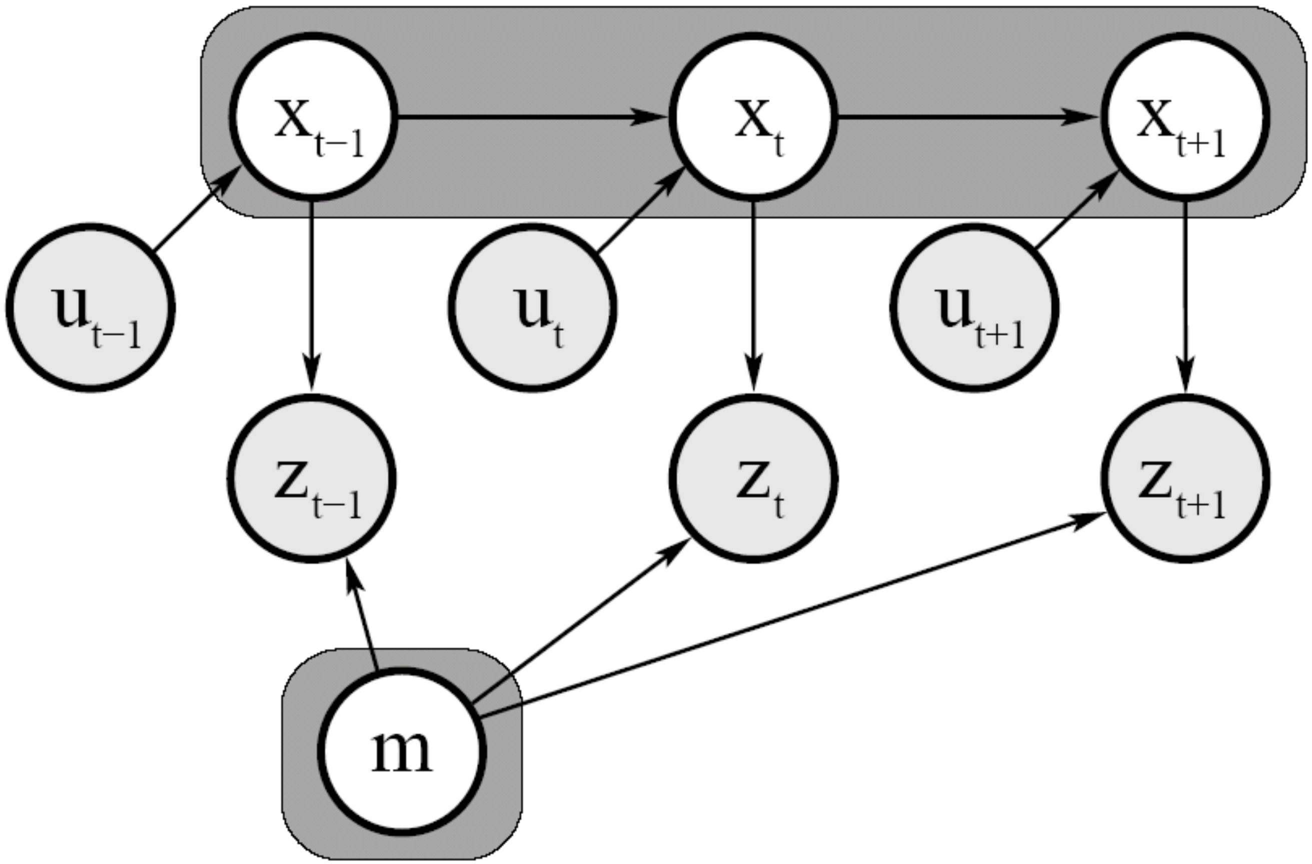
Estimates most recent pose and map!

Graphical Model of Online SLAM:



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

Graphical Model of Full SLAM:



