Introduction to Mobile Robotics

SLAM: Simultaneous Localization and Mapping

Kshitij Sirohi

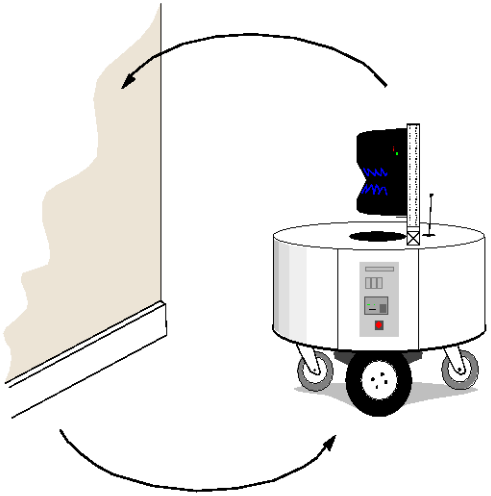
What is SLAM?

- Estimate the pose of a robot and the map of the environment at the same time
- SLAM is hard, because
  - a map is needed for localization and
  - a good pose estimate is needed for mapping

- **Localization:** inferring location given a map
- **Mapping:** inferring a map given locations
- **SLAM:** learning a map and locating the robot simultaneously

The SLAM Problem

- SLAM has long been regarded as a **chicken-or-egg** problem:
  → a map is needed for localization and
  → a pose estimate is needed for mapping



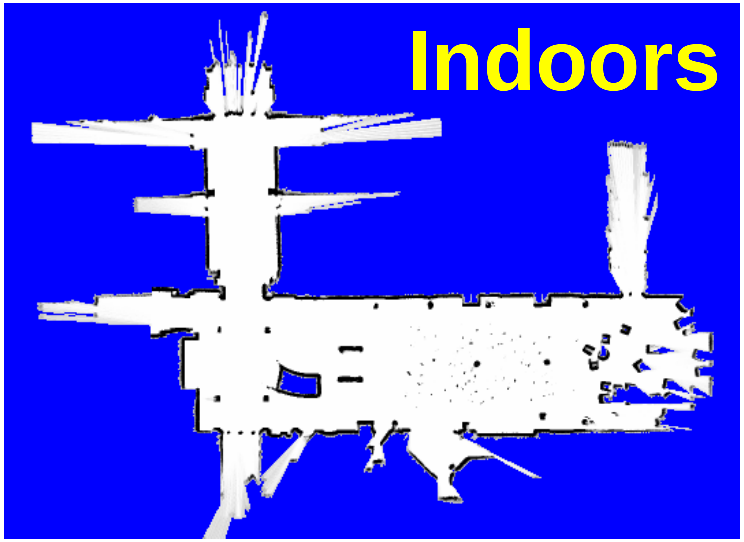
SLAM Applications

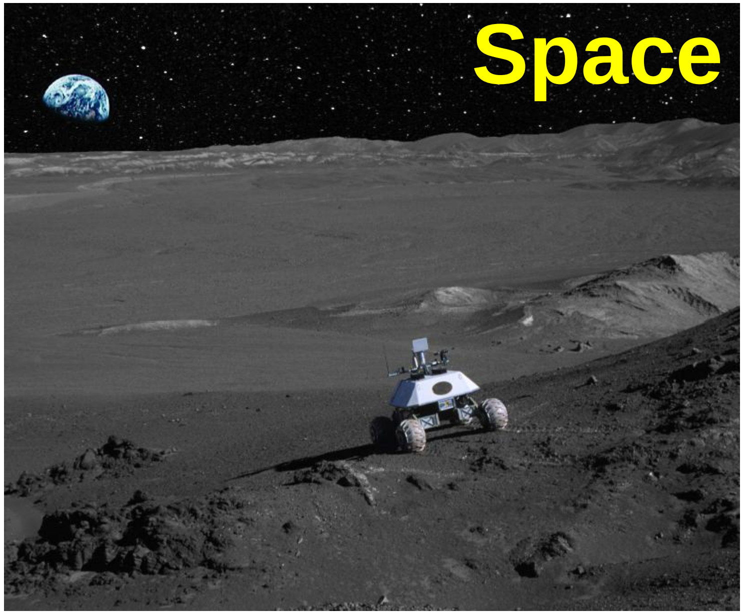
- SLAM is central to a range of indoor, outdoor, in-air and underwater applications for both manned and autonomous vehicles.

Examples:

- At home: vacuum cleaner, lawn mower
- Air: surveillance with unmanned air vehicles
- Underwater: reef monitoring
- Underground: exploration of mines
- Space: terrain mapping for localization

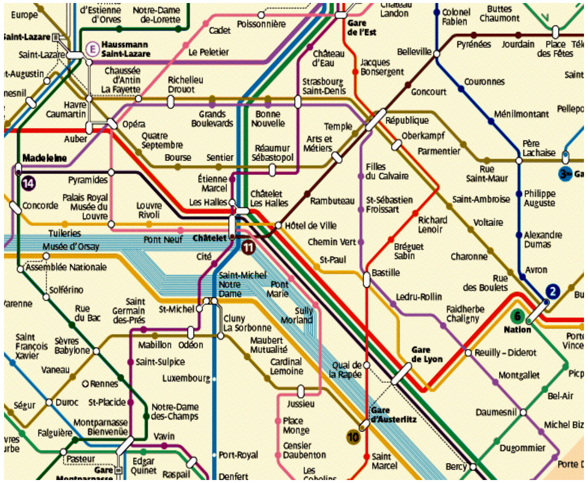
SLAM Applications



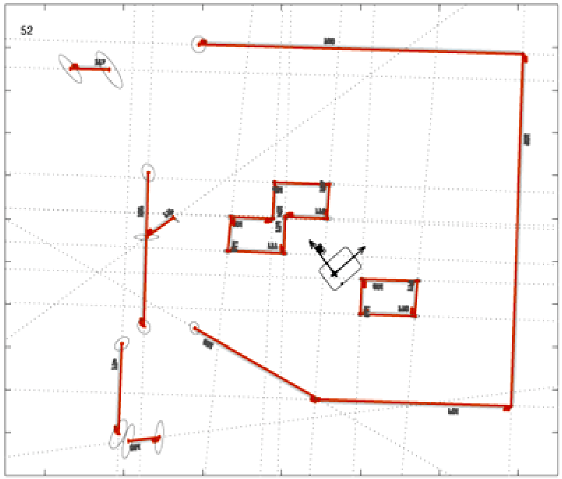


Map Representations

Examples: Subway map, city map, landmark-based map



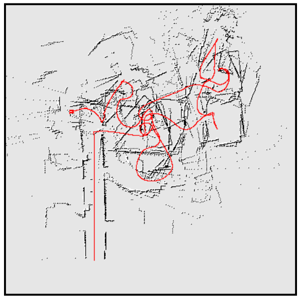
537
501
534
495
Mor
447
474

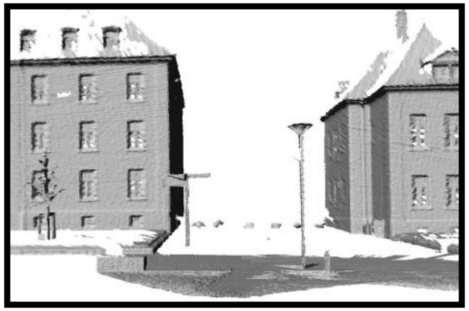


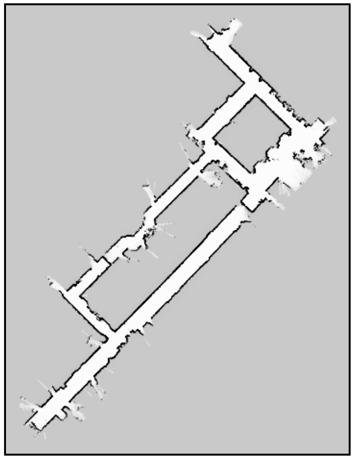
Maps are **topological** and/or **metric models** of the environment

Map Representations in Robotics

- Grid maps or scans, 2d, 3d

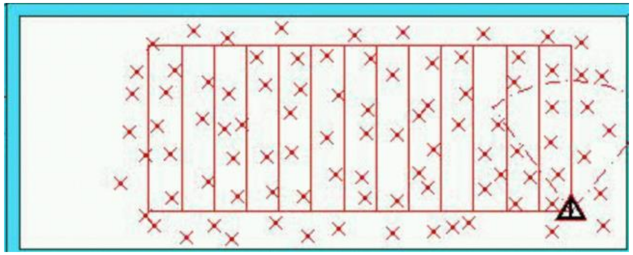


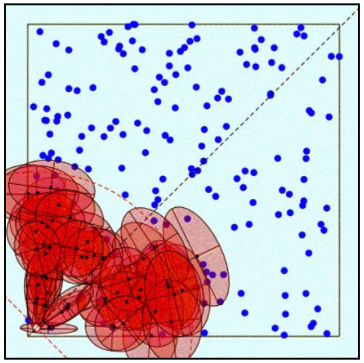




[Lu & Milios, 97; Gutmann, 98: Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01; Grisetti et al., 05; …]

- Landmark-based





[Leonard et al., 98; Castelanos et al., 99: Dissanayake et al., 2001; Montemerlo et al., 2002;…

The SLAM Problem

- SLAM is considered a fundamental problems for robots to become truly autonomous
- Large variety of different SLAM approaches have been developed
- The majority uses probabilistic concepts
- History of SLAM dates back to the mid-eighties

Feature-Based SLAM

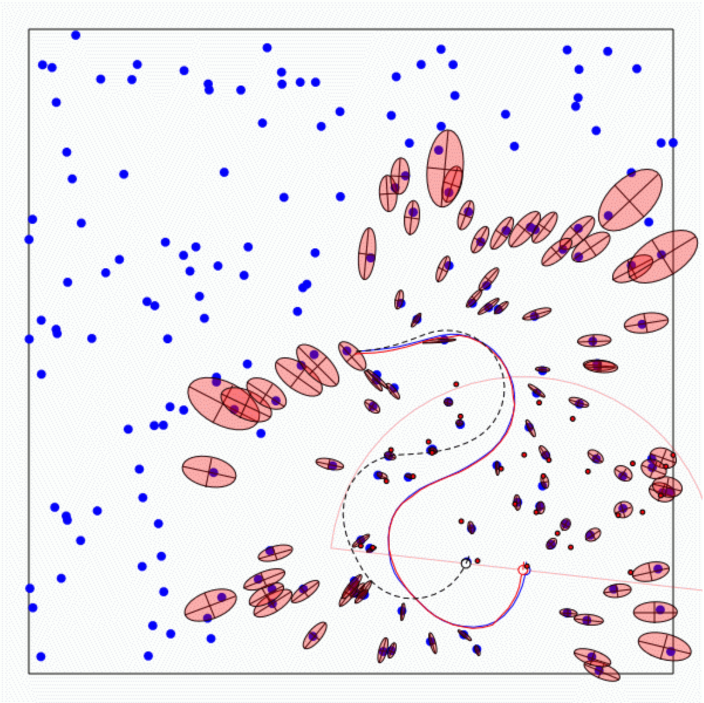
Given:

- The robot's controls

$$U_{1:k} = \{u_1, u_2, \ldots, u_k\}$$

- Relative observations

$$Z_{1:k} = \{z_1, z_2, \ldots, z_k\}$$



Wanted:

