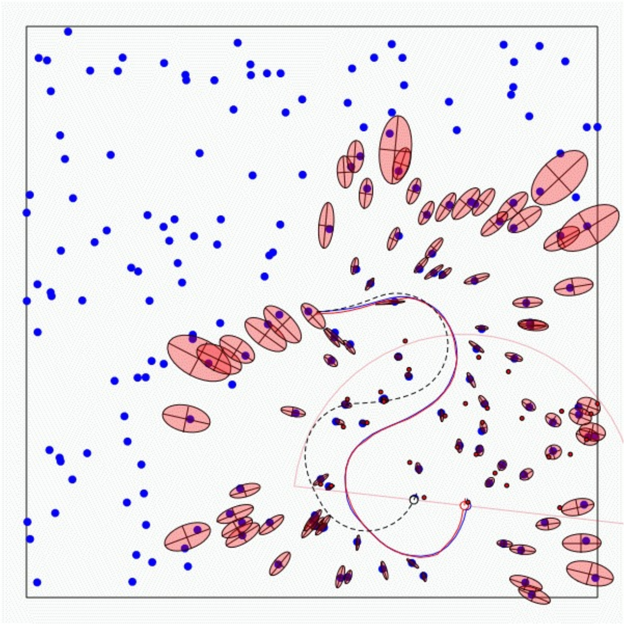
CSE-571
Robotics

SLAM: Simultaneous Localization and Mapping

Many slides courtesy of Ryan Eustice, Cyrill Stachniss, John Leonard

The SLAM Problem



A robot is exploring an unknown, static environment.

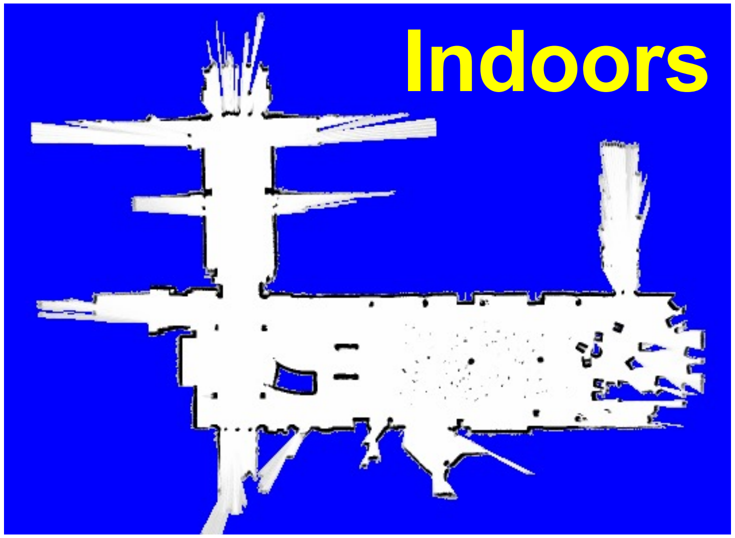
Given:

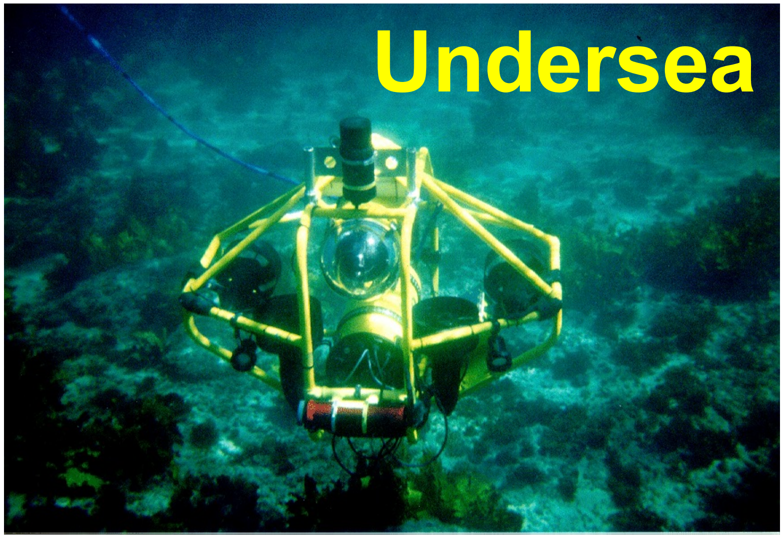
- The robot's controls
- Observations of nearby features

Estimate:

- Map of features
- Path of the robot

SLAM Applications





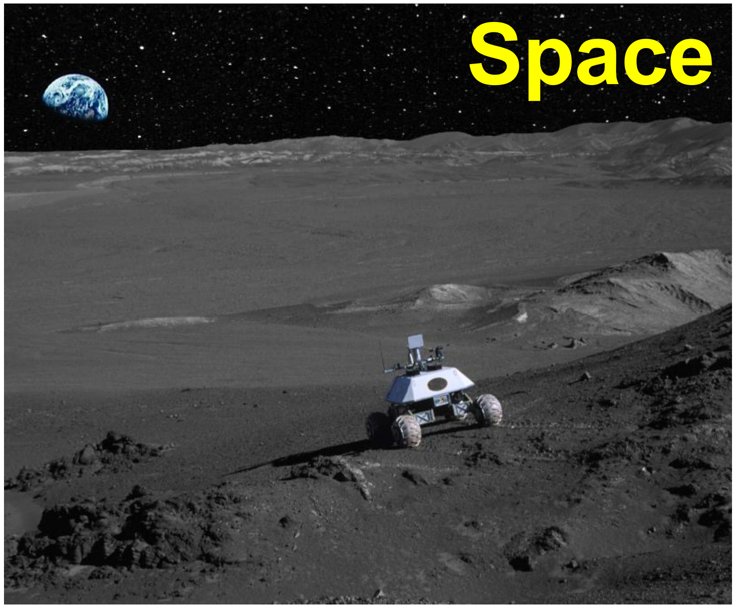
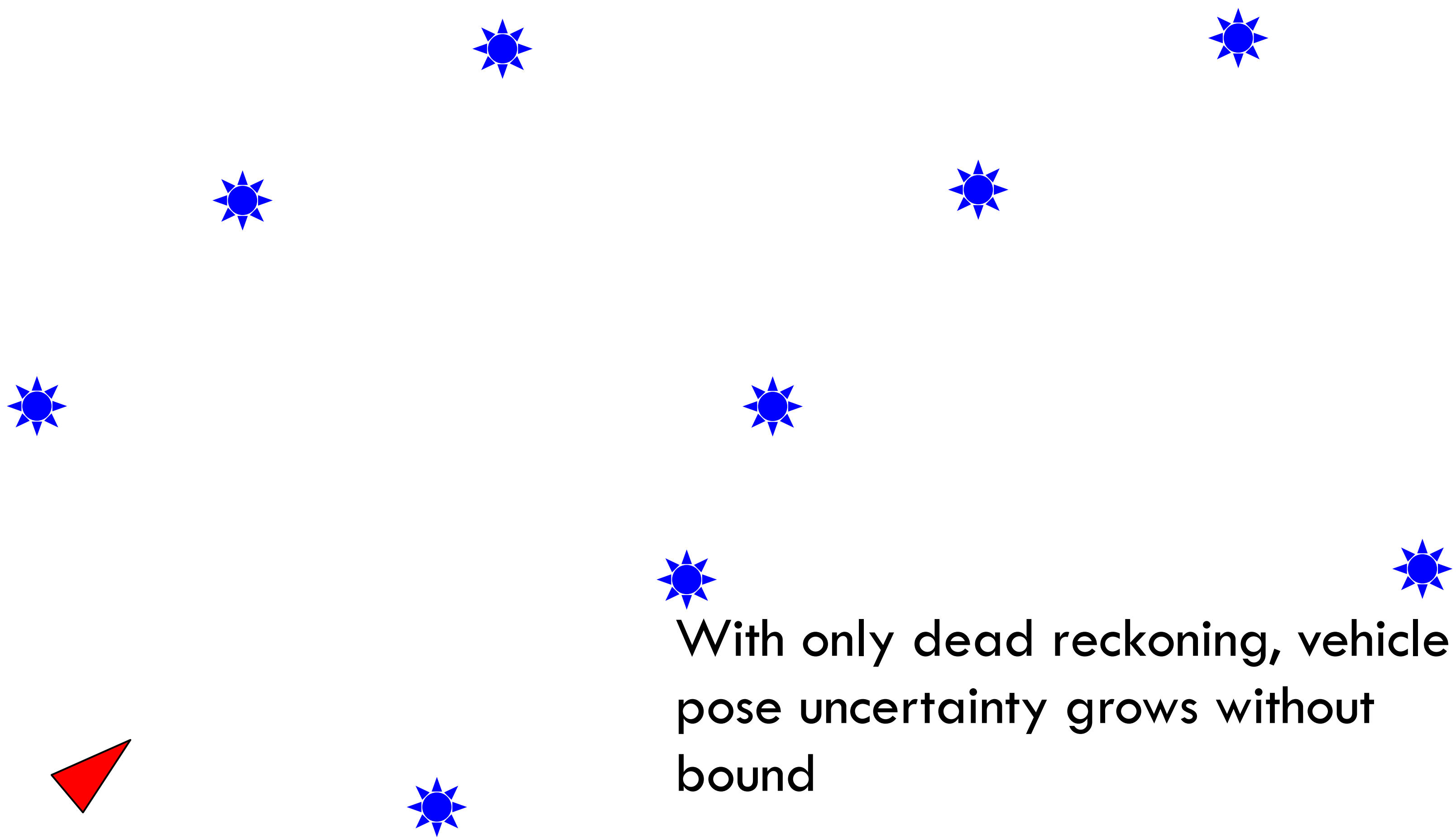