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

Techniques for Generating Consistent Maps

- Scan matching
- EKF SLAM
- Graph-SLAM, SEIF
- Fast-SLAM
- MAP estimation

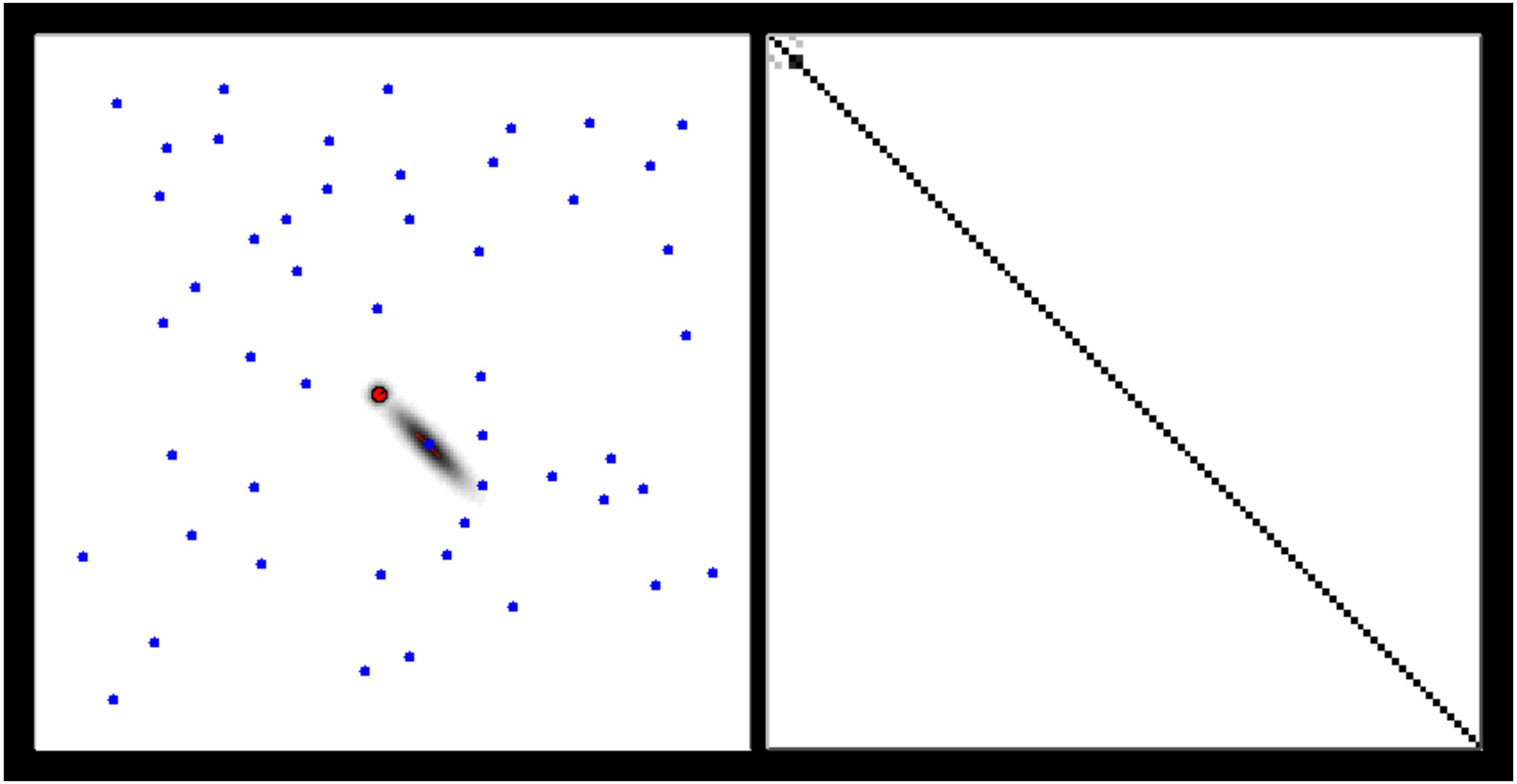
(E)KF-SLAM

- Map with N landmarks:(3+2N)-dimensional Gaussian

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

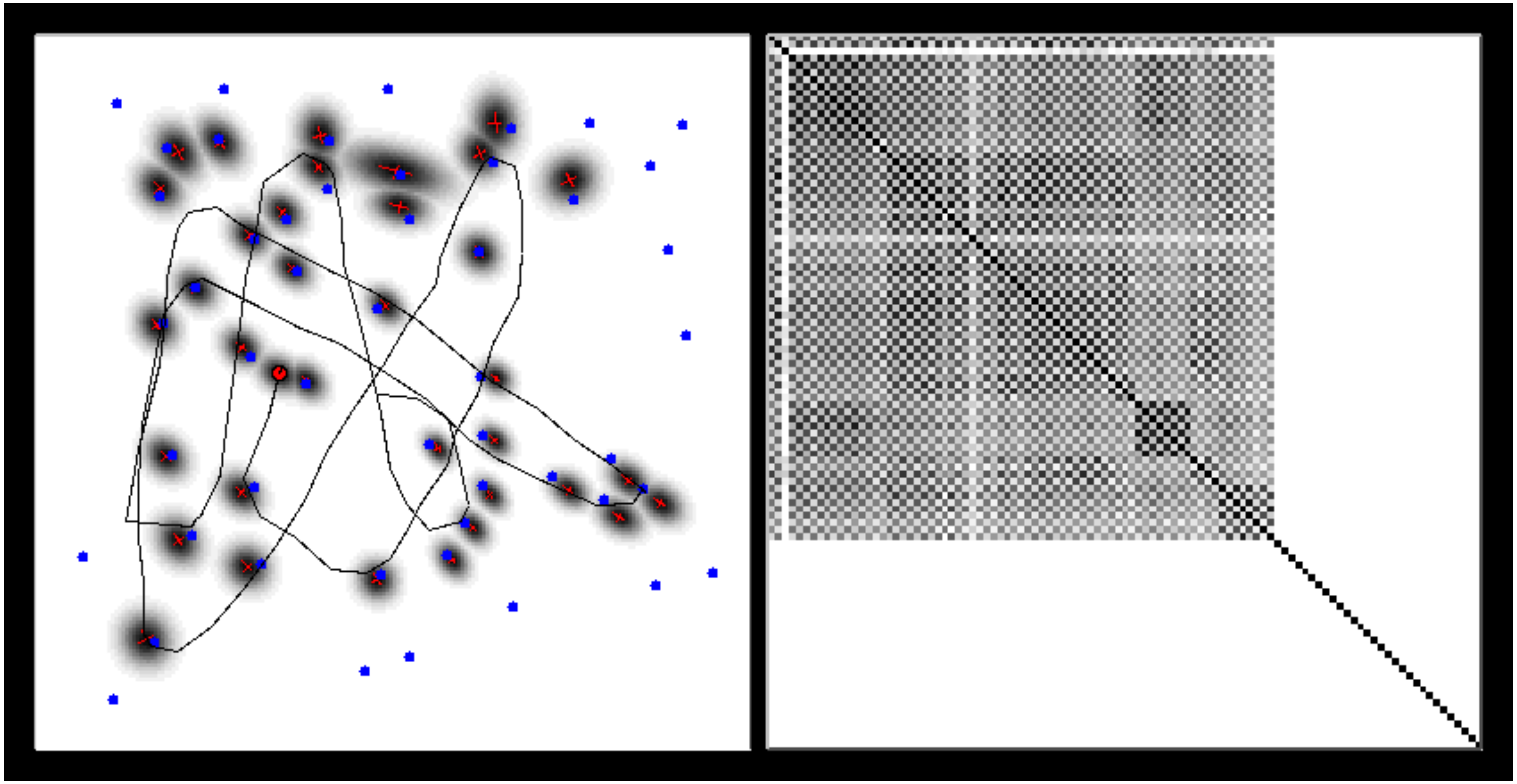
- Can handle hundreds of dimensions

EKF-SLAM



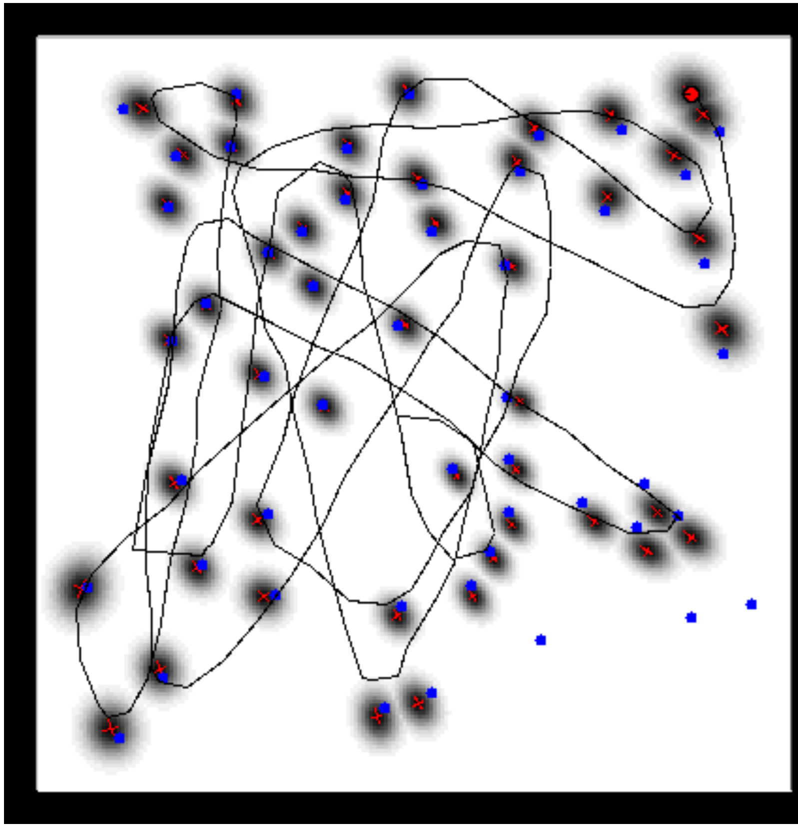
Map Correlation matrix

EKF-SLAM

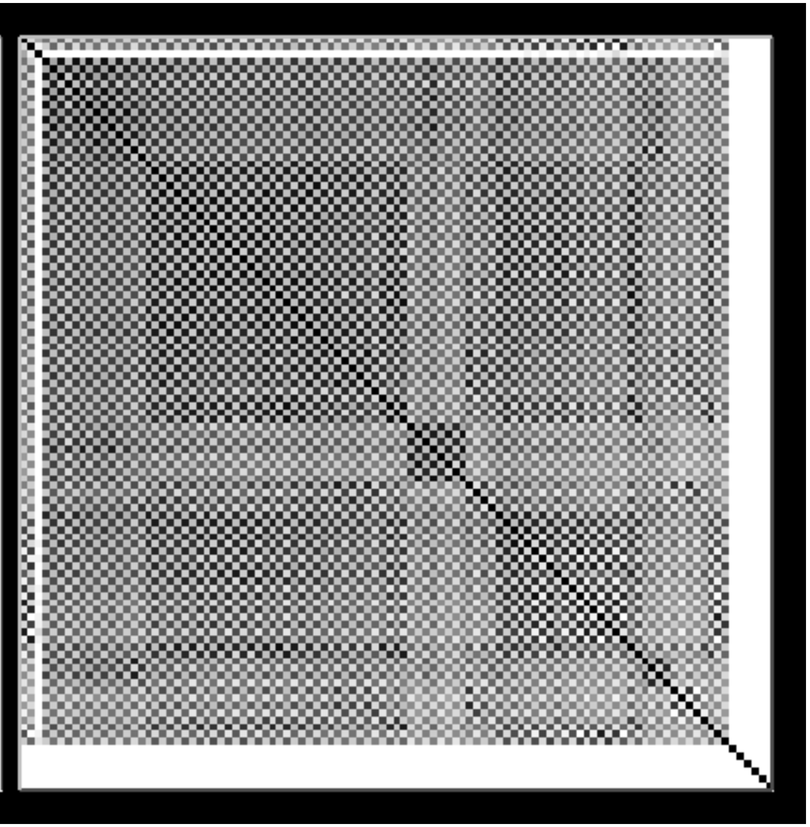


Map Correlation matrix

EKF-SLAM

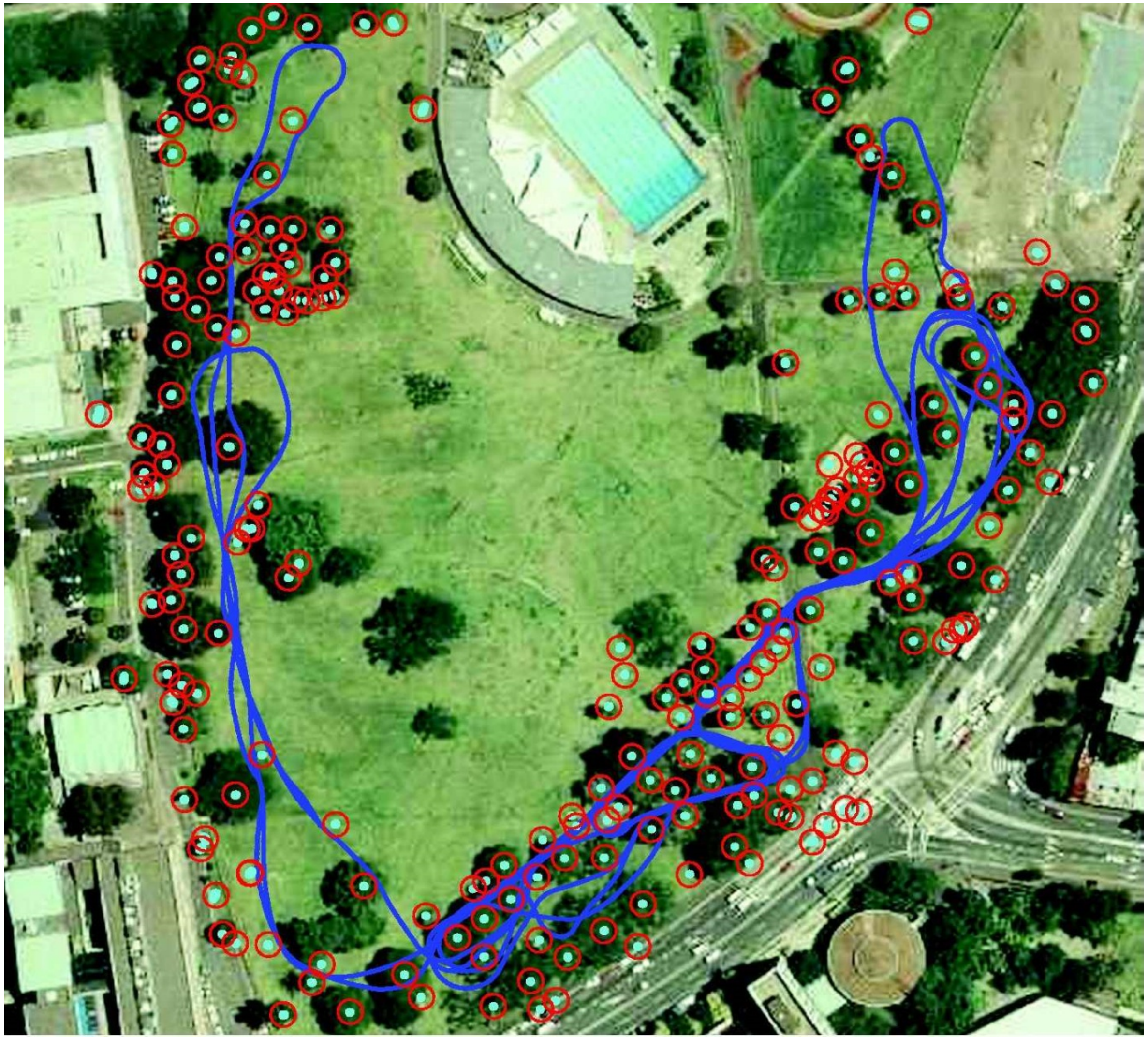


Map



Correlation matrix

Victoria Park Data Set



[courtesy of E. Nebot]