- Map of features

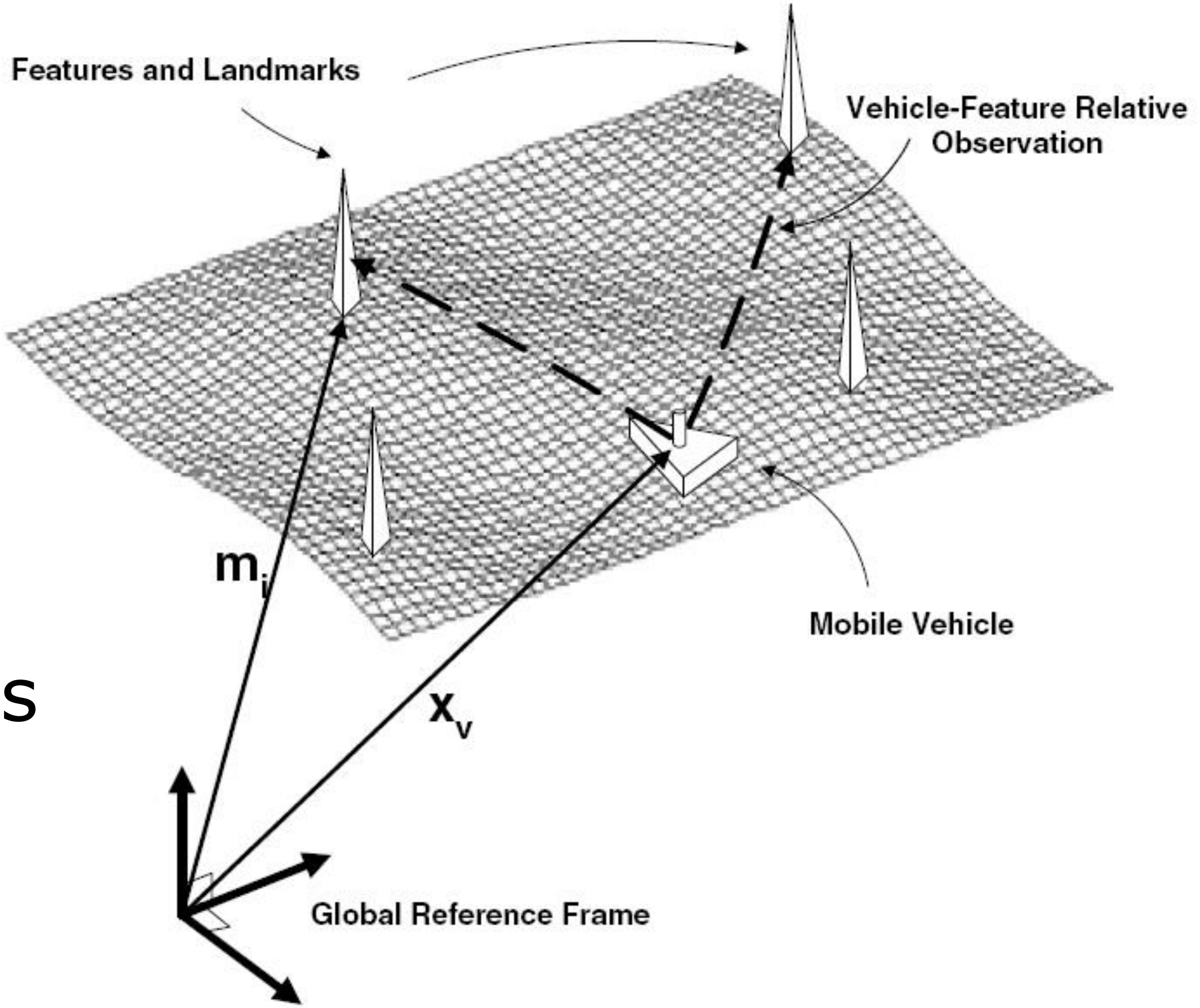
$$m = \{m_1, m_2, \ldots, m_n\}$$

- Path of the robot

$$X_{1:k} = \{x_1, x_2, \ldots, x_k\}$$

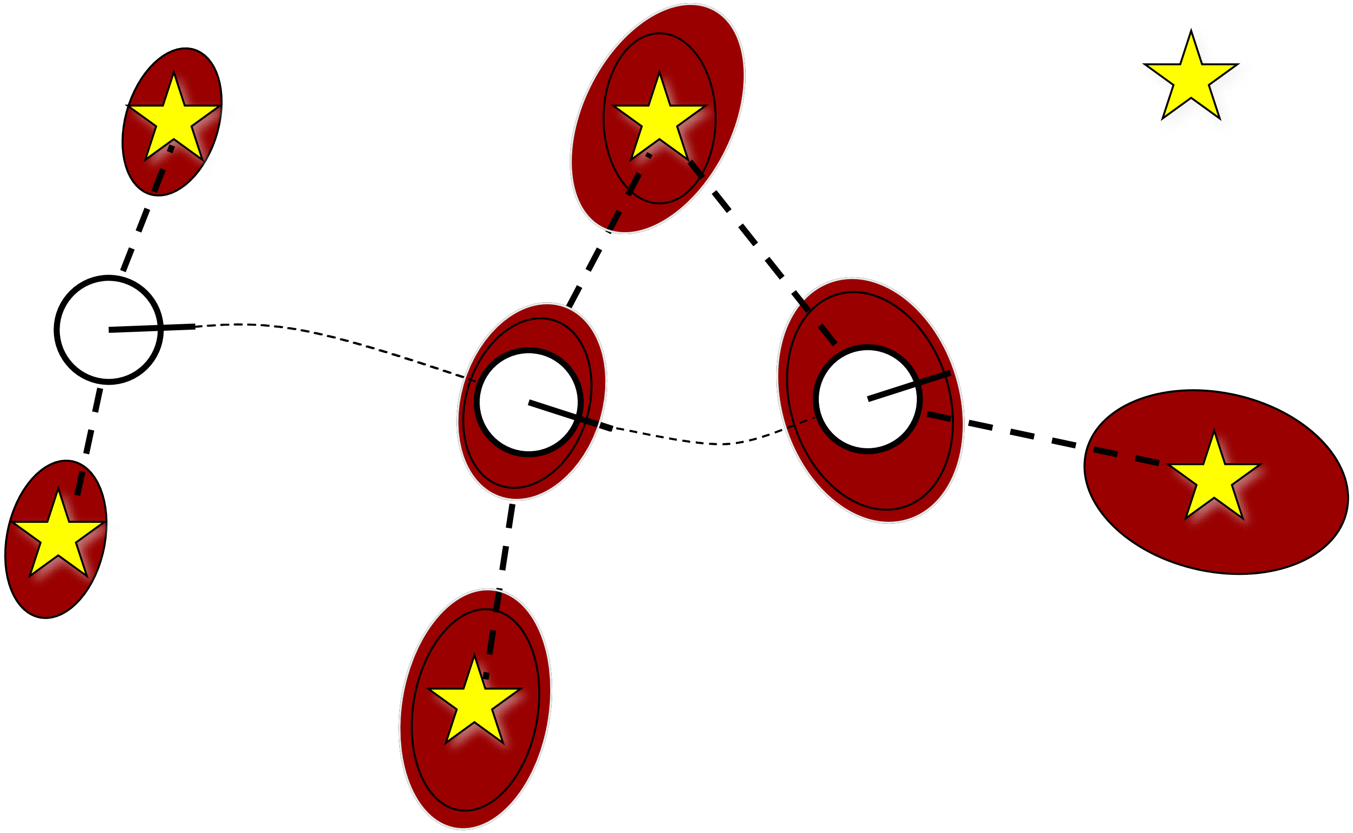
Feature-Based SLAM

- **Absolute** robot poses
- **Absolute** landmark positions
- But only **relative** measurements of landmarks



Why is SLAM a hard problem?

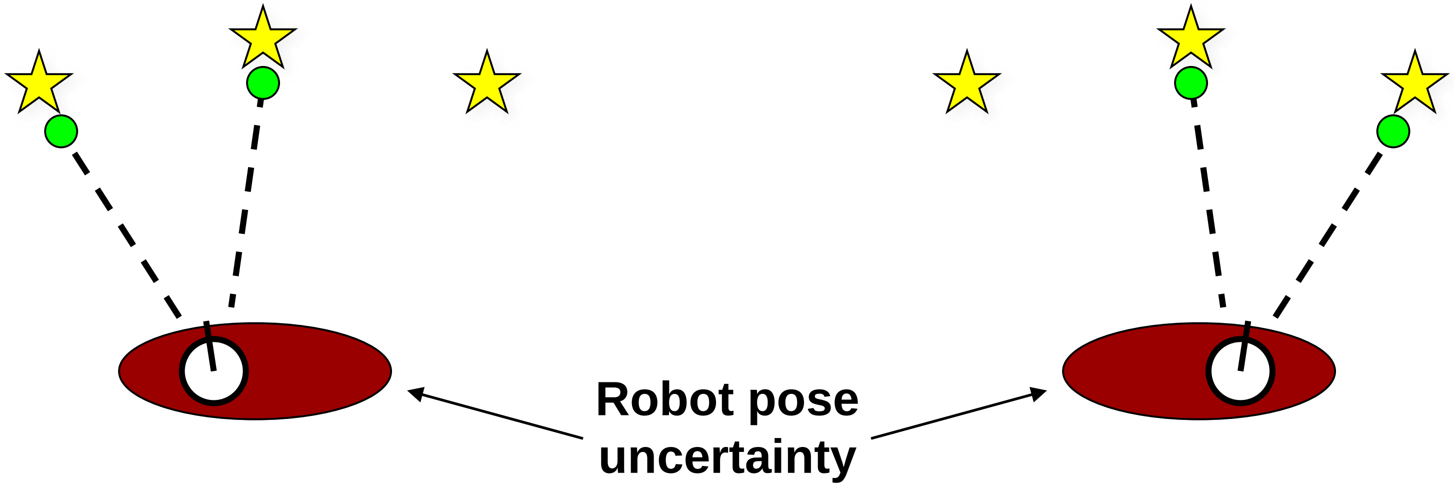
1. Robot path and map are both **unknown**



2. Errors in map and pose estimates correlated

Why is SLAM a hard problem?

- The **mapping between observations and landmarks is unknown**
- Picking **wrong** data associations can have **catastrophic** consequences (divergence)



SLAM: Simultaneous Localization And Mapping

- Full SLAM:

$$p(x_{0:t}, m \mid z_{1:t}, u_{1:t})$$

Estimates entire path and map!

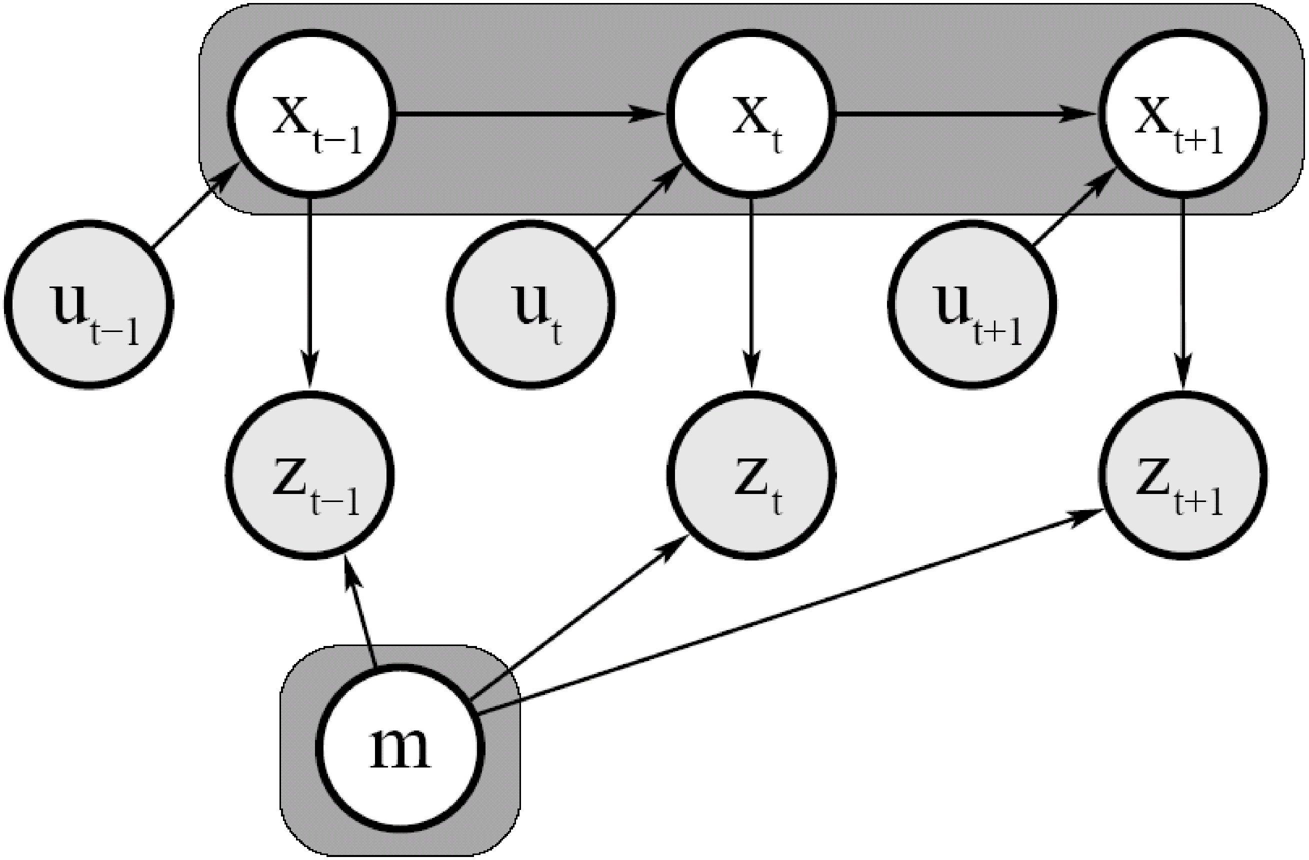
- Online SLAM:

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

Estimates most recent pose and map!

