

Probabilistic Robotics

SLAM and FastSLAM

**(lightly modified version of the
slideset accompanying the
Thrun, Burgard, Fox book)**

The SLAM Problem

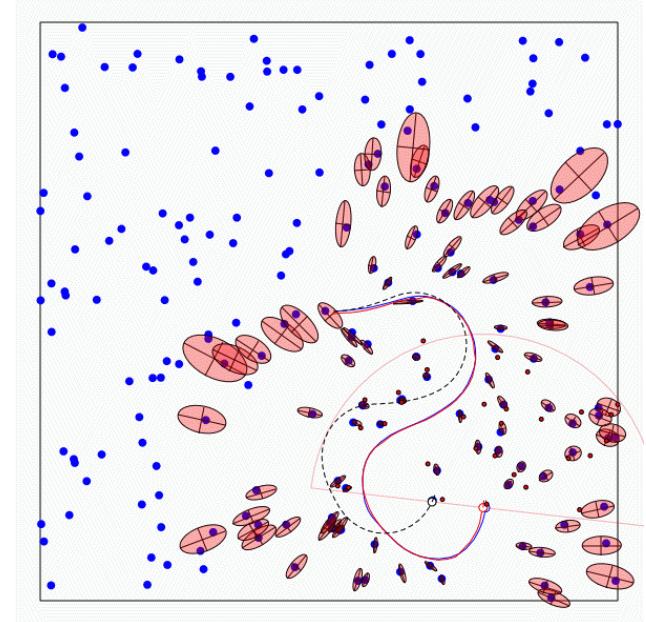
A robot is exploring an unknown, static environment.

Given:

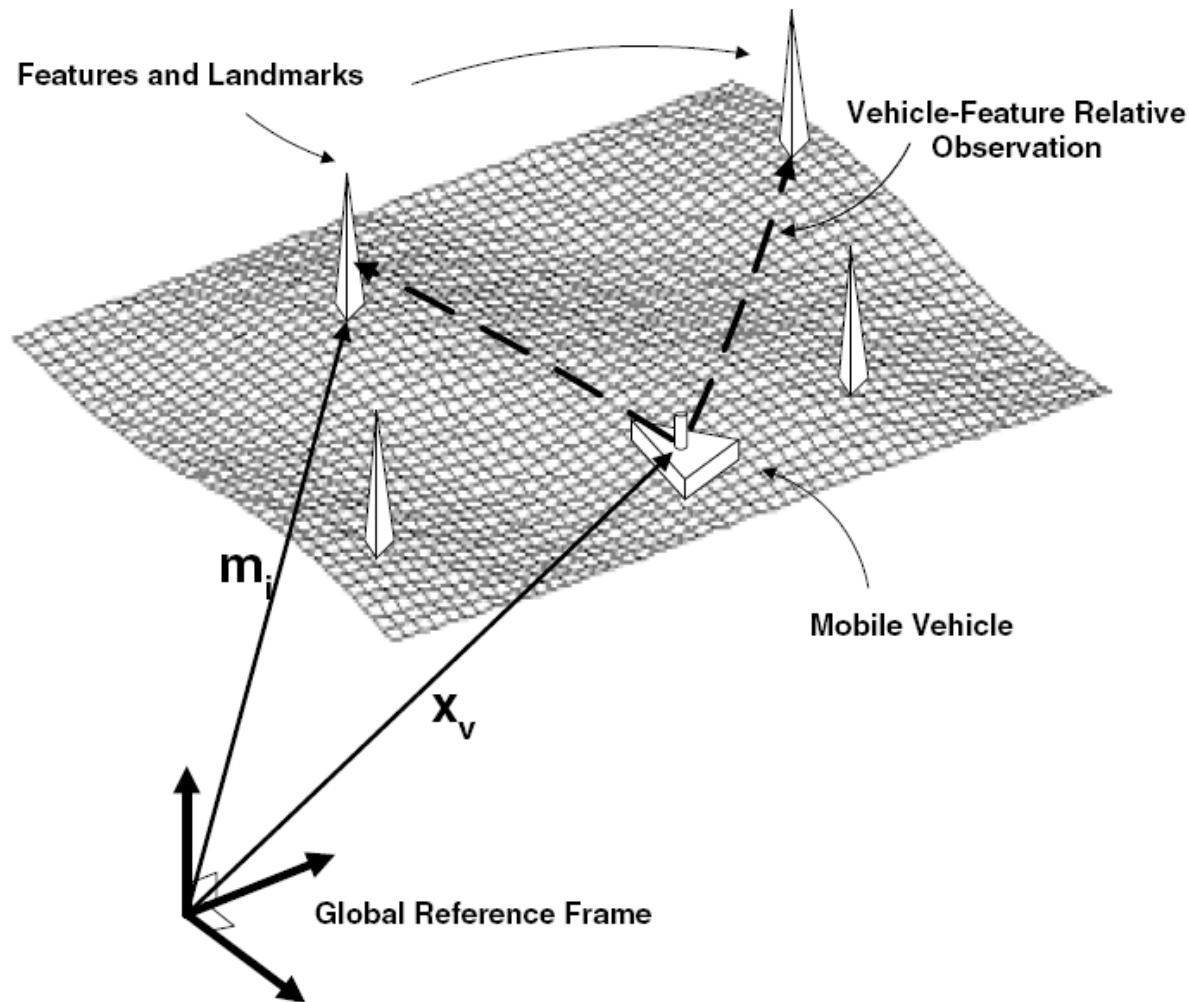
- The robot's controls
- Observations of nearby features

Estimate:

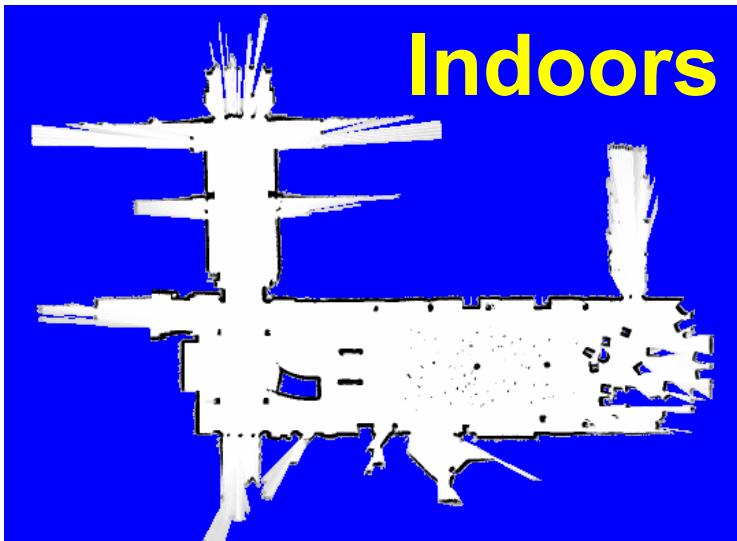
- Map of features
- Path of the robot



Structure of the Landmark-based SLAM-Problem



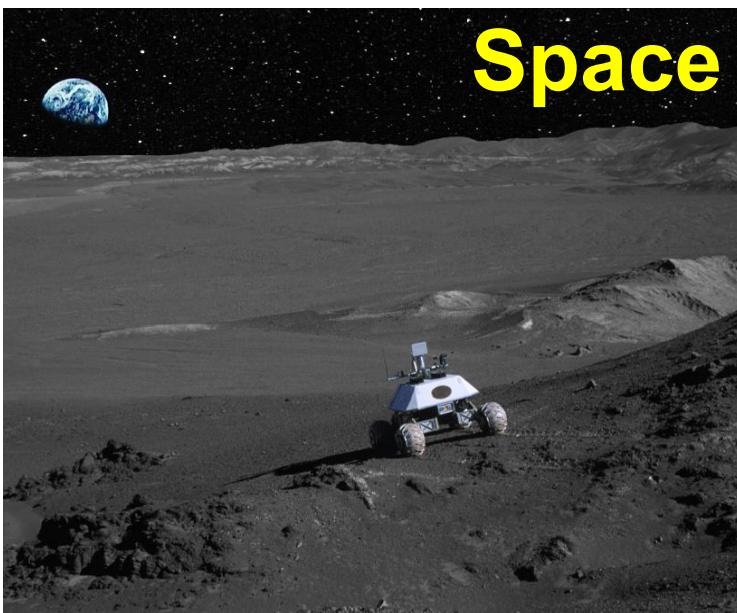
SLAM Applications



Indoors



Undersea



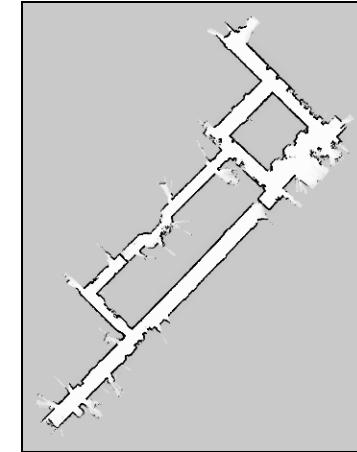
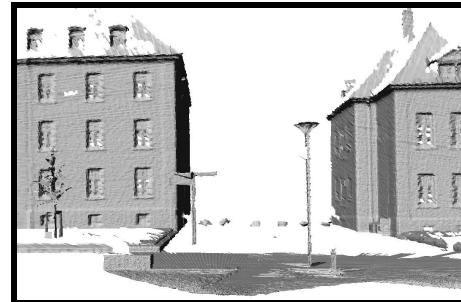
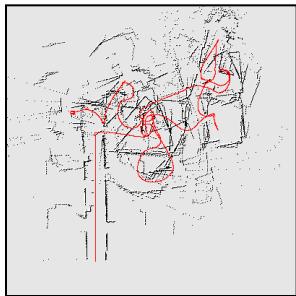
Space



Underground

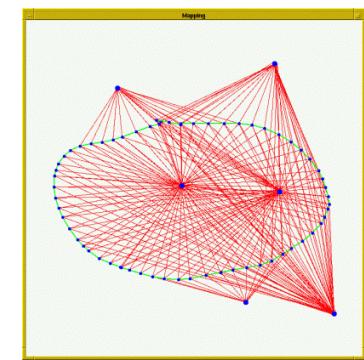
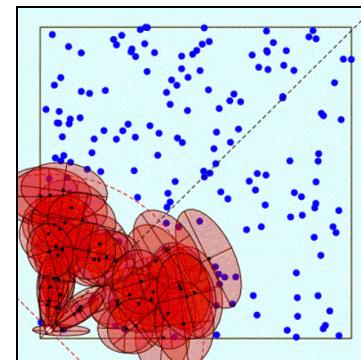
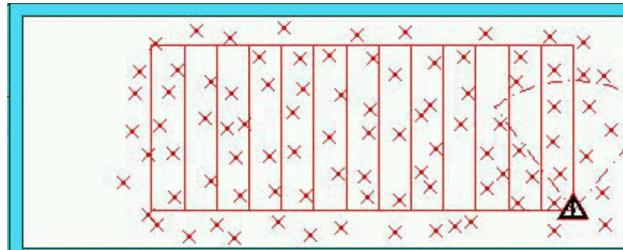
Representations

- Grid maps or scans



[Lu & Milios, 97; Gutmann, 98; Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01;...]