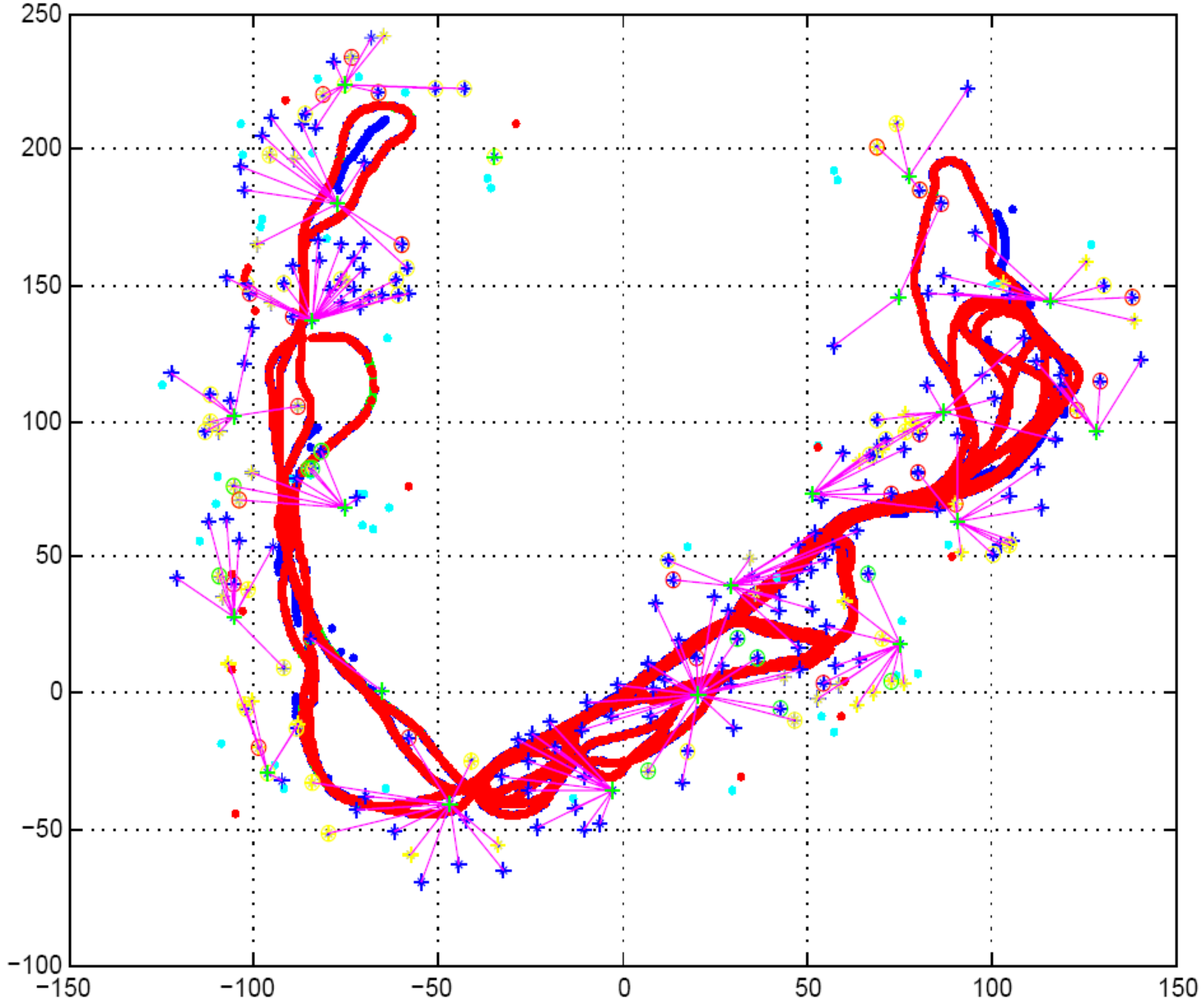
Victoria Park Data Set Vehicle

[courtesy of E. Nebot]

Data Acquisition

[courtesy of E. Nebot]

Estimated Trajectory



[courtesy of E. Nebot]

EKF-SLAM: Complexity

- Cost per step: quadratic in the number of landmarks: $O(n^2)$
- Total cost to build a map with n landmarks: $O(n^3)$
- Memory: $O(n^2)$

Approaches exist that make EKF-SLAM $O(n)$ / $O(n^2)$ / $O(n^2)$

Approximations for SLAM

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

- Sparse links (correlations)
  [Lu & Milios 97, Guivant & Nebot 01]

- Sparse extended information filters
  [Frese et al. 01, Thrun et al. 02]

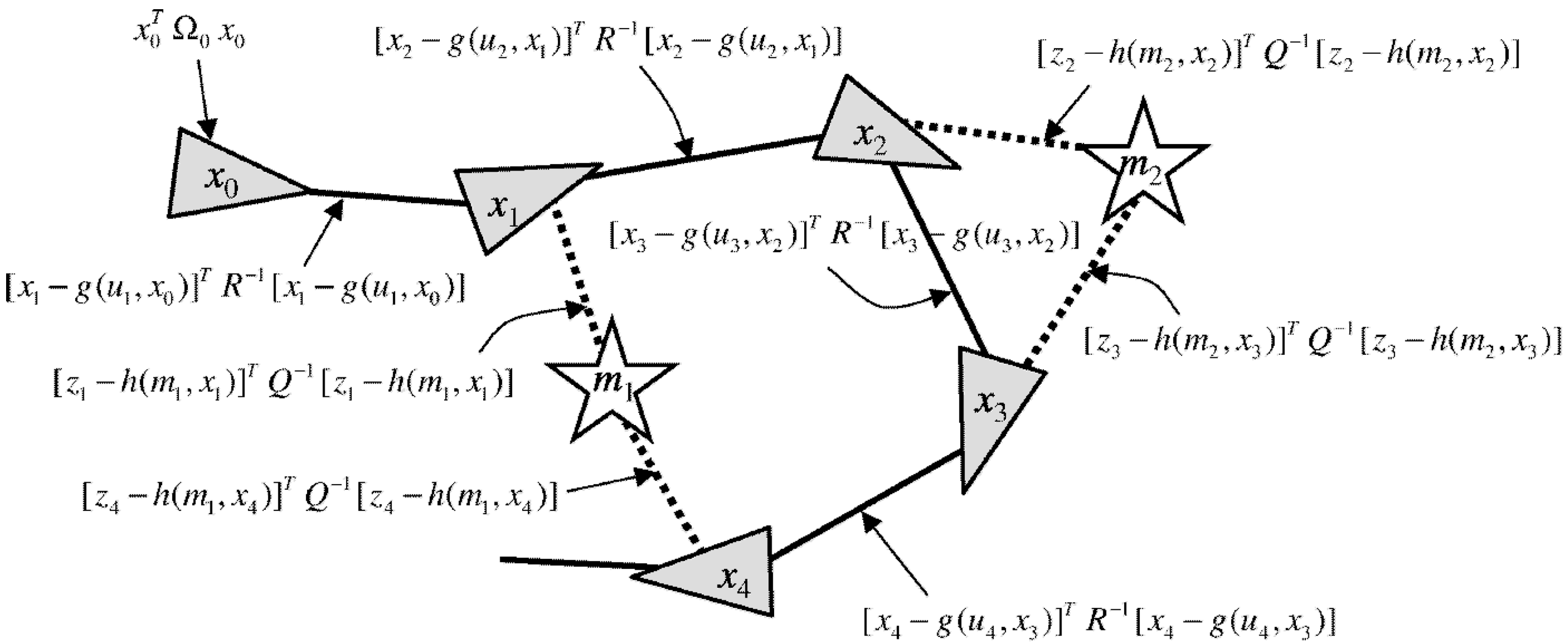
- Thin junction tree filters
  [Paskin 03]