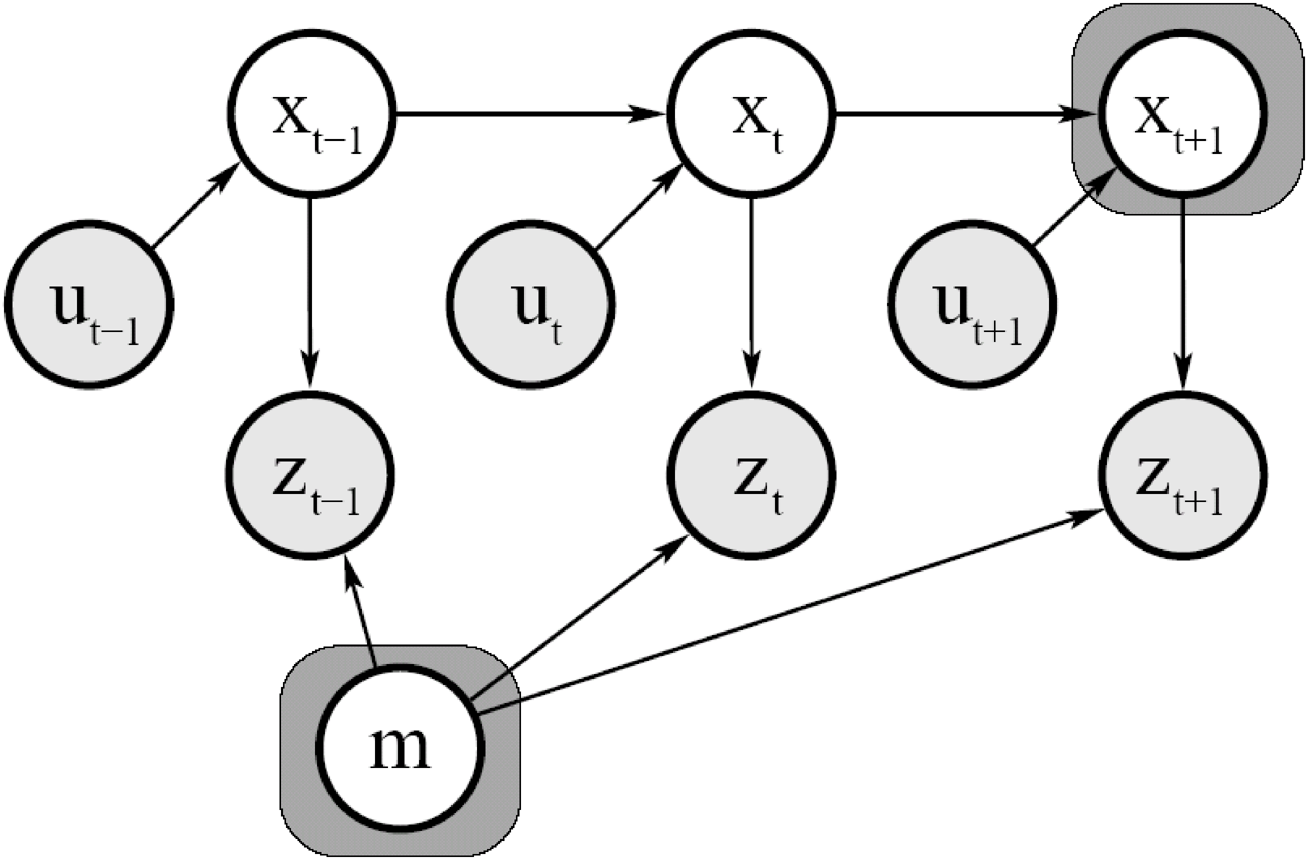
- Integrations (marginalization) typically done recursively, one at a time

Graphical Model of Full SLAM



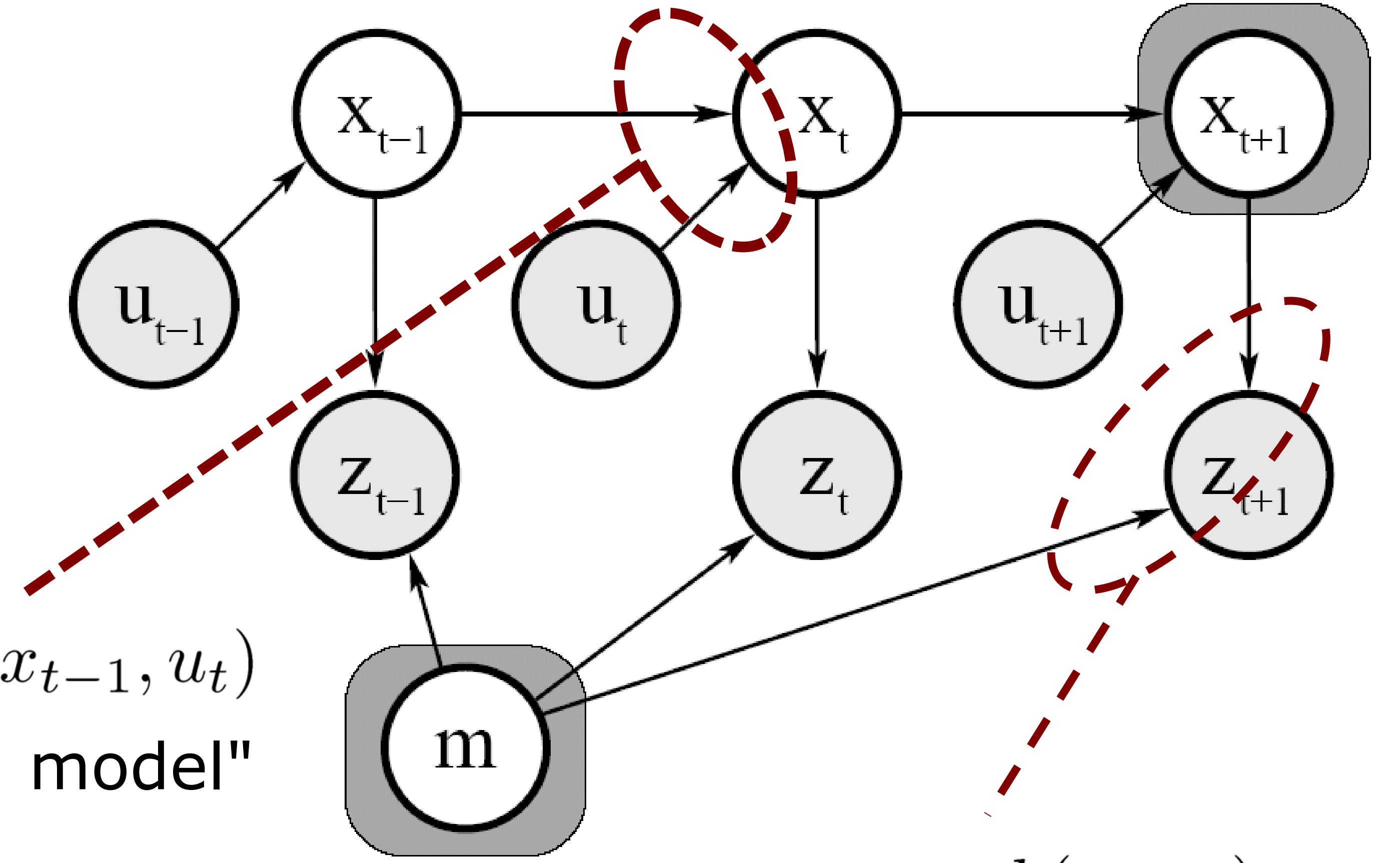
$$p(x_{1:t+1}, m \mid z_{1:t+1}, u_{1:t+1})$$

Graphical Model of Online SLAM



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

Motion and Observation Model



$$x_t = f(x_{t-1}, u_t)$$

"Motion model"

$$z_t = h(x_t, m)$$

"Observation model"

Remember the KF Algorithm

```
Algorithm **Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):
Prediction:
    $\bar{m}_t = A_t m_{t-1} + B_t u_t$
    $\bar{S}_t = A_t S_{t-1} A_t^T + R_t$

Correction:
    $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
    $m_t = \bar{m}_t + K_t(z_t - C_t \bar{m}_t)$
    $S_t = (I - K_t C_t) \bar{S}_t$
Return $\mu_t, \Sigma_t$
```

EKF SLAM: State representation

- **Localization**

  3x1 pose vector

  3x3 cov. matrix

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \qquad \Sigma_k = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 & \sigma_{x\theta}^2 \\ \sigma_{yx}^2 & \sigma_y^2 & \sigma_{y\theta}^2 \\ \sigma_{\theta x}^2 & \sigma_{\theta y}^2 & \sigma_\theta^2 \end{bmatrix}$$

- **SLAM**

  Landmarks **simply extend** the state.

  Growing state vector and covariance matrix!

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_R \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_n \end{bmatrix} \qquad \Sigma_k = \begin{bmatrix} \Sigma_R & \Sigma_{RM_1} & \Sigma_{RM_2} & \cdots & \Sigma_{RM_n} \\ \Sigma_{M_1R} & \Sigma_{M_1} & \Sigma_{M_1M_2} & \cdots & \Sigma_{M_1M_n} \\ \Sigma_{M_2R} & \Sigma_{M_2M_1} & \Sigma_{M_2} & \cdots & \Sigma_{M_2M_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Sigma_{M_nR} & \Sigma_{M_nM_1} & \Sigma_{M_nM_2} & \cdots & \Sigma_{M_n} \end{bmatrix}$$

EKF SLAM: State representation

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

$$\underbrace{\begin{bmatrix} x \\ y \\ \theta \\ m_{1,x} \\ m_{1,y} \\ \vdots \\ m_{n,x} \\ m_{n,y} \end{bmatrix}}_{\mu} \quad \underbrace{\begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xm_{1,x}} & \sigma_{xm_{1,y}} & \cdots & \sigma_{xm_{n,x}} & \sigma_{xm_{n,y}} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{y\theta} & \sigma_{ym_{1,x}} & \sigma_{ym_{1,y}} & \cdots & \sigma_{m_{n,x}} & \sigma_{m_{n,y}} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta\theta} & \sigma_{\theta m_{1,x}} & \sigma_{\theta m_{1,y}} & \cdots & \sigma_{\theta m_{n,x}} & \sigma_{\theta m_{n,y}} \\ \sigma_{m_{1,x}x} & \sigma_{m_{1,x}y} & \sigma_{\theta} & \sigma_{m_{1,x}m_{1,x}} & \sigma_{m_{1,x}m_{1,y}} & \cdots & \sigma_{m_{1,x}m_{n,x}} & \sigma_{m_{1,x}m_{n,y}} \\ \sigma_{m_{1,y}x} & \sigma_{m_{1,y}y} & \sigma_{\theta} & \sigma_{m_{1,y}m_{1,x}} & \sigma_{m_{1,y}m_{1,y}} & \cdots & \sigma_{m_{1,y}m_{n,x}} & \sigma_{m_{1,y}m_{n,y}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_{m_{n,x}x} & \sigma_{m_{n,x}y} & \sigma_{\theta} & \sigma_{m_{n,x}m_{1,x}} & \sigma_{m_{n,x}m_{1,y}} & \cdots & \sigma_{m_{n,x}m_{n,x}} & \sigma_{m_{n,x}m_{n,y}} \\ \sigma_{m_{n,y}x} & \sigma_{m_{n,y}y} & \sigma_{\theta} & \sigma_{m_{n,y}m_{1,x}} & \sigma_{m_{n,y}m_{1,y}} & \cdots & \sigma_{m_{n,y}m_{n,x}} & \sigma_{m_{n,y}m_{n,y}} \end{bmatrix}}_{\Sigma}$$