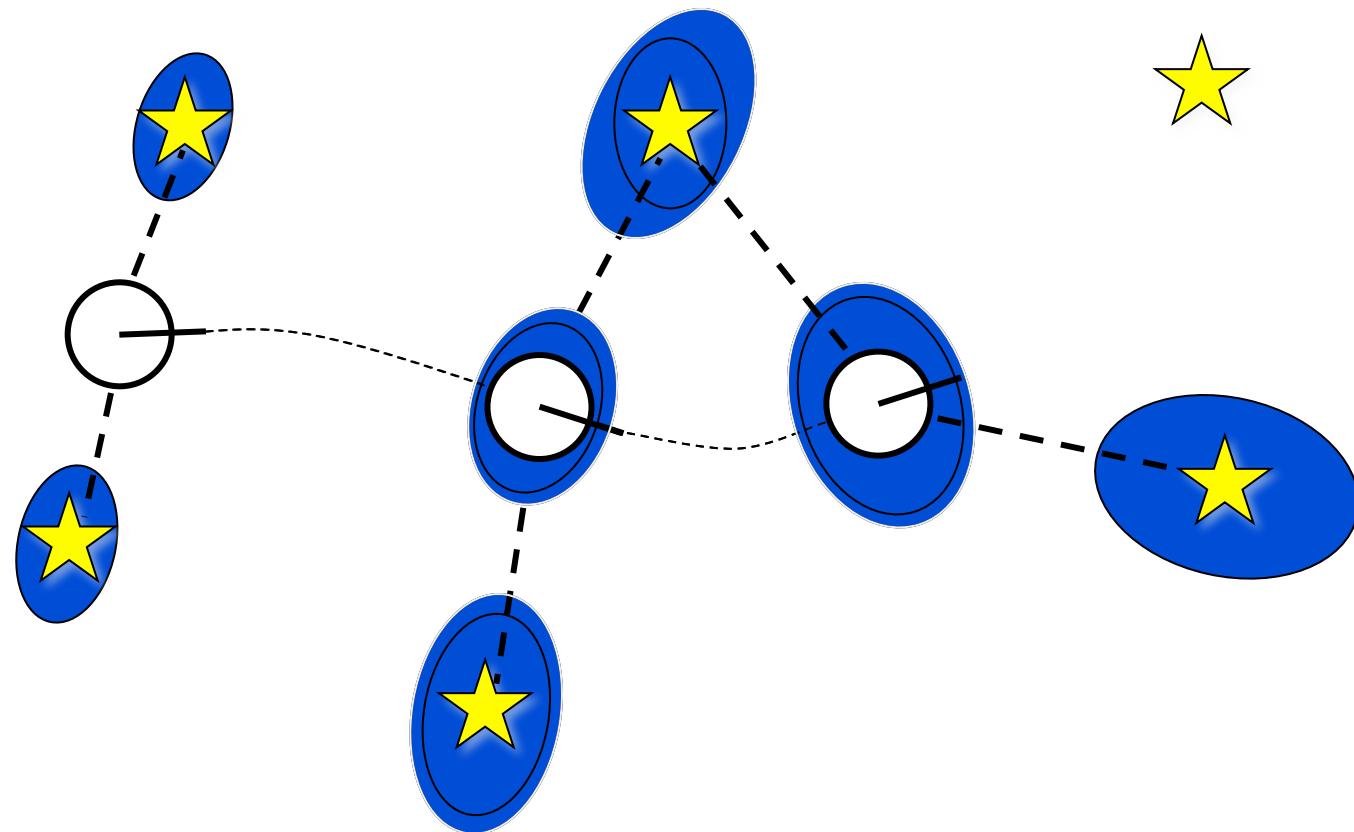
- Landmark-based



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

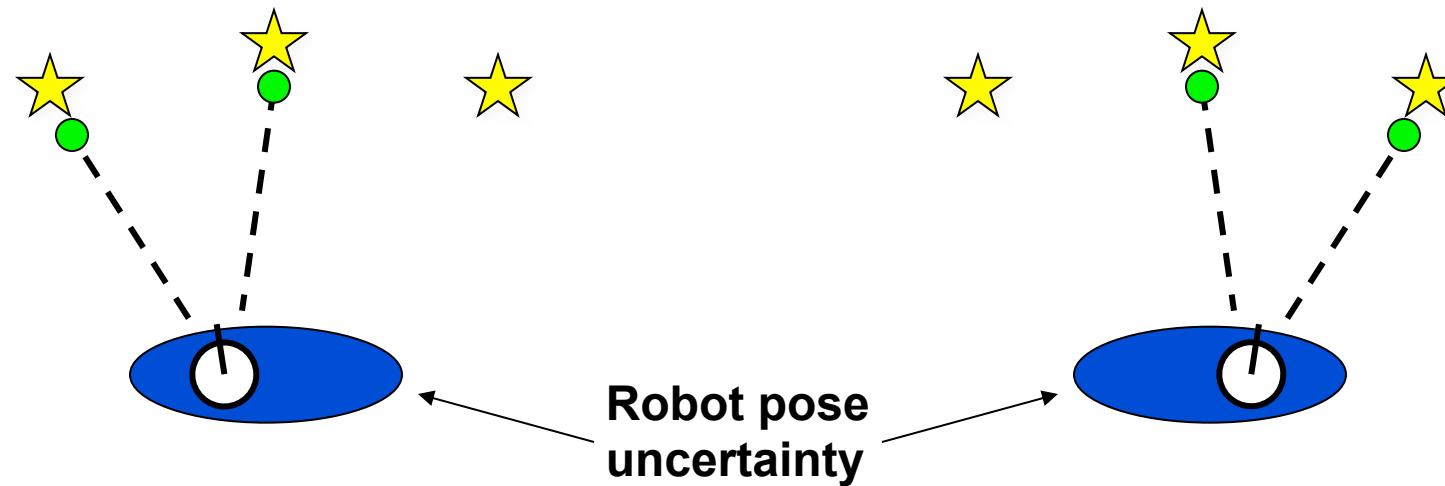
Why is SLAM a hard problem?

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!

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

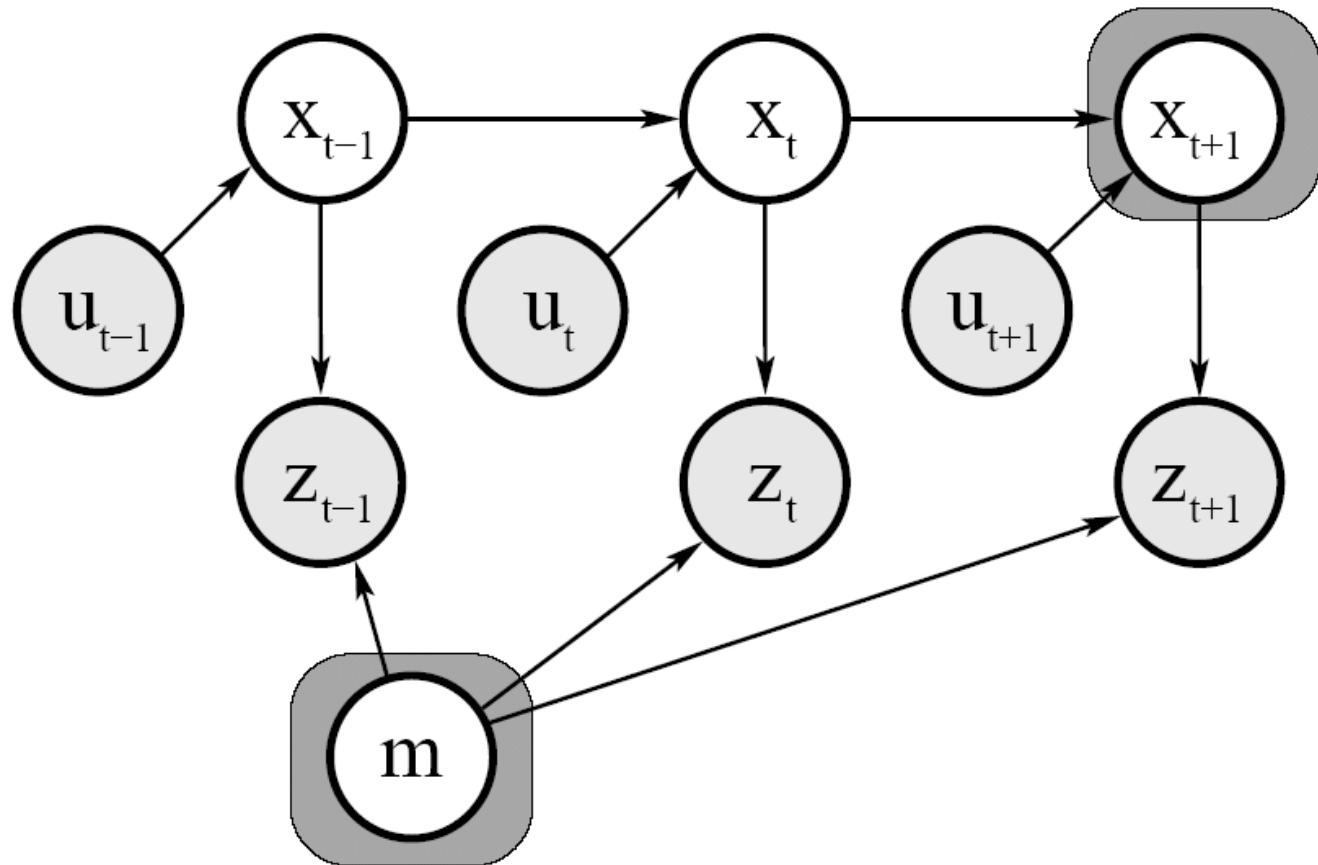
- Online SLAM:

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

Integrations typically done one at a time

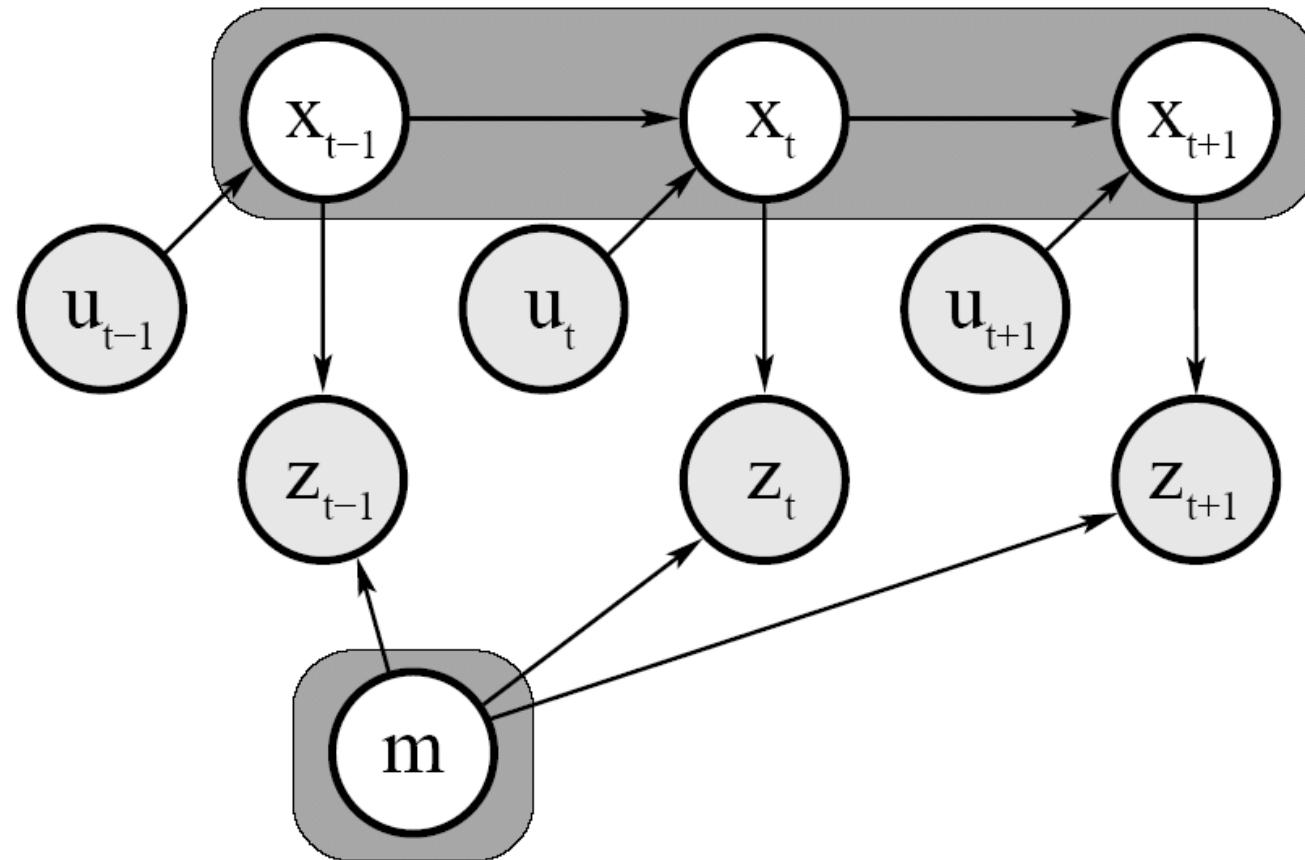
Estimates most recent pose and map!

Graphical Model of Online SLAM:



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

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
- Fast-SLAM
- Probabilistic mapping with a single map and a posterior about poses
 Mapping + Localization
- Graph-SLAM, SEIFs

Scan Matching

Maximize the likelihood of the i-th pose and map relative to the (i-1)-th pose and map.

$$\hat{x}_t = \arg \max_{x_t} \left\{ p(z_t | x_t, \hat{m}^{[t-1]}) \cdot p(x_t | u_{t-1}, \hat{x}_{t-1}) \right\}$$

↑
current measurement ↑
map constructed so far ↑
robot motion

Calculate the map, $\hat{m}^{[t]}$ according to “mapping with known poses” based on the poses and observations.