- Rao-Blackwellisation (FastSLAM)
  [Murphy 99, Montemerlo et al. 02, Eliazar et al. 03, Haehnel et al. 03]

Graph-SLAM

- Full SLAM technique
- Generates probabilistic links
- Computes map only occasionally
- Based on Information Filter form

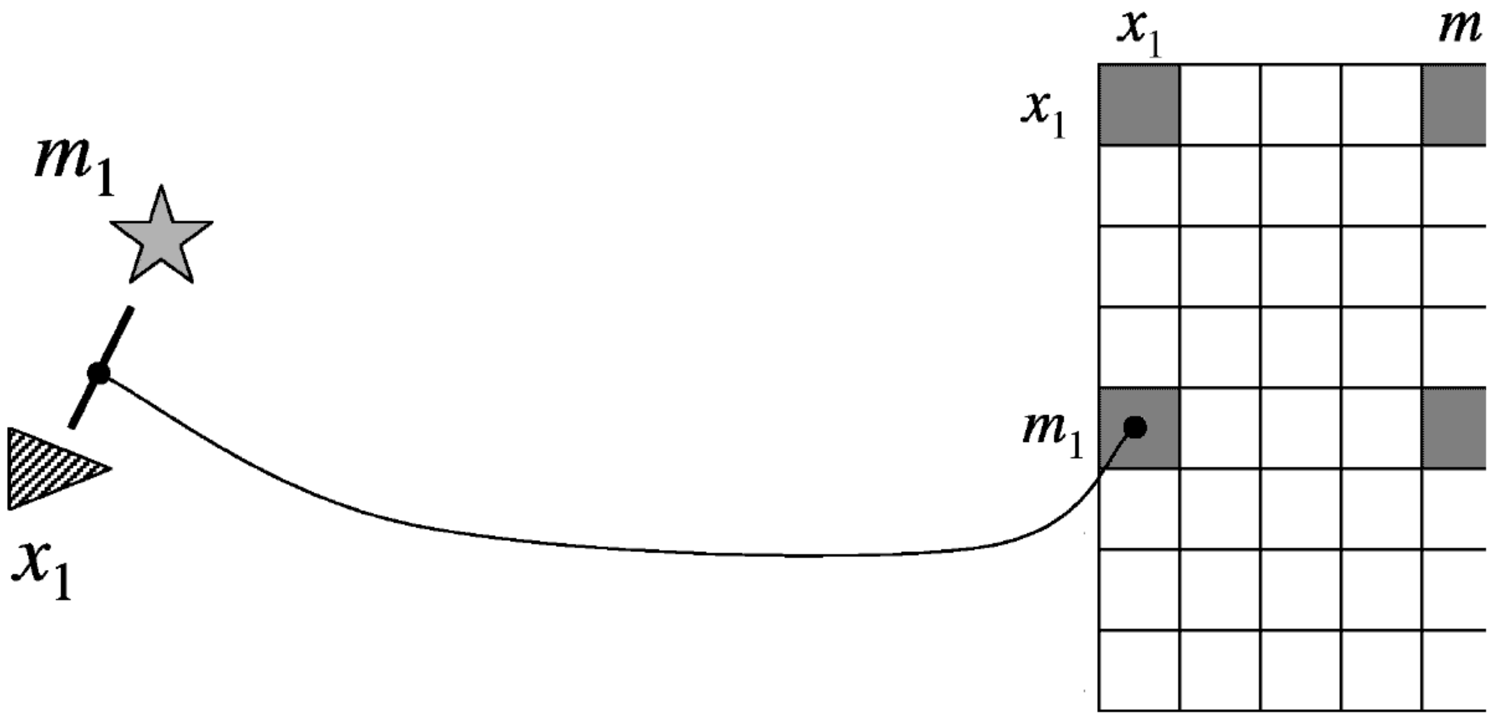
Graph-SLAM Idea



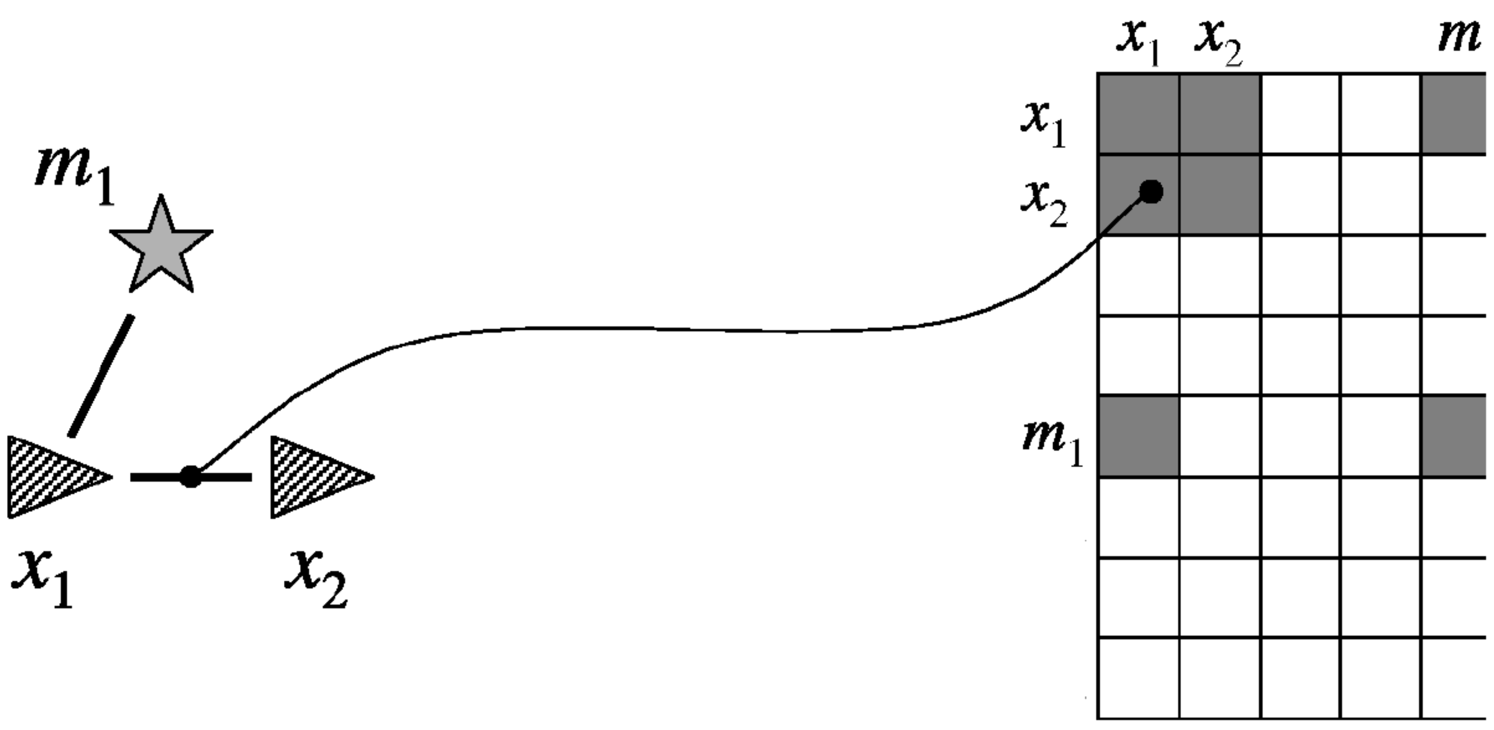
Sum of all constraints:

$$J_{\text{GraphSLAM}} = x_0^T \, \Omega_0 \, x_0 + \sum_t [x_t - g(u_t, x_{t-1})]^T R^{-1} [x_t - g(u_t, x_{t-1})] + \sum_t [z_t - h(m_{c_t}, x_t)]^T Q^{-1} [z_t - h(m_{c_t}, x_t)]$$