- Can handle hundreds of dimensions

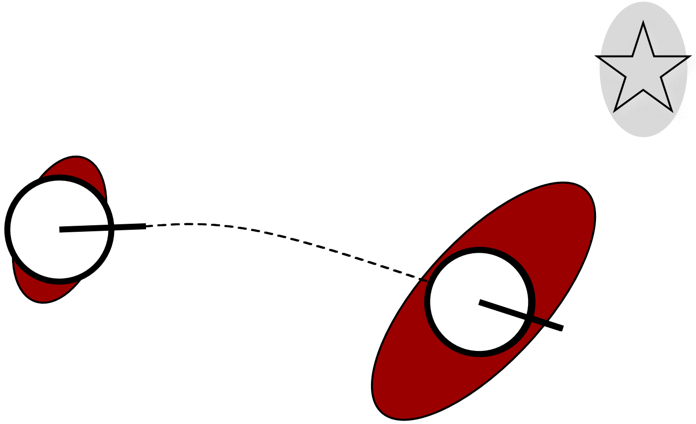
EKF SLAM: Filter Cycle

1. State prediction (odometry)
2. Measurement prediction
3. Measurement
4. Data association
5. Update
6. Integration of new landmarks

EKF SLAM: Filter Cycle

1. State prediction (odometry)
2. Measurement prediction
3. Measurement
4. Data association
5. Update
6. Integration of new landmarks

EKF SLAM: State Prediction



Odometry:

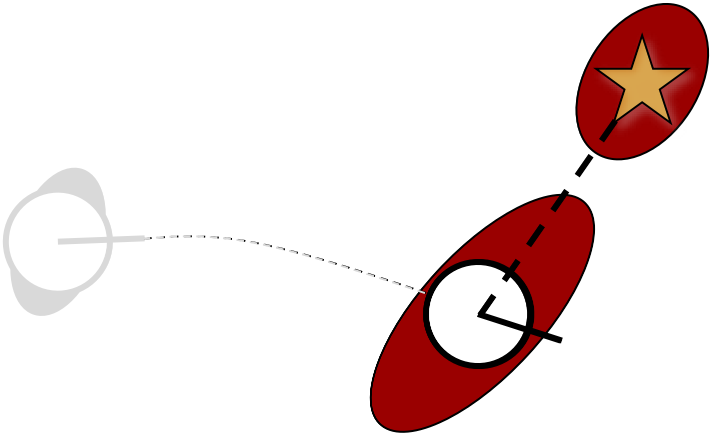
$$\hat{\mathbf{x}}_R = f(\mathbf{x}_R, \mathbf{u})$$
$$\hat{\Sigma}_R = F_x \Sigma_R F_x^T + F_u U F_u^T$$

Robot-landmark cross-covariance prediction:

$$\hat{\Sigma}_{RM_i} = F_x \Sigma_{RM_i}$$

$$\underbrace{\begin{bmatrix} \mathbf{x}_R \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_n \end{bmatrix}}_{\mu} \quad \underbrace{\begin{bmatrix} \Sigma_R & \Sigma_{RM_1} & \cdots & \Sigma_{RM_n} \\ \Sigma_{M_1R} & \Sigma_{M_1} & \cdots & \Sigma_{M_1M_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{M_nR} & \Sigma_{M_nM_1} & \cdots & \Sigma_{M_n} \end{bmatrix}}_{\Sigma}$$

EKF SLAM: Measurement Prediction



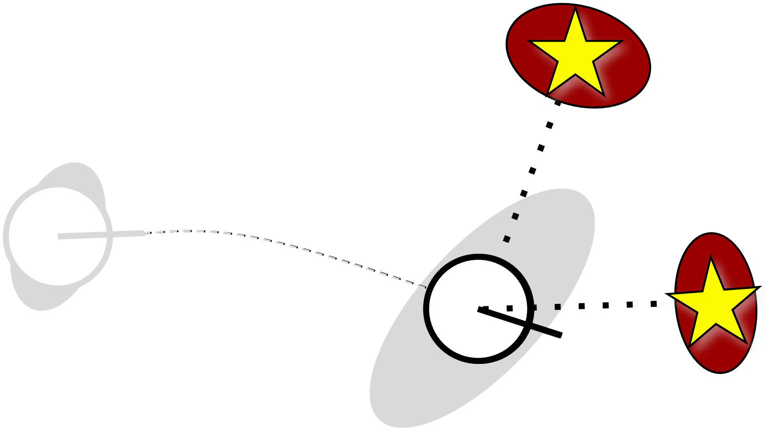
Global-to-local frame transform h

$$\hat{\mathbf{z}}_k = h(\hat{\mathbf{x}}_k)$$

$$\underbrace{\begin{bmatrix} \mathbf{x}_R \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_n \end{bmatrix}}_{\mu} \quad \underbrace{\begin{bmatrix} \Sigma_R & \Sigma_{RM_1} & \cdots & \Sigma_{RM_n} \\ \Sigma_{M_1R} & \Sigma_{M_1} & \cdots & \Sigma_{M_1M_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{M_nR} & \Sigma_{M_nM_1} & \cdots & \Sigma_{M_n} \end{bmatrix}}_{\Sigma}$$

EKF SLAM: Obtained Measurement

(x,y)-point landmarks

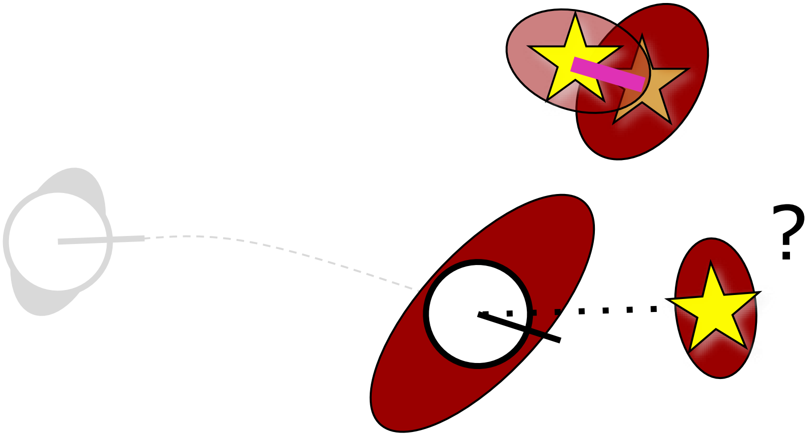


$$\mathbf{z}_k = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}$$

$$R_k = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} \mathbf{x}_R \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_n \end{bmatrix}}_{\mu} \quad \underbrace{\begin{bmatrix} \Sigma_R & \Sigma_{RM_1} & \cdots & \Sigma_{RM_n} \\ \Sigma_{M_1R} & \Sigma_{M_1} & \cdots & \Sigma_{M_1M_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{M_nR} & \Sigma_{M_nM_1} & \cdots & \Sigma_{M_n} \end{bmatrix}}_{\Sigma}$$

EKF SLAM: Data Association

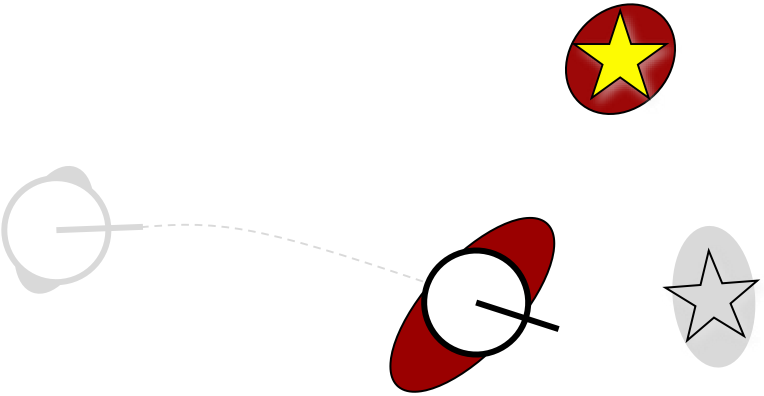


Associates predicted measurements $\hat{\mathbf{z}}_k^i$ with observation $\mathbf{z}_k^j$

$$\begin{aligned} \nu_k^{ij} &= \mathbf{z}_k^j - \hat{\mathbf{z}}_k^i \\ S_k^{ij} &= R_k^j + H^i \, \hat{\Sigma}_k \, H^{i\,T} \end{aligned}$$

$$\underbrace{\begin{bmatrix} \mathbf{x}_R \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_n \end{bmatrix}}_{\mu} \quad \underbrace{\begin{bmatrix} \Sigma_R & \Sigma_{RM_1} & \cdots & \Sigma_{RM_n} \\ \Sigma_{M_1 R} & \Sigma_{M_1} & \cdots & \Sigma_{M_1 M_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{M_n R} & \Sigma_{M_n M_1} & \cdots & \Sigma_{M_n} \end{bmatrix}}_{\Sigma}$$

EKF SLAM: Update Step



The usual Kalman filter expressions

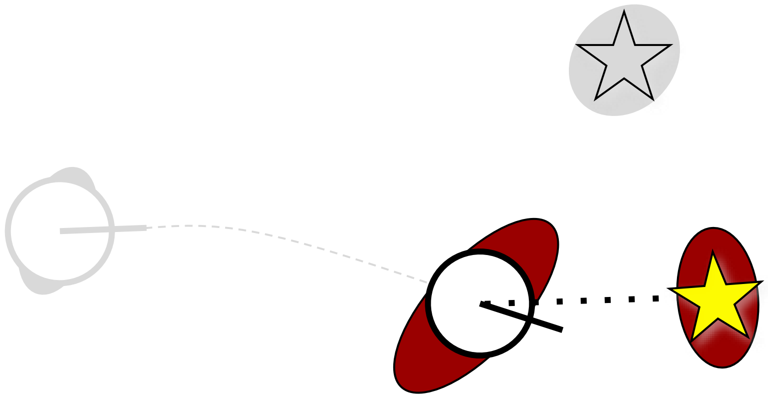
$$K_k = \hat{\Sigma}_k H^T S_k^{-1}$$

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + K_k \nu_k$$

$$C_k = (I - K_k H) \hat{\Sigma}_k$$

$$\underbrace{\begin{bmatrix} \mathbf{x}_R \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_n \end{bmatrix}}_{\mu} \quad \underbrace{\begin{bmatrix} \Sigma_R & \Sigma_{RM_1} & \cdots & \Sigma_{RM_n} \\ \Sigma_{M_1 R} & \Sigma_{M_1} & \cdots & \Sigma_{M_1 M_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{M_n R} & \Sigma_{M_n M_1} & \cdots & \Sigma_{M_n} \end{bmatrix}}_{\Sigma}$$