Kalman Filter Algorithm

1. Algorithm **Kalman_filter**(μ_{t-1} , Σ_{t-1} , U_t , Z_t):
2. Prediction:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$
5. Correction:
6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$
9. Return μ_t , Σ_t

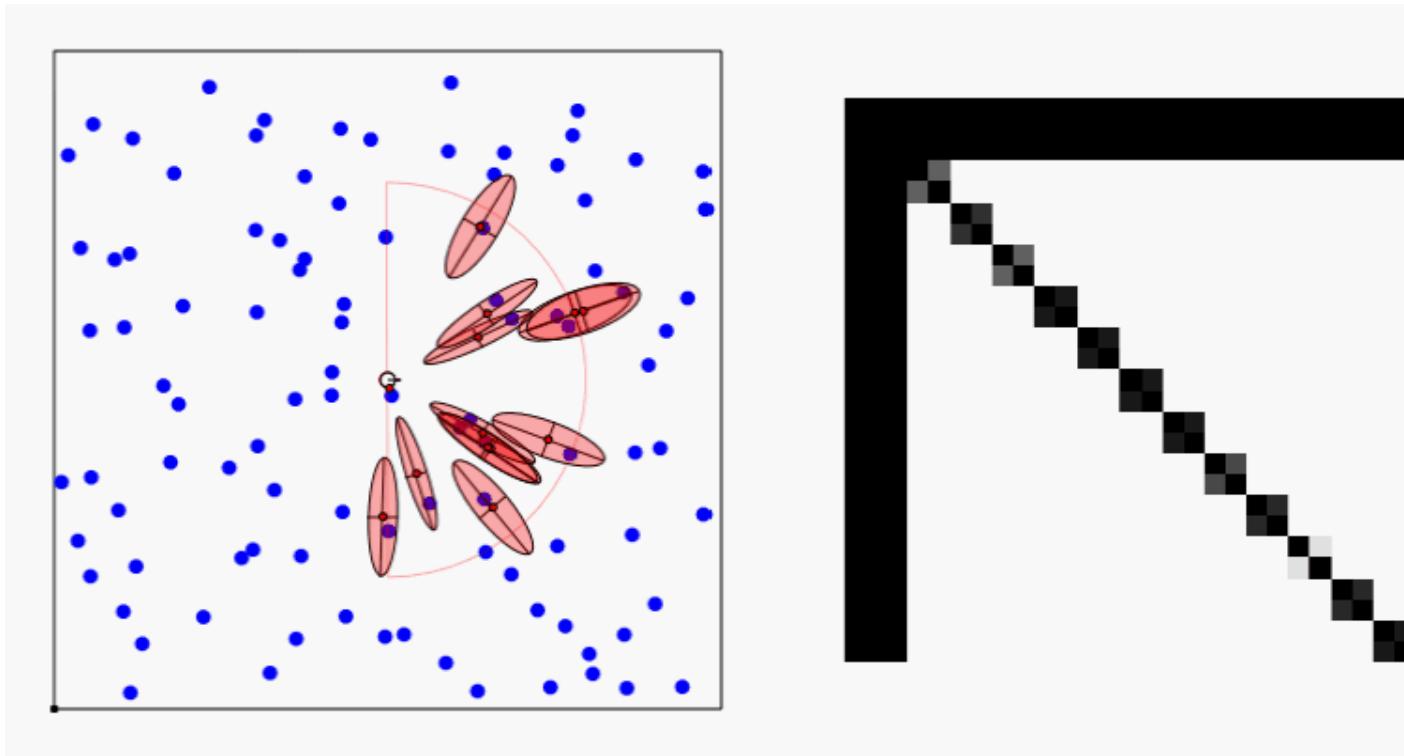
(E)KF-SLAM

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

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

- Can handle hundreds of dimensions

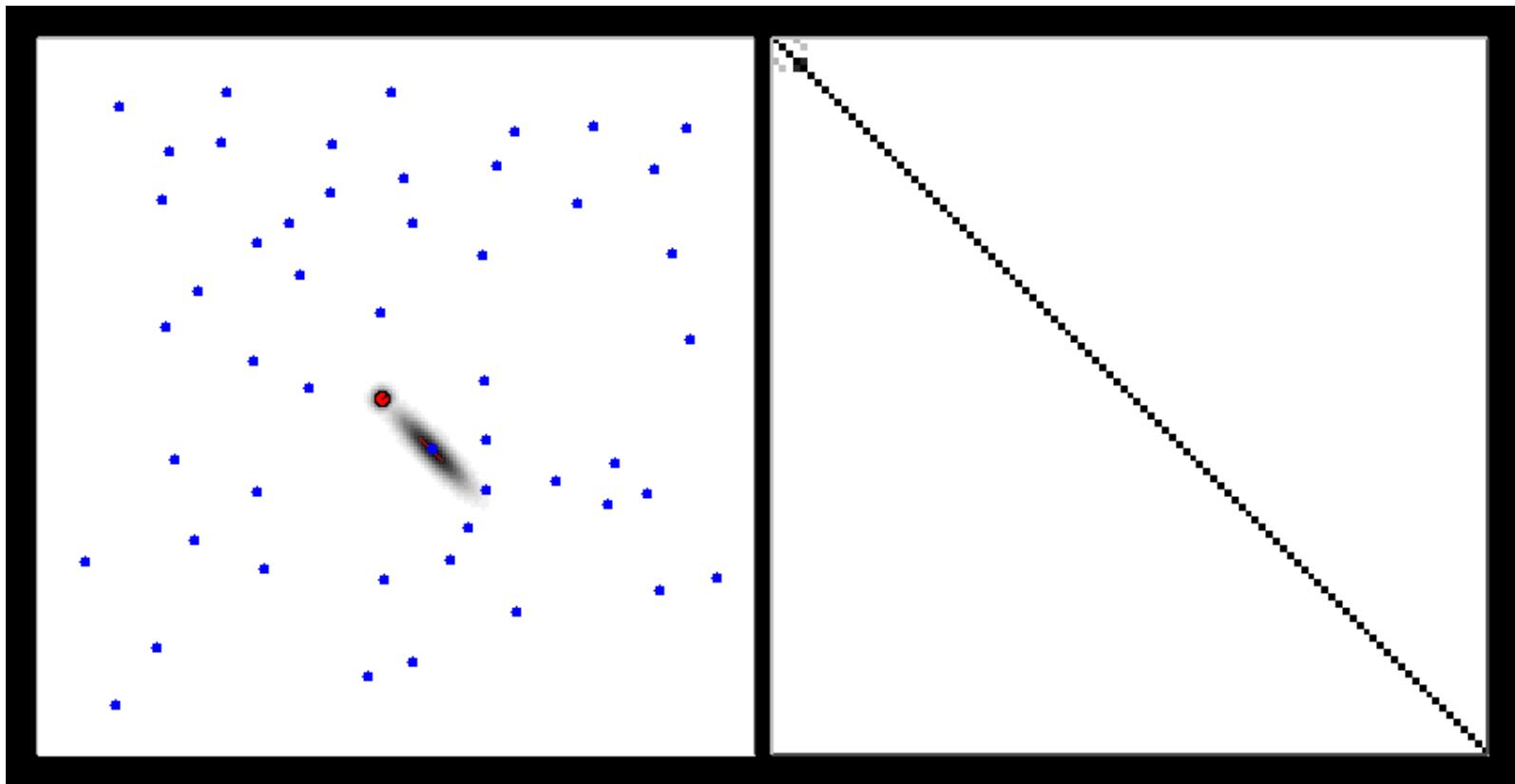
Classical Solution – The EKF



Blue path = true path Red path = estimated path Black path = odometry

- Approximate the SLAM posterior with a high-dimensional Gaussian [Smith & Cheesman, 1986] ...
- Single hypothesis data association

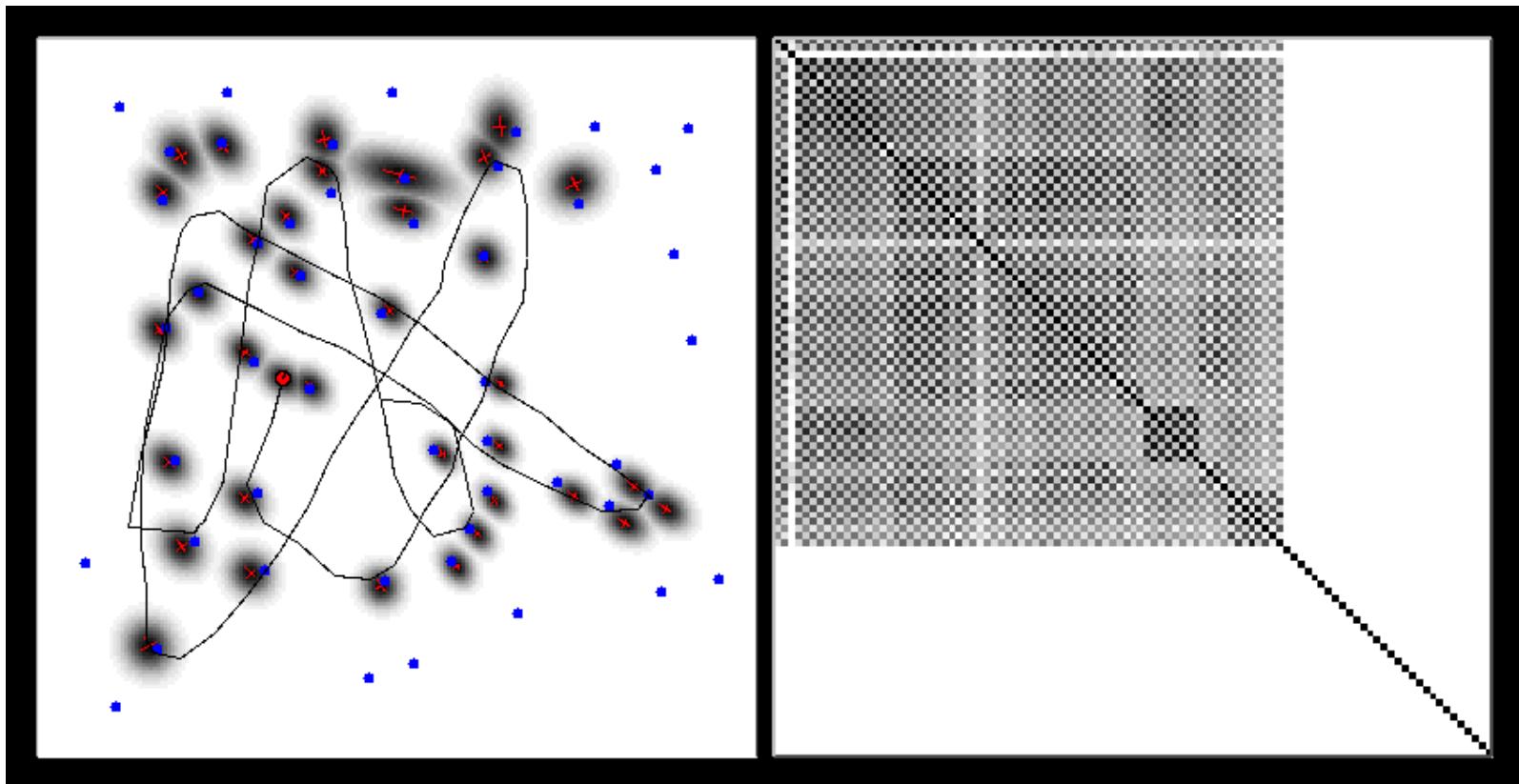
EKF-SLAM



Map

Correlation matrix

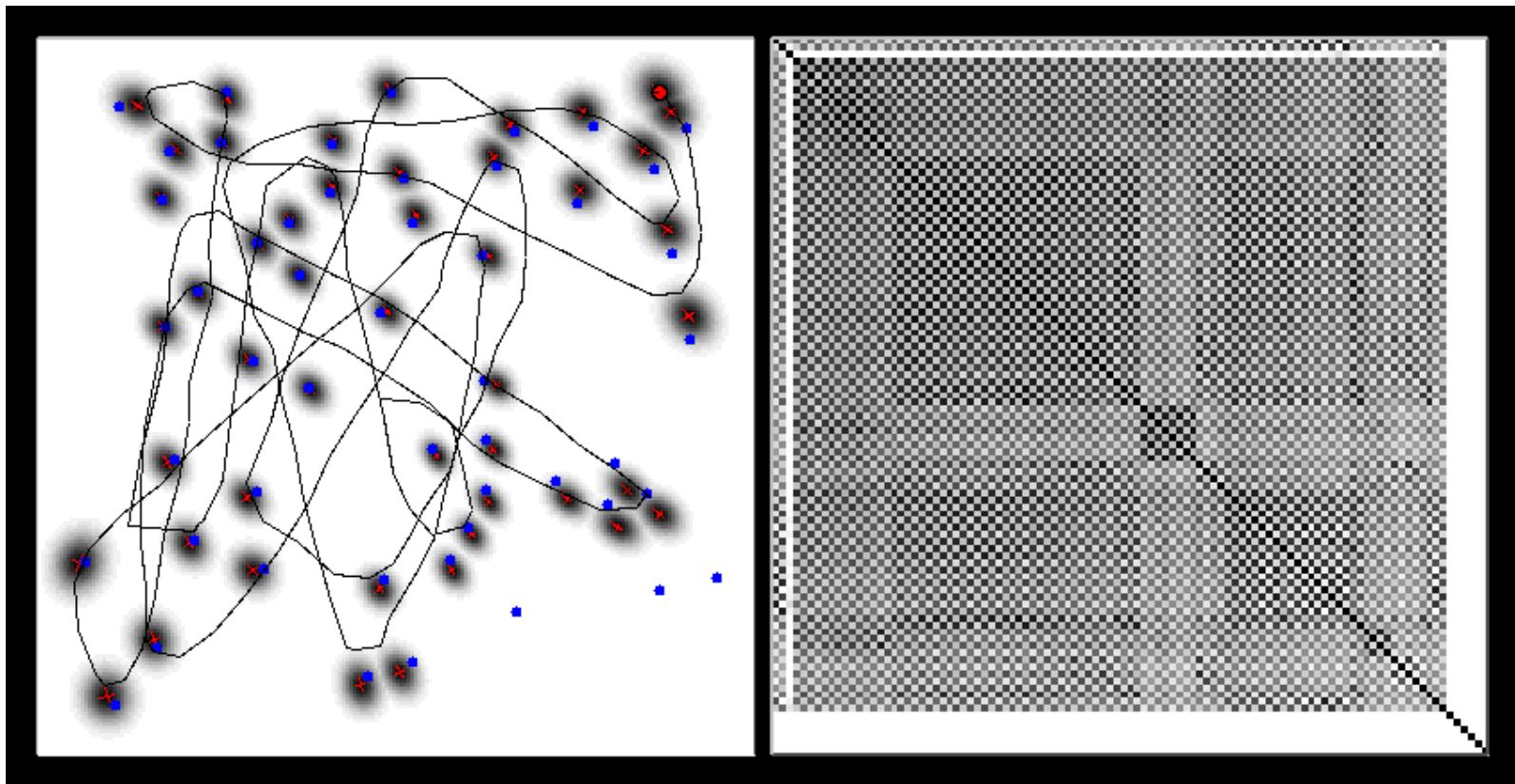
EKF-SLAM



Map

Correlation matrix

EKF-SLAM



Map

Correlation matrix

Properties of KF-SLAM (Linear Case)