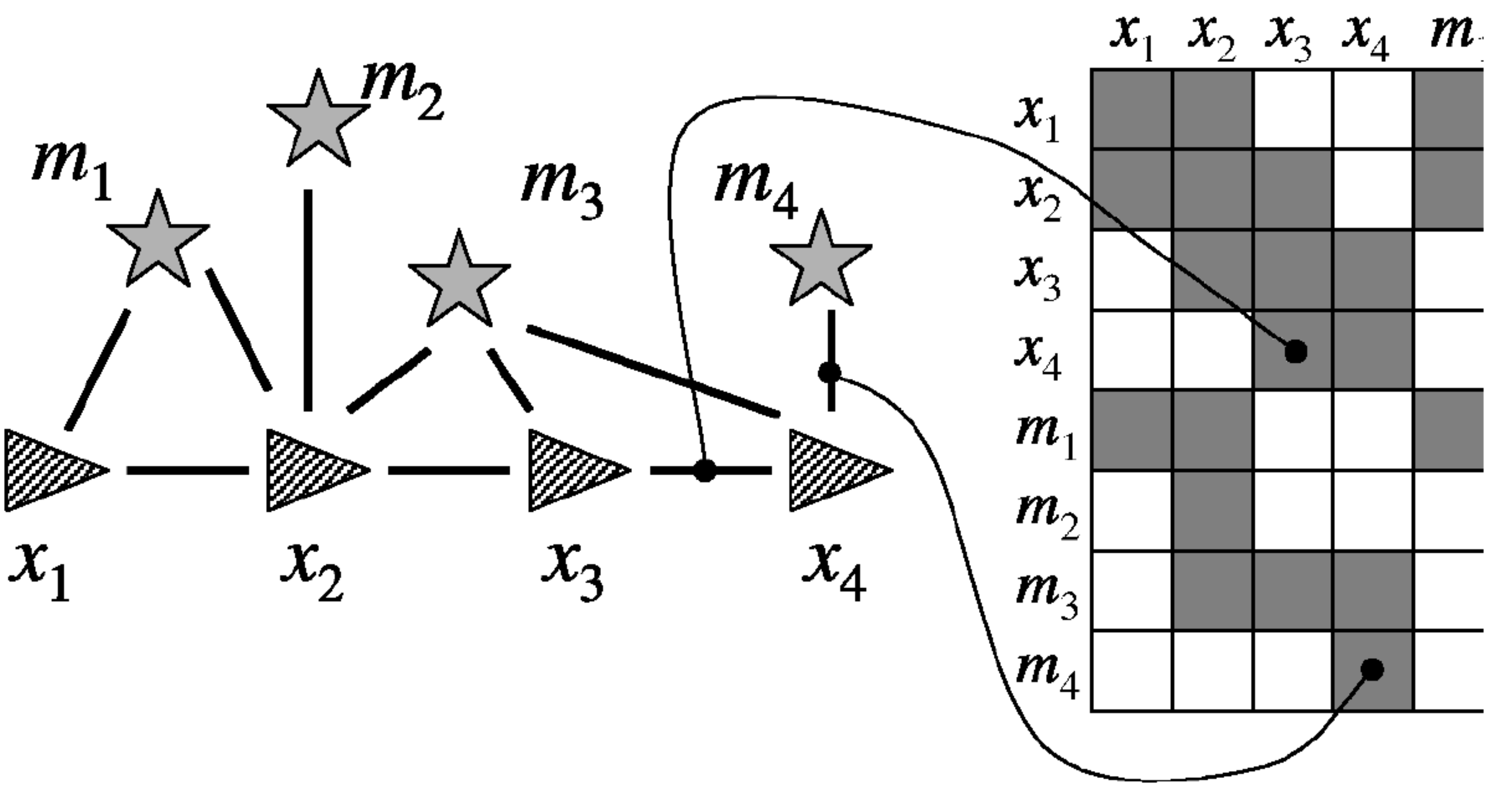
Graph-SLAM Idea (1)



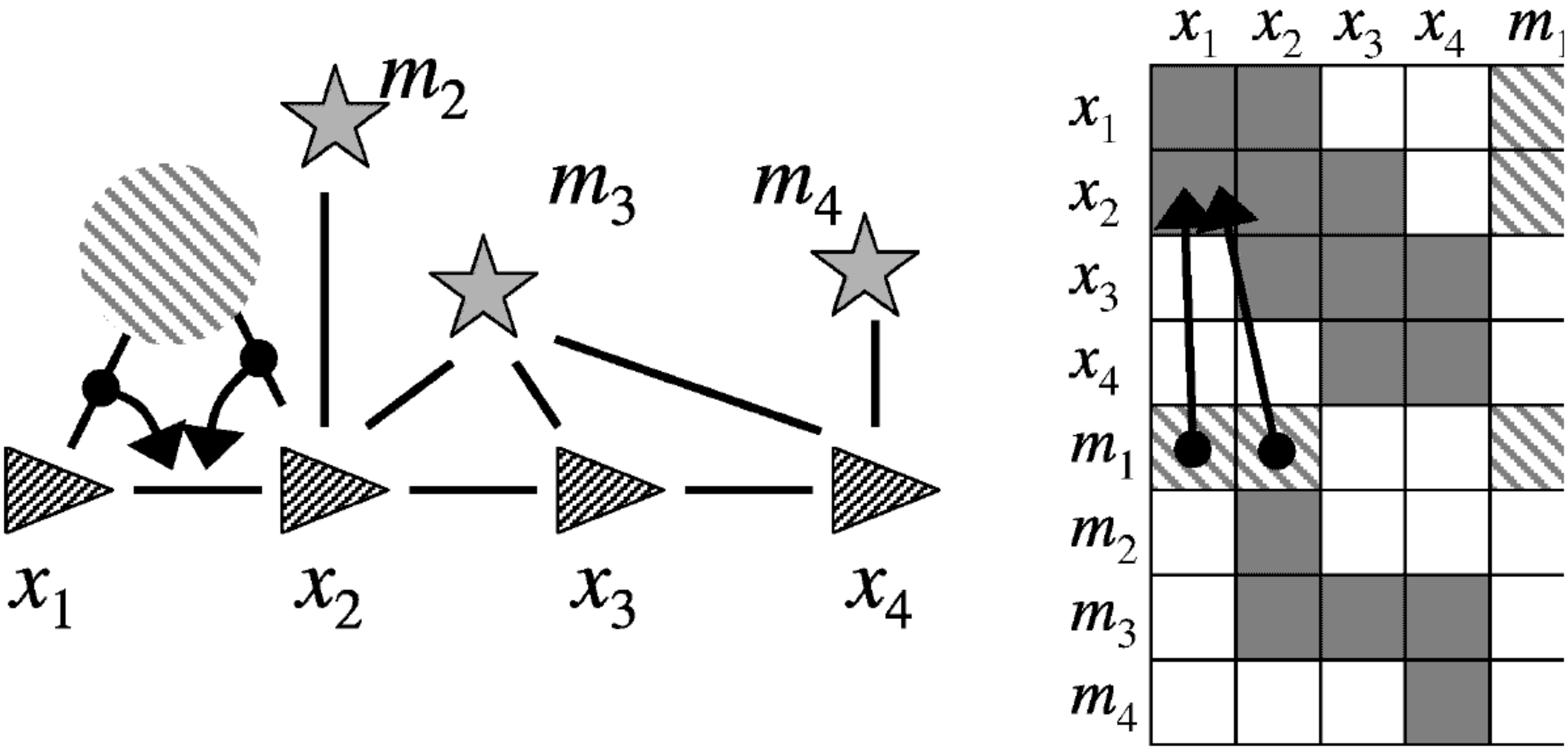
Graph-SLAM Idea (2)



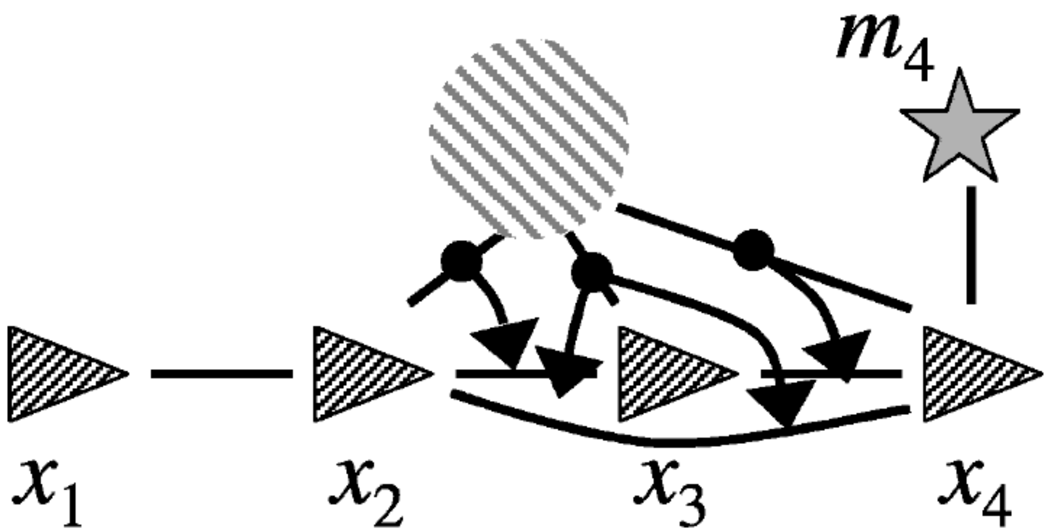
Graph-SLAM Idea (3)