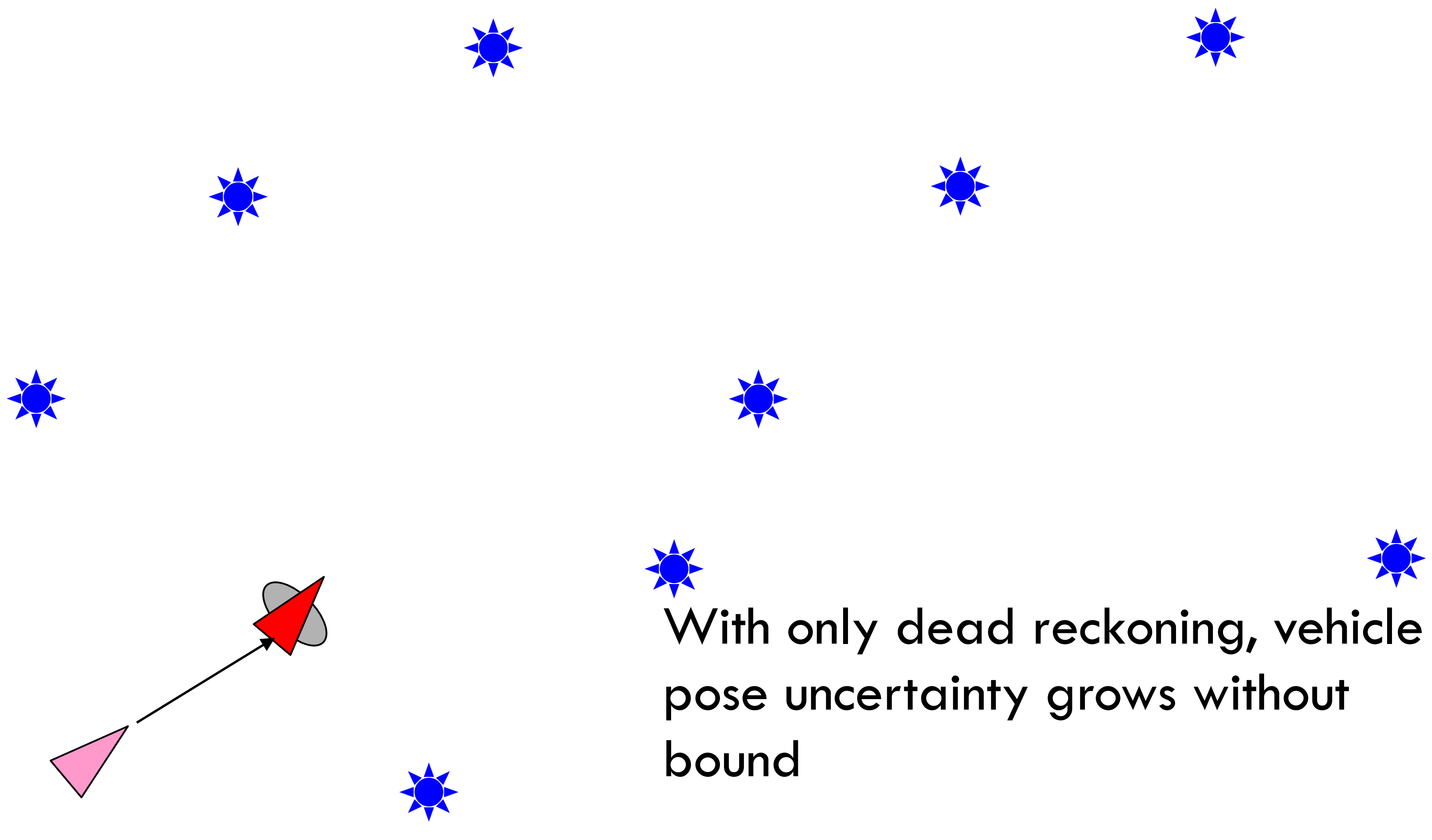
Illustration of SLAM without Landmarks



With only dead reckoning, vehicle pose uncertainty grows without bound

Courtesy J. Leonard

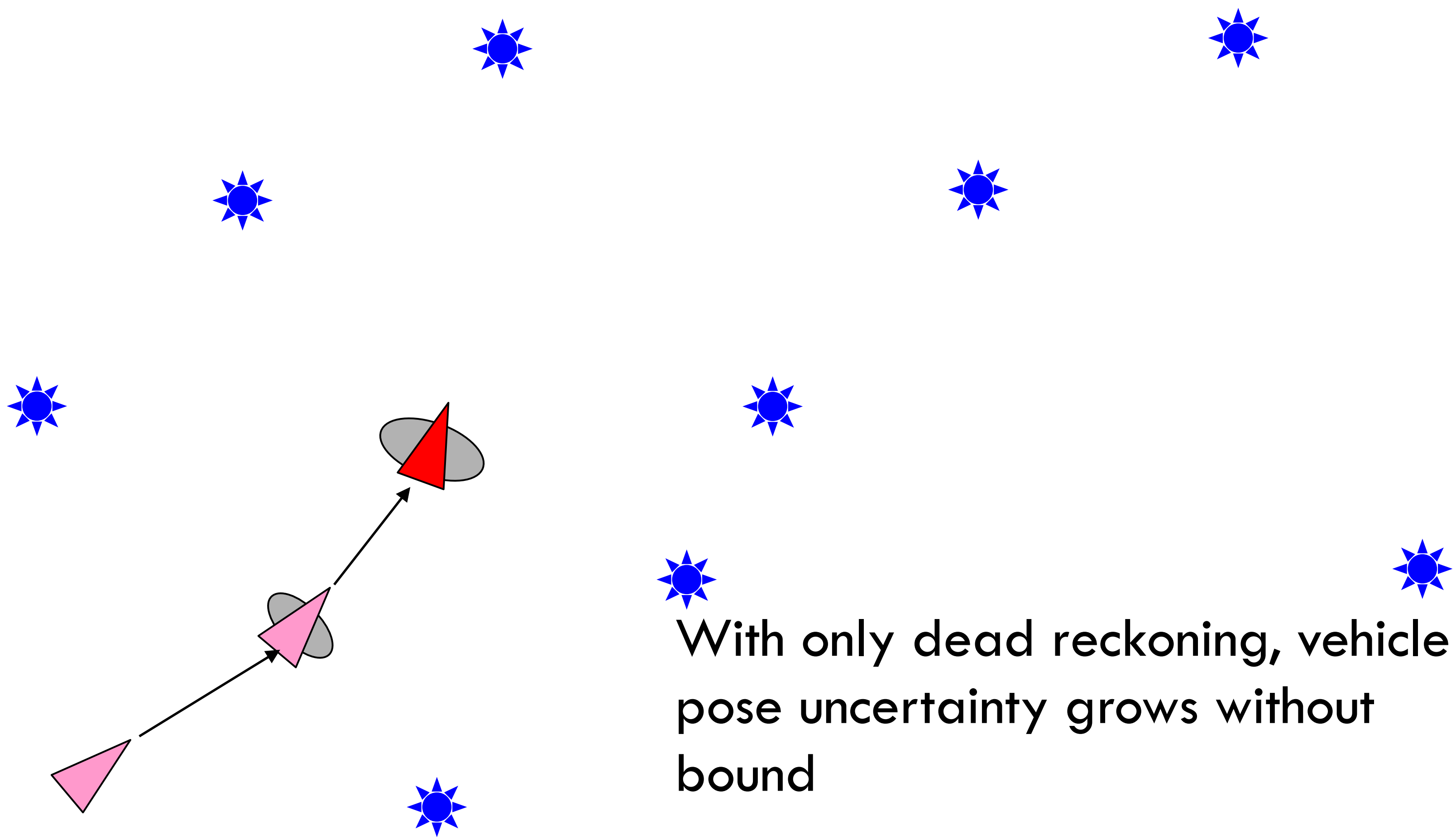
Illustration of SLAM without Landmarks



With only dead reckoning, vehicle pose uncertainty grows without bound

Courtesy J. Leonard

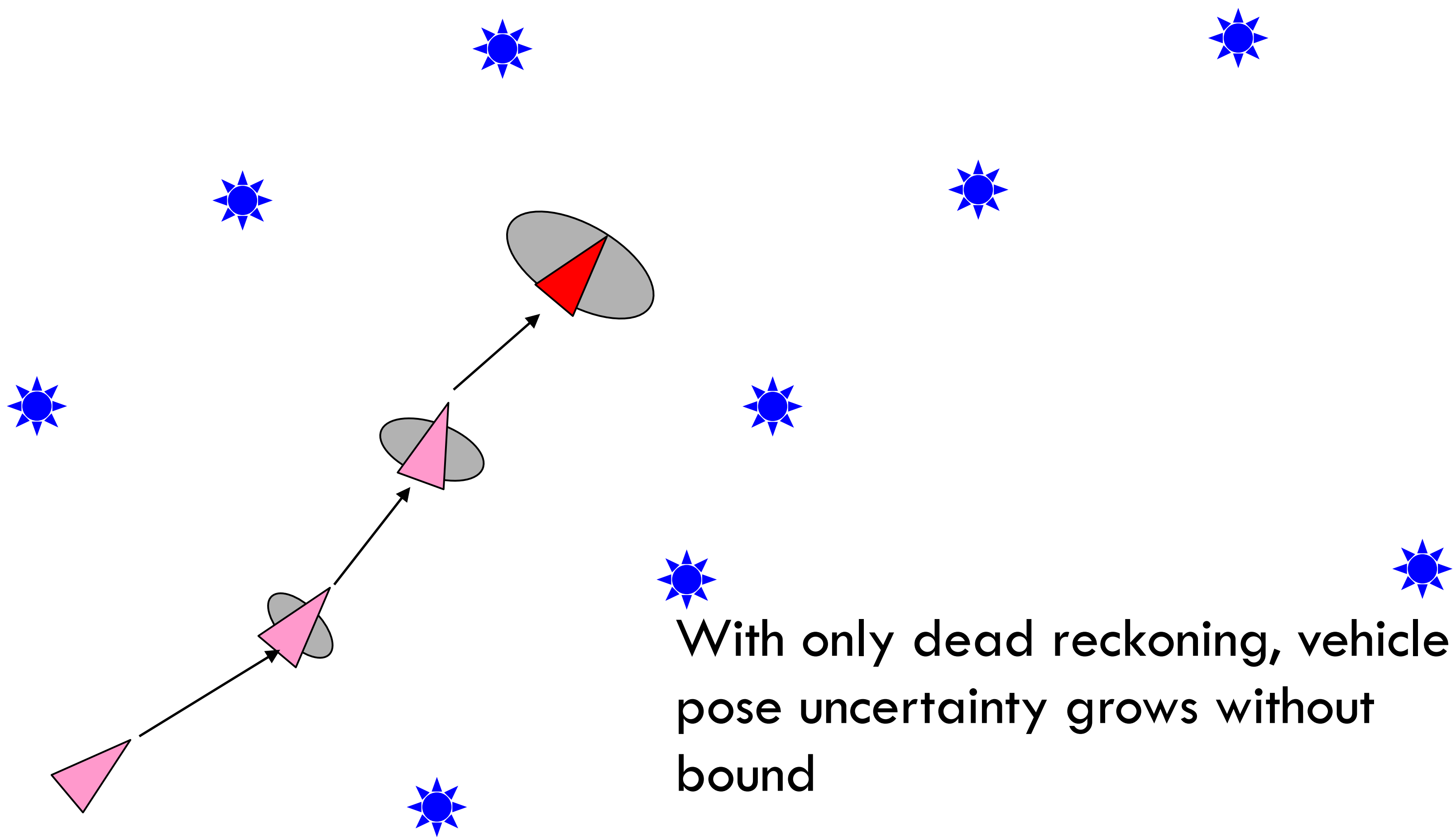
Illustration of SLAM without Landmarks



With only dead reckoning, vehicle pose uncertainty grows without bound

Courtesy J. Leonard

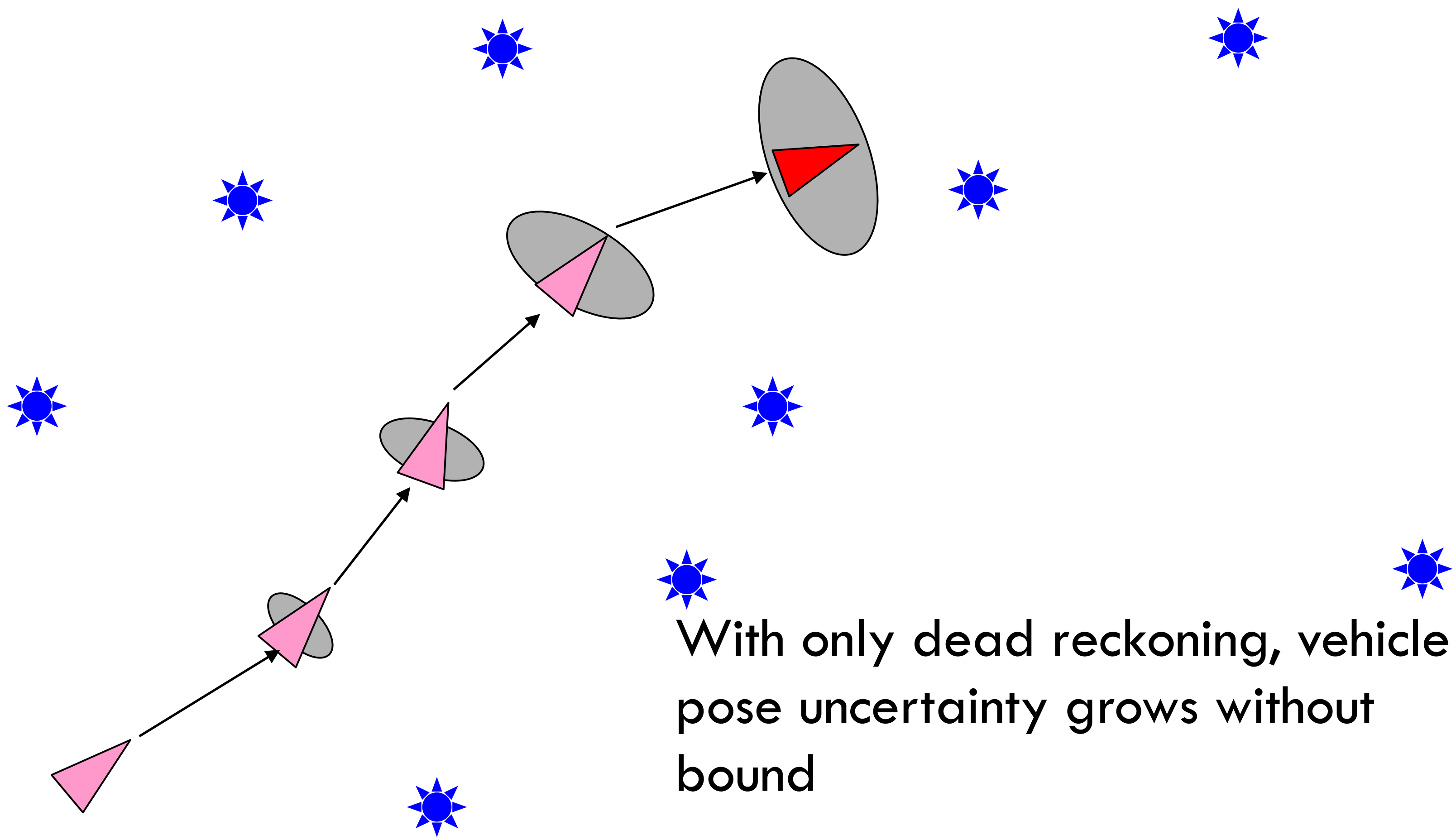
Illustration of SLAM without Landmarks



With only dead reckoning, vehicle pose uncertainty grows without bound

Courtesy J. Leonard

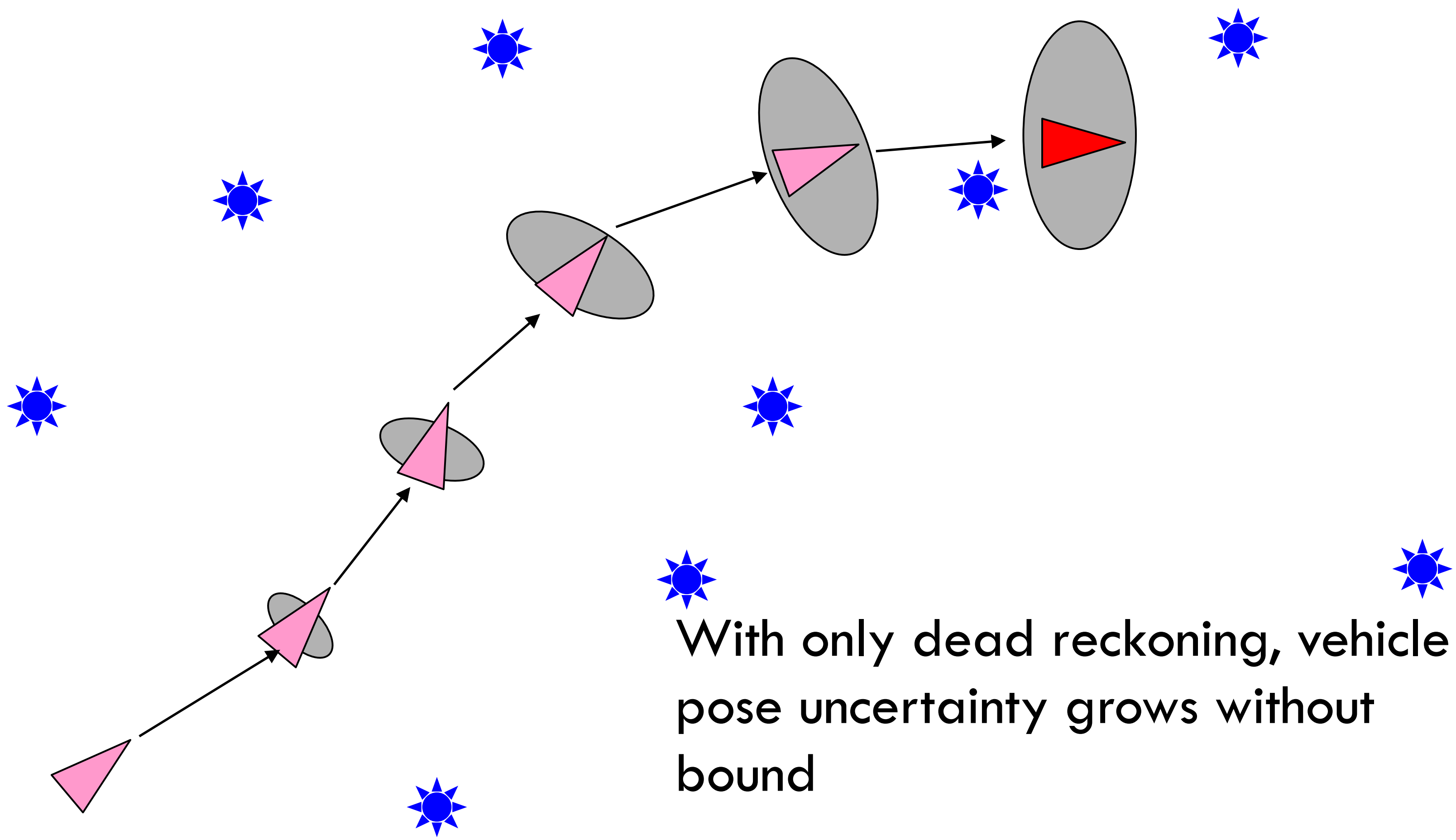
Illustration of SLAM without Landmarks



With only dead reckoning, vehicle pose uncertainty grows without bound

Courtesy J. Leonard

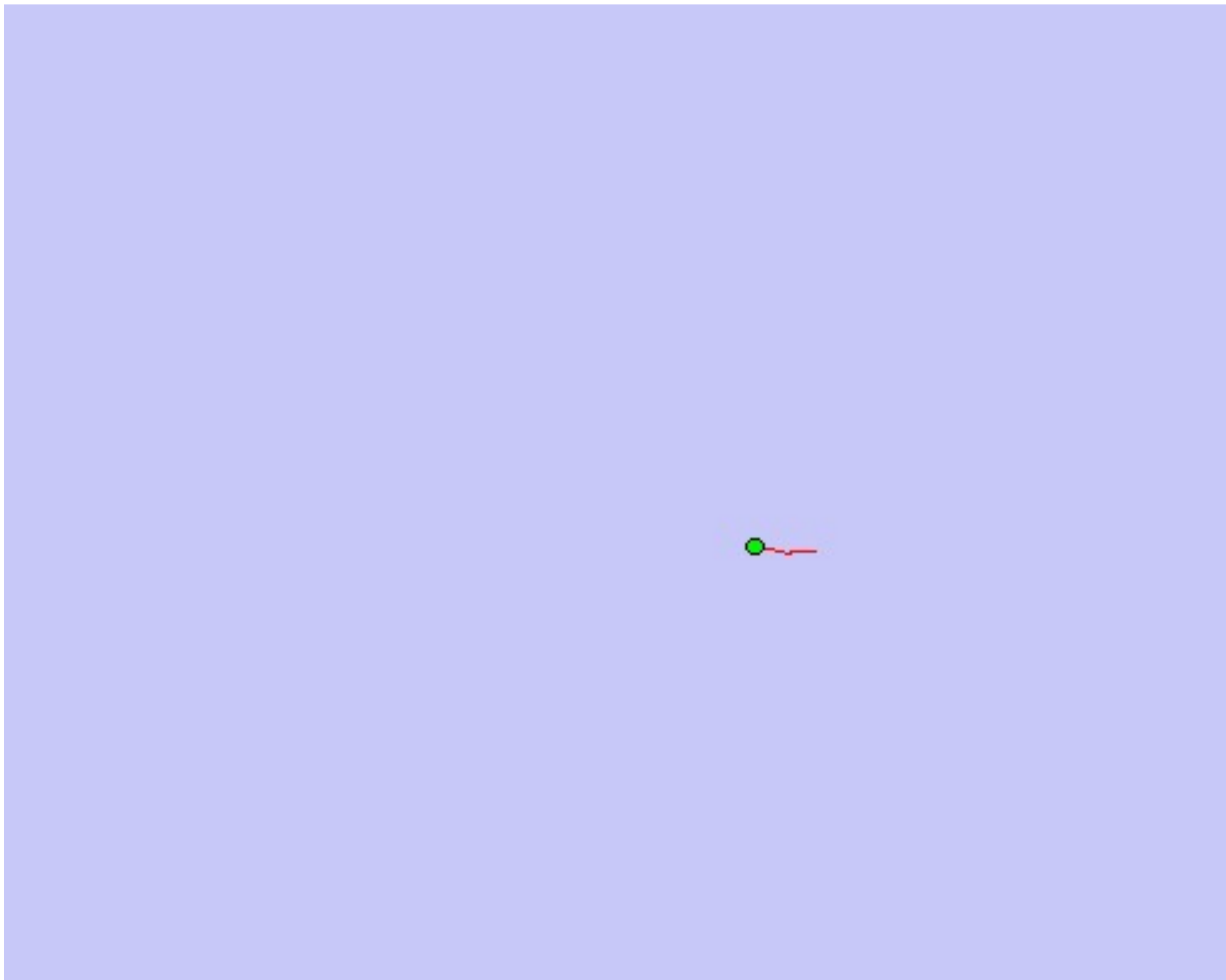
Illustration of SLAM without Landmarks



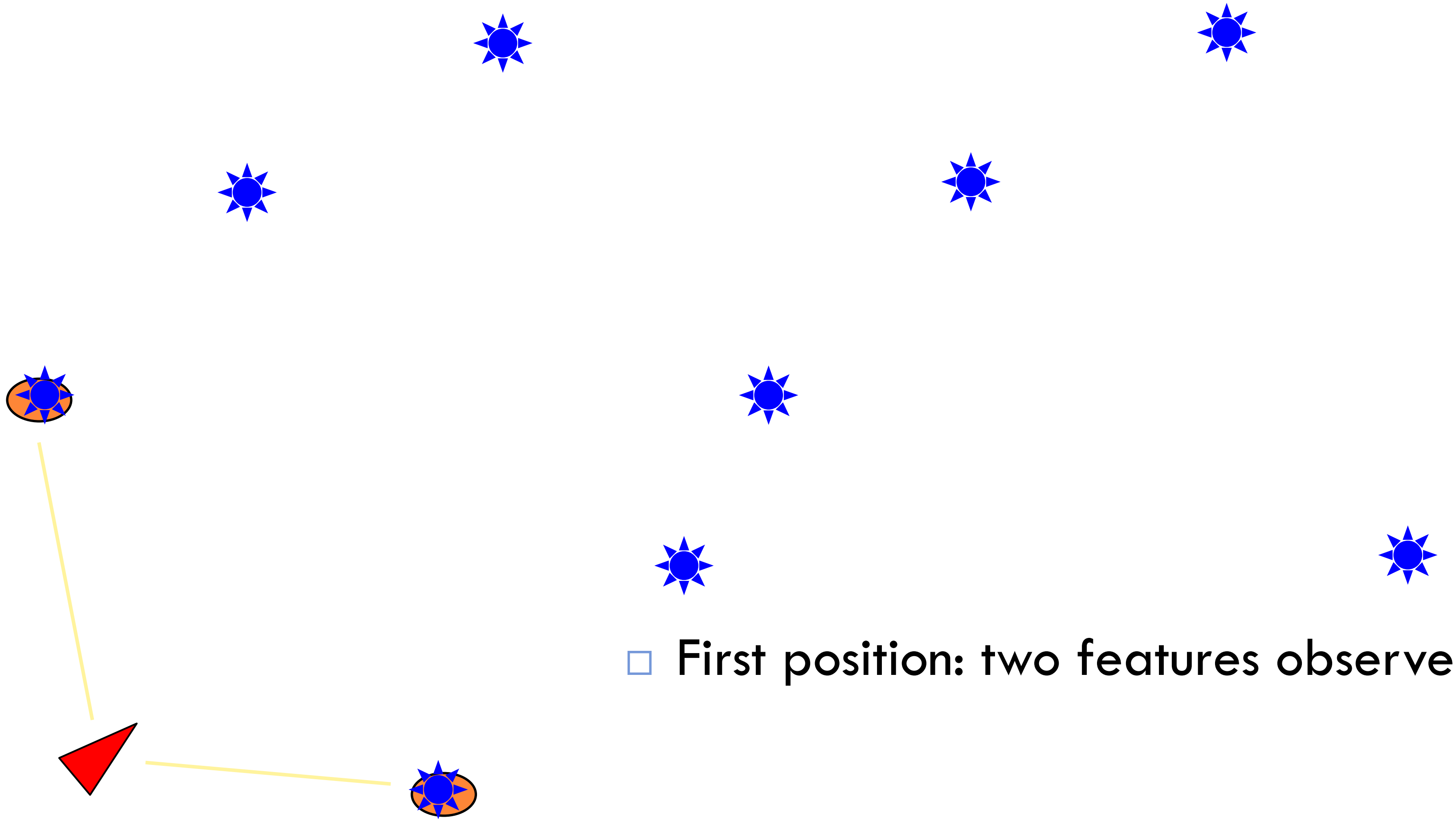
With only dead reckoning, vehicle pose uncertainty grows without bound