EKF SLAM: New Landmarks



State augmented by

$$\mathbf{m}_{n+1} = g(\mathbf{x}_R, \mathbf{z}_j)$$

$$\Sigma_{M_{n+1}} = G_R \Sigma_R G_R^T + G_z R_j G_z^T$$

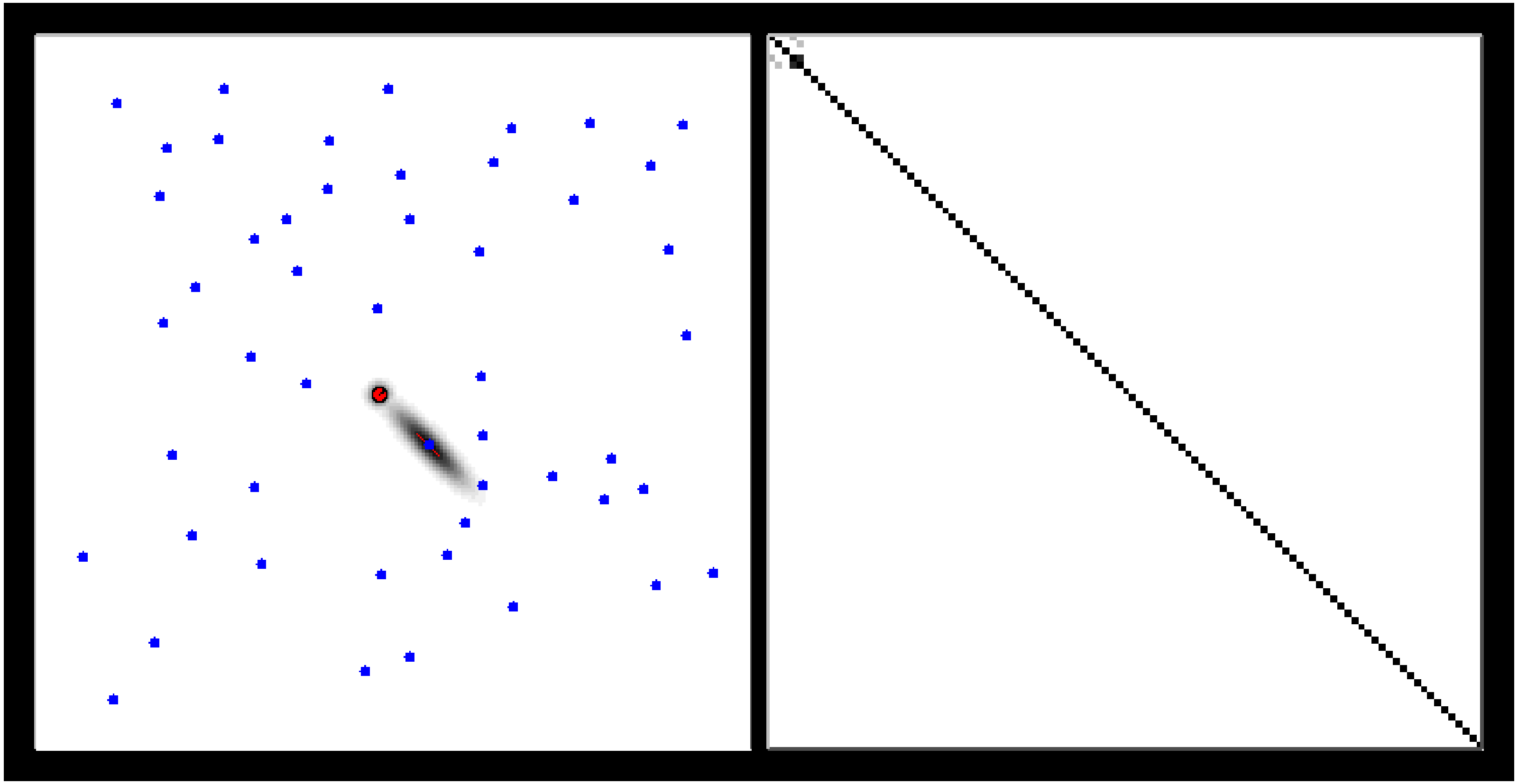
Cross-covariances:

$$\Sigma_{M_{n+1} M_i} = G_R \Sigma_{R M_i}$$

$$\Sigma_{M_{n+1} R} = G_R \Sigma_R$$

$$\underbrace{\begin{bmatrix} \mathbf{x}_R \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_n \\ \mathbf{m}_{n+1} \end{bmatrix}}_{\mu} \qquad \underbrace{\begin{bmatrix} \Sigma_R & \Sigma_{RM_1} & \cdots & \Sigma_{RM_n} & \Sigma_{RM_{n+1}} \\ \Sigma_{M_1 R} & \Sigma_{M_1} & \cdots & \Sigma_{M_1 M_n} & \Sigma_{M_1 M_{n+1}} \\ \vdots & \vdots & \ddots & \vdots & \\ \Sigma_{M_n R} & \Sigma_{M_n M_1} & \cdots & \Sigma_{M_n} & \Sigma_{M_n M_{n+1}} \\ \Sigma_{M_{n+1} R} & \Sigma_{M_{n+1} M_1} & \cdots & \Sigma_{M_{n+1} M_n} & \Sigma_{M_{n+1}} \end{bmatrix}}_{\Sigma}$$

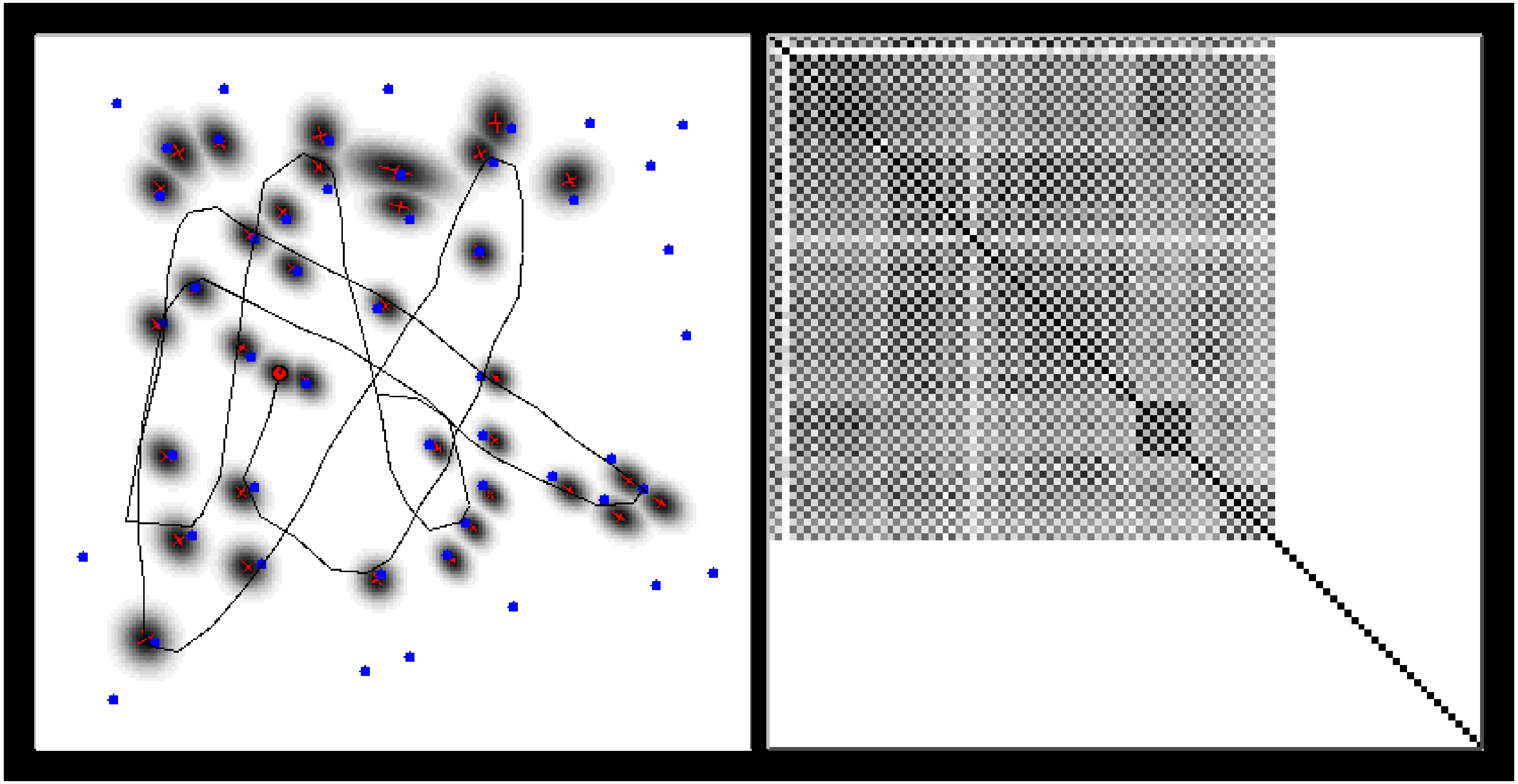
EKF SLAM



Map

Correlation matrix

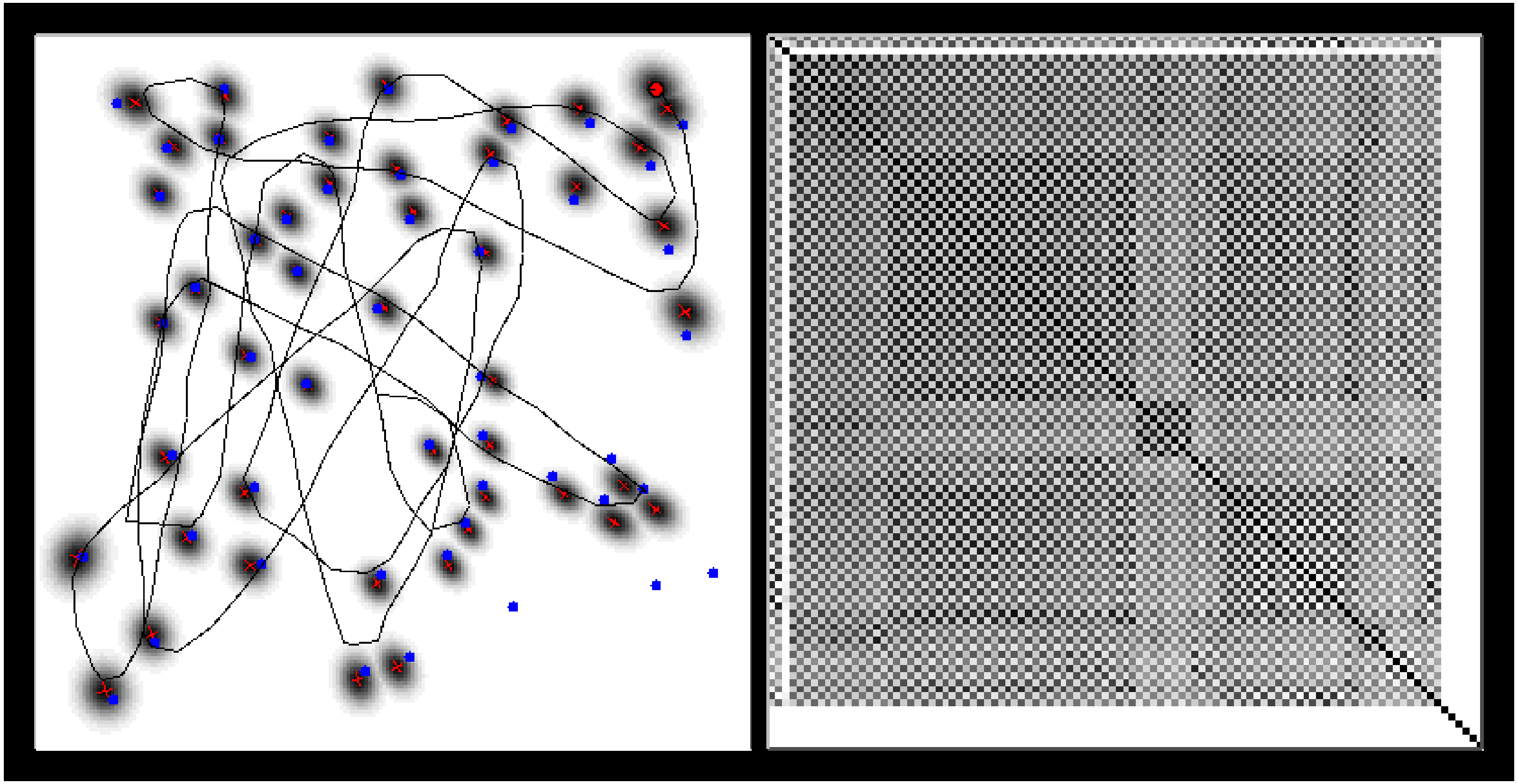
EKF SLAM



Map

Correlation matrix

EKF SLAM



Map Correlation matrix

EKF SLAM: Correlations Matter

- What if we neglected cross-correlations?