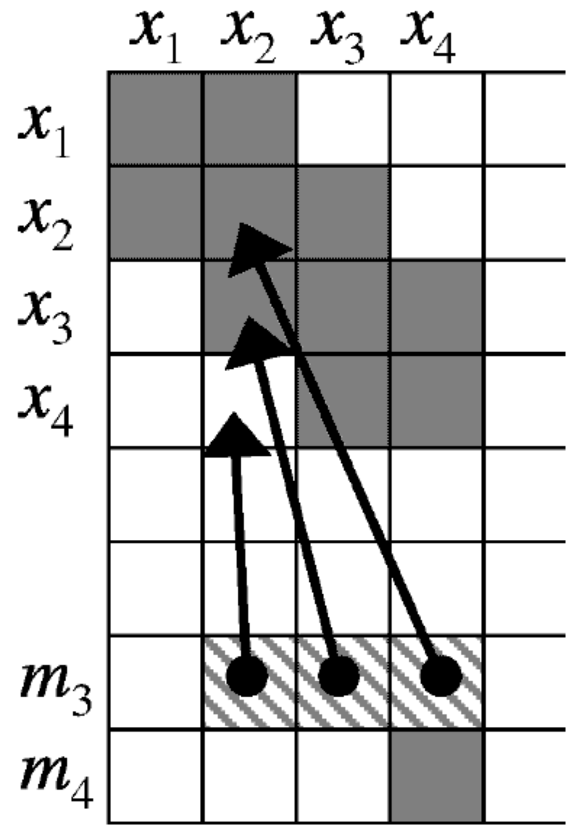


Graph-SLAM Inference (1)

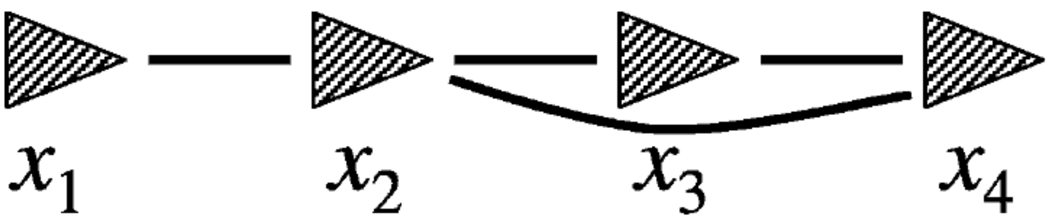


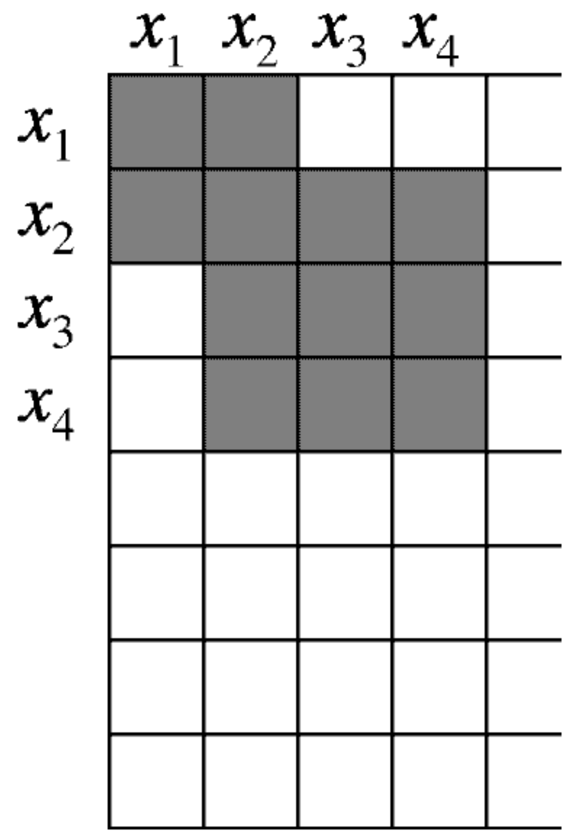
Graph-SLAM Inference (2)





Graph-SLAM Inference (3)



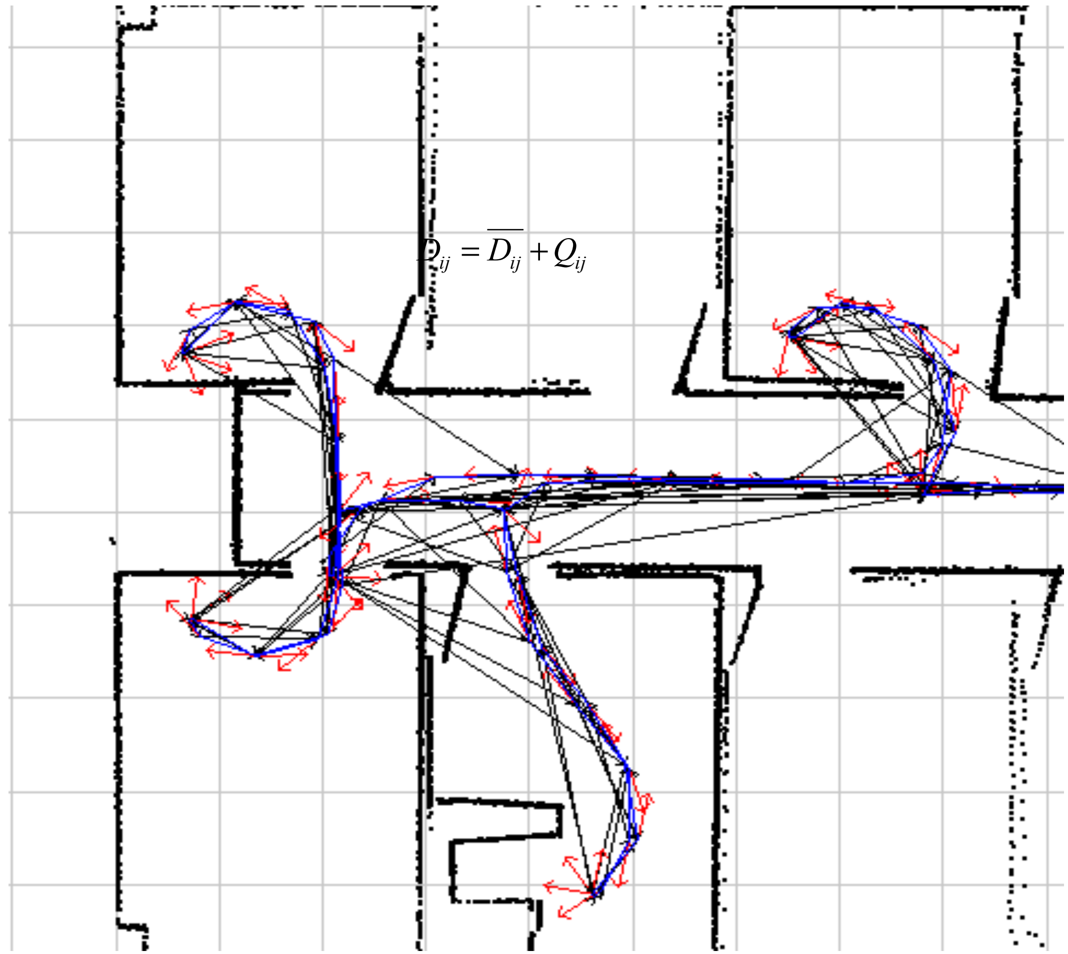


Robot Poses and Scans [Lu and Milios 1997]

- Successive robot poses connected by odometry
- Sensor readings yield constraints between poses
- Constraints represented by Gaussians