Courtesy J. Leonard

Mapping with Raw Odometry



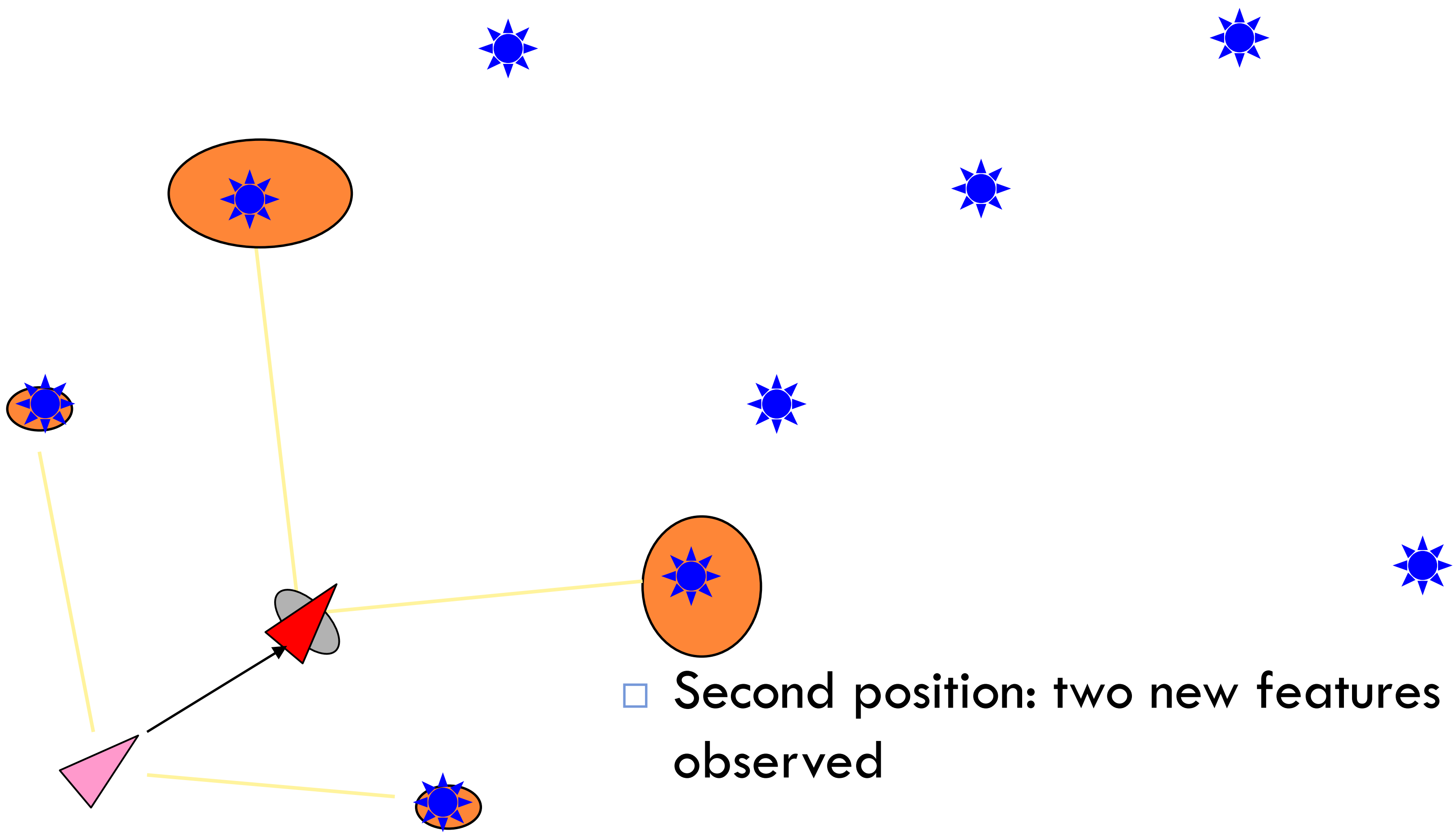
Repeat, with Measurements of Landmarks



- First position: two features observed

Courtesy J. Leonard

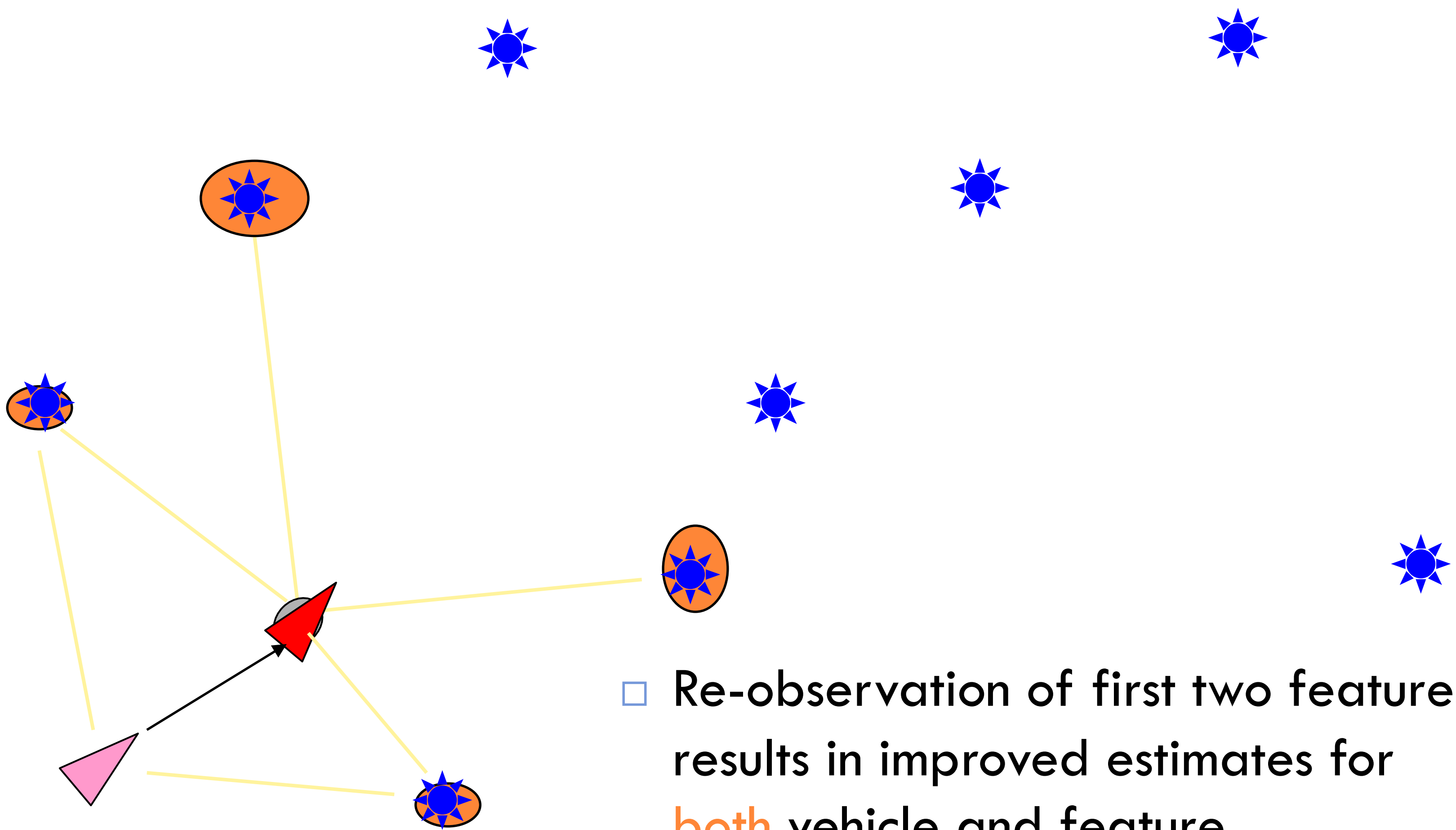
Illustration of SLAM with Landmarks



- Second position: two new features observed

Courtesy J. Leonard

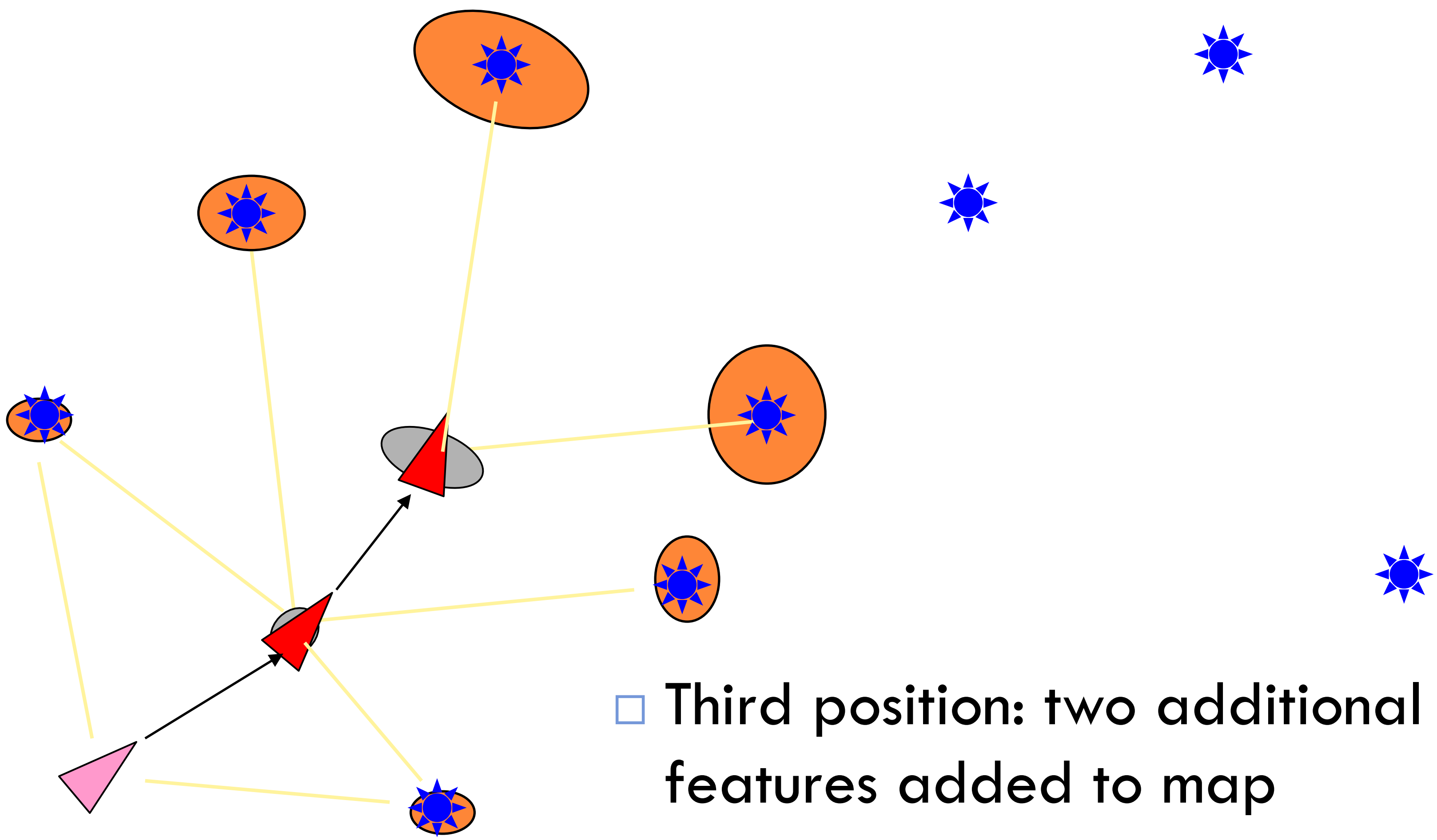
Illustration of SLAM with Landmarks



- Re-observation of first two features results in improved estimates for both vehicle <u>and</u> feature

Courtesy J. Leonard

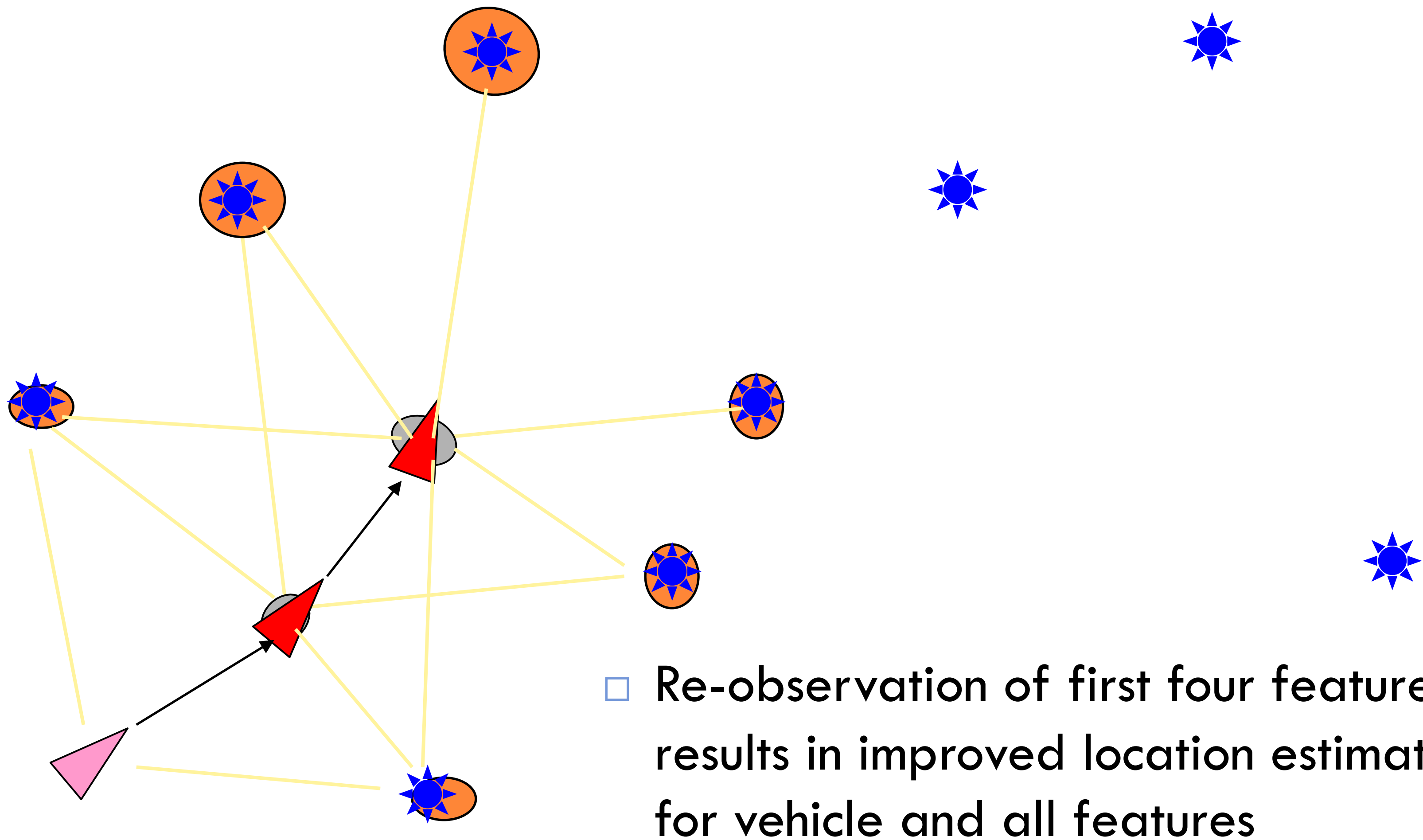
Illustration of SLAM with Landmarks



- Third position: two additional features added to map

Courtesy J. Leonard

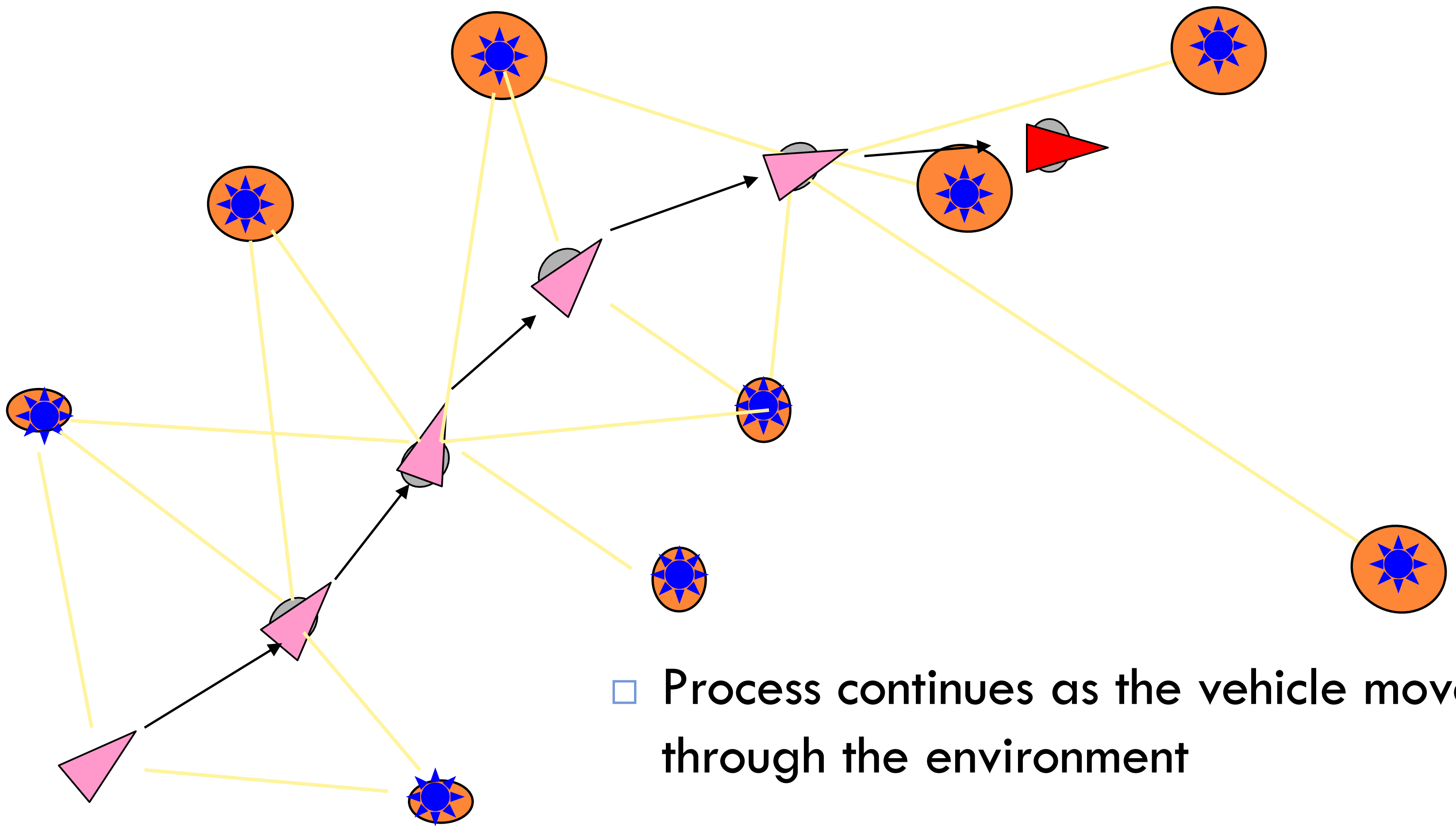
Illustration of SLAM with Landmarks



- Re-observation of first four features results in improved location estimates for vehicle and all features

Courtesy J. Leonard

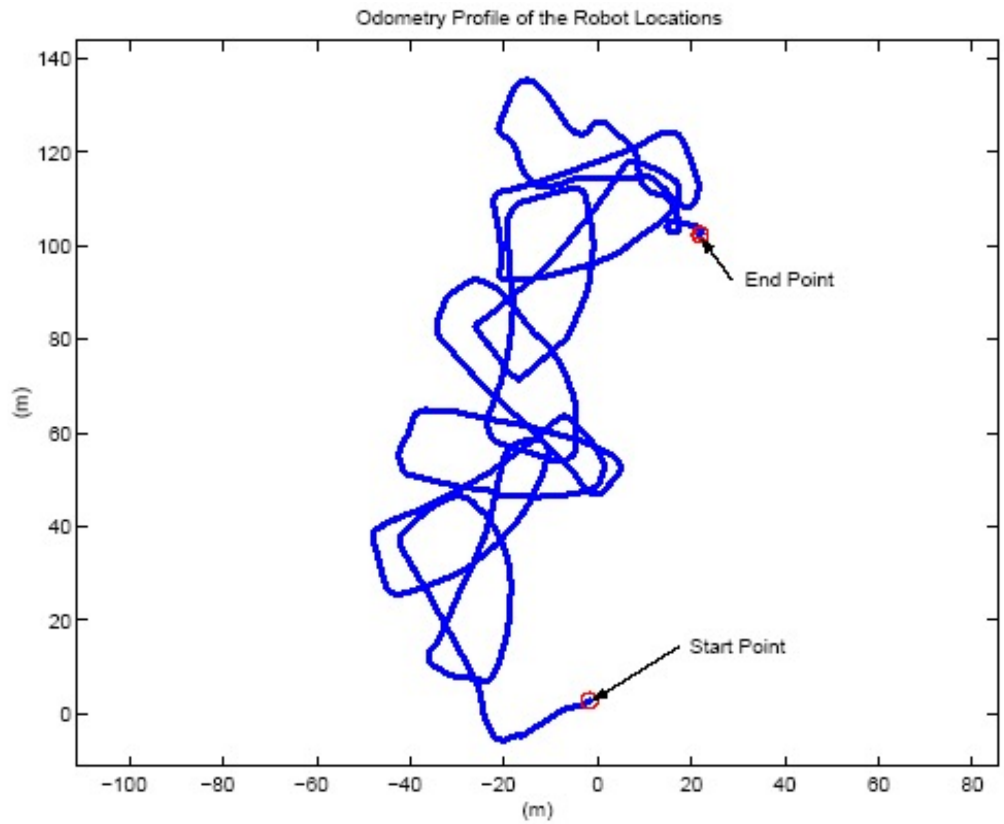
Illustration of SLAM with Landmarks



- Process continues as the vehicle moves through the environment

Courtesy J. Leonard

SLAM Using Landmarks



Courtesy J. Leonard

Test Environment (Point Landmarks)

Courtesy J. Leonard

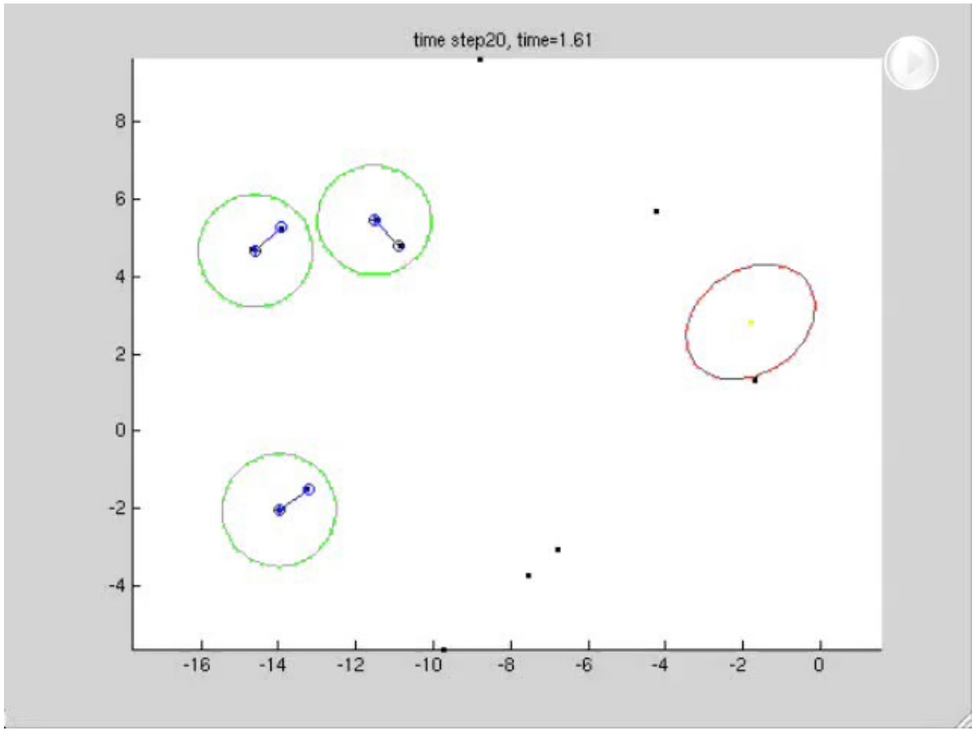
View from Vehicle

Courtesy J. Leonard

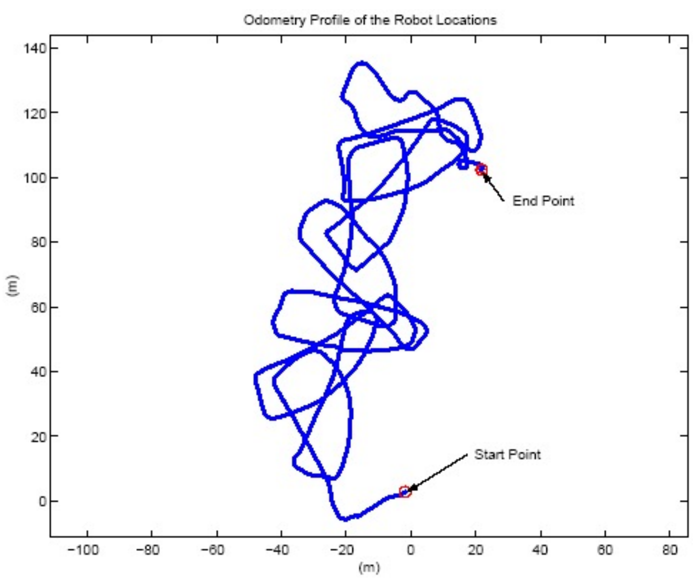
SLAM Using Landmarks

1. Move
2. Sense
3. Associate measurements with known features
4. Update state estimates for robot and previously mapped features
5. Find new features from unassociated measurements
6. Initialize new features
7. Repeat

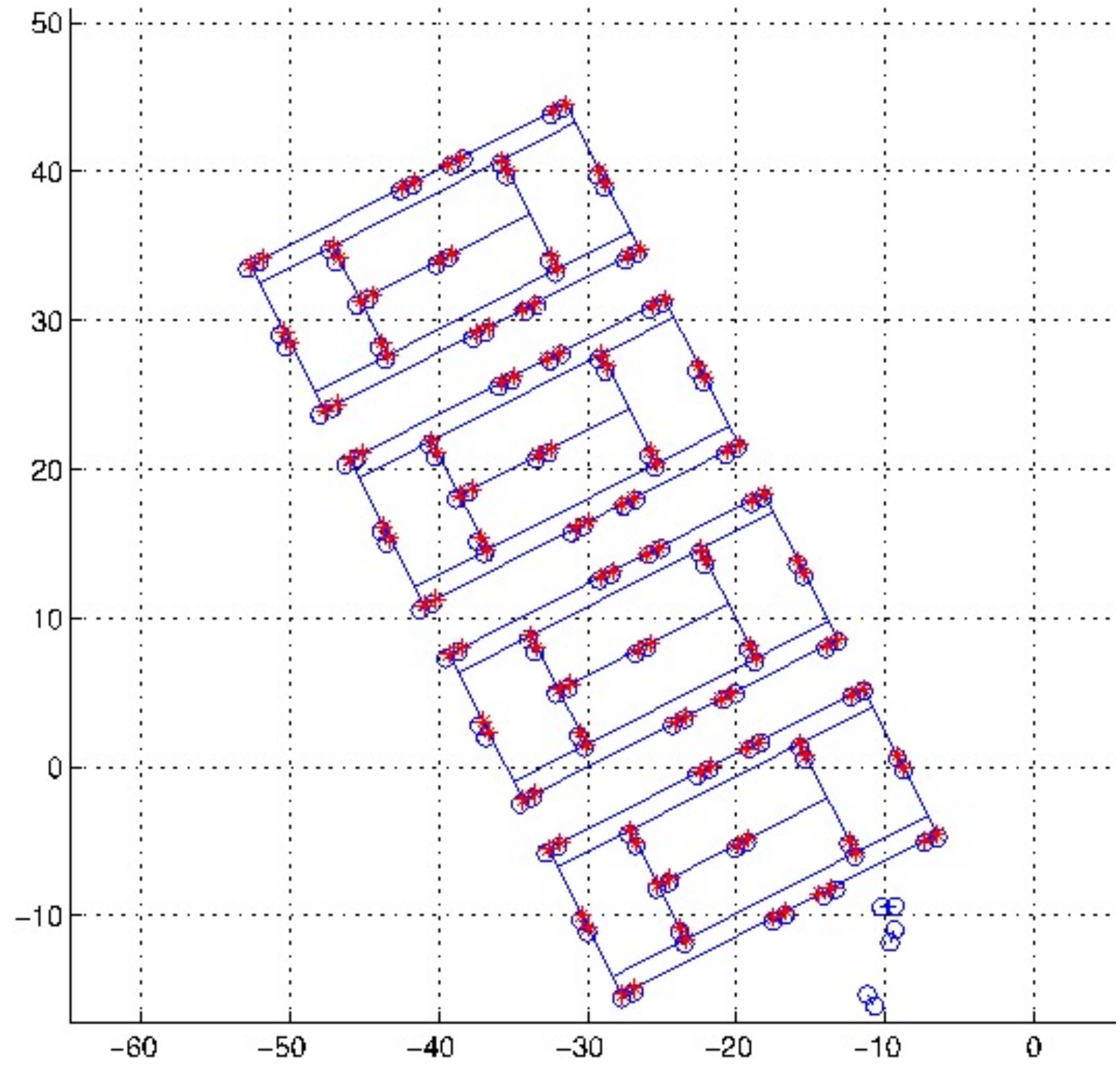
MIT Indoor Track



Comparison with Ground Truth



odometry

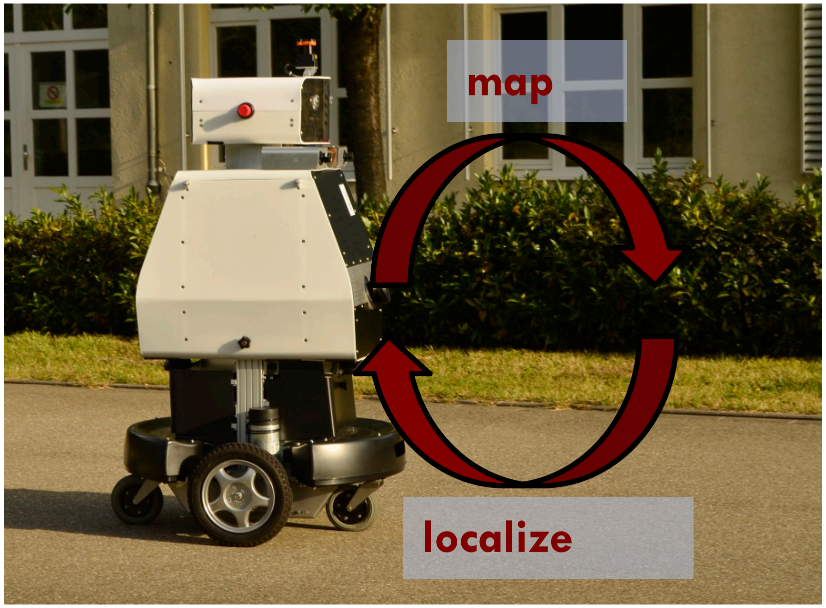


SLAM result

Courtesy J. Leonard

Simultaneous Localization and Mapping (SLAM)