[Dissanayake et al., 2001]

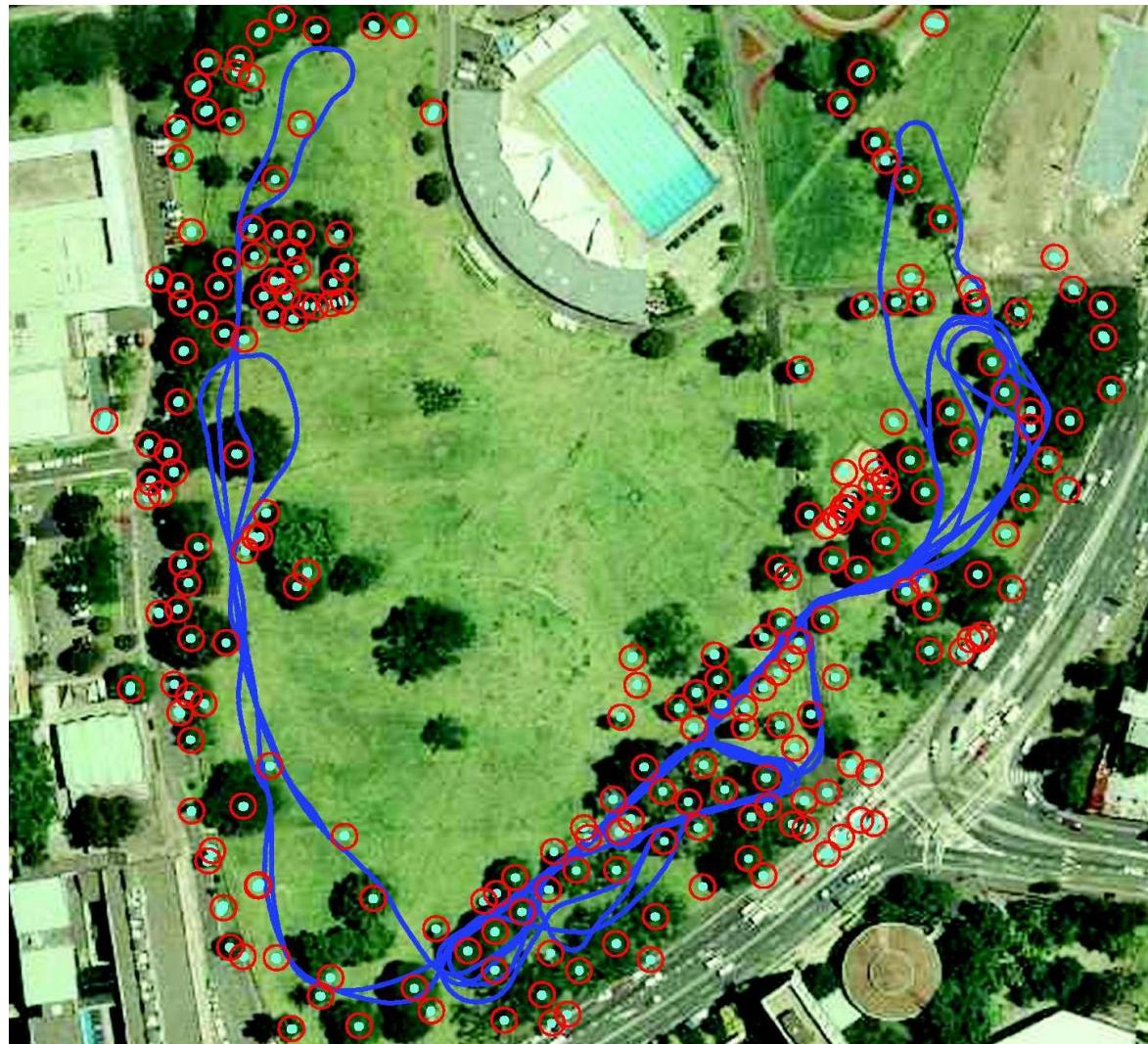
Theorem:

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

Theorem:

In the limit the landmark estimates become fully correlated

Victoria Park Data Set



[courtesy by E. Nebot]

Victoria Park Data Set Vehicle



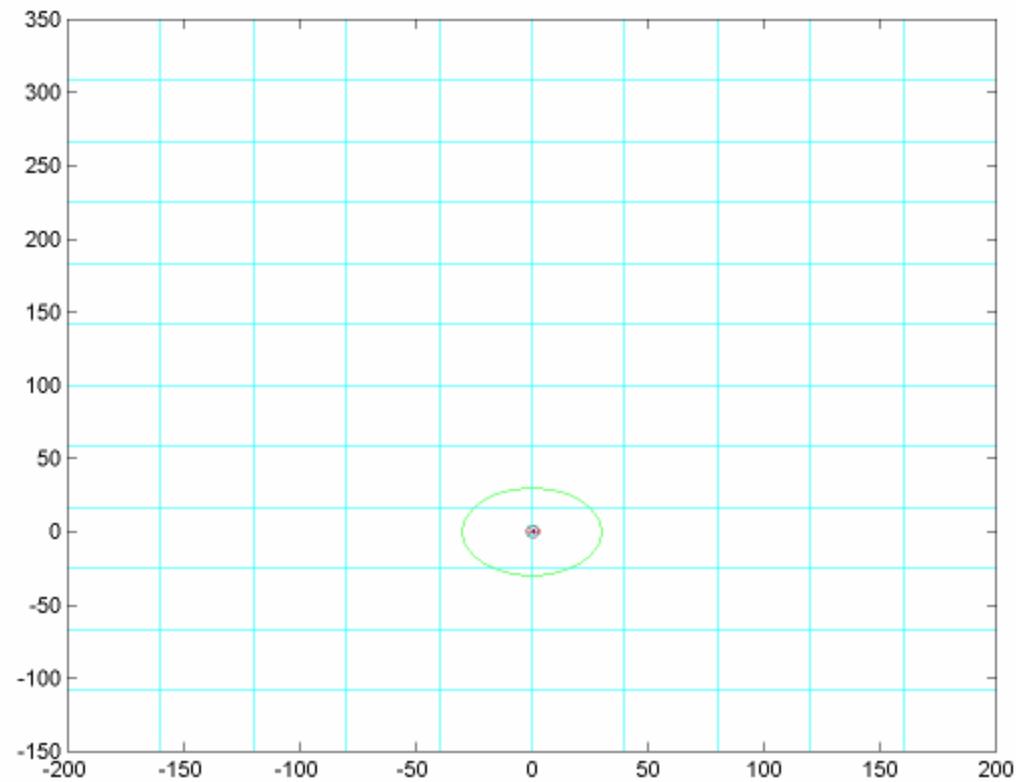
[courtesy by E. Nebot]

Data Acquisition



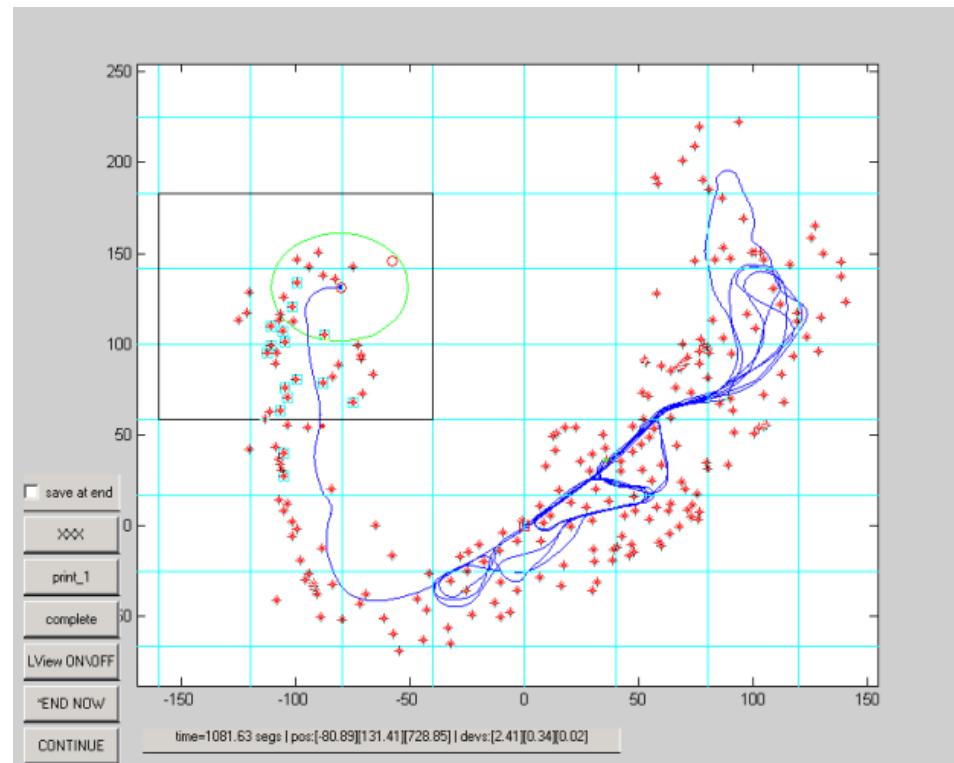
[courtesy by E. Nebot]

SLAM



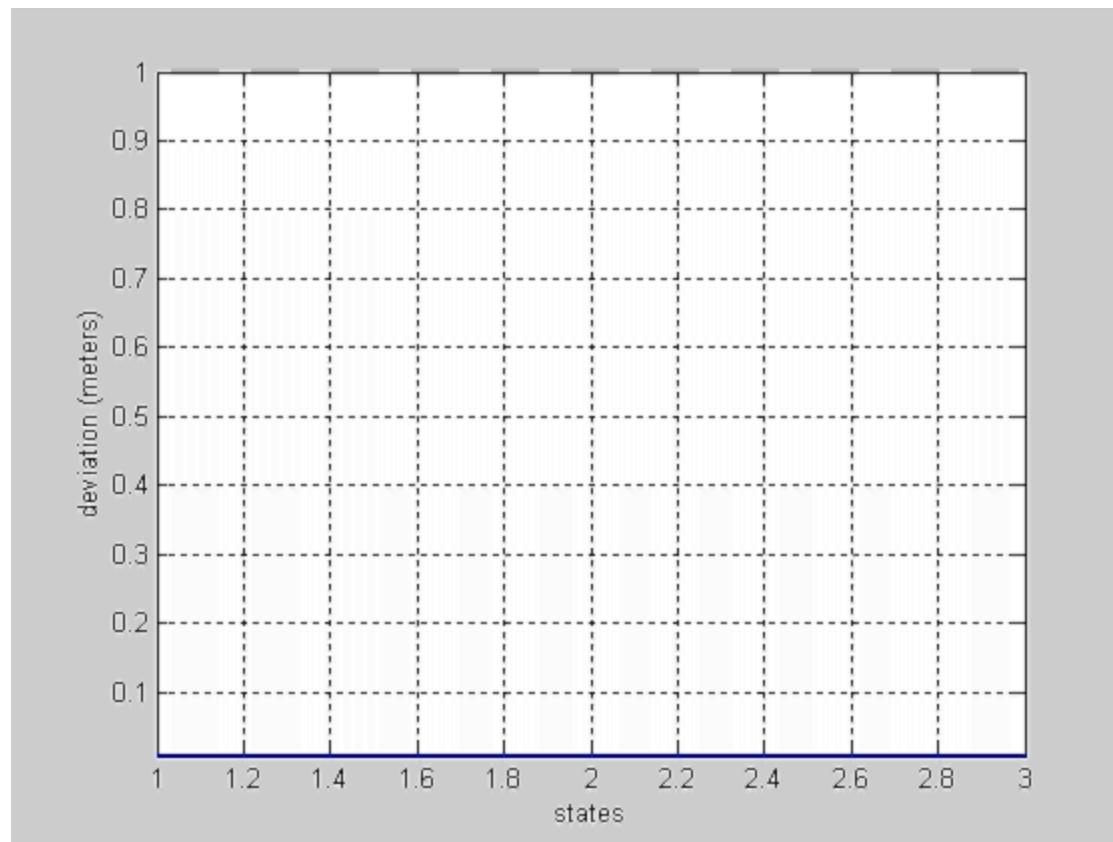
[courtesy by E. Nebot]