$$D_{ij} = \overline{D_{ij}} + Q_{ij}$$

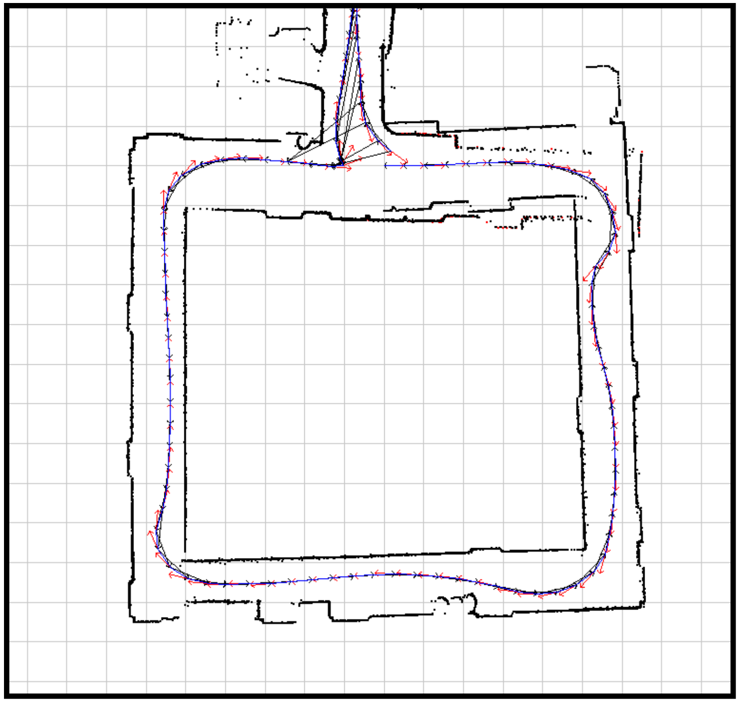
- Globally optimal estimate



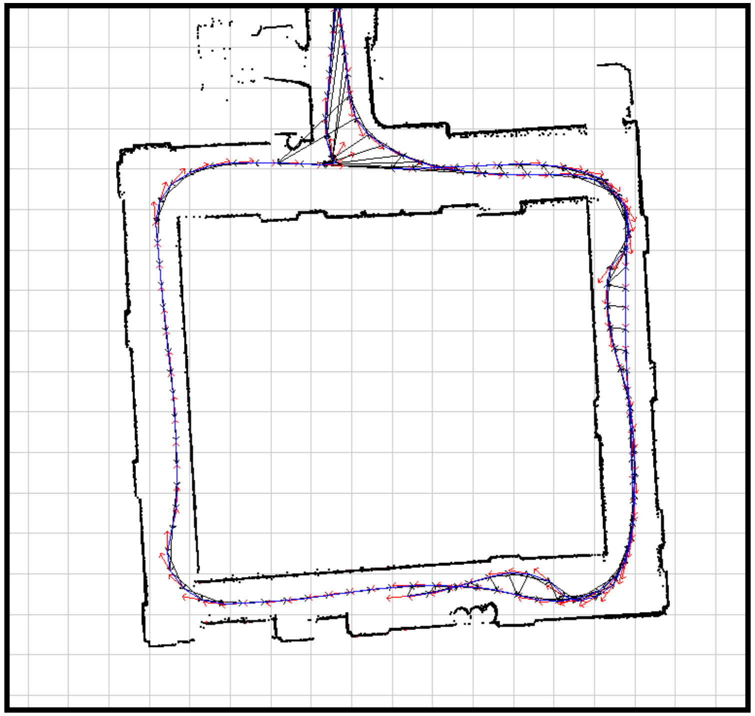
$$\arg\max_{X_i} \left[P(D_{ij} \mid \overline{D_{ij}}) \right]$$

Loop Closure

- Use scan patches to detect loop closure
- Add new position constraints
- Deform the network based on covariances of matches