- Building a map and locating the robot in the map at the same time
- Chicken-and-egg problem



Courtesy: Cyrill Stachniss

Definition of the SLAM Problem

Given

- The robot's controls

  $$u_{1:T} = \{u_1, u_2, u_3, \ldots, u_T\}$$

- Observations

  $$z_{1:T} = \{z_1, z_2, z_3, \ldots, z_T\}$$

Wanted

- Map of the environment

  $$m$$

- Path of the robot

  $$x_{0:T} = \{x_0, x_1, x_2, \ldots, x_T\}$$

Courtesy: Cyrill Stachniss

Three Main Paradigms

Kalman filter

Graph-based

Particle filter

Courtesy: Cyrill Stachniss

Bayes Filter

- Recursive filter with prediction and correction step

- Prediction

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\ bel(x_{t-1})\, dx_{t-1}$$

- Correction

$$bel(x_t) = \eta\, p(z_t \mid x_t)\ \overline{bel}(x_t)$$

Courtesy: Cyrill Stachniss

EKF SLAM

- Application of the EKF to SLAM
- Estimate robot's pose and locations of landmarks in the environment
- Assumption: known correspondences
- State space (for the 2D plane) is

$$x_t = (\underbrace{x, y, \theta}_{\text{robot's pose}}, \underbrace{m_{1,x}, m_{1,y}}_{\text{landmark 1}}, \ldots, \underbrace{m_{n,x}, m_{n,y}}_{\text{landmark n}})^T$$

Courtesy: Cyrill Stachniss

EKF SLAM: State Representation

- Map with n landmarks: (3+2n)-dimensional Gaussian
- Belief is represented by

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

Courtesy: Cyrill Stachniss

EKF SLAM: State Representation

- More compactly

$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \dots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \dots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \dots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

Courtesy: Cyrill Stachniss

EKF SLAM: State Representation

- Even more compactly (note: $x_R \longrightarrow x$)

$$\underbrace{\begin{pmatrix} x \\ m \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{xx} & \Sigma_{xm} \\ \Sigma_{mx} & \Sigma_{mm} \end{pmatrix}}_{\Sigma}$$

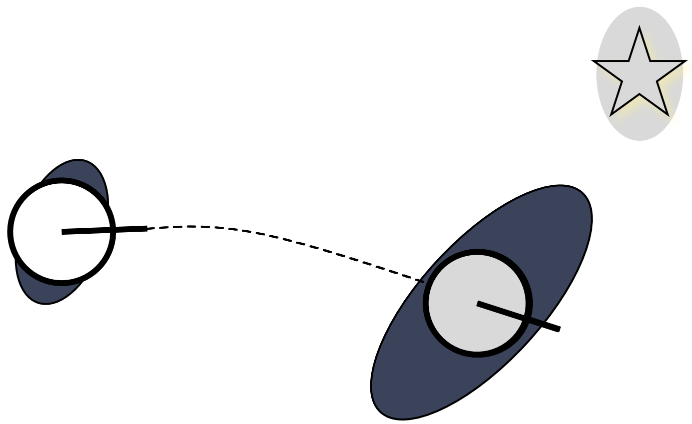
Courtesy: Cyrill Stachniss

EKF SLAM: Filter Cycle

1. State prediction
2. Measurement prediction
3. Measurement
4. Data association
5. Update

Courtesy: Cyrill Stachniss

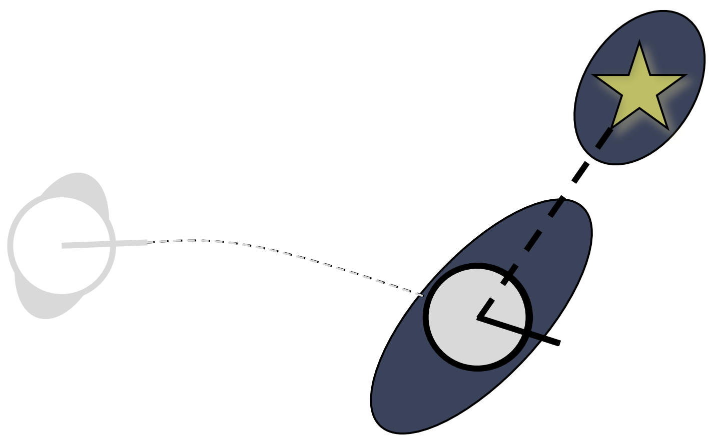
EKF SLAM: State Prediction



$$
\underbrace{\begin{pmatrix} \boxed{x_R} \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu}
\quad
\underbrace{\begin{pmatrix}
\Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\
\Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n}
\end{pmatrix}}_{\Sigma}
$$

Courtesy: Cyrill Stachniss

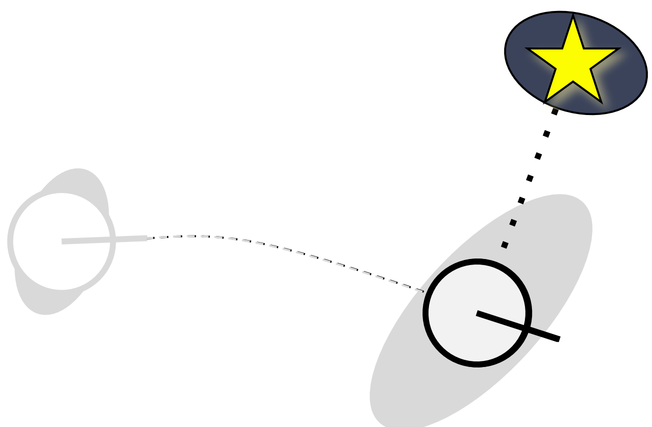
EKF SLAM: Measurement Prediction



$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

Courtesy: Cyrill Stachniss

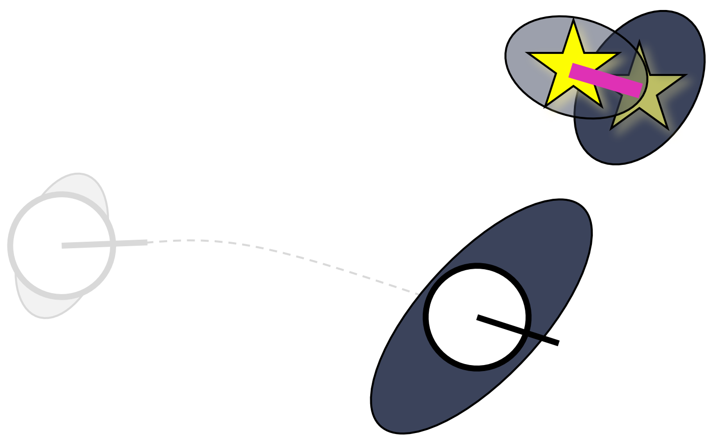
EKF SLAM: Obtained Measurement



$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

Courtesy: Cyrill Stachniss

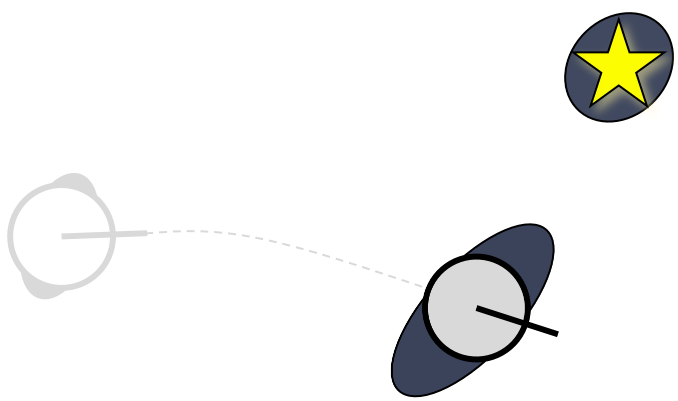
EKF SLAM: Data Association and Difference Between h(x) and z



$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

Courtesy: Cyrill Stachniss

EKF SLAM: Update Step



$$\underbrace{\begin{pmatrix} x_R \\ m_1 \\ \vdots \\ m_n \end{pmatrix}}_{\mu} \quad \underbrace{\begin{pmatrix} \Sigma_{x_R x_R} & \Sigma_{x_R m_1} & \cdots & \Sigma_{x_R m_n} \\ \Sigma_{m_1 x_R} & \Sigma_{m_1 m_1} & \cdots & \Sigma_{m_1 m_n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{m_n x_R} & \Sigma_{m_n m_1} & \cdots & \Sigma_{m_n m_n} \end{pmatrix}}_{\Sigma}$$

Courtesy: Cyrill Stachniss

EKF SLAM: Concrete Example

Setup

- Robot moves in the 2D plane
- Velocity-based motion model
- Robot observes point landmarks
- Range-bearing sensor
- Known data association
- Known number of landmarks

Courtesy: Cyrill Stachniss

Initialization

- Robot starts in its own reference frame (all landmarks unknown)
- 2N+3 dimensions

$$\mu_0 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \end{pmatrix}^T$$

$$\Sigma_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \infty & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \infty \end{pmatrix}$$

Courtesy: Cyrill Stachniss

Extended Kalman Filter Algorithm

1: $\textbf{Extended\_Kalman\_filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\textbf{:}$

2: $\bar{\mu}_t = g(u_t, \mu_{t-1})$

3: $\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$

4: $K_t = \bar{\Sigma}_t \, H_t^T (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1}$

5: $\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$

6: $\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

7: $return \ \mu_t, \Sigma_t$

Courtesy: Cyrill Stachniss

Prediction Step (Motion)

- Goal: Update state space based on the robot's motion
- Robot motion in the plane

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \underbrace{\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}}_{g_{x,y,\theta}(u_t,(x,y,\theta)^T)}$$

- How to map that to the 2N+3 dim space?

Courtesy: Cyrill Stachniss

Update the State Space

- From the motion in the plane

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}$$

- to the 2N+3 dimensional space

$$\begin{pmatrix} x' \\ y' \\ \theta' \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} x \\ y \\ \theta \\ \vdots \end{pmatrix} + \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & \underbrace{0 \dots 0}_{2N cols} \end{pmatrix}^T}_{F_x^T} \begin{pmatrix} -\frac{v_t}{\omega_t}\sin\theta + \frac{v_t}{\omega_t}\sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t}\cos\theta - \frac{v_t}{\omega_t}\cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}}_{g(u_t, x_t)}$$

Courtesy: Cyrill Stachniss

Extended Kalman Filter Algorithm

$$\textbf{Extended\_Kalman\_filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\textbf{:}$$

$$\cancel{\bar{\mu}_t = g(u_t, \mu_{t-1})}$$ **DONE**

$$\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$$

$$K_t = \bar{\Sigma}_t \, H_t^T (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$$

$$\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$$

$$\mathit{return} \; \mu_t, \Sigma_t$$

Courtesy: Cyrill Stachniss

Update Covariance

- The function g only affects the robot's motion and not the landmarks

Jacobian of the motion (3x3)

$$G_t = \begin{pmatrix} G_t^x & 0 \\ 0 & I \end{pmatrix}$$

Identity (2N x 2N)

Courtesy: Cyrill Stachniss

This Leads to the Time Propagation

1: $\textbf{Extended\_Kalman\_filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

2: $\cancel{\bar{\mu}_t = g(u_t, \mu_{t-1})}$ **Apply & DONE**

3: $\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$

$$
\begin{aligned}
\bar{\Sigma}_t &= G_t \Sigma_{t-1} G_t^T + R_t \\
&= \begin{pmatrix} G_t^x & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \Sigma_{xx} & \Sigma_{xm} \\ \Sigma_{mx} & \Sigma_{mm} \end{pmatrix} \begin{pmatrix} (G_t^x)^T & 0 \\ 0 & I \end{pmatrix} + R_t \\
&= \begin{pmatrix} G_t^x \Sigma_{xx} (G_t^x)^T & G_t^x \Sigma_{xm} \\ (G_t^x \Sigma_{xm})^T & \Sigma_{mm} \end{pmatrix} + R_t
\end{aligned}
$$

Courtesy: Cyrill Stachniss

Extended Kalman Filter Algorithm

$$\textbf{Extended\_Kalman\_filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\textbf{:}$$

$$\cancel{\bar{\mu}_t = g(u_t, \mu_{t-1})}$$ **DONE**

$$\cancel{\bar{\Sigma}_t = G_t\, \Sigma_{t-1}\, G_t^T + R_t}$$ **DONE**

$$K_t = \bar{\Sigma}_t\, H_t^T (H_t\, \bar{\Sigma}_t\, H_t^T + Q_t)^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$$

$$\Sigma_t = (I - K_t\, H_t)\, \bar{\Sigma}_t$$

$$return\ \mu_t, \Sigma_t$$

Courtesy: Cyrill Stachniss

EKF SLAM: Correction Step

- Known data association
- $c_t^i = j$: i-th measurement at time t observes the landmark with index j
- Initialize landmark if unobserved
- Compute the expected observation
- Compute the Jacobian of h
- Proceed with computing the Kalman gain

Courtesy: Cyrill Stachniss

Range-Bearing Observation

- Range-Bearing observation $z_t^i = (r_t^i, \phi_t^i)^T$
- If landmark has not been observed

$$\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{t,x} \\ \bar{\mu}_{t,y} \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\phi_t^i + \bar{\mu}_{t,\theta}) \\ r_t^i \sin(\phi_t^i + \bar{\mu}_{t,\theta}) \end{pmatrix}$$

observed location of landmark j

estimated robot's location

relative measurement

Courtesy: Cyrill Stachniss

Jacobian for the Observation

- Based on

$$\begin{aligned} \delta &= \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \bar{\mu}_{j,x} - \bar{\mu}_{t,x} \\ \bar{\mu}_{j,y} - \bar{\mu}_{t,y} \end{pmatrix} \\ q &= \delta^T \delta \\ \hat{z}_t^i &= \begin{pmatrix} \sqrt{q} \\ \operatorname{atan2}(\delta_y, \delta_x) - \bar{\mu}_{t,\theta} \end{pmatrix} \end{aligned}$$

- Compute the Jacobian

$$\begin{aligned} {}^{\text{low}}H_t^i &= \frac{\partial h(\bar{\mu}_t)}{\partial \bar{\mu}_t} \\ &= \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & +\sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{pmatrix} \end{aligned}$$

Courtesy: Cyrill Stachniss

Jacobian for the Observation

- Use the computed Jacobian

$$
{}^{\text{low}}H_t^i = \frac{1}{q}\begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & +\sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{pmatrix}
$$

- map it to the high dimensional space

$$
H_t^i = {}^{\text{low}}H_t^i \; F_{x,j}
$$

$$
F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 1 & 0 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 1 & 0\cdots0 & 0 & 0 & 0\cdots0 \\ 0 & 0 & 0 & 0\cdots0 & 1 & 0 & 0\cdots0 \\ 0 & 0 & 0 & \underbrace{0\cdots0}_{2j-2} & 0 & 1 & \underbrace{0\cdots0}_{2N-2j} \end{pmatrix}
$$

Courtesy: Cyrill Stachniss

Next Steps as Specified…

1: **Extended_Kalman_filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$**:**

2: $\cancel{\bar{\mu}_t = g(u_t, \mu_{t-1})}$ **DONE**

3: $\cancel{\bar{\Sigma}_t = G_t\, \Sigma_{t-1}\, G_t^T + R_t}$ **DONE**

4: ➡ $K_t = \bar{\Sigma}_t\, H_t^T (H_t\, \bar{\Sigma}_t\, H_t^T + Q_t)^{-1}$

5: $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

6: $\Sigma_t = (I - K_t\, H_t)\, \bar{\Sigma}_t$

7: $return\ \mu_t, \Sigma_t$

Courtesy: Cyrill Stachniss

Extended Kalman Filter Algorithm

$\textbf{Extended\_Kalman\_filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$**:**

$\cancel{\bar{\mu}_t = g(u_t, \mu_{t-1})}$ **DONE**

$\cancel{\bar{\Sigma}_t = G_t\, \Sigma_{t-1}\, G_t^T + R_t}$ **DONE**

$\cancel{K_t = \bar{\Sigma}_t\, H_t^T (H_t\, \bar{\Sigma}_t\, H_t^T + Q_t)^{-1}}$ **Apply & DONE**

$\cancel{\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))}$ **Apply & DONE**

$\cancel{\Sigma_t = (I - K_t\, H_t)\, \bar{\Sigma}_t}$ **Apply & DONE**

return μ_t, Σ_t

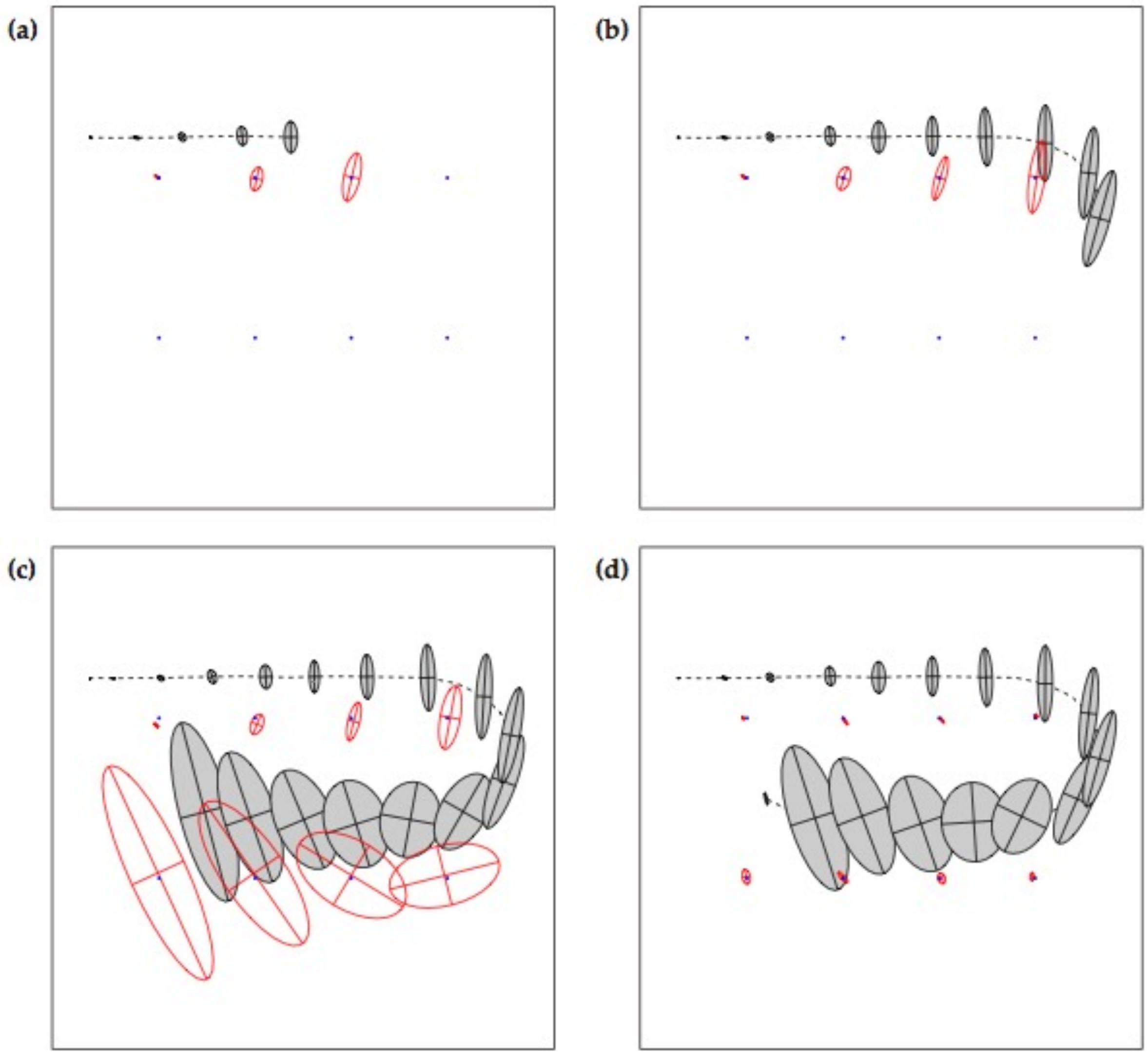
Courtesy: Cyrill Stachniss

EKF SLAM Complexity

- Cubic complexity depends only on the measurement dimensionality
- Cost per step: dominated by the number of landmarks: $O(n^2)$
- Memory consumption: $O(n^2)$
- The EKF becomes computationally intractable for large maps!