Map and Trajectory



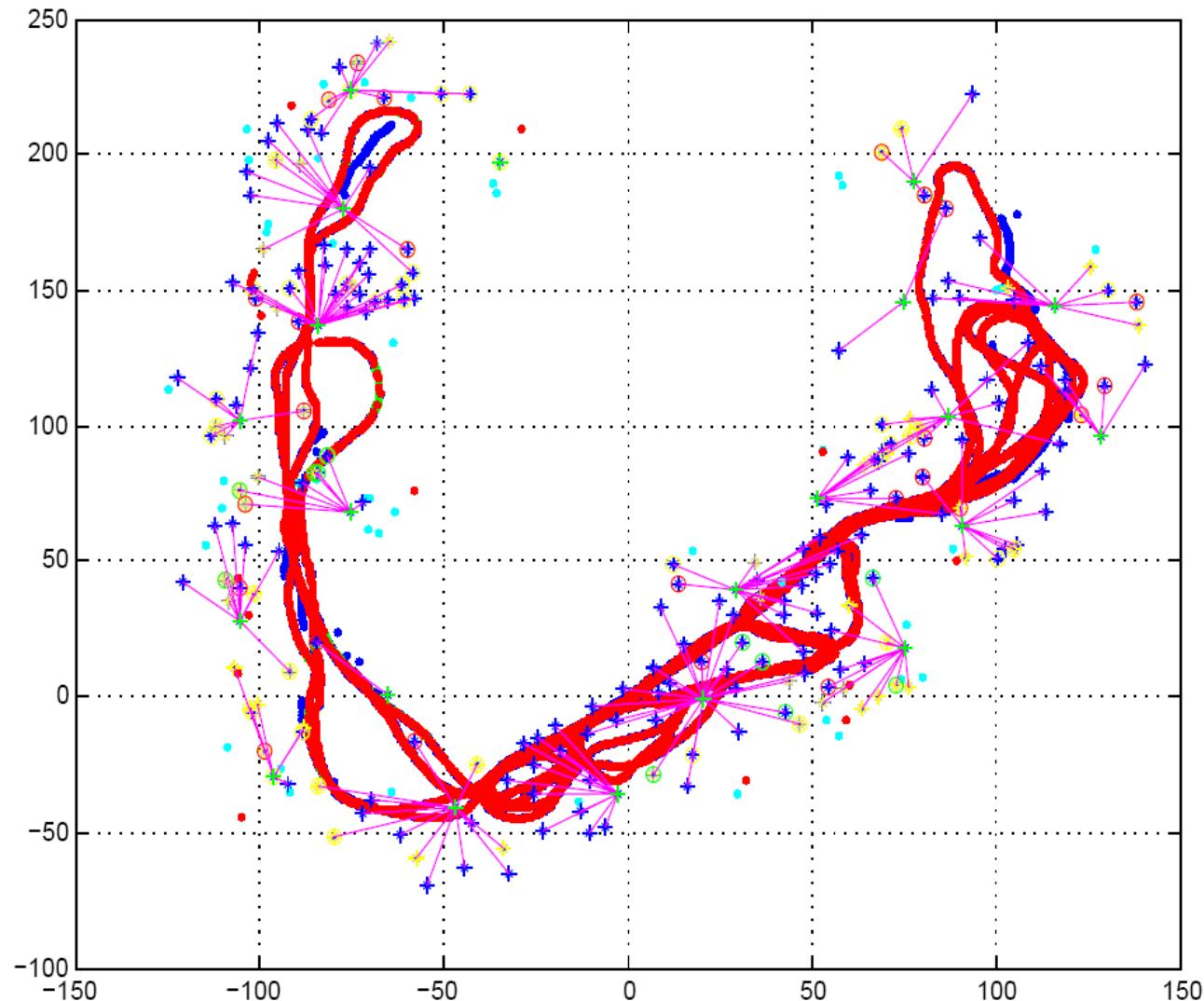
[courtesy by E. Nebot]

Landmark Covariance



[courtesy by E. Nebot]

Estimated Trajectory



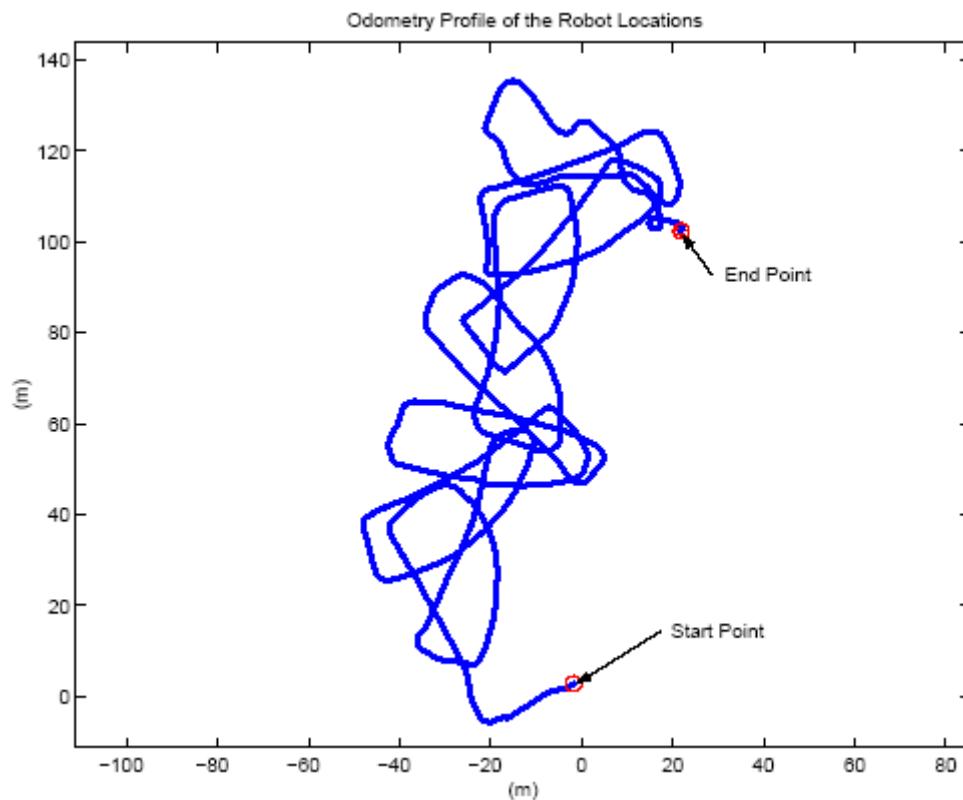
[courtesy by E. Nebot]

EKF SLAM Application

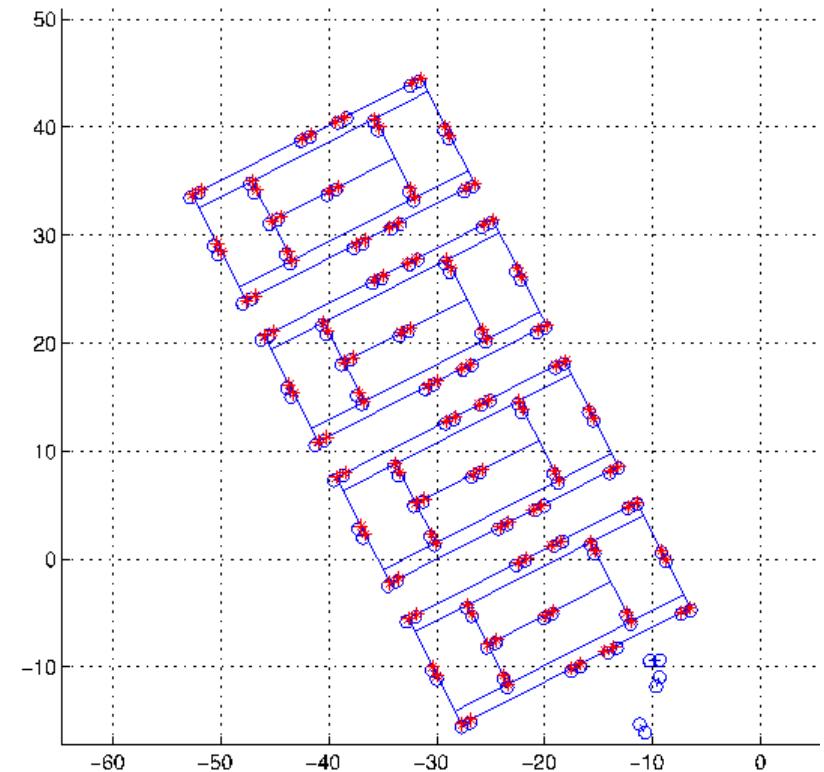


[courtesy by John Leonard]

EKF SLAM Application



odometry



estimated trajectory

[courtesy by John Leonard]

Approximations for SLAM

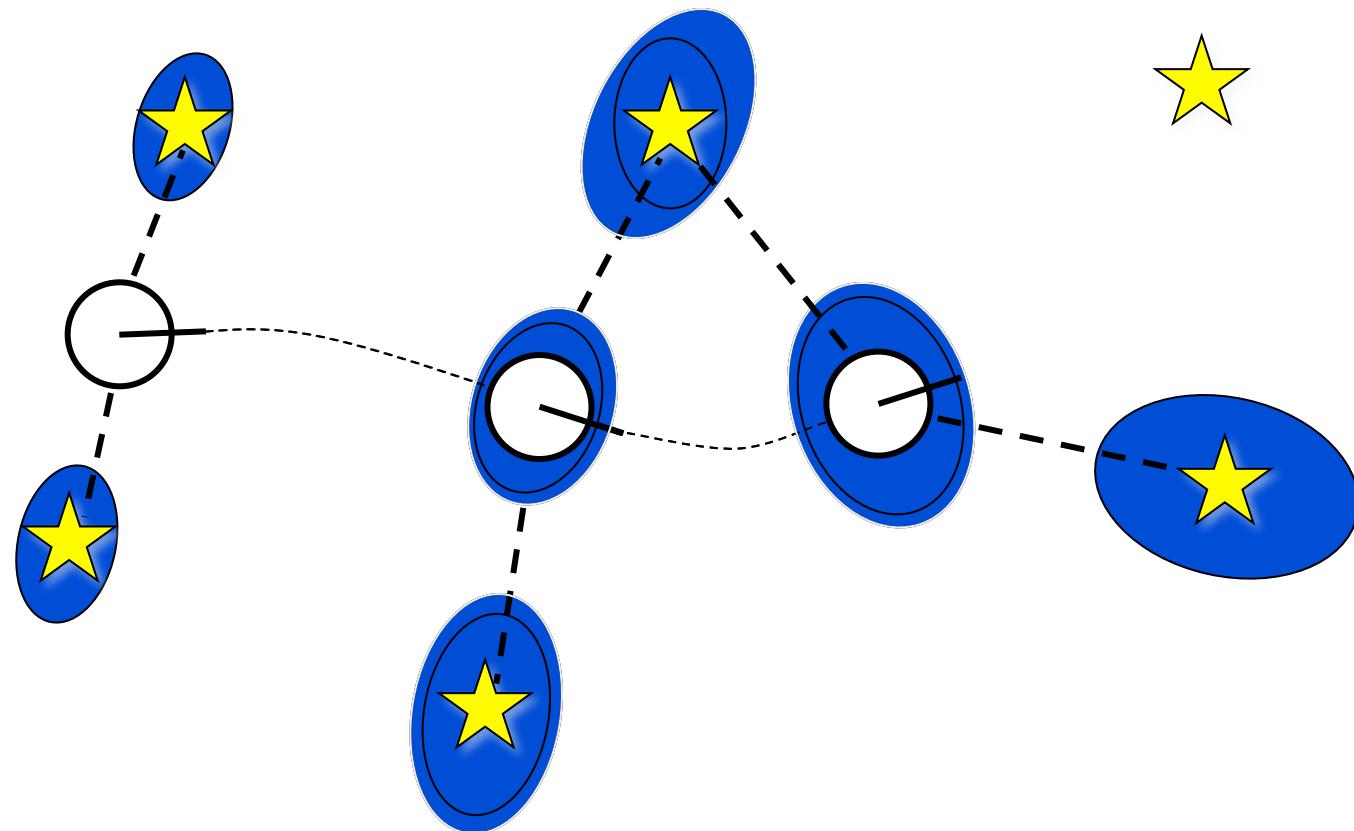
- Local submaps
[Leonard et al. 99, Bosse et al. 02, Newman et al. 03]
- Sparse links (correlations)
[Lu & Milios 97, Guivant & Nebot 01]
- Sparse extended information filters
[Frese et al. 01, Thrun et al. 02]
- Thin junction tree filters
[Paskin 03]
- Rao-Blackwellisation (FastSLAM)
[Murphy 99, Montemerlo et al. 02, Eliazar et al. 03, Haehnel et al. 03]

EKF-SLAM Summary

- Quadratic in the number of landmarks: $O(n^2)$
- Convergence results for the linear case.
- Can diverge if nonlinearities are large!
- Have been applied successfully in large-scale environments.
- Approximations reduce the computational complexity.

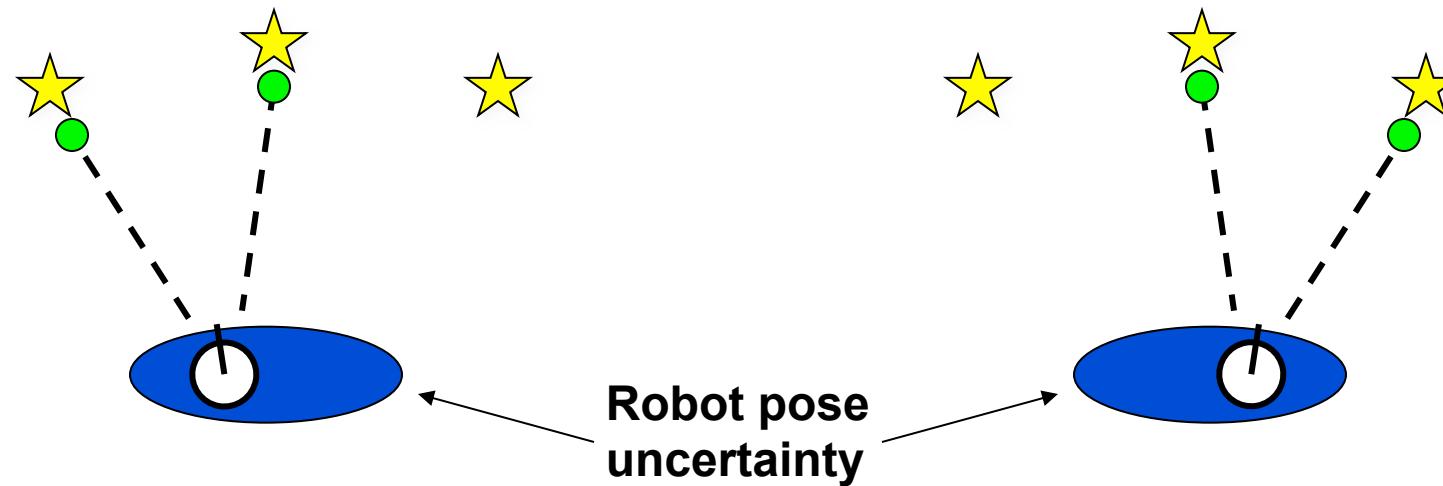
Why is SLAM a hard problem?

