CSE-P590a
Robotics
Mapping
Types of SLAM-Problems

Grid maps or scans

Sparse landmarks

RGB / Depth Maps
Problems in Mapping

• Sensor interpretation
  • How do we extract relevant information from raw sensor data?
  • How do we represent and integrate this information over time?

• Robot locations have to be known
  • How can we estimate them during mapping?
Occupancy Grid Maps

- Introduced by Moravec and Elfes in 1985
- Represent environment by a grid.
- Estimate the probability that a location is occupied by an obstacle.

**Key assumptions**
- Occupancy of individual cells is independent
- Robot positions are known!

\[
Bel(m_t) = P(m_t \mid u_1, z_2, \ldots, u_{t-1}, z_t) = \prod_{x,y} Bel(m_{t}^{[xy]})
\]

- Robot positions are known!
Inverse Sensor Model for Occupancy Grid Maps

Combination of linear function and Gaussian:
Incremental Updating of Occupancy Grids (Example)
Alternative for Lidar: Counting

• For every cell count
  • hits\((x,y)\): number of cases where a beam ended at \(\langle x,y\rangle\)
  • misses\((x,y)\): number of cases where a beam passed through \(\langle x,y\rangle\)

\[
Bel(m^{[xy]}) = \frac{\text{hits}(x, y)}{\text{hits}(x, y) + \text{misses}(x, y)}
\]

• Assumption: \(P(\text{occupied}(x,y)) = P(\text{reflects}(x,y))\)
Occupancy Grids: From scans to maps
Tech Museum, San Jose

CAD map

occupancy grid map
OctoMap
A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems

K.M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, W. Burgard

University of Freiburg, Germany

http://octomap.sf.net
Robots in 3D Environments

Mobile manipulation

Humanoid robots

Outdoor navigation

Flying robots
3D Map Requirements

- Full 3D Model
  - Volumetric representation
  - Free-space
  - Unknown areas (e.g. for exploration)

- Can be updated
  - Probabilistic model
    (sensor noise, changes in the environment)
  - Update of previously recorded maps

- Flexible
  - Map is dynamically expanded
  - Multi-resolution map queries

- Compact
  - Memory efficient
  - Map files for storage and exchange
Map Representations

Pointclouds

- **Pro:**
  - No discretization of data
  - Mapped area not limited

- **Contra:**
  - Unbounded memory usage
  - No direct representation of free or unknown space
Map Representations

3D voxel grids

- **Pro:**
  - Probabilistic update
  - Constant access time

- **Contra:**
  - Memory requirement
    - Extent of map has to be known
    - Complete map is allocated in memory
Map Representations

Octrees

- Tree-based data structure
- Recursive subdivision of space into octants
- Volumes allocated as needed
- Multi-resolution
Map Representations

Octrees

- **Pro:**
  - Full 3D model
  - Probabilistic
  - Flexible, multi-resolution
  - Memory efficient

- **Contra:**
  - Implementation can be tricky (memory, update, map files, ...)

- Open source implementation as C++ library available at [http://octomap.sf.net](http://octomap.sf.net)
Probabilistic Map Update

- **Clamping policy** ensures updatability [Yguel ‘07]
  \[ L(n) \in [l_{\text{min}}, l_{\text{max}}] \]

- Update of inner nodes enables **multi-resolution queries**
  \[ L(n) = \max_{i=1..8} L(n_i) \]
Examples

- Cluttered office environment

Map resolution: 2 cm
Examples: Office Building

- Freiburg, building 079
Examples: Large Outdoor Areas

- Freiburg computer science campus
  (292 x 167 x 28 m³, 20 cm resolution)
Examples: Tabletop
# Memory Usage

<table>
<thead>
<tr>
<th>Map dataset</th>
<th>Mapped area [m³]</th>
<th>Resolution [m]</th>
<th>Memory consumption [MB]</th>
<th>File size [MB]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Full grid</td>
<td>No compr.</td>
</tr>
<tr>
<td>FR-079 corridor</td>
<td>43.8 × 18.2 × 3.3</td>
<td>0.05</td>
<td>80.54</td>
<td>73.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1</td>
<td>10.42</td>
<td>10.90</td>
</tr>
<tr>
<td>Freiburg outdoor</td>
<td>292 × 167 × 28</td>
<td>0.20</td>
<td>654.42</td>
<td>188.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.80</td>
<td>10.96</td>
<td>4.56</td>
</tr>
<tr>
<td>New College (Epoch C)</td>
<td>250 × 161 × 33</td>
<td>0.20</td>
<td>637.48</td>
<td>91.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.80</td>
<td>10.21</td>
<td>2.35</td>
</tr>
</tbody>
</table>
CSE-P590a
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

Indoors

Undersea

Space

Underground
Illustration of SLAM without Landmarks

With only dead reckoning, vehicle pose uncertainty grows without bound.
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 vehicle and 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