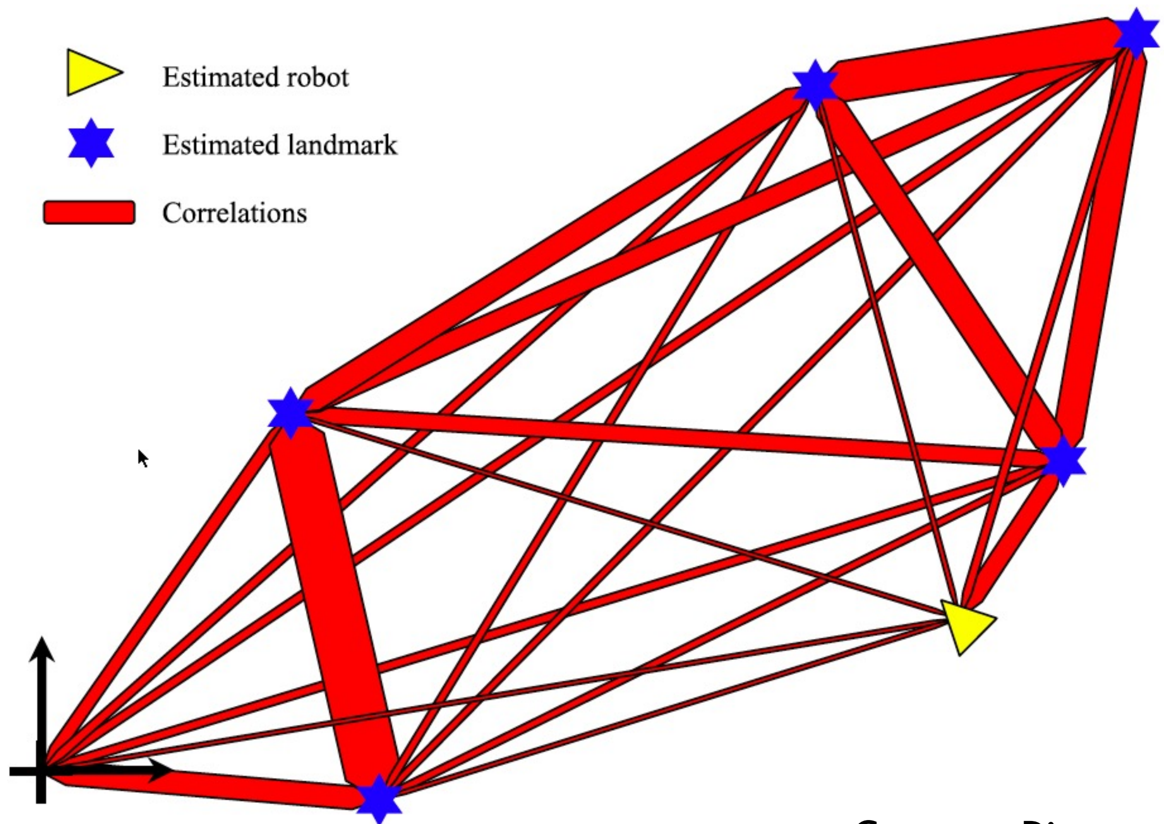
Courtesy: Cyrill Stachniss

Online SLAM Example



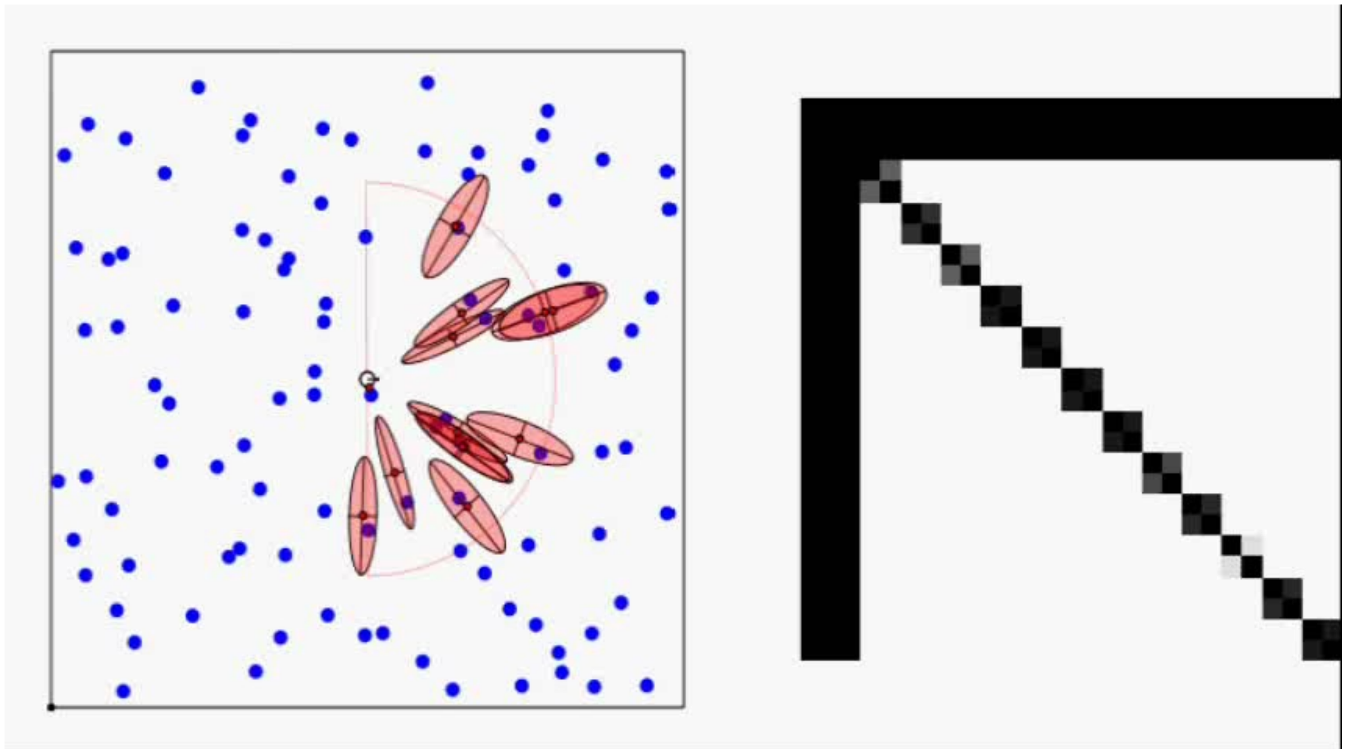
EKF SLAM Correlations

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



Courtesy: Dissanayake

EKF SLAM Correlations

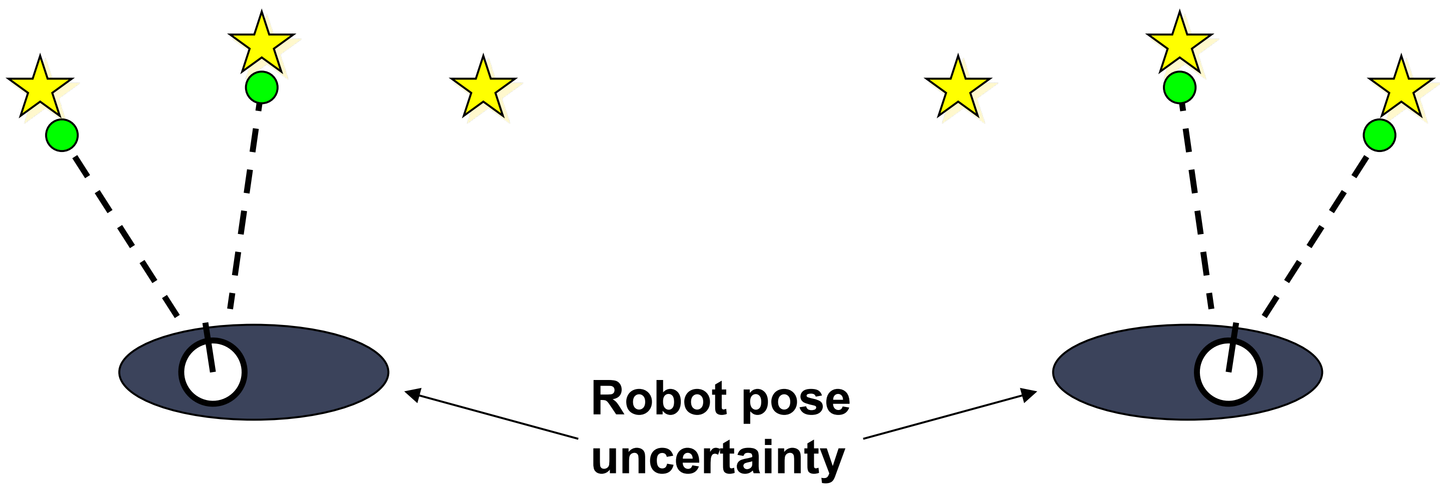


Blue path = true path   **Red path** = estimated path   **Black path** = odometry

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

Courtesy: M. Montemerlo

Data Association in SLAM



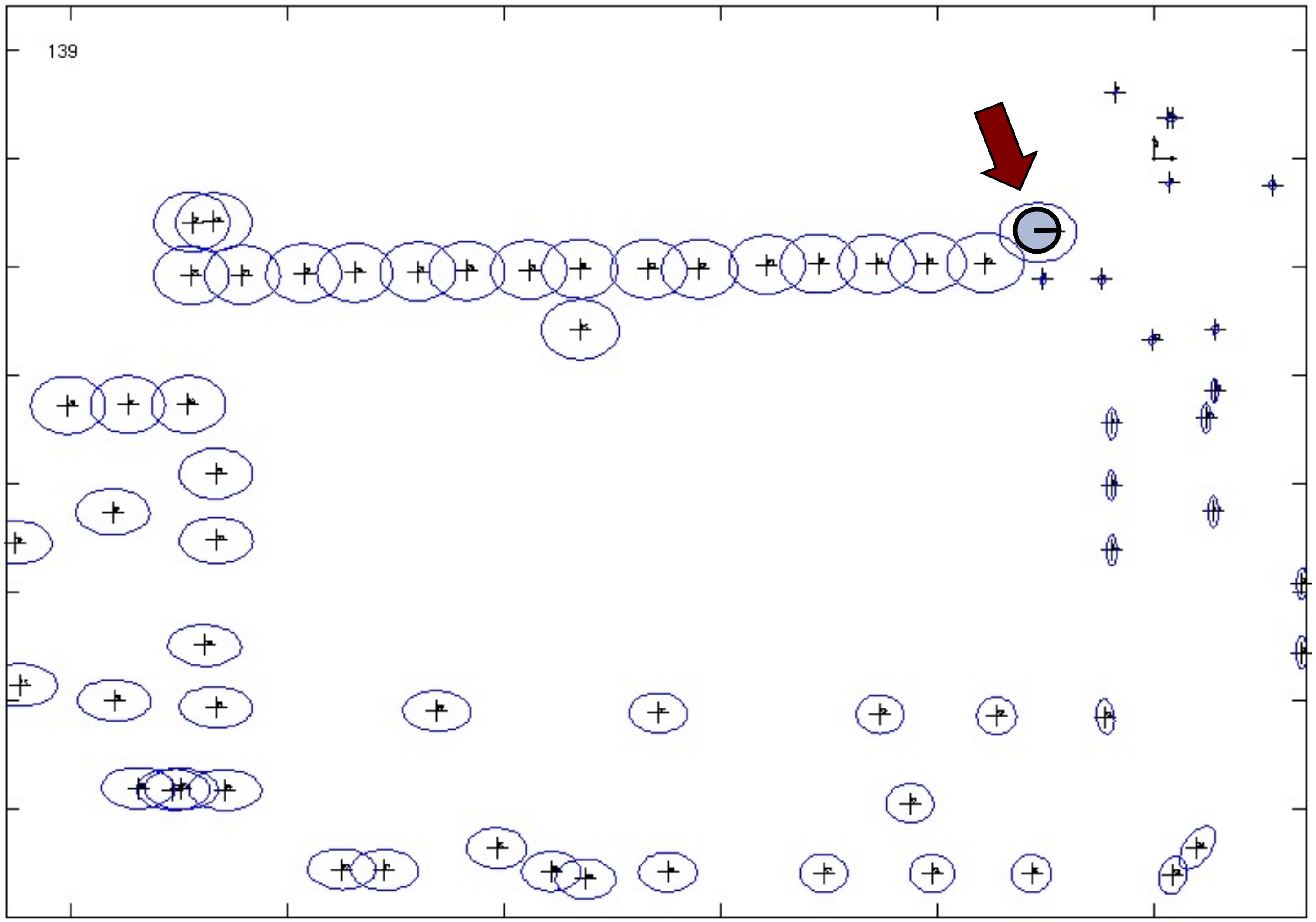
- In the real world, the mapping between observations and landmarks is unknown
- Picking wrong data associations can have catastrophic consequences
  - EKF SLAM is brittle in this regard
- Pose error correlates data associations

Loop-Closing

- Loop-closing means recognizing an already mapped area
- Data association under
  - high ambiguity
  - possible environment symmetries
- Uncertainties **collapse** after a loop-closure (whether the closure was correct or not)