$$\Sigma_k = \begin{bmatrix} \Sigma_R & 0 & \cdots & 0 \\ 0 & \Sigma_{M_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{M_n} \end{bmatrix}$$

$$\Sigma_{RM_i} = \mathbf{0}_{3\times 2}$$

$$\Sigma_{M_i M_{i+1}} = \mathbf{0}_{2\times 2}$$

EKF SLAM: Correlations Matter

- What if we neglected cross-correlations?

$$\Sigma_k = \begin{bmatrix} \Sigma_R & 0 & \cdots & 0 \\ 0 & \Sigma_{M_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{M_n} \end{bmatrix}$$

$$\Sigma_{RM_i} = \mathbf{0}_{3\times 2}$$

$$\Sigma_{M_i M_{i+1}} = \mathbf{0}_{2\times 2}$$

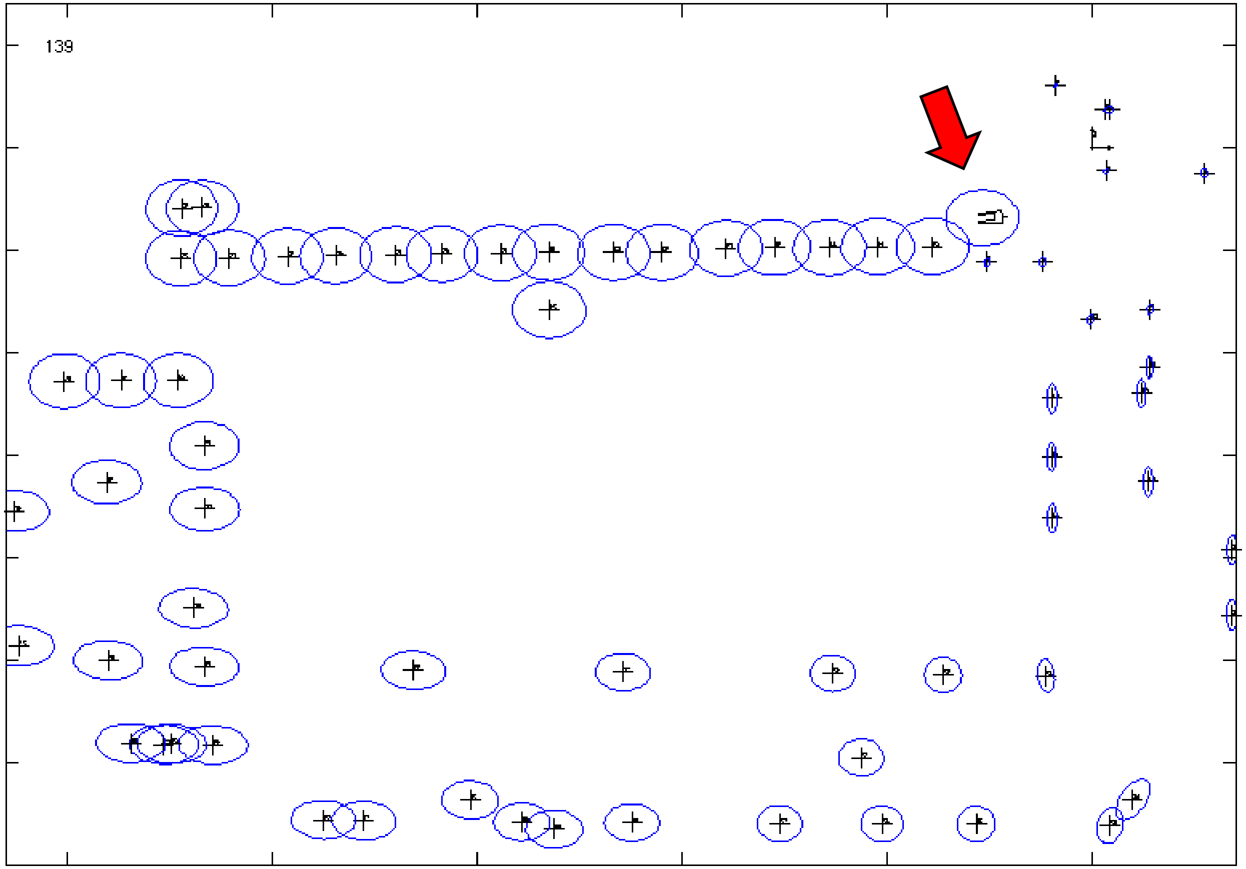
- Landmark and robot uncertainties would become overly optimistic
- Data association would fail
- Multiple map entries of the same landmark
- Inconsistent map

SLAM: Loop Closure

- **Recognizing an already mapped area**, typically after a long exploration path (the robot "closes a loop")
- Structurally identical to data association, but
  - high levels of ambiguity
  - possibly useless validation gates
  - environment symmetries
- Uncertainties **collapse** after a loop closure (whether the closure was correct or not)

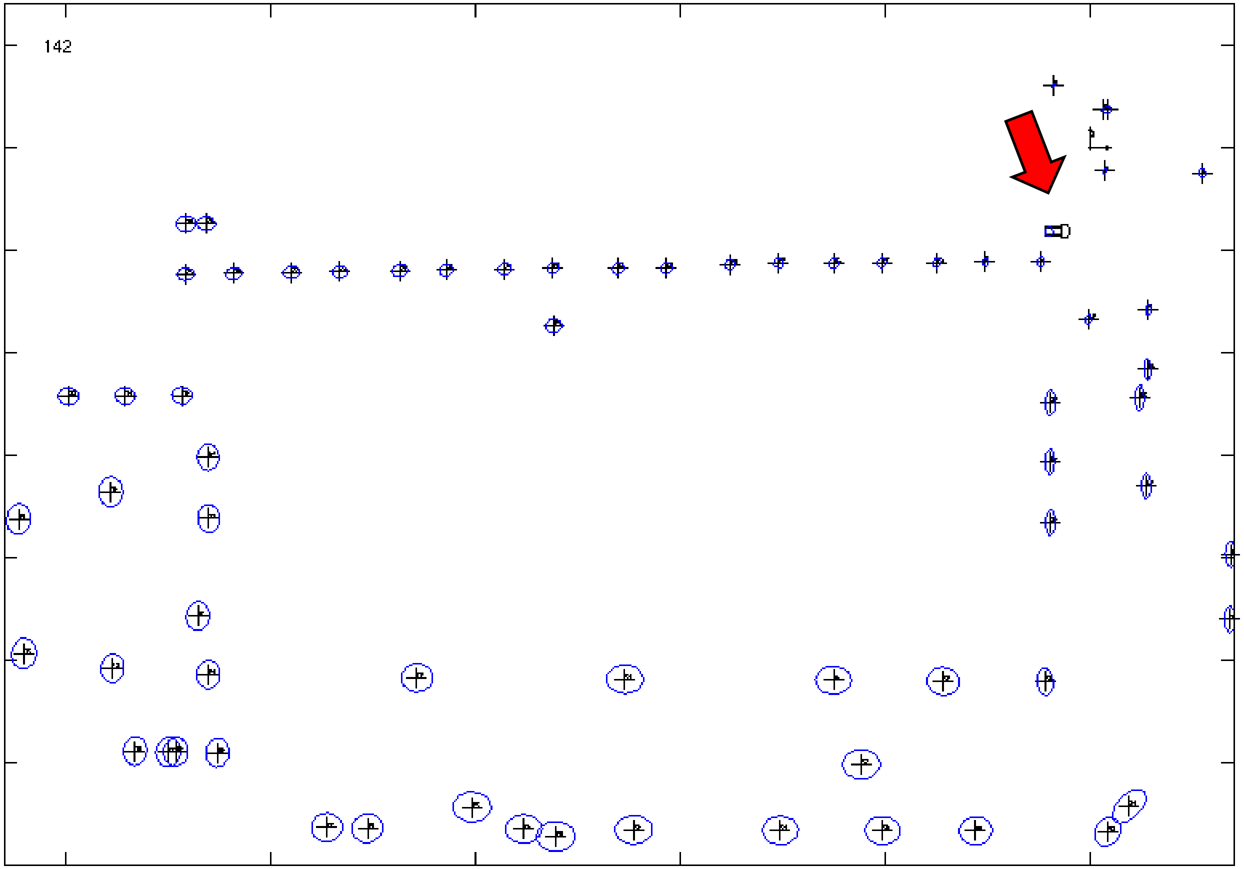
SLAM: Loop Closure

- Before loop closure



SLAM: Loop Closure

- After loop closure



SLAM: Loop Closure

- By revisiting already mapped areas, uncertainties in robot and landmark estimates can be **reduced**

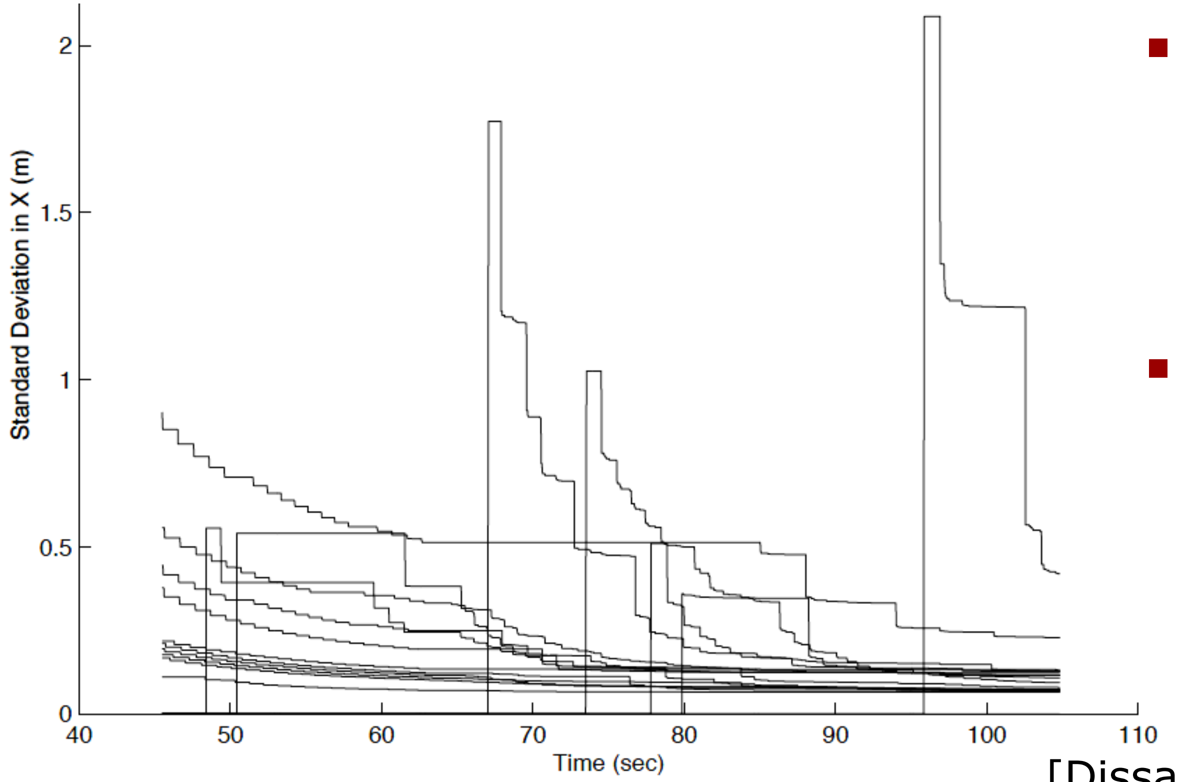
- This can be exploited when **exploring** an environment for the sake of better (e.g. more accurate) maps