MIT Indoor Track

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 Stachni
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
  \[ x_{0:T} = \{ x_0, x_1, x_2, \ldots, x_T \} \]
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

\[ x_t = \begin{pmatrix} x, y, \theta \, , \, m_1,x, m_1,y, \, \ldots, \, m_n,x, m_n,y \end{pmatrix}^T \]

Courtesy: Cyrill Stachni
EKF SLAM: State Representation

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

\[
\begin{pmatrix}
x \\
y \\
\theta \\
\end{pmatrix}
\begin{pmatrix}
\sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} \\
\sigma_{yx} & \sigma_{yy} & \sigma_{y\theta} \\
\sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta \theta} \\
\end{pmatrix}
\begin{pmatrix}
m_{1,x} \\
m_{1,y} \\
\vdots \\
m_{n,x} \\
m_{n,y} \\
\end{pmatrix}
\begin{pmatrix}
\sigma_{m_{1,x}x} & \sigma_{m_{1,x}y} & \sigma_{m_{1,x}\theta} \\
\sigma_{m_{1,y}x} & \sigma_{m_{1,y}y} & \sigma_{m_{1,y}\theta} \\
\vdots & \vdots & \vdots \\
\sigma_{m_{n,x}x} & \sigma_{m_{n,x}y} & \sigma_{m_{n,x}\theta} \\
\sigma_{m_{n,y}x} & \sigma_{m_{n,y}y} & \sigma_{m_{n,y}\theta} \\
\end{pmatrix}
\begin{pmatrix}
\sigma_{x_{m_{1,x}x}} & \sigma_{x_{m_{1,y}y}} & \vdots \\
\sigma_{y_{m_{1,x}y}} & \sigma_{y_{m_{1,y}y}} & \vdots \\
\vdots & \vdots & \vdots \\
\sigma_{\theta_{m_{1,x}\theta}} & \sigma_{\theta_{m_{1,y}\theta}} & \vdots \\
\sigma_{\theta_{m_{n,x}\theta}} & \sigma_{\theta_{m_{n,y}\theta}} & \vdots \\
\end{pmatrix}
\begin{pmatrix}
\sigma_{x_{m_{1,x}}} & \sigma_{x_{m_{1,y}}} & \cdots & \sigma_{x_{m_{n,x}}} & \sigma_{x_{m_{n,y}}} \\
\sigma_{y_{m_{1,x}}} & \sigma_{y_{m_{1,y}}} & \cdots & \sigma_{y_{m_{n,x}}} & \sigma_{y_{m_{n,y}}} \\
\vdots & \vdots & \cdots & \vdots & \vdots \\
\sigma_{\theta_{m_{1,x}}} & \sigma_{\theta_{m_{1,y}}} & \cdots & \sigma_{\theta_{m_{n,x}}} & \sigma_{\theta_{m_{n,y}}} \\
\sigma_{\theta_{m_{1,x}}} & \sigma_{\theta_{m_{1,y}}} & \cdots & \sigma_{\theta_{m_{n,x}}} & \sigma_{\theta_{m_{n,y}}} \\
\end{pmatrix}
\begin{pmatrix}
\mu \\
\Sigma \\
\end{pmatrix}

Courtesy: Cyrill Stachniss
EKF SLAM: State Representation

- More compactly

\[
\begin{bmatrix}
\sum x_R x_R \\
\sum m_1 x_R \\
\sum m_n x_R \\
\end{bmatrix}
\begin{bmatrix}
x_R \\
m_1 \\
m_n \\
\end{bmatrix}
+ 
\begin{bmatrix}
\sum x_R m_1 \\
\sum m_1 m_1 \\
\sum m_n m_1 \\
\end{bmatrix}
= 
\begin{bmatrix}
\sum x_R m_n \\
\sum m_1 m_n \\
\sum m_n m_n \\
\end{bmatrix}
\]

Courtesy: Cyrill Stachni
EKF SLAM: State Representation

- Even more compactly (note: )

\[ x_R \rightarrow x \]

\[
\begin{pmatrix}
  x \\
  m
\end{pmatrix}
\]

\[
\begin{pmatrix}
  \Sigma_{xx} \\
  \Sigma_{mx} \\
  \Sigma_{mm}
\end{pmatrix}
\]

\[ \mu \rightarrow \Sigma \]

Courtesy: Cyrill Stachni
EKF SLAM: Filter Cycle

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

Courtesy: Cyrill Stachni
EKF SLAM: State Prediction

\[
\begin{pmatrix}
    x_R \\
    m_1 \\
    \vdots \\
    m_n \\
\end{pmatrix}
\begin{pmatrix}
    \sum x_R x_R \\
    \sum m_1 x_R \\
    \vdots \\
    \sum m_n x_R \\
\end{pmatrix}
\begin{pmatrix}
    \sum x_R m_1 & \cdots & \sum x_R m_n \\
    \sum m_1 m_1 & \cdots & \sum m_1 m_n \\
    \vdots & \cdots & \vdots \\
    \sum m_n m_1 & \cdots & \sum m_n m_n \\
\end{pmatrix}
\mu 