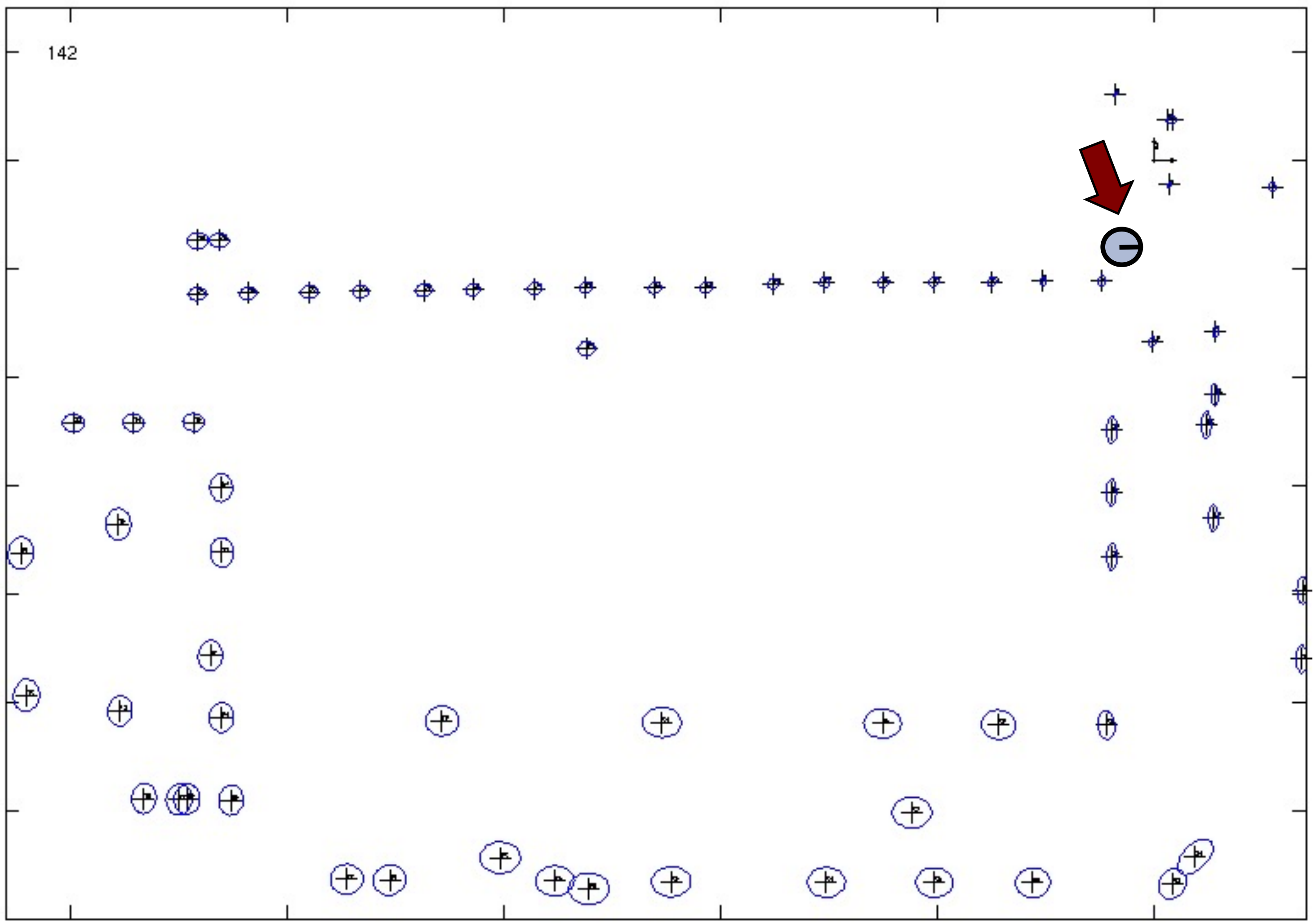
Courtesy: Cyrill Stachniss

Before the Loop-Closure



Courtesy: K. Arras

After the Loop-Closure



Courtesy: K. Arras

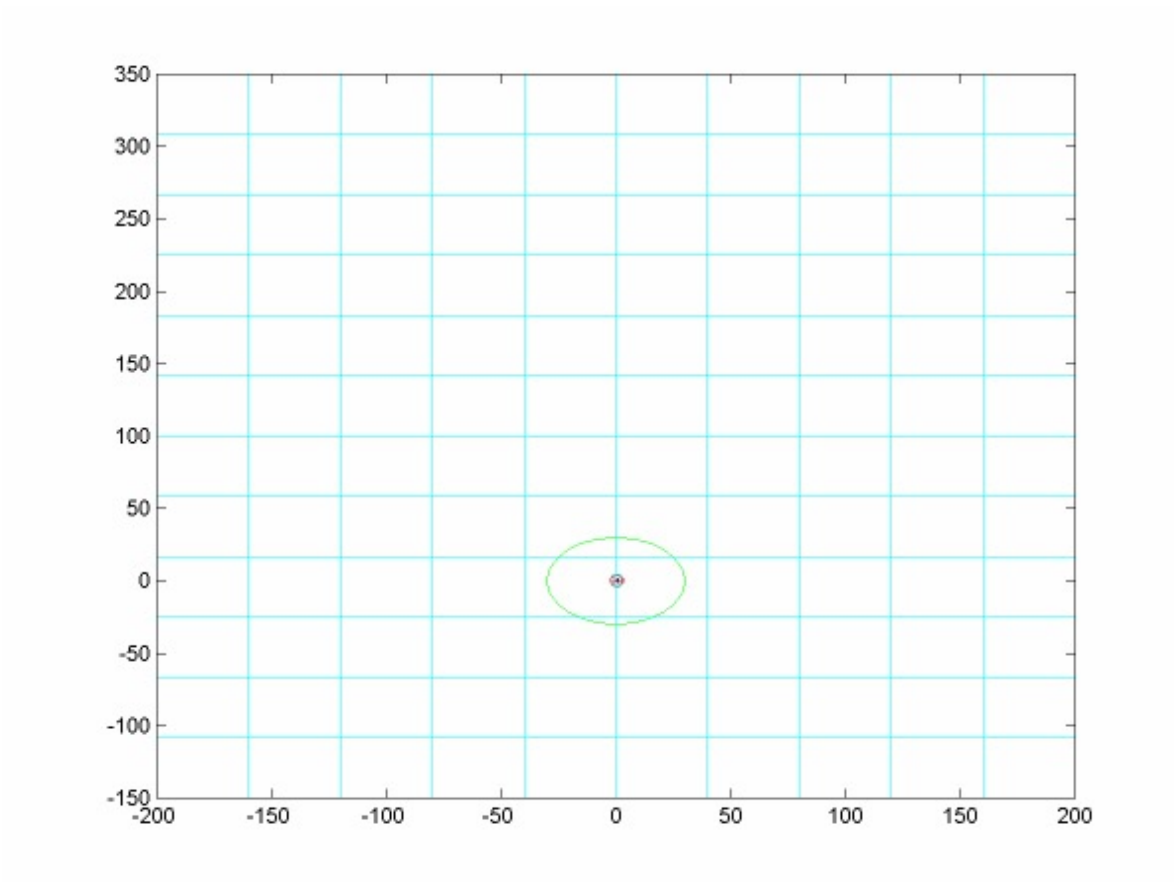
Example: Victoria Park Dataset

Courtesy: E. Nebot

Victoria Park: Data Acquisition

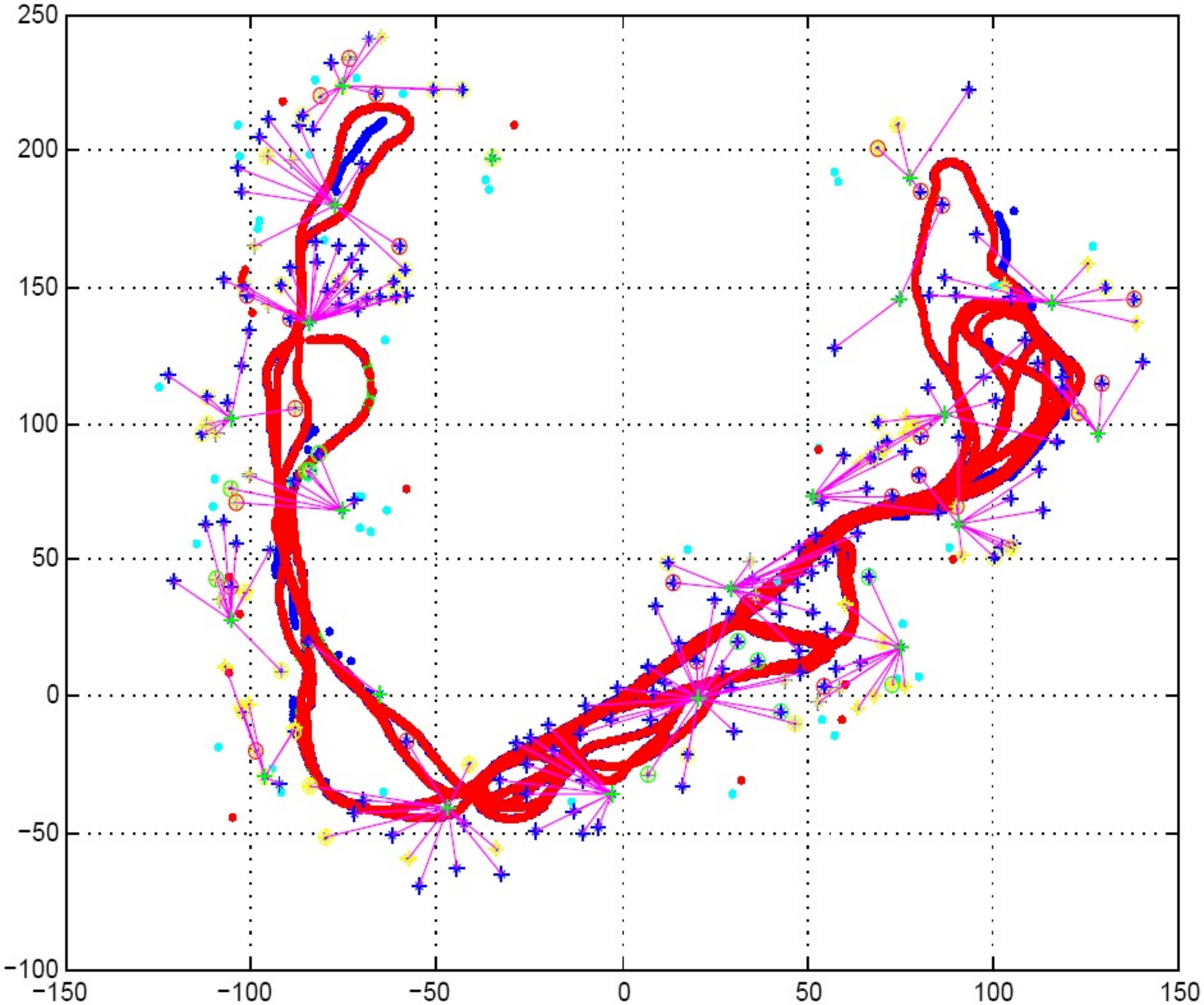
Courtesy: E. Nebot

Victoria Park: EKF Estimate



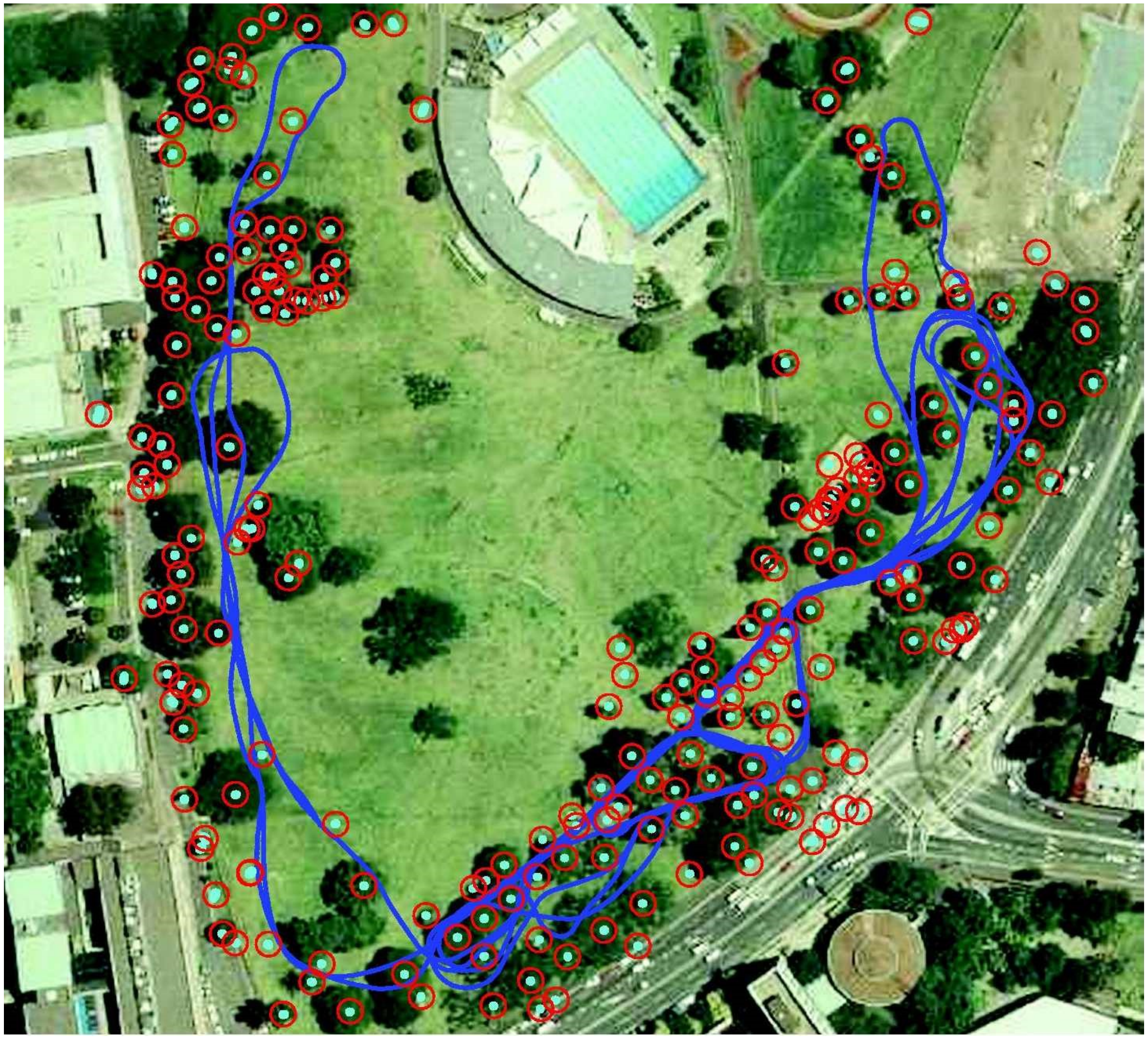
Courtesy: E. Nebot

Victoria Park: EKF Estimate



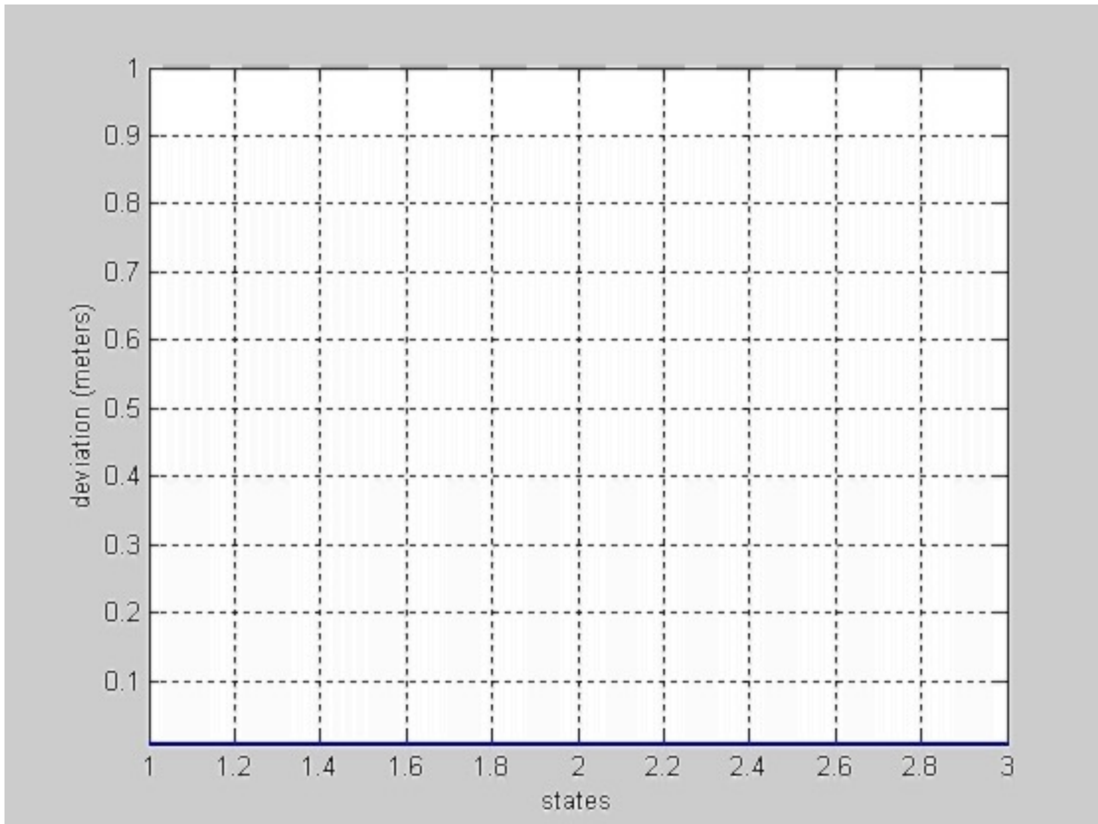
Courtesy: E. Nebot

Victoria Park: Landmarks



Courtesy: E. Nebot

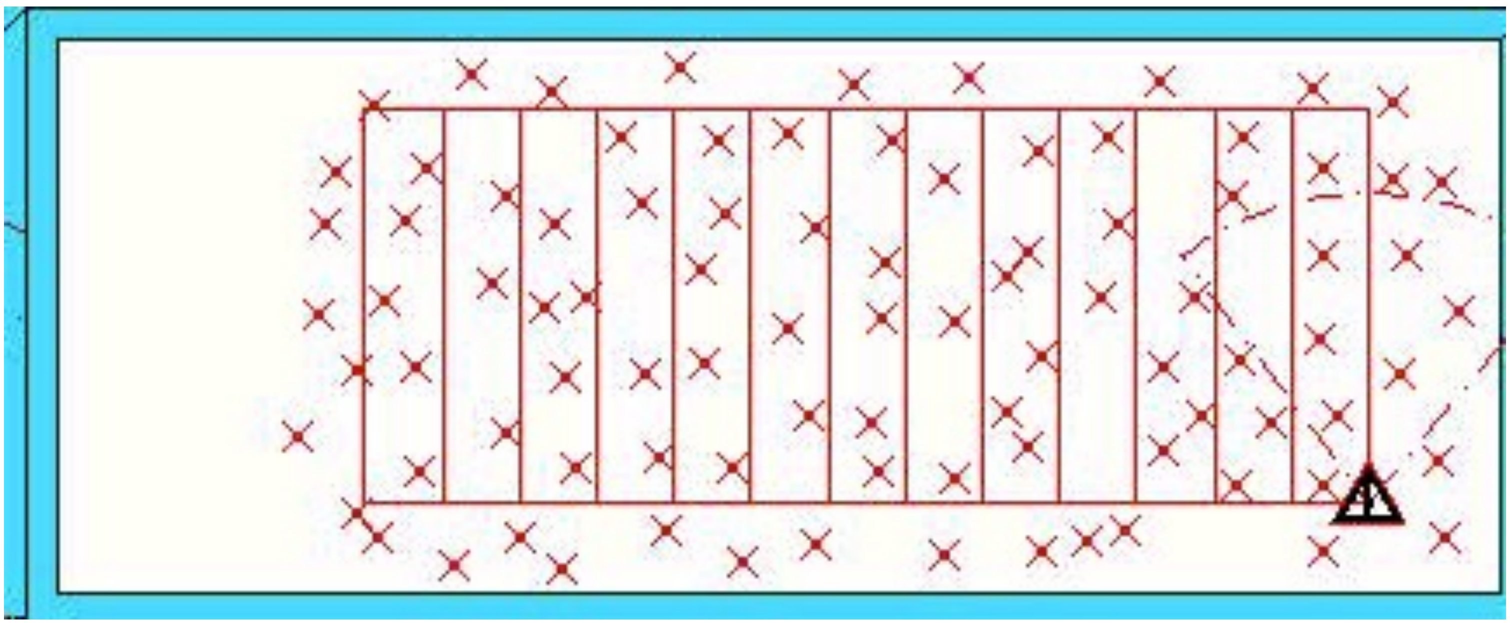
Victoria Park: Landmark Covariance



Courtesy: E. Nebot

Andrew Davison: MonoSLAM

Sub-maps for EKF SLAM



[Leonard et al 1998]

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.

Literature

EKF SLAM

- "Probabilistic Robotics", Chapter 10
- Smith, Self, & Cheeseman: "Estimating Uncertain Spatial Relationships in Robotics"
- Dissanayake et al.: "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem"
- Durrant-Whyte & Bailey: "SLAM Part 1" and "SLAM Part 2" tutorials

Courtesy: Cyrill Stachniss

Graph-SLAM

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

Information Form

- Represent posterior in canonical form

$$\Omega = \Sigma^{-1} \quad \text{Information matrix}$$

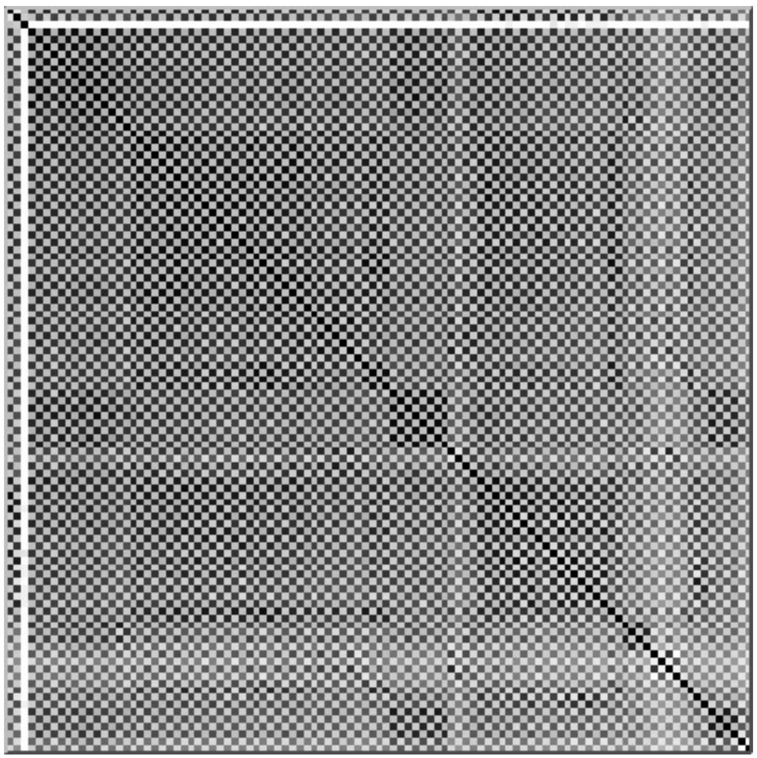
$$\xi = \Sigma^{-1}\mu \quad \text{Information vector}$$