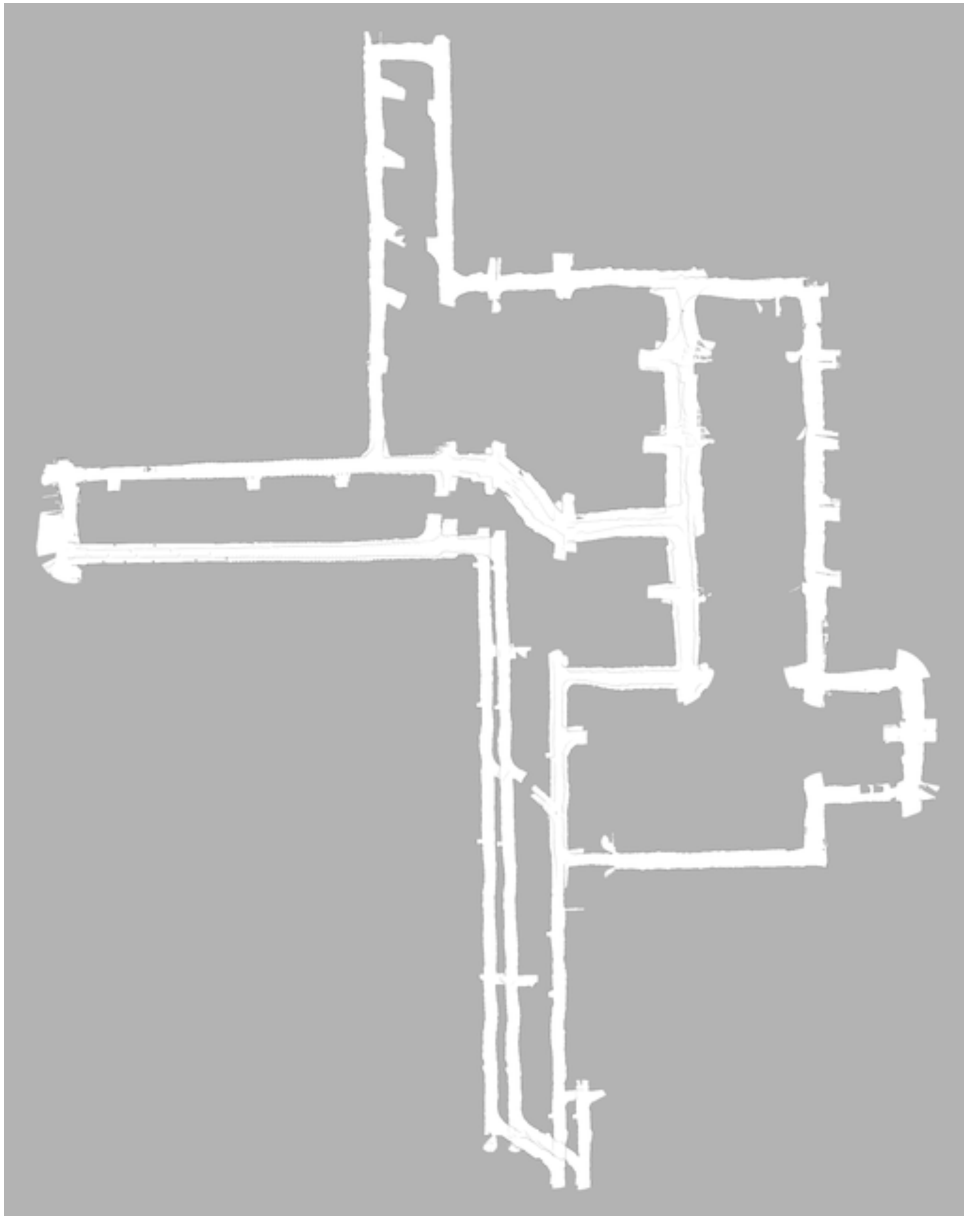
Before loop closure



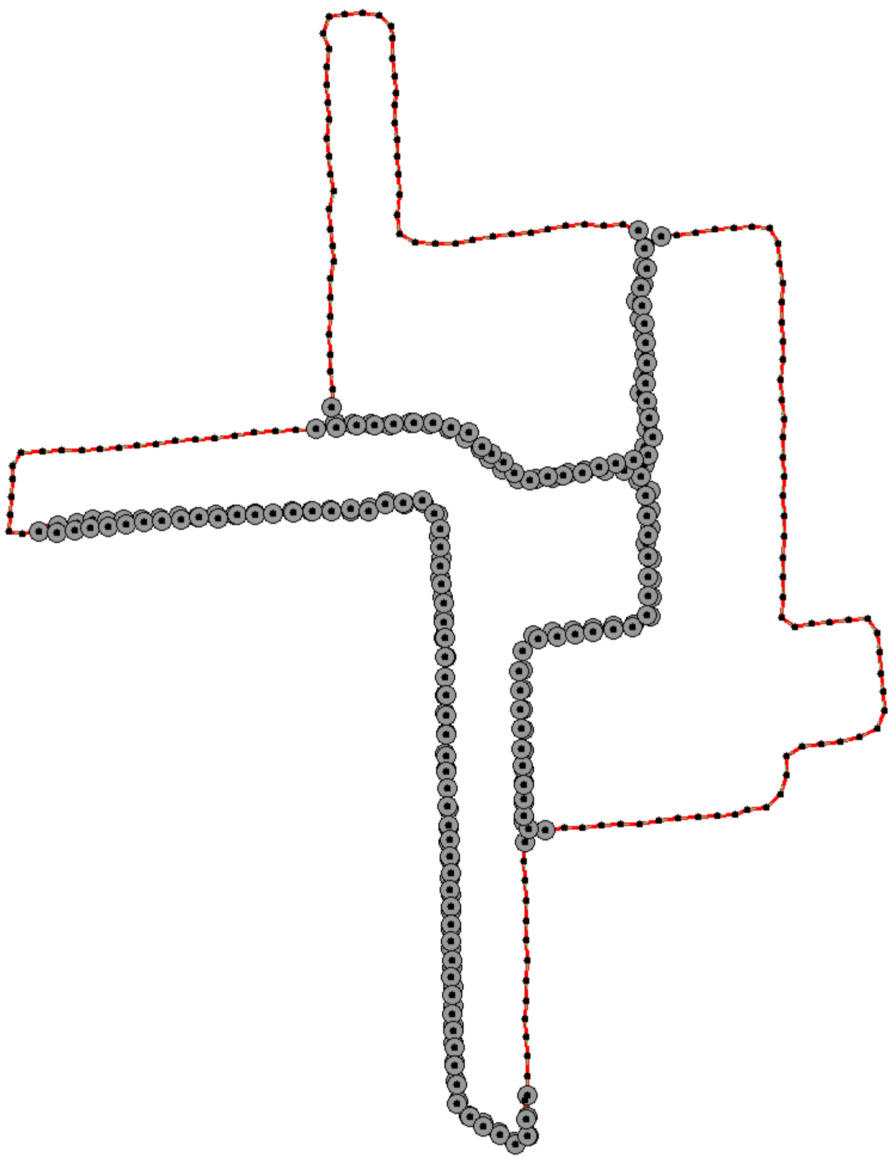
After loop closure

Mapping the Allen Center

Mine Mapping



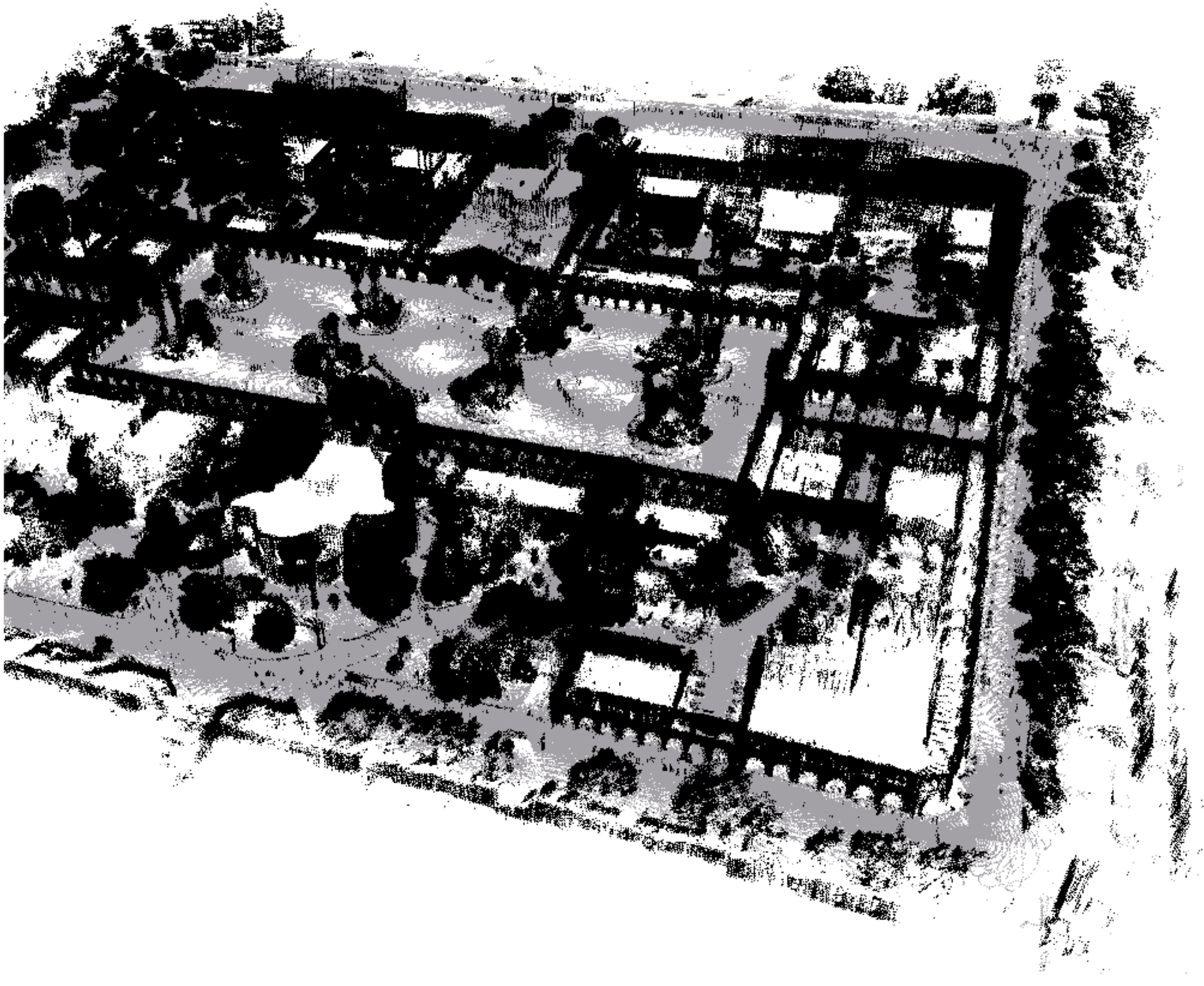
Mine Mapping: Data Associations



Efficient Map Recovery

- Information matrix inversion can be avoided if only best map estimate is required
- Minimize constraint function $J_{GraphSLAM}$ using standard optimization techniques (gradient descent, Levenberg Marquardt, conjugate gradient)

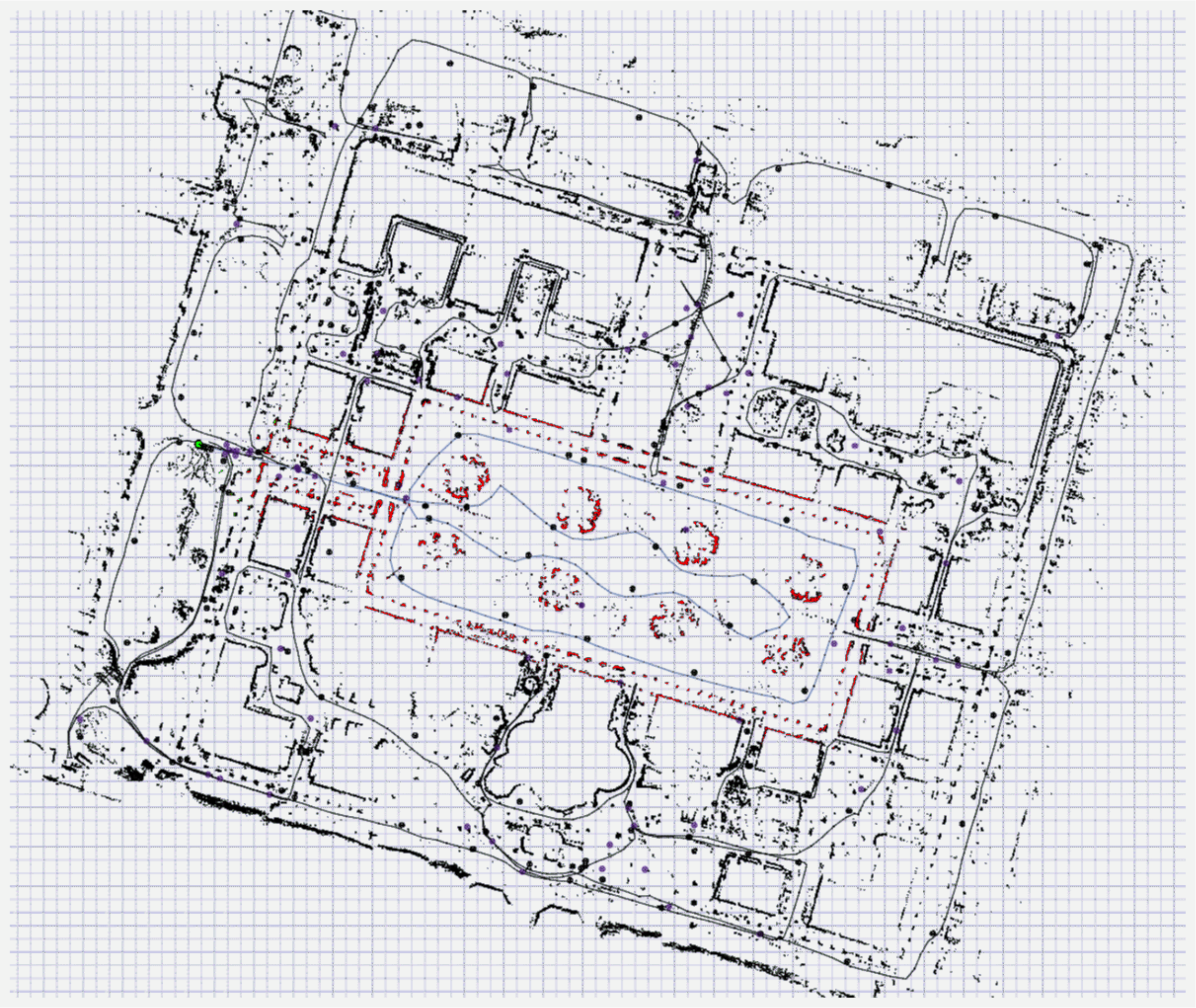
3D Outdoor Mapping



10^8 features, 10^5 poses, only few secs using cg.

Map Before Optimization

Map After Optimization



Graph-SLAM Summary

- Adresses full SLAM problem
- Constructs link graph between poses and poses/ landmarks
- Graph is sparse: number of edges linear in number of nodes
- Inference performed by building information matrix and vector (linearized form)
- Map recovered by reduction to robot poses, followed by conversion to moment representation, followed by estimation of landmark positions
- ML estimate by minimization of $J_{GraphSLAM}$
- Data association by iterative greedy search