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



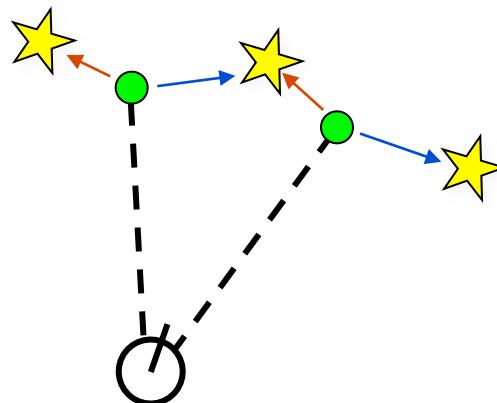
Robot path error correlates errors in the map

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

Data Association Problem



- A data association is an assignment of observations to landmarks
- In general there are more than $\binom{n}{m}$ (n observations, m landmarks) possible associations
- Also called “assignment problem”

Particle Filters

- Represent belief by random samples
- Estimation of non-Gaussian, nonlinear processes
- Sampling Importance Resampling (SIR) principle
 - Draw the new generation of particles
 - Assign an importance weight to each particle
 - Resampling
- Typical application scenarios are tracking, localization, ...

Localization vs. SLAM

- A particle filter can be used to solve both problems
- Localization: state space $\langle x, y, \theta \rangle$
- SLAM: state space $\langle x, y, \theta, map \rangle$
 - for landmark maps = $\langle l_1, l_2, \dots, l_m \rangle$
 - for grid maps = $\langle c_{11}, c_{12}, \dots, c_{1n}, c_{21}, \dots, c_{nm} \rangle$
- **Problem:** The number of particles needed to represent a posterior grows exponentially with the dimension of the state space!

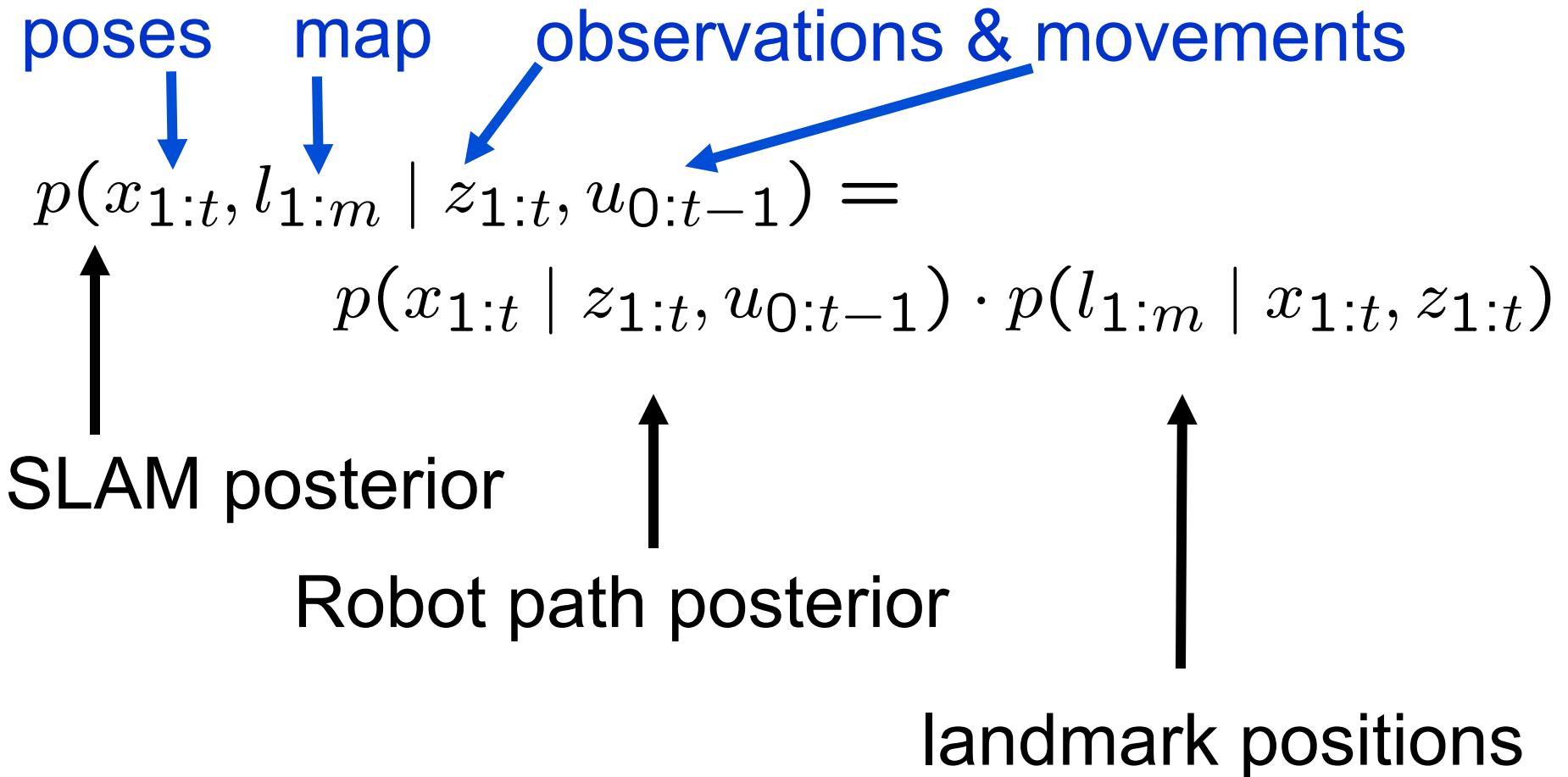
Dependencies

- Is there a dependency between the dimensions of the state space?
- If so, can we use the dependency to solve the problem more efficiently?

Dependencies

- Is there a dependency between the dimensions of the state space?
- If so, can we use the dependency to solve the problem more efficiently?
- In the SLAM context
 - The map depends on the poses of the robot.
 - We know how to build a map given the position of the sensor is known.

Factored Posterior (Landmarks)



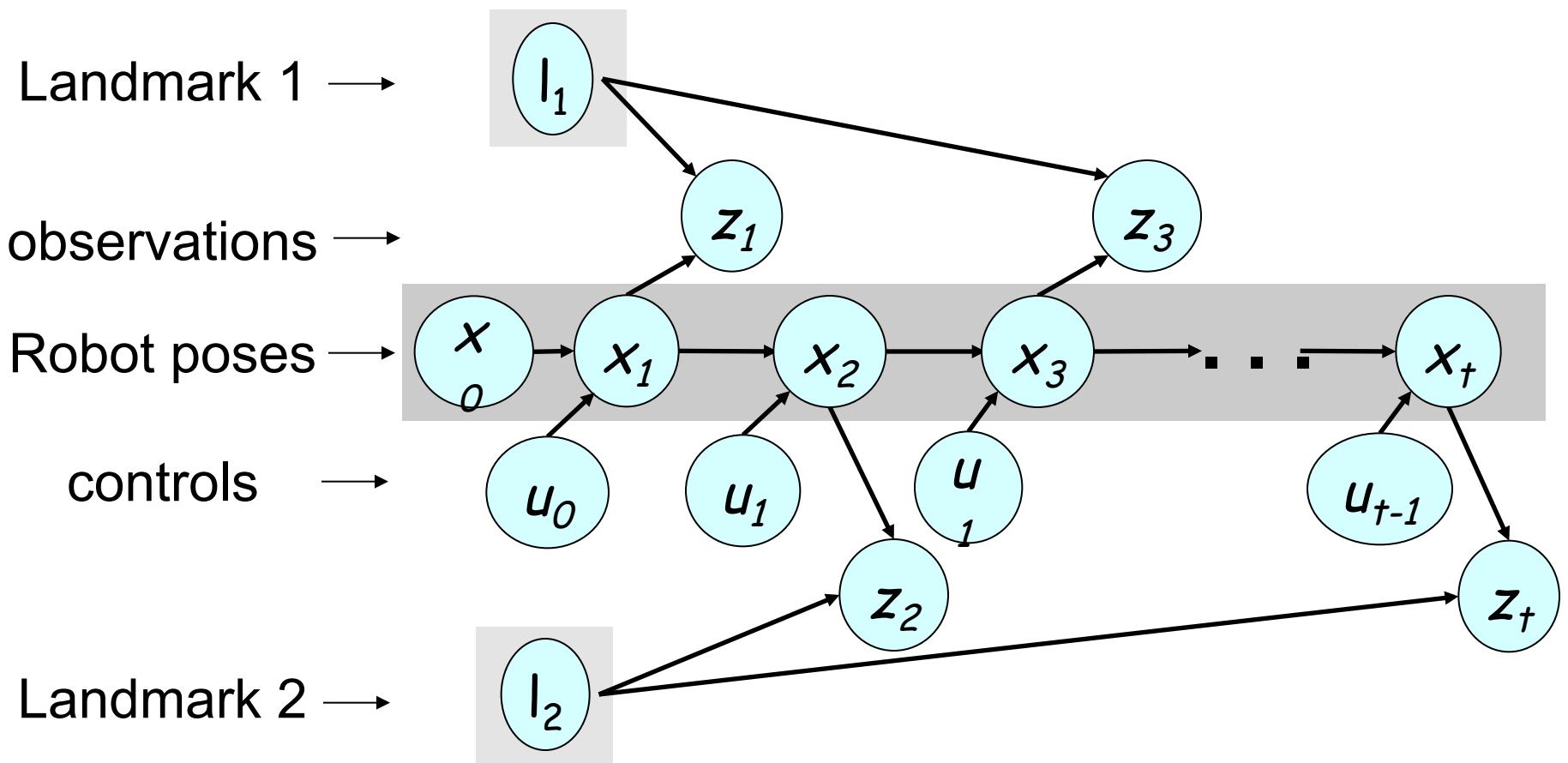
Does this help to solve the problem?

Factored Posterior (Landmarks)

poses map observations & movements

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

Mapping using Landmarks



Knowledge of the robot's true path renders landmark positions conditionally independent

Factored Posterior

$$\begin{aligned} p(x_{1:t}, l_{1:m} \mid z_{1:t}, u_{0:t-1}) \\ = p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot p(l_{1:m} \mid x_{1:t}, z_{1:t}) \\ = p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot \prod_{i=1}^M p(l_i \mid x_{1:t}, z_{1:t}) \end{aligned}$$

Robot path posterior
(localization problem)

Conditionally
independent
landmark positions

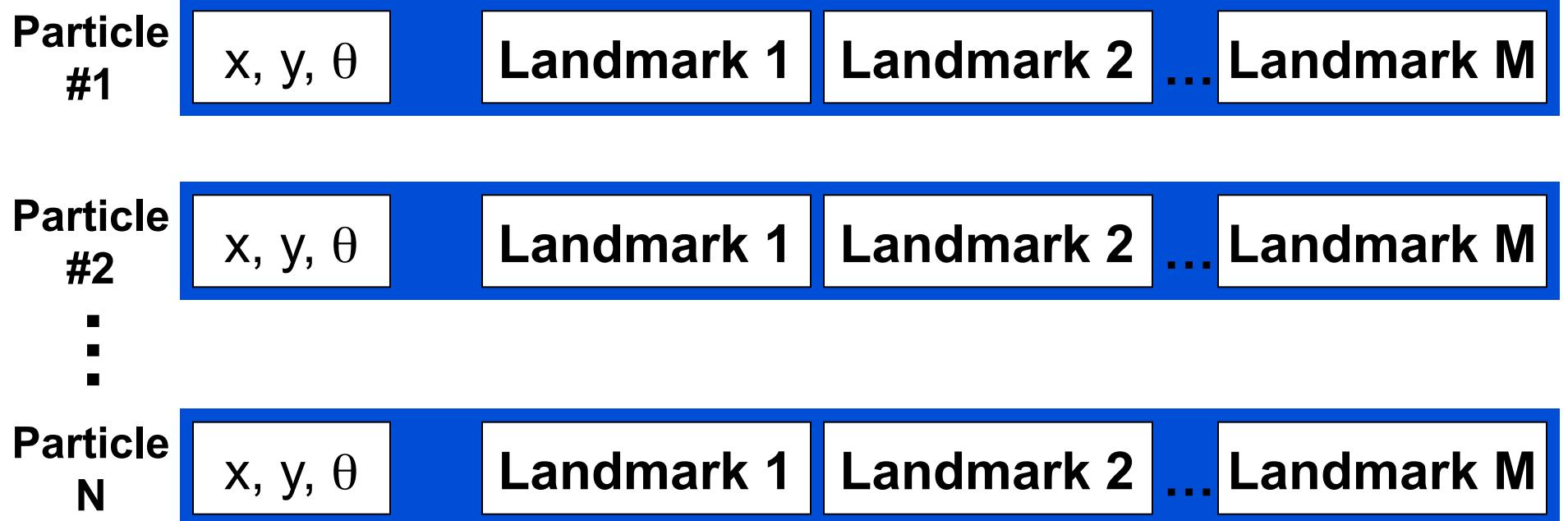
Rao-Blackwellization

$$p(x_{1:t}, l_{1:m} \mid z_{1:t}, u_{0:t-1}) = \\ p(x_{1:t} \mid z_{1:t}, u_{0:t-1}) \cdot \prod_{i=1}^M p(l_i \mid x_{1:t}, z_{1:t})$$

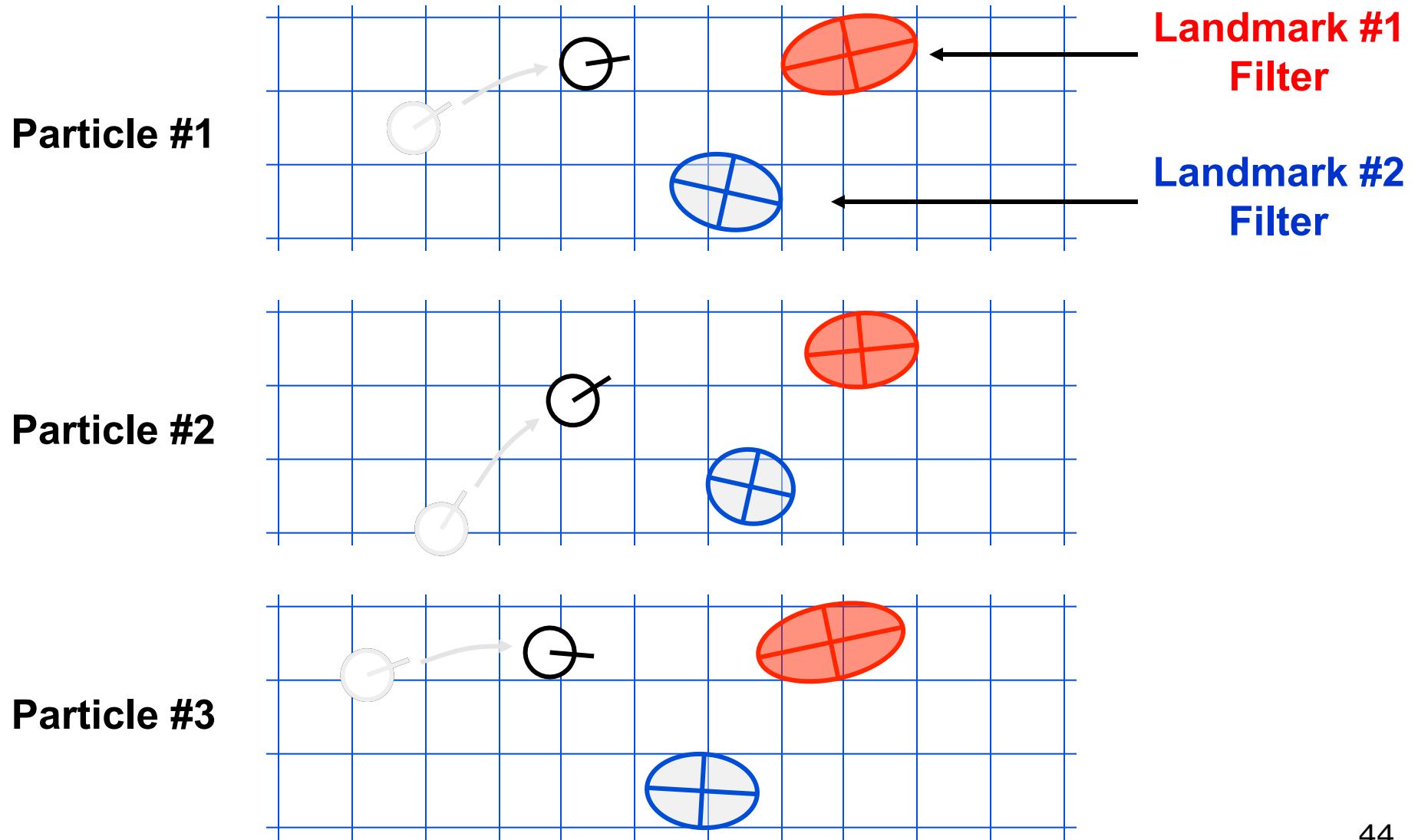
- This factorization is also called Rao-Blackwellization
- Given that the second term can be computed efficiently, particle filtering becomes possible!

FastSLAM

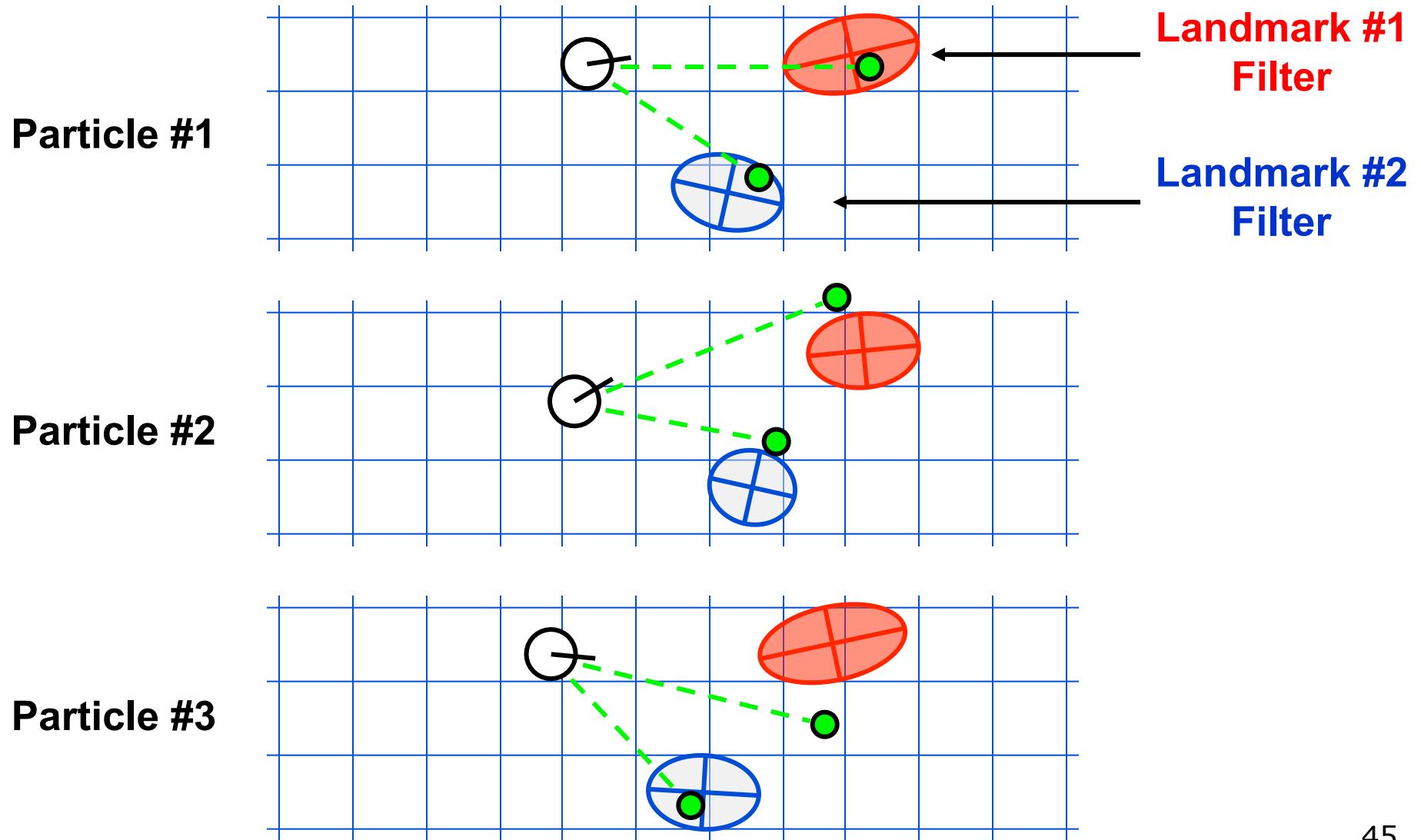
- Rao-Blackwellized particle filtering based on landmarks [Montemerlo et al., 2002]
- Each landmark is represented by a 2×2 Extended Kalman Filter (EKF)
- Each particle therefore has to maintain M EKFs



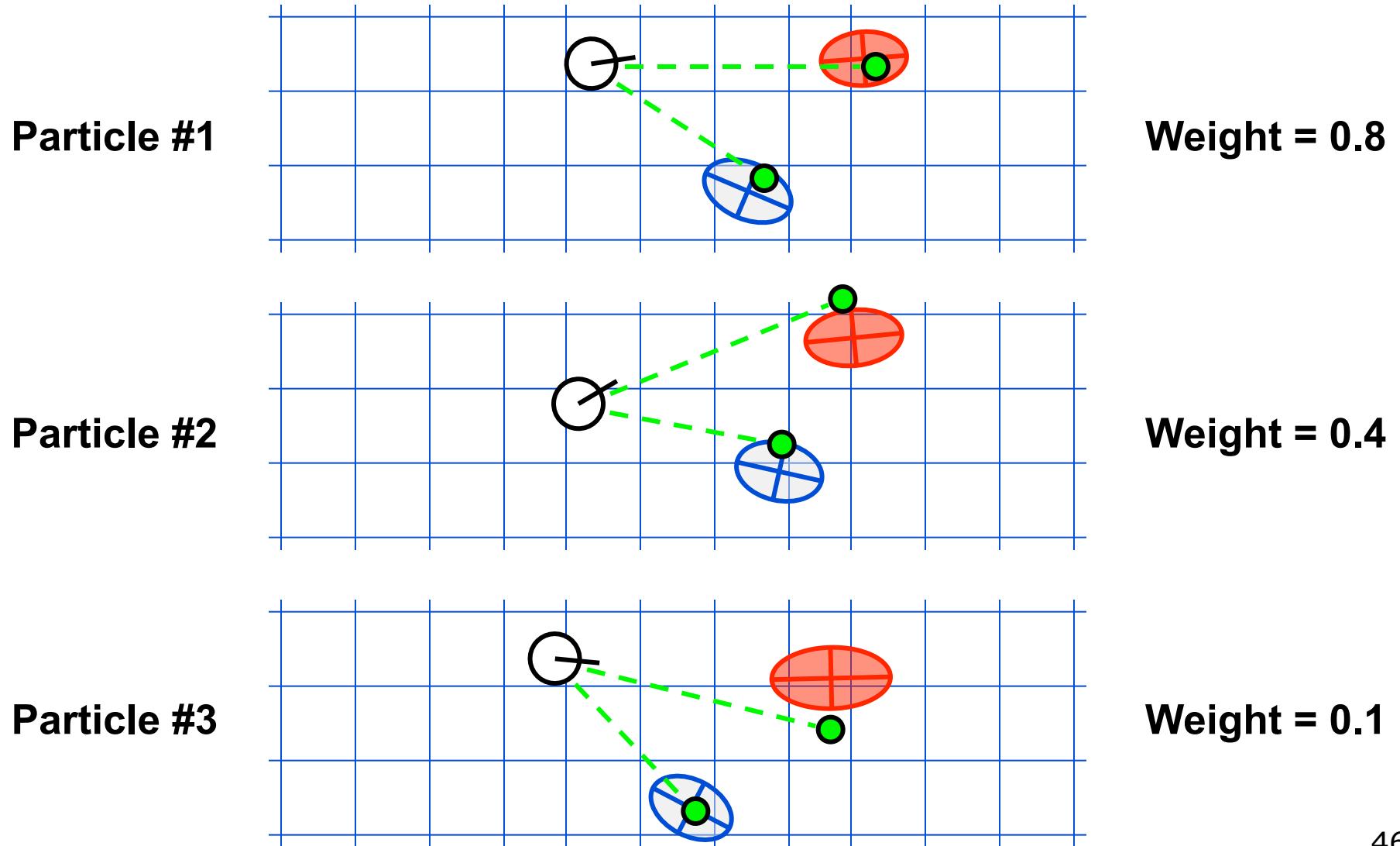
FastSLAM – Action Update



FastSLAM – Sensor Update



FastSLAM – Sensor Update



FastSLAM Complexity

- Update robot particles based on control u_{t-1} $O(N)$
Constant time per particle
- Incorporate observation z_t into Kalman filters $O(N \cdot \log(M))$
Log time per particle
- Resample particle set $O(N \cdot \log(M))$
Log time per particle