- One-to-one transform between canonical and moment representation

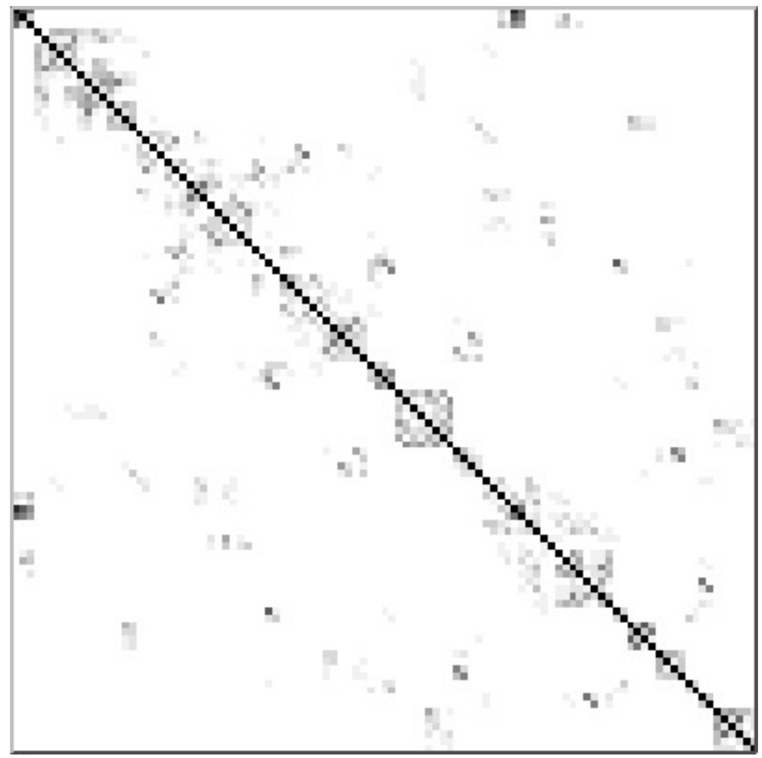
$$\Sigma = \Omega^{-1}$$

$$\mu = \Omega^{-1}\xi$$

Information vs. Moment Form

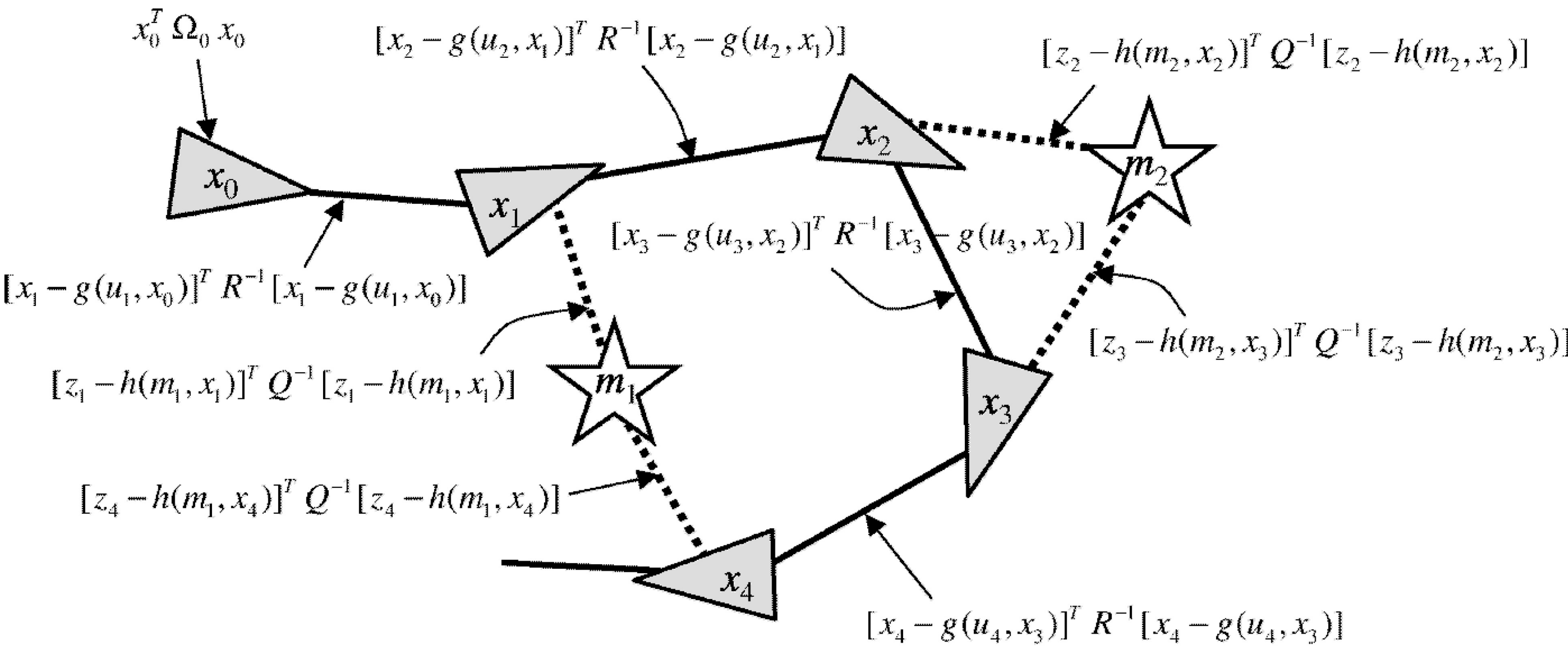


Correlation matrix



Information matrix

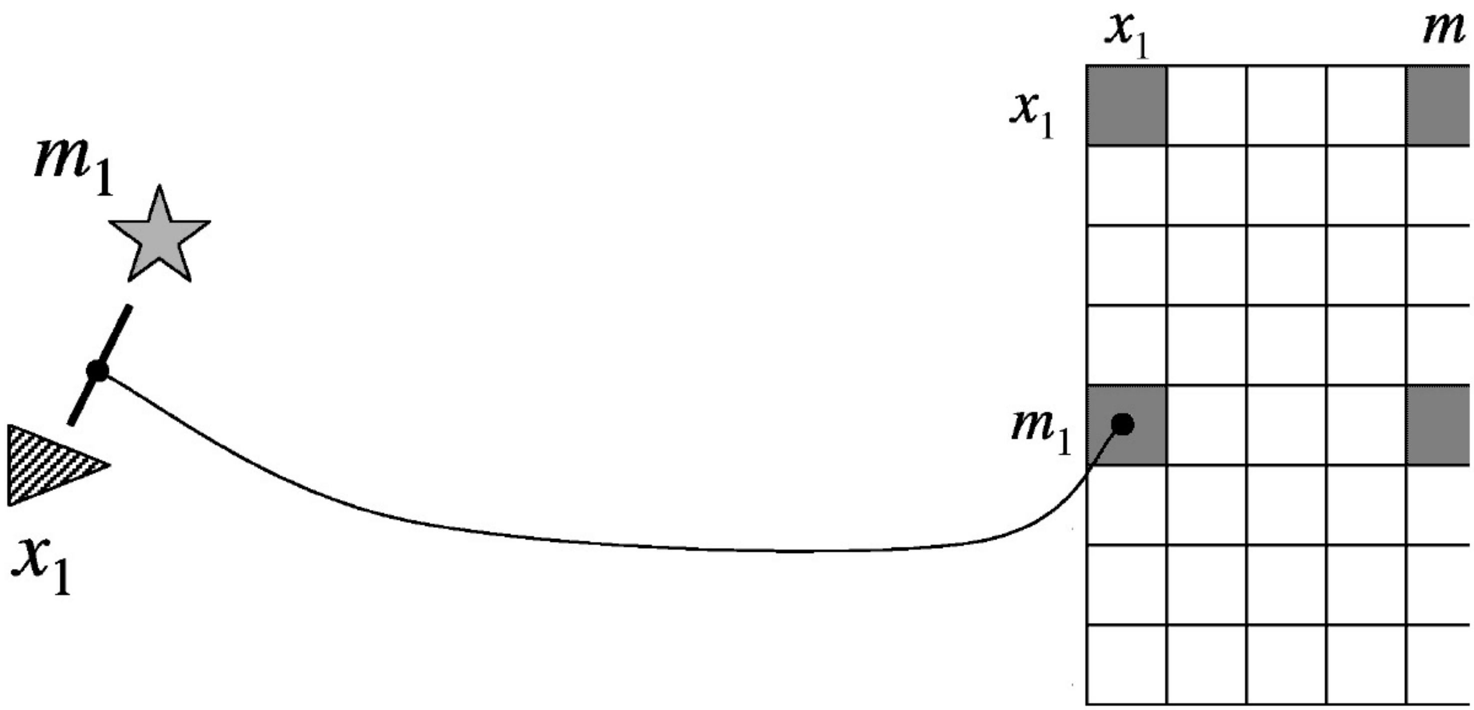
Graph-SLAM Idea



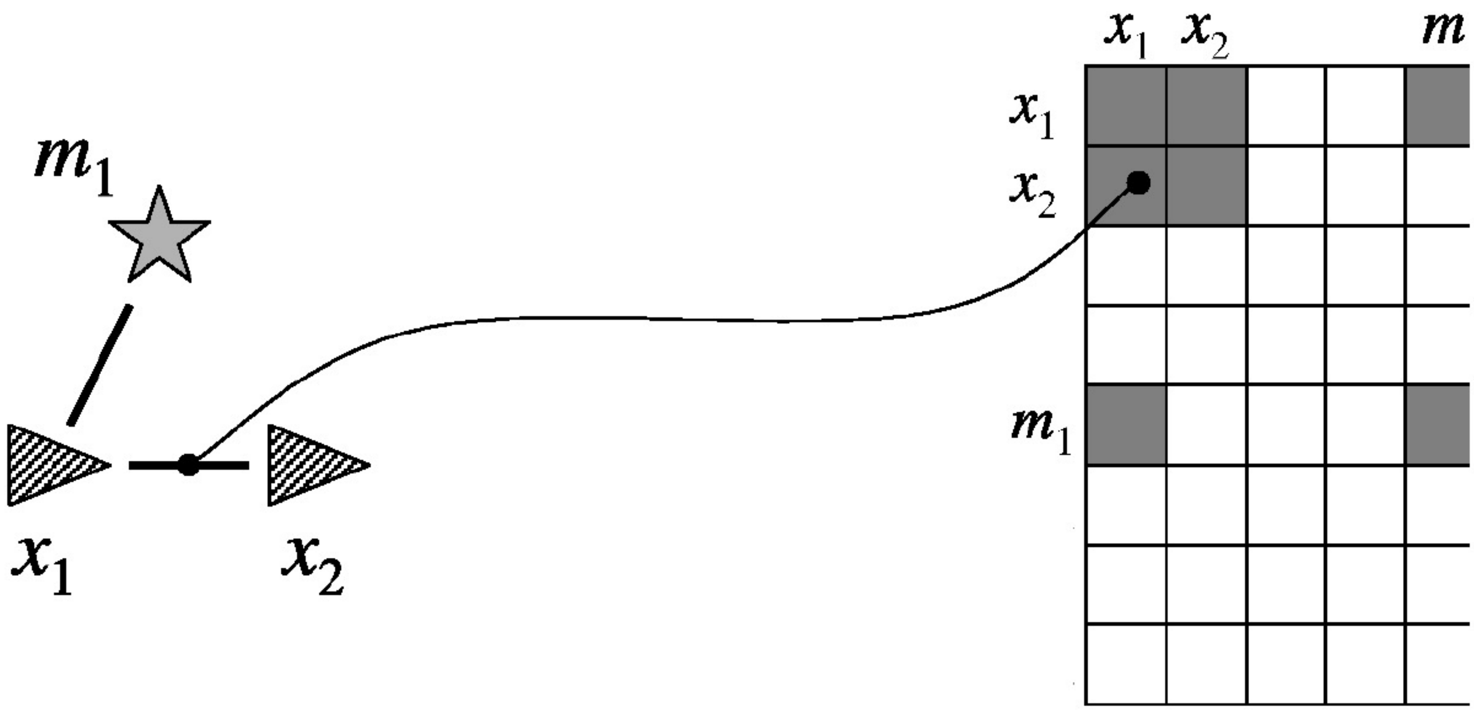
Sum of all constraints:

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

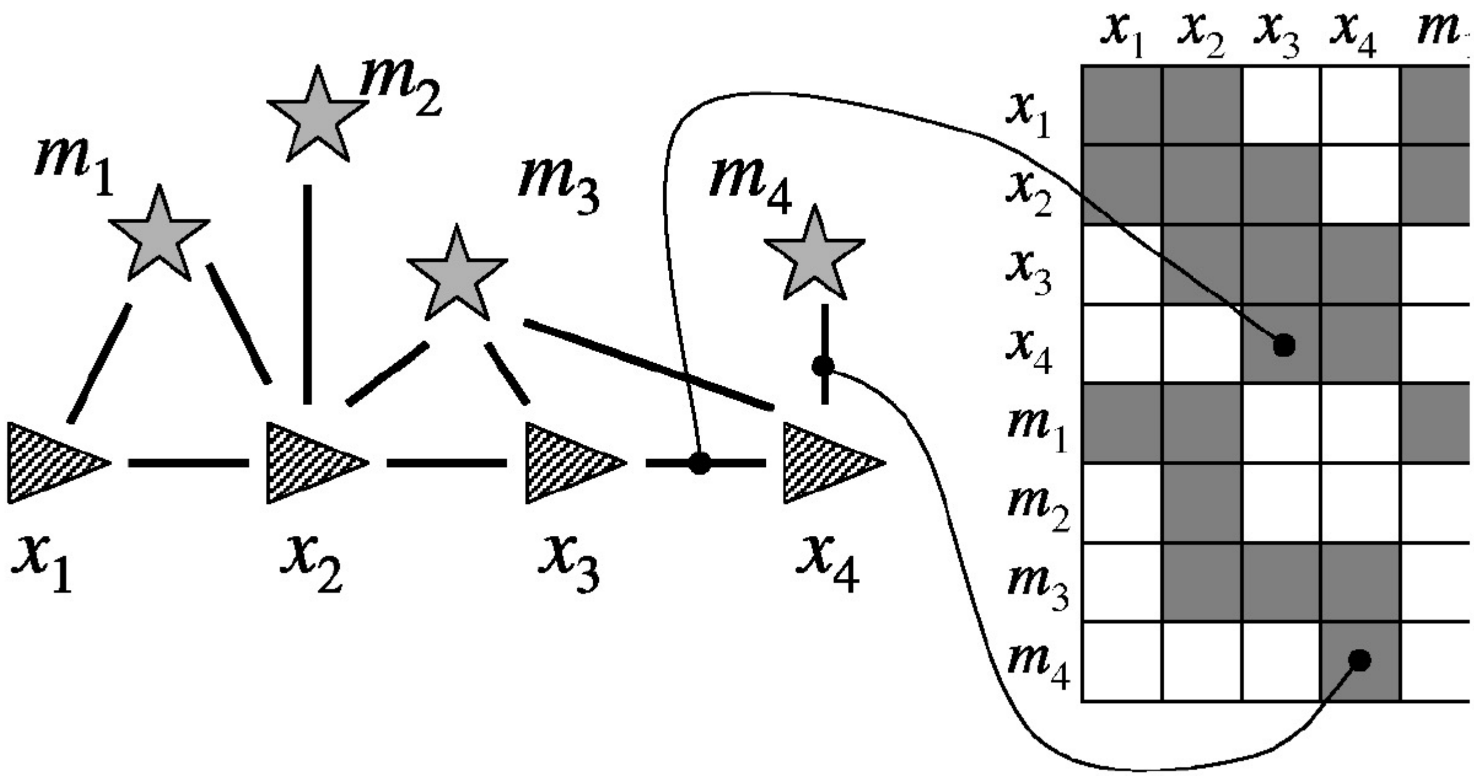
Graph-SLAM Idea (1)



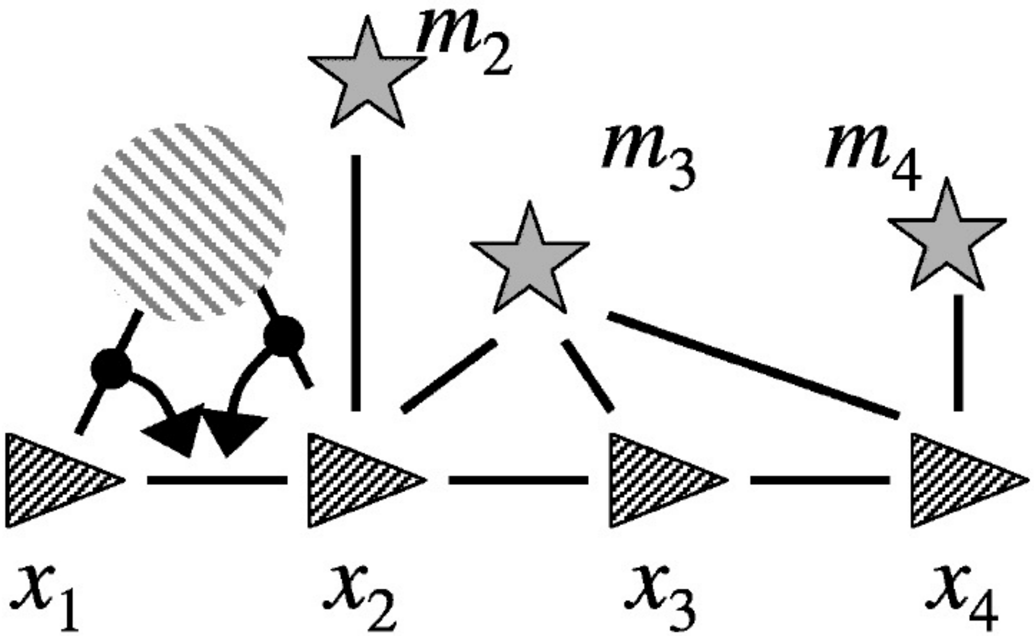
Graph-SLAM Idea (2)

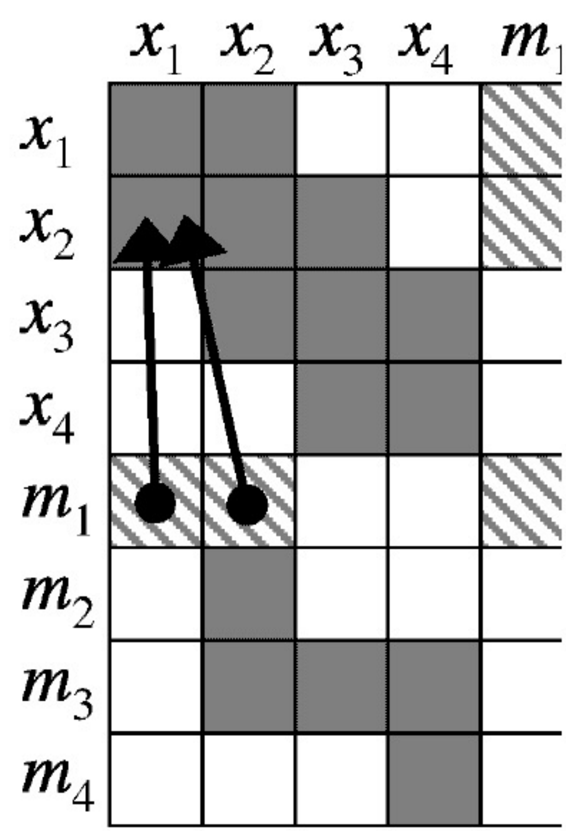


Graph-SLAM Idea (3)

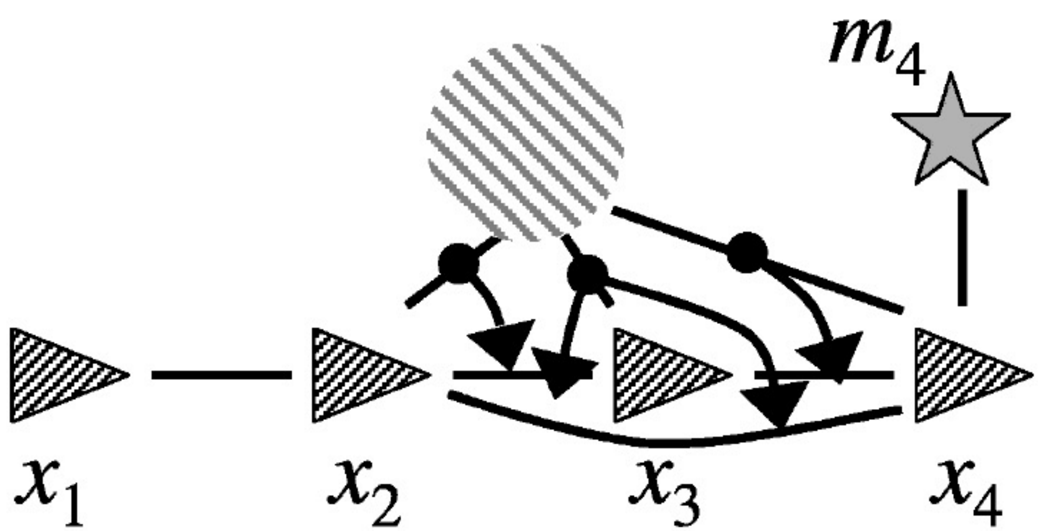


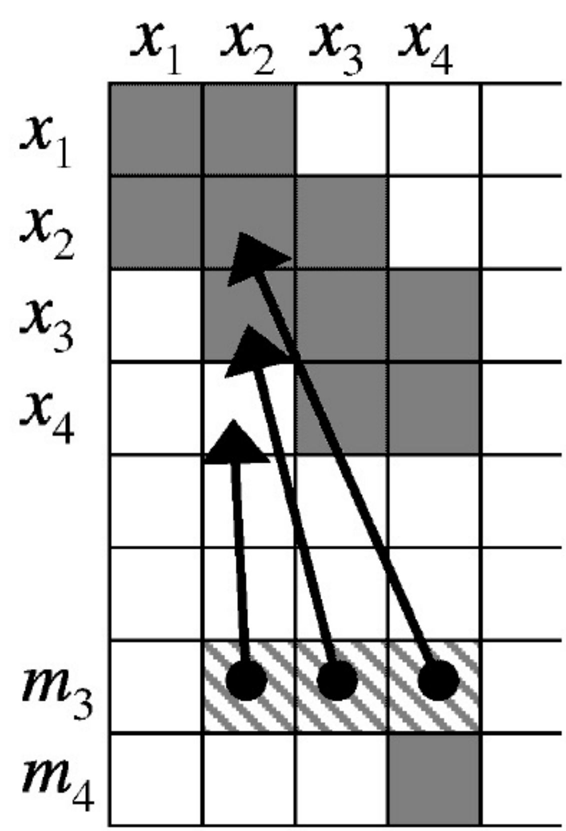
Graph-SLAM Inference (1)



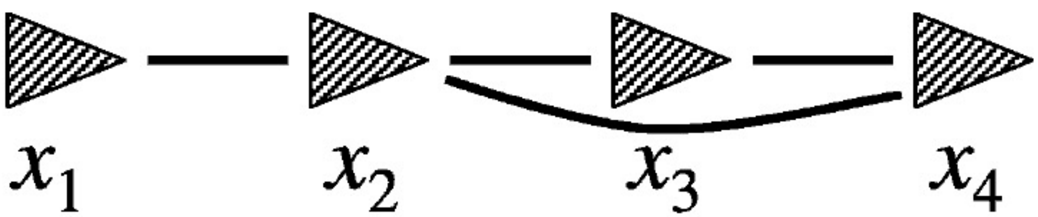


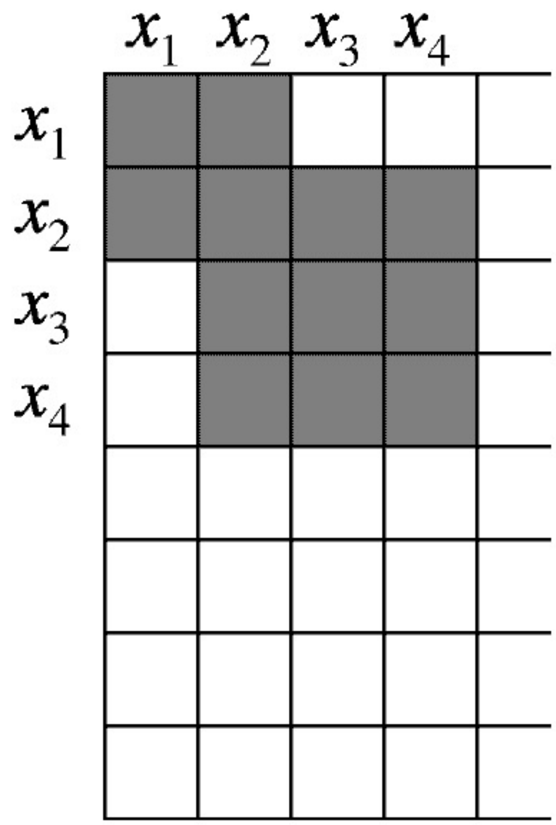
Graph-SLAM Inference (2)



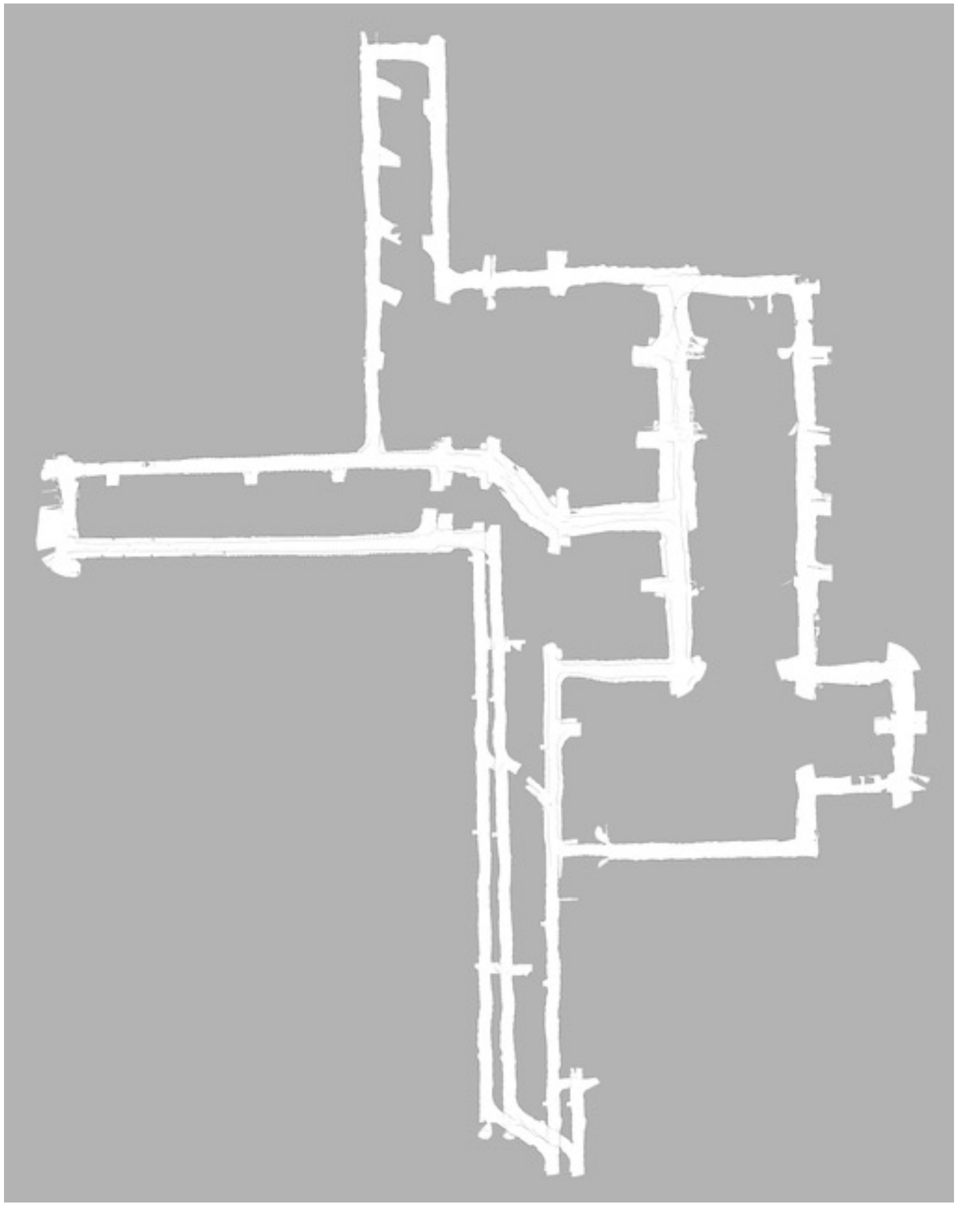


Graph-SLAM Inference (3)