- Exploration: the problem of ***where* to acquire new information**

→ See separate chapter on exploration

KF-SLAM Properties (Linear Case)

- The **determinant** of any sub-matrix of the map covariance matrix **decreases monotonically** as successive observations are made



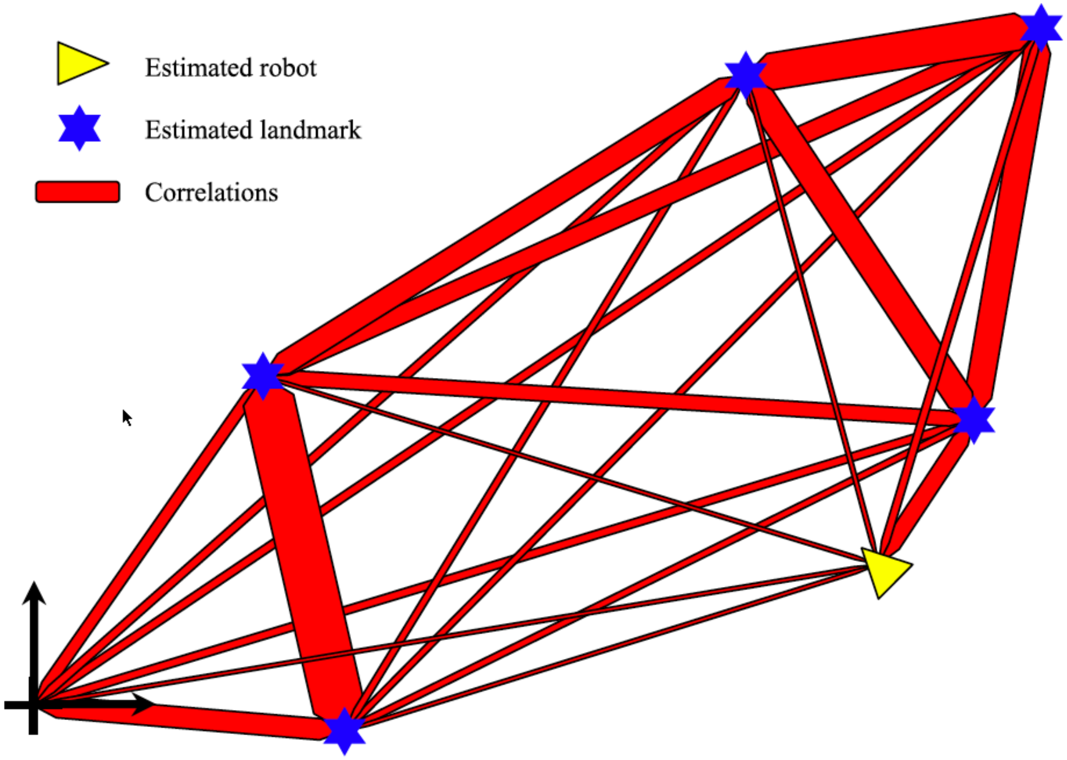
- When a new landmark is initialized, its **uncertainty** is **maximal**

- Landmark uncertainty **decreases monotonically** with each new observation

[Dissanayake et al., 2001]

KF-SLAM Properties (Linear Case)

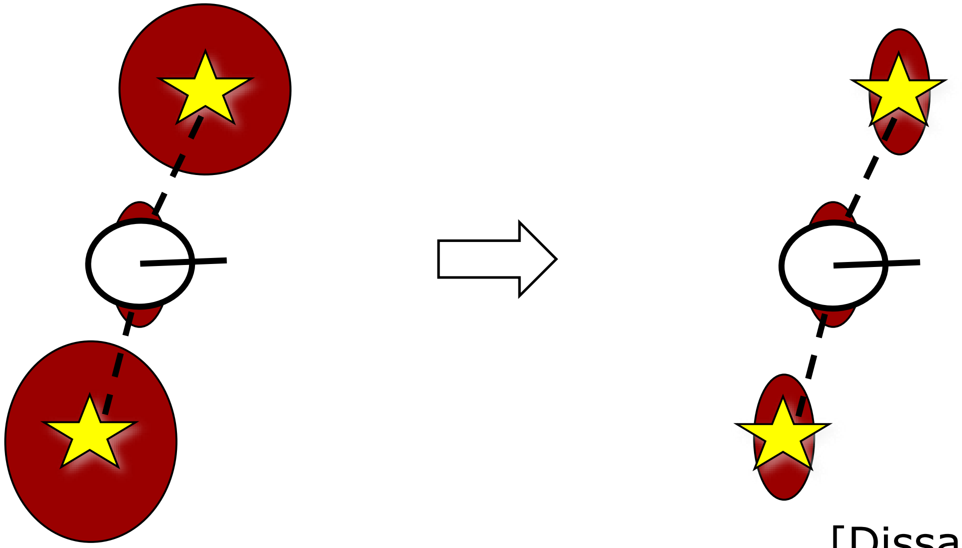
- In the limit, the landmark estimates become **fully correlated**



[Dissanayake et al., 2001]

KF-SLAM Properties (Linear Case)

- In the limit, the **covariance** associated with any single landmark location estimate is determined only by the **initial covariance in the vehicle location estimate**.



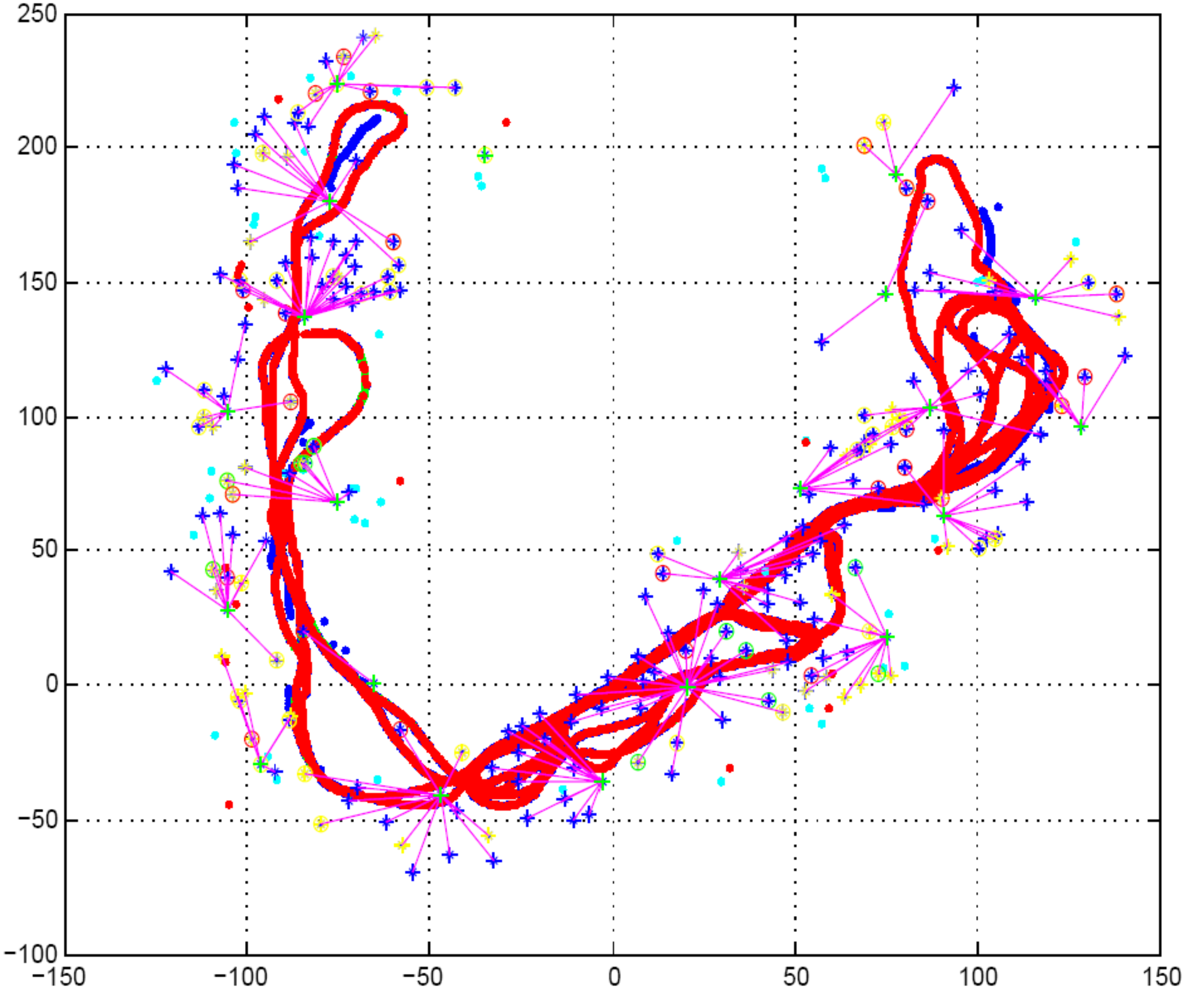
[Dissanayake et al., 2001]

EKF SLAM Example: Victoria Park Dataset

Victoria Park: Data Acquisition

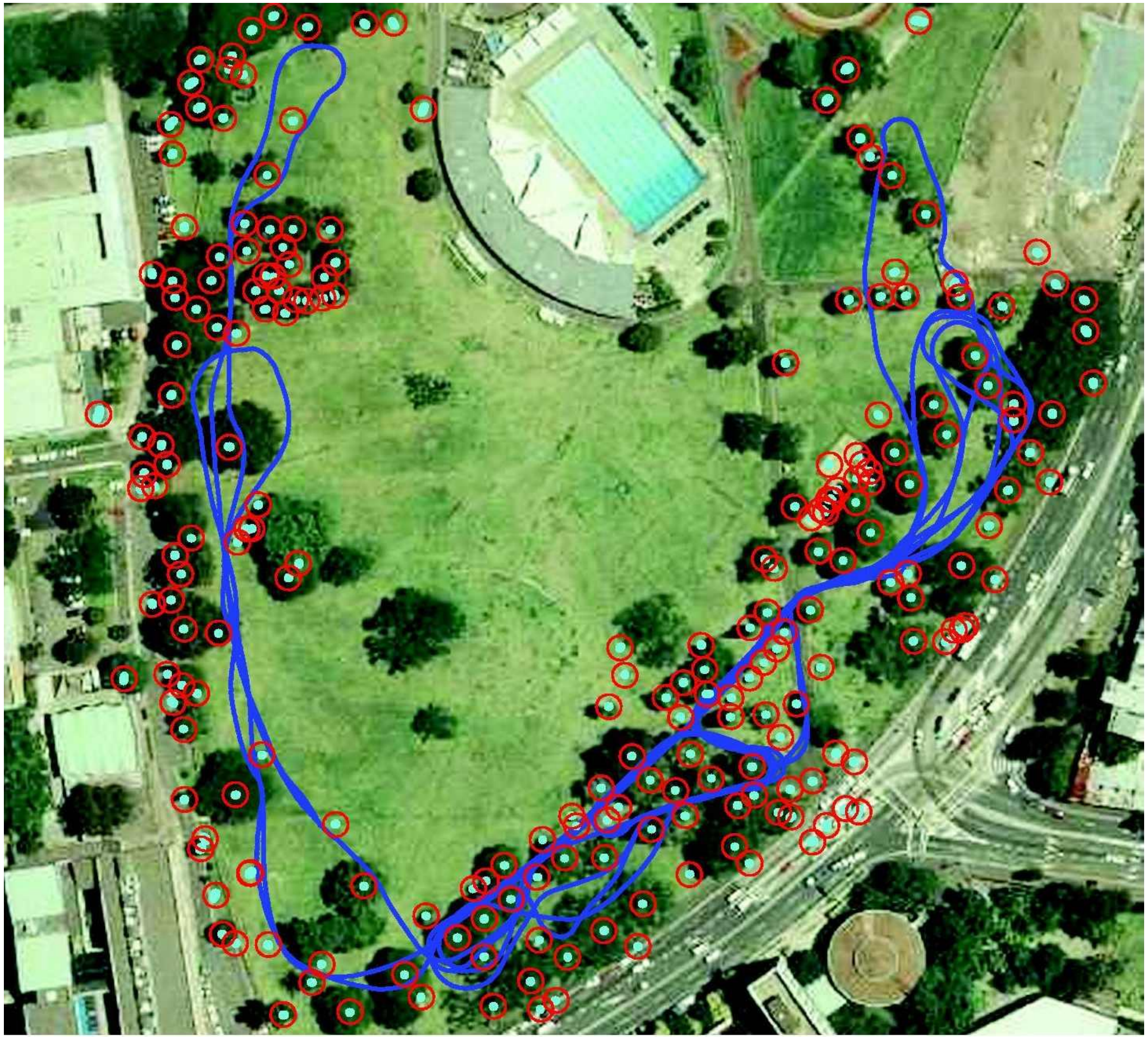
[courtesy by E. Nebot]

Victoria Park: Estimated Trajectory



[courtesy by E. Nebot]

Victoria Park: Landmarks

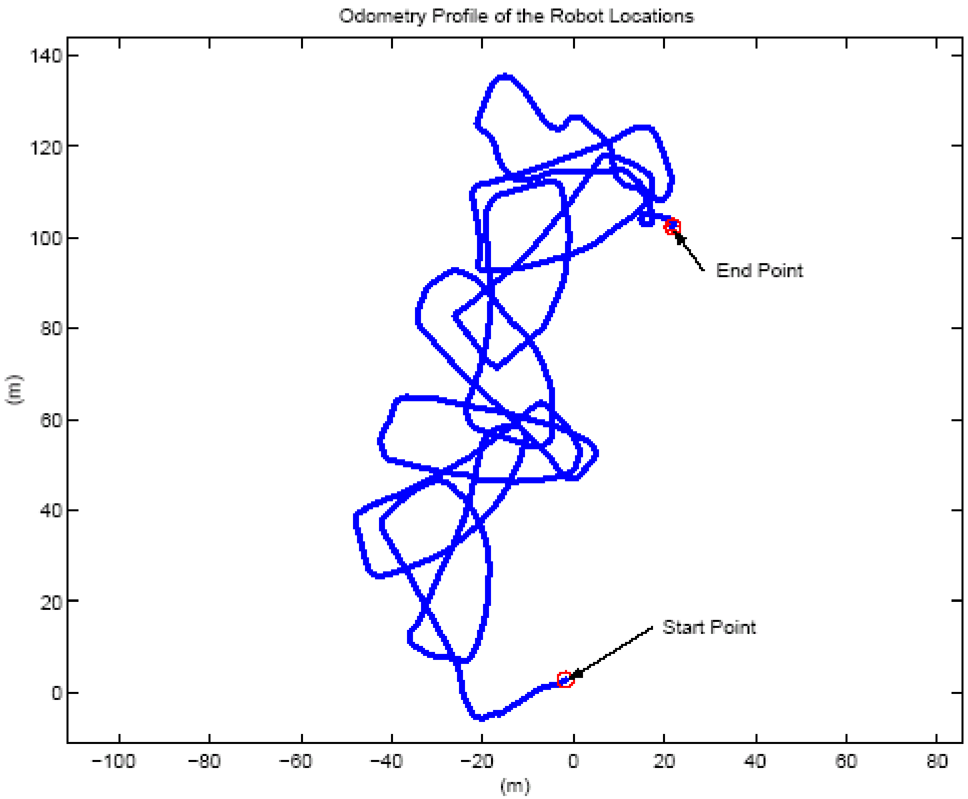


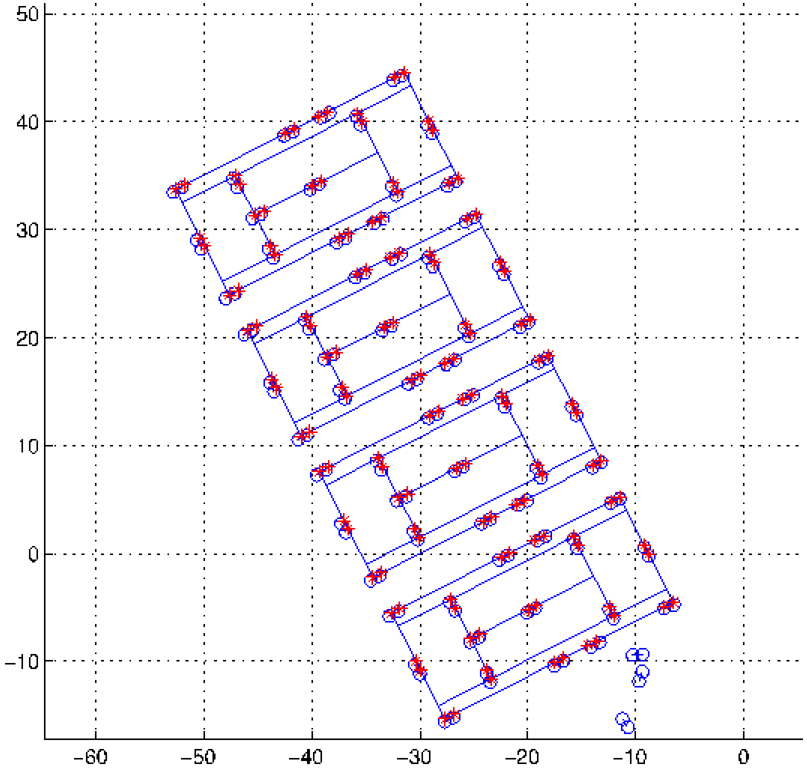
[courtesy by E. Nebot]

EKF SLAM Example: Tennis Court

[courtesy by J. Leonard]

EKF SLAM Example: Tennis Court





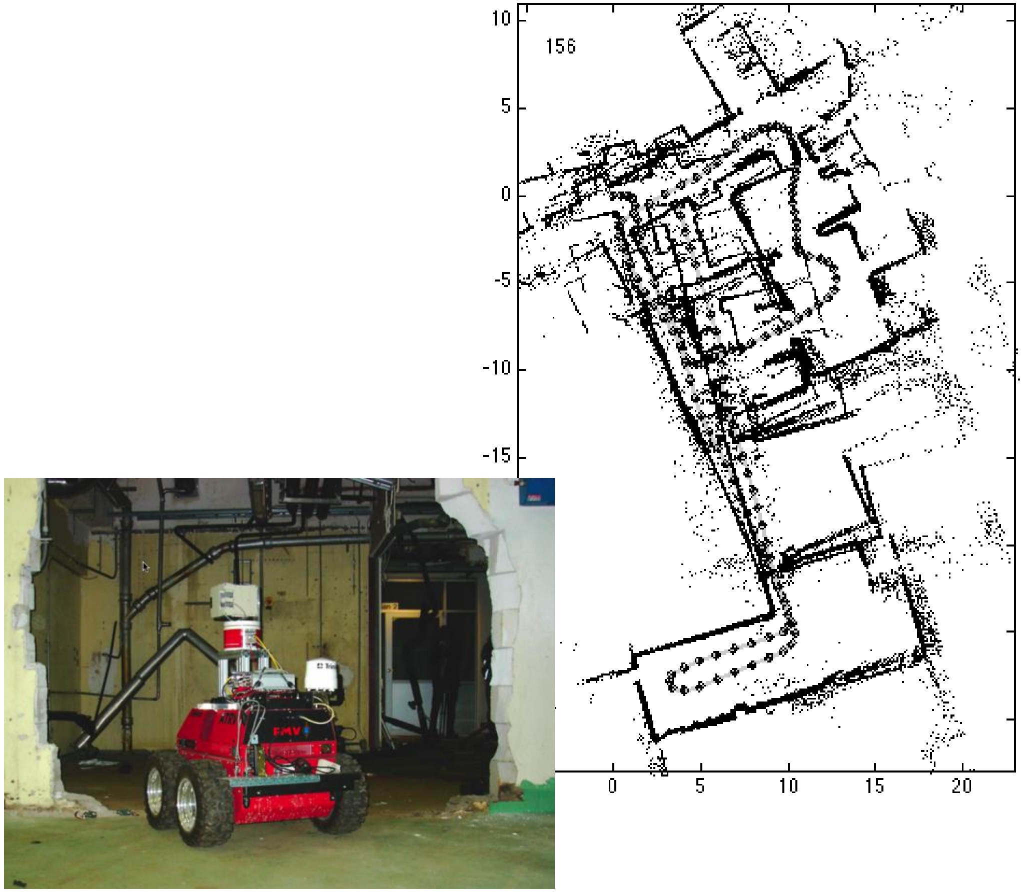
odometry

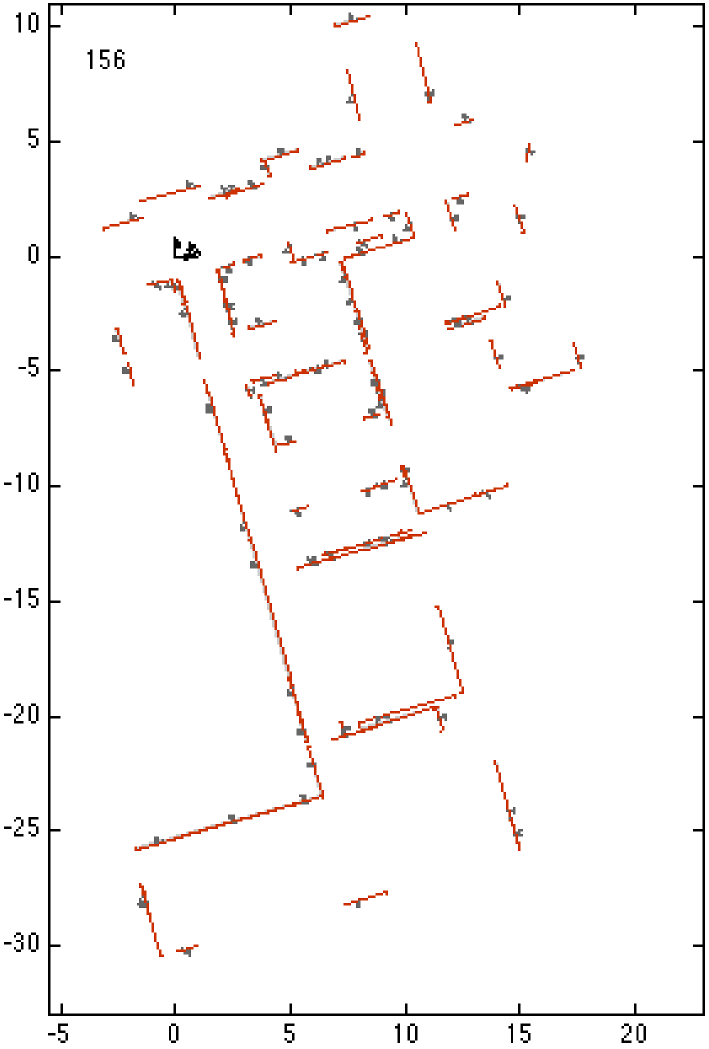
estimated trajectory

[courtesy by John Leonard]

EKF SLAM Example: Line Features

- KTH Bakery Data Set





[Wulf et al., ICRA 04]

EKF-SLAM: Complexity

- Cost per step: quadratic in n, the number of landmarks: $O(n^2)$
- Total cost to build a map with n landmarks: $O(n^3)$
- Memory consumption: $O(n^2)$
- Problem: becomes computationally intractable for large maps!
- There exists variants to circumvent these problems

SLAM Techniques

- EKF SLAM
- FastSLAM
- Graph-based SLAM
- Topological SLAM
  (mainly place recognition)
- Scan Matching / Visual Odometry
  (only locally consistent maps)
- Approximations for SLAM: Local submaps, Sparse extended information filters, Sparse links, Thin junction tree filters, etc.
- …

EKF-SLAM: Summary

- The first SLAM solution
- Convergence proof for linear Gaussian case
- Can diverge if nonlinearities are large (and the real world is nonlinear ...)
- Can deal only with a single mode
- Successful in medium-scale scenes
- Approximations exists to reduce the computational complexity