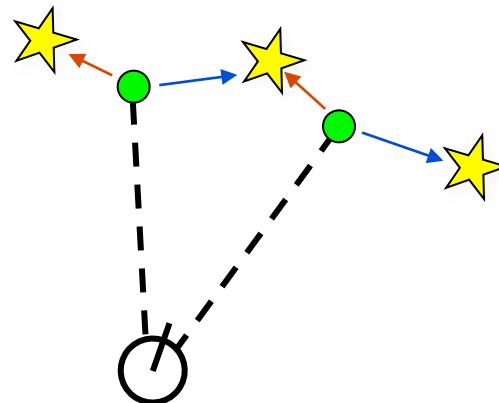
N = Number of particles

M = Number of map features

O(N \cdot log(M))
Log time per particle

Data Association Problem

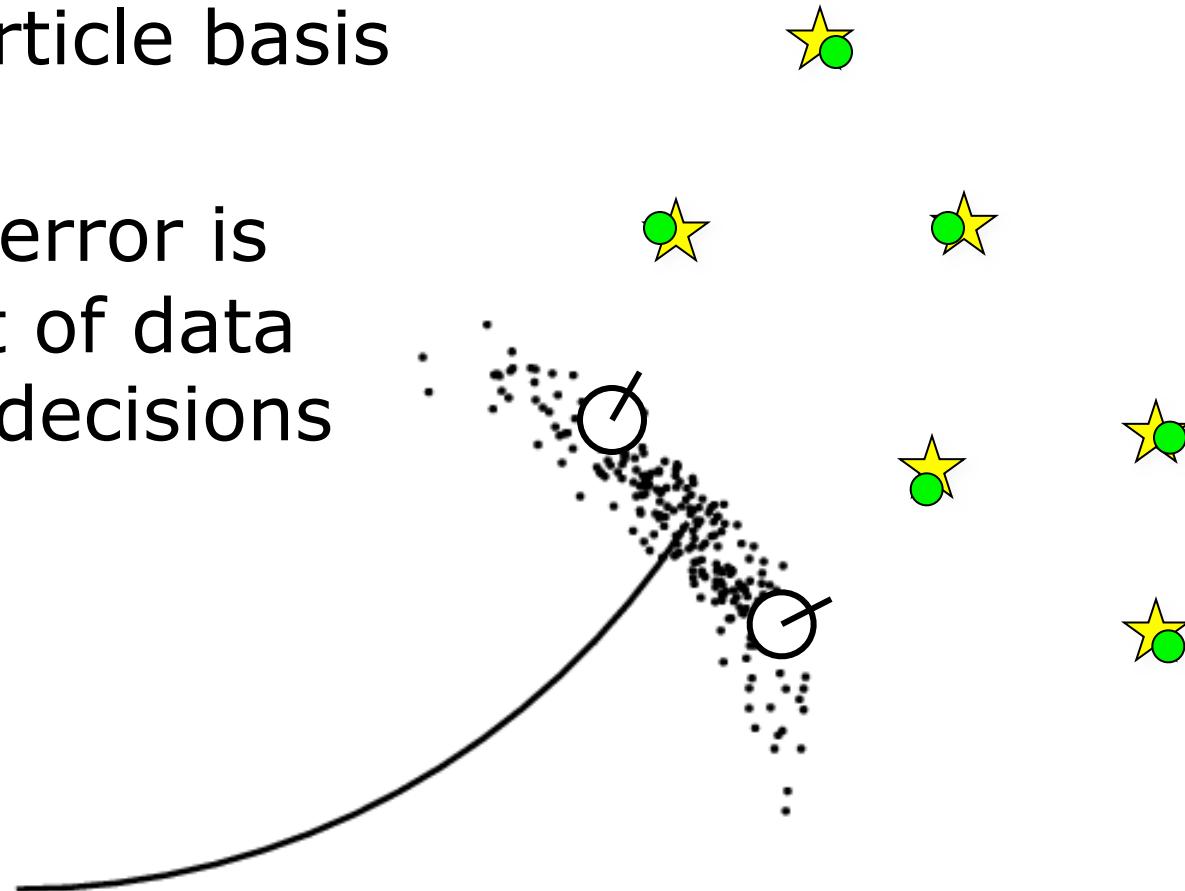
- Which observation belongs to which landmark?



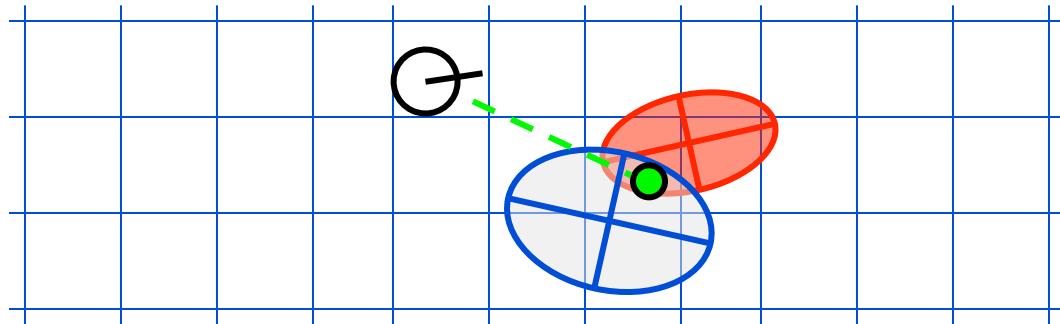
- A robust SLAM must consider possible data associations
- Potential data associations depend also on the pose of the robot

Multi-Hypothesis Data Association

- Data association is done on a per-particle basis
- Robot pose error is factored out of data association decisions



Per-Particle Data Association



$$P(\text{observation}|\text{red}) = 0.3$$

$$P(\text{observation}|\text{blue}) = 0.7$$

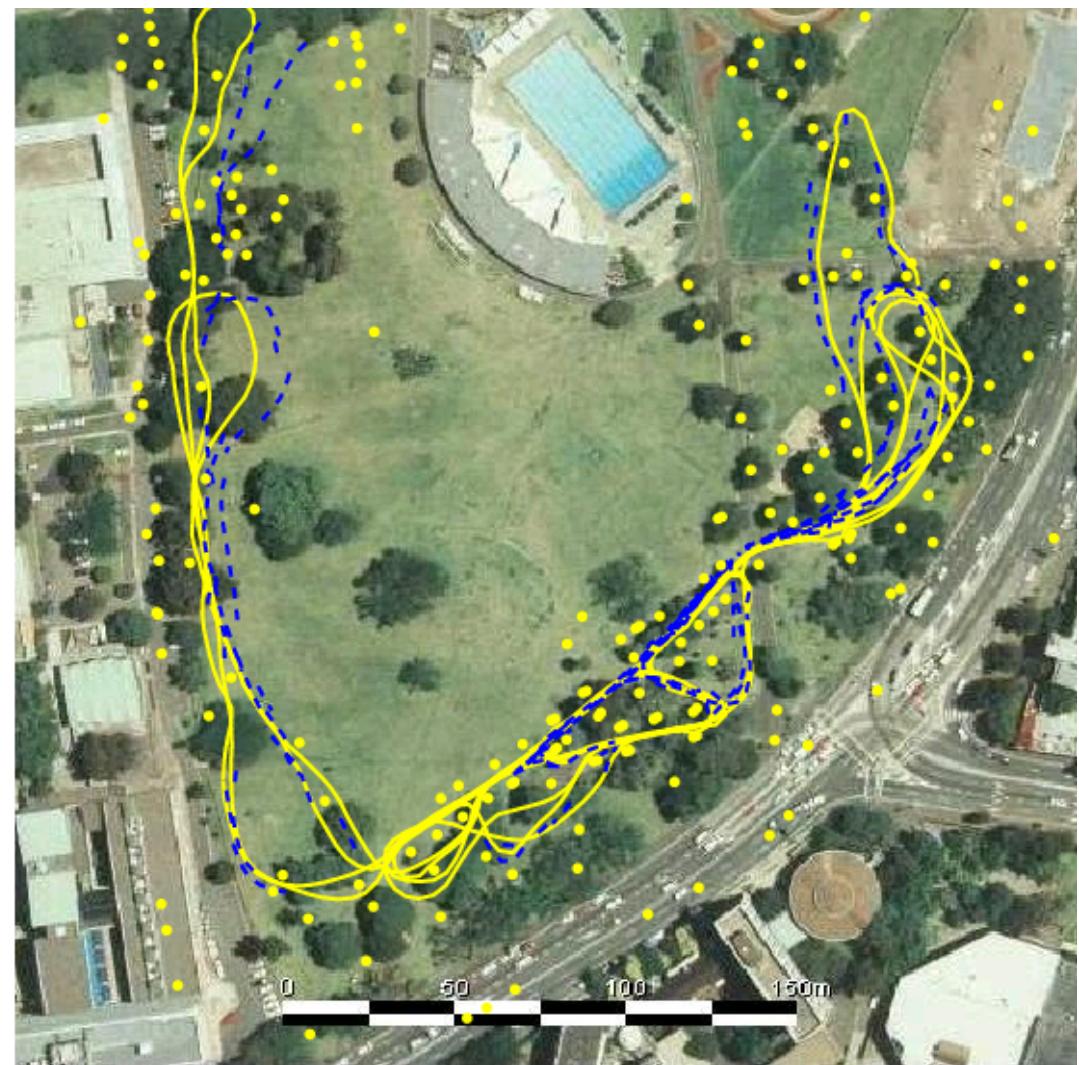
Was the observation generated by the red or the blue landmark?

- Two options for per-particle data association
 - Pick the most probable match
 - Pick an random association weighted by the observation likelihoods
- If the probability is too low, generate a new landmark

Results – Victoria Park

- 4 km traverse
- < 5 m RMS position error
- 100 particles

Blue = GPS
Yellow = FastSLAM



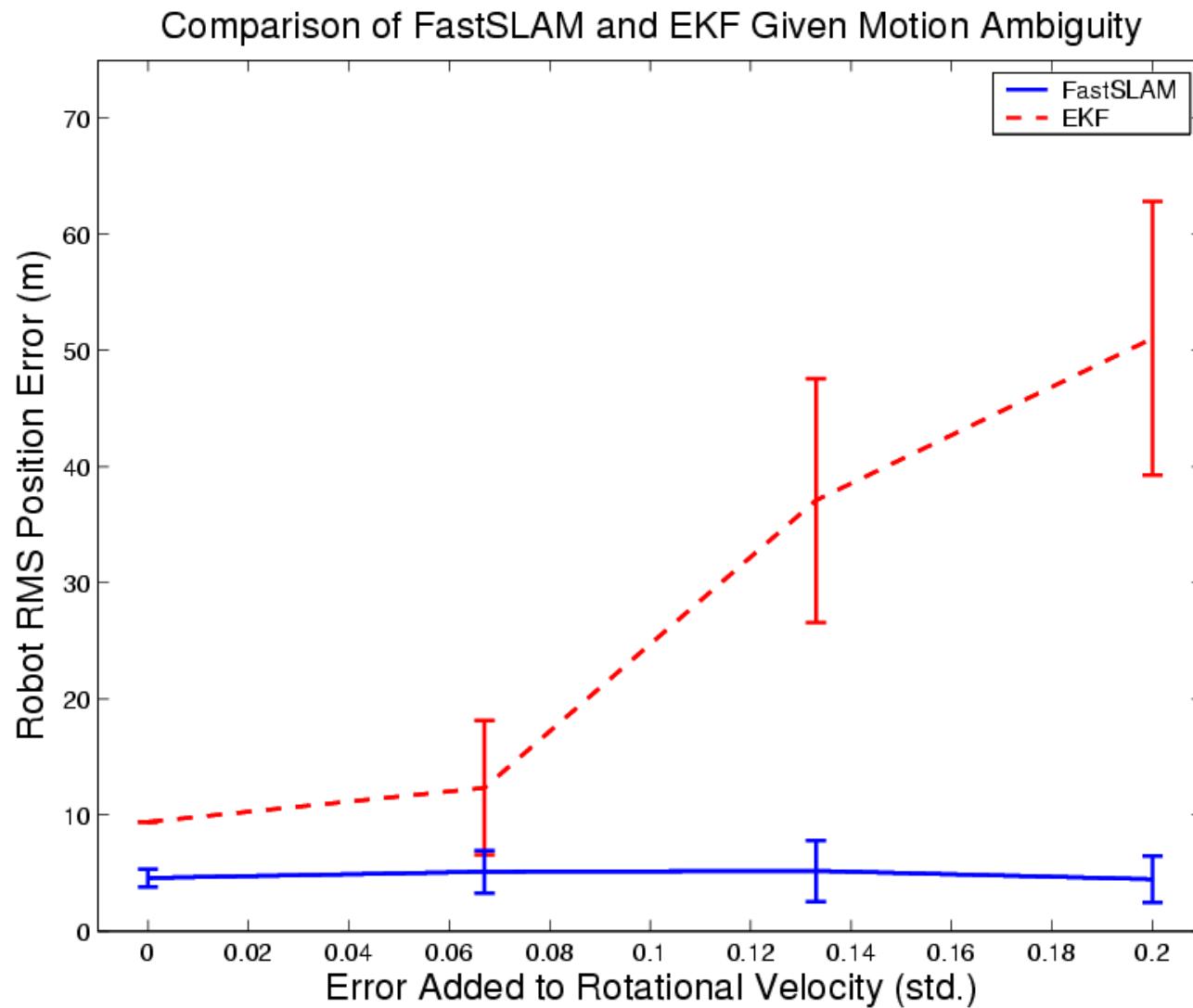
Dataset courtesy of University of Sydney 51

Results – Victoria Park



Dataset courtesy of University of Sydney 52

Results – Data Association



Results – Accuracy