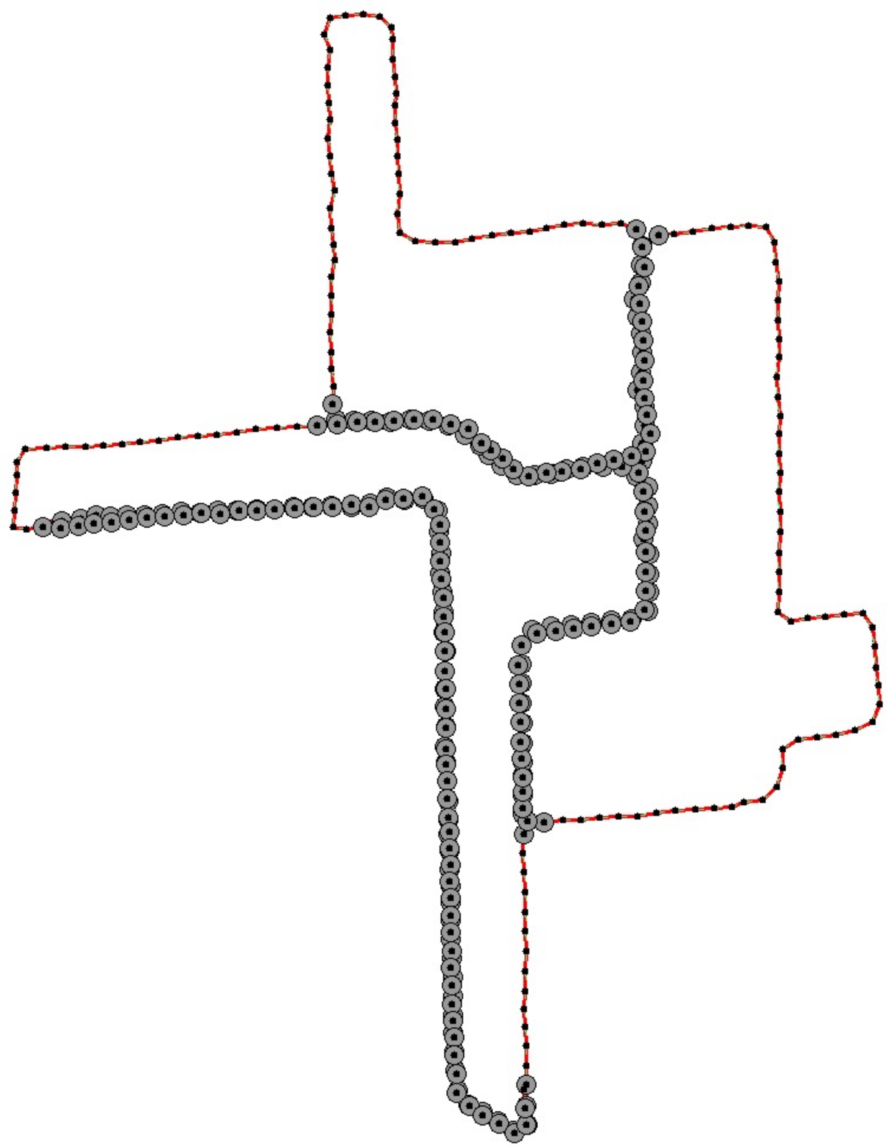




Mine Mapping



Mine Mapping: Data Associations

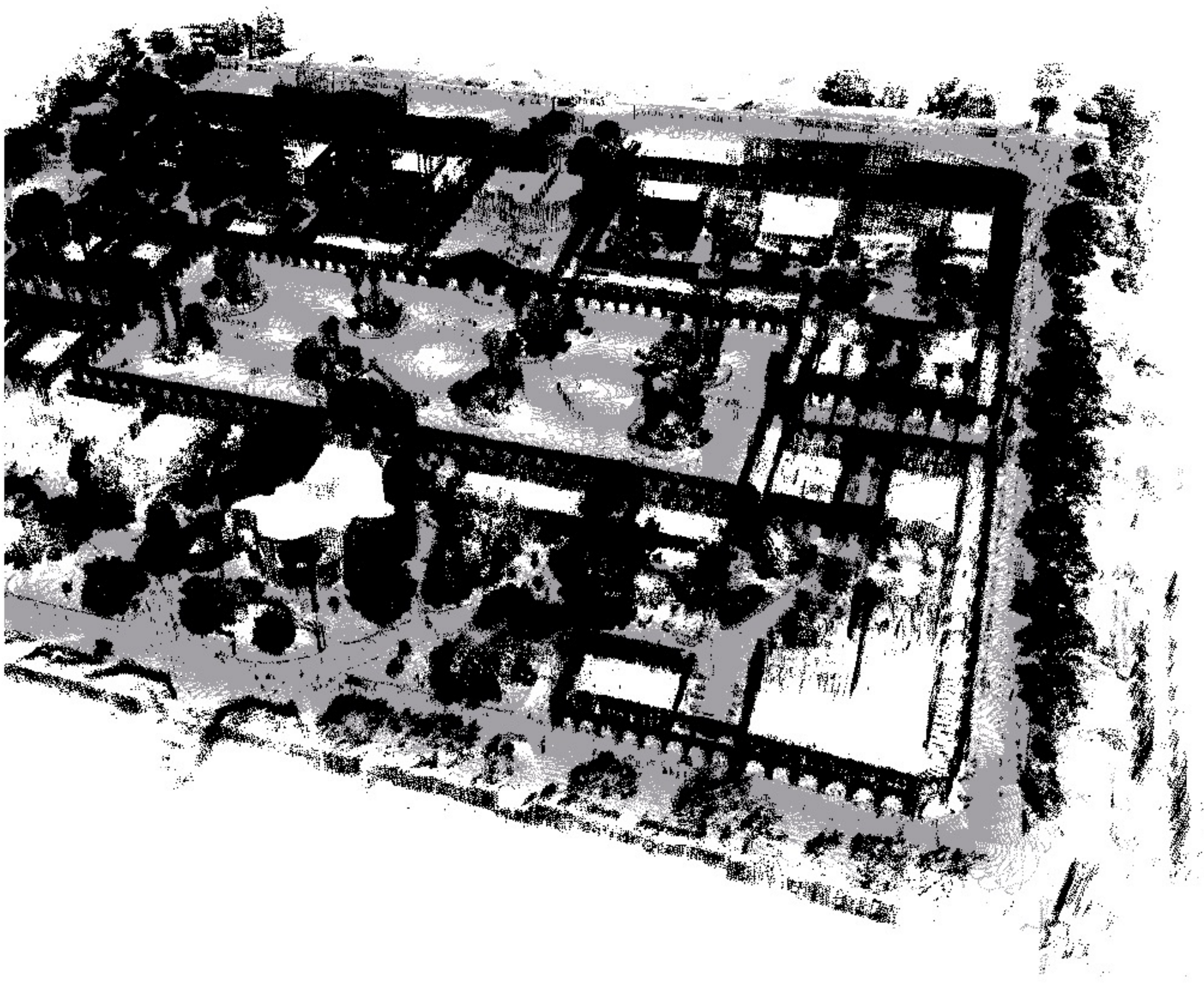


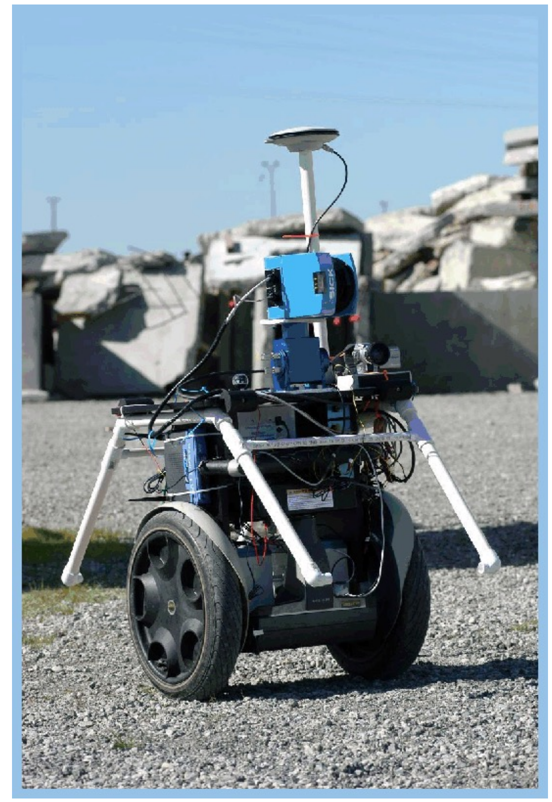
Efficient Map Recovery

- 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)

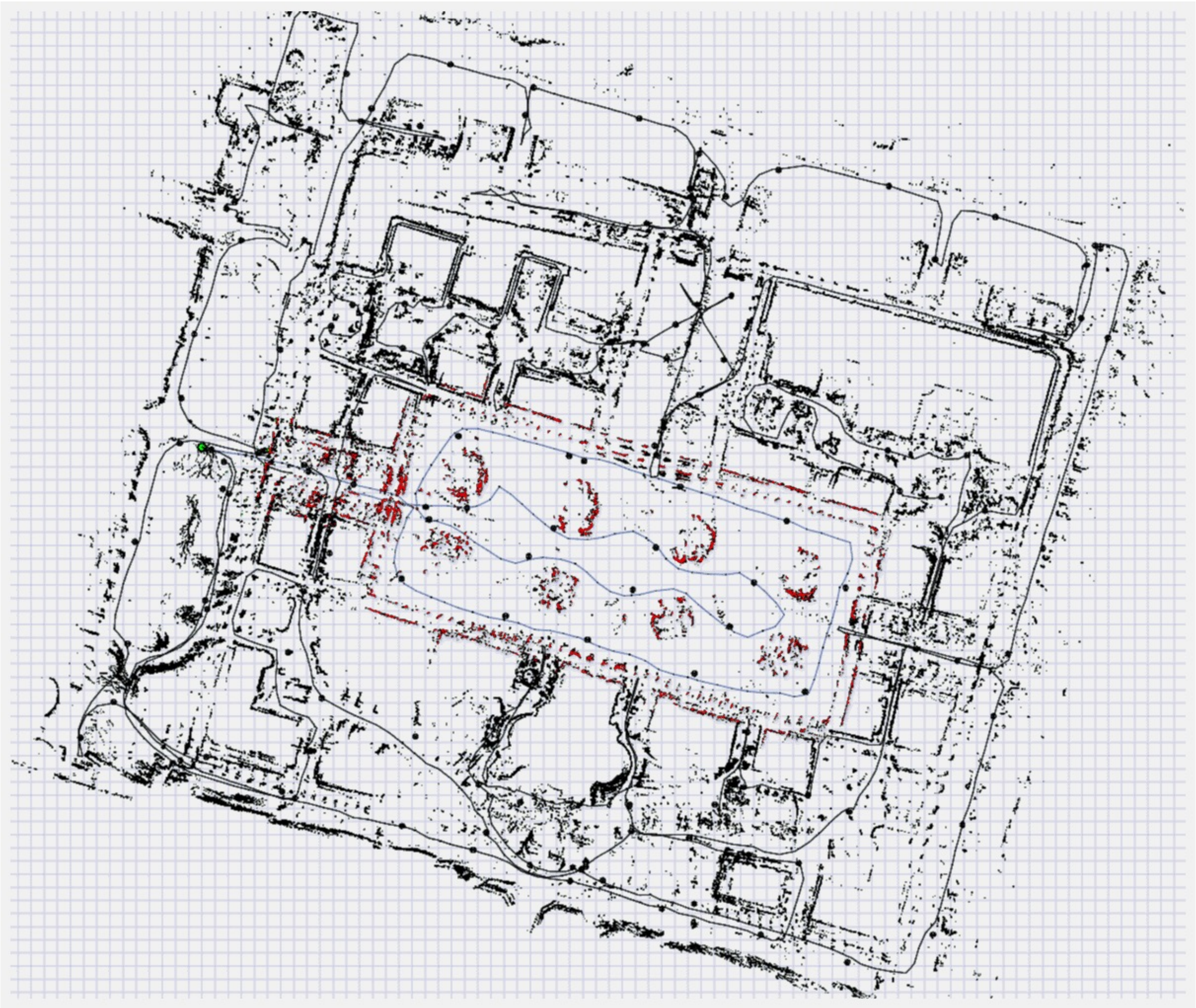
3D Outdoor Mapping



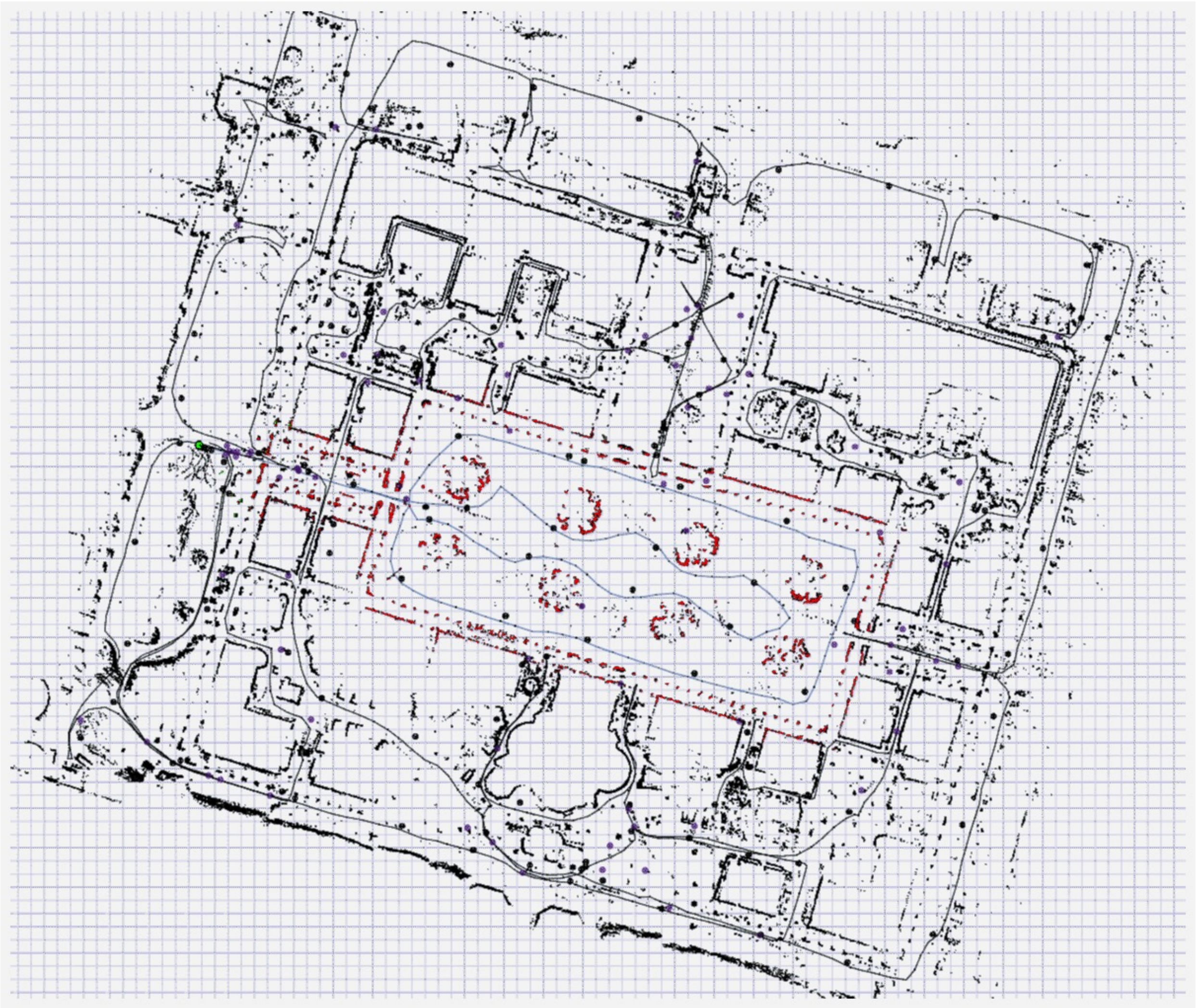


10^8 features, 10^5 poses, only few secs using cg.

Map Before Optimization

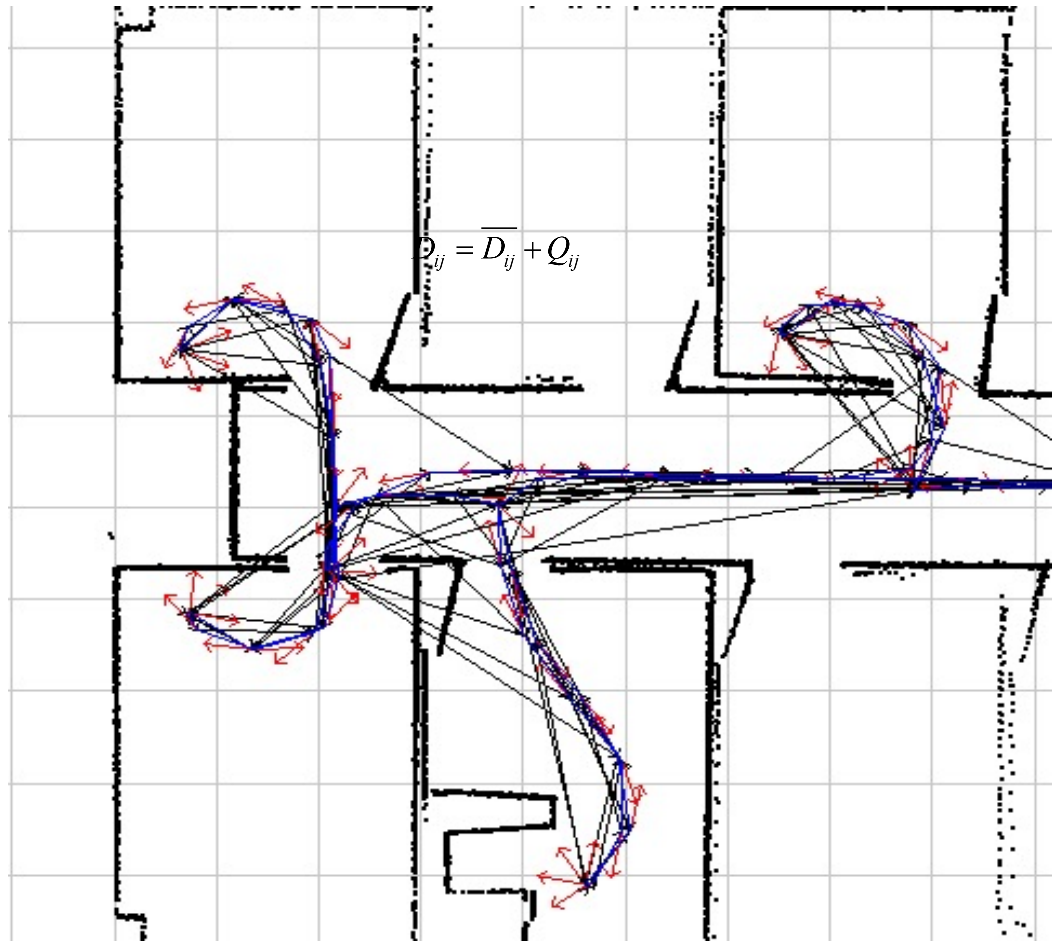


Map After Optimization



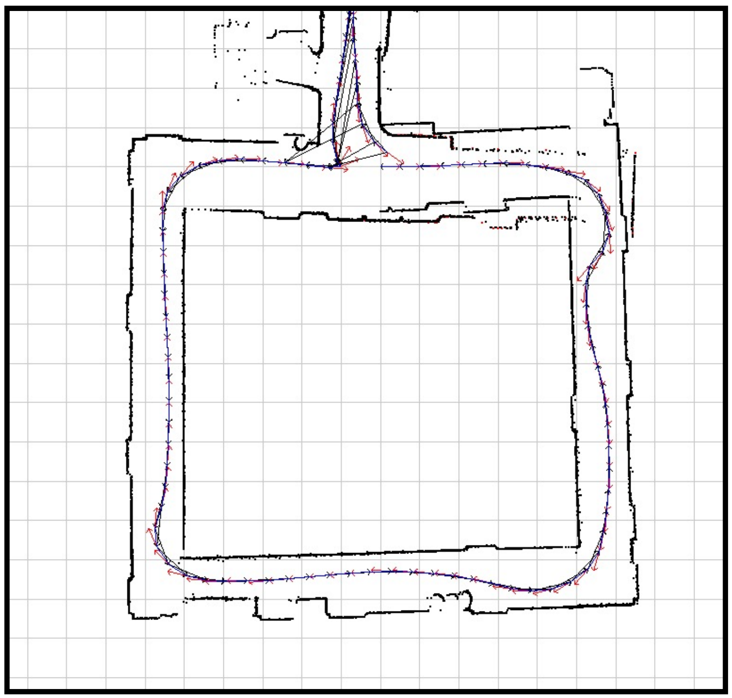
Robot Poses and Scans [Lu and Milios 1997]

- Successive robot poses connected by odometry
- Laser scan matching yields constraints between poses
- Loop closure based on map patches created from multiple scans

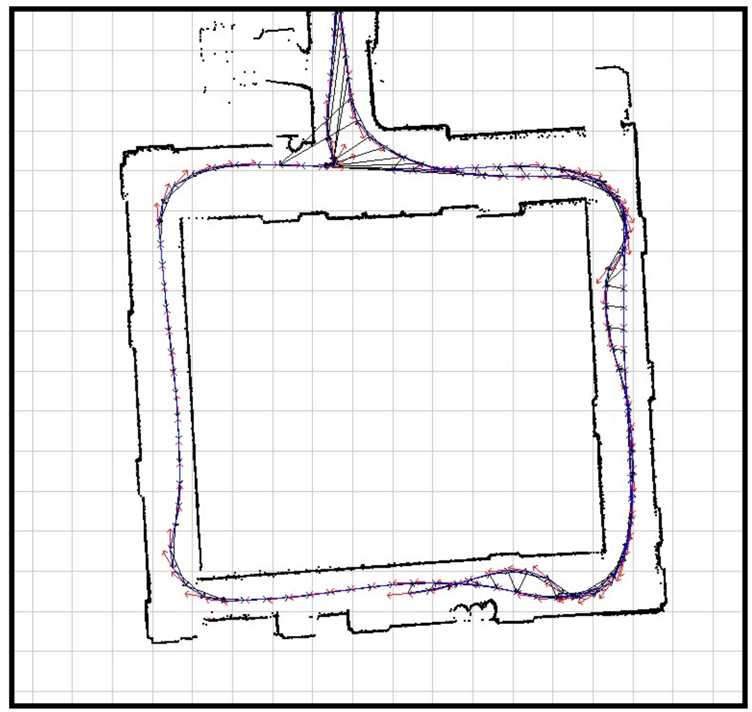


Loop Closure

- 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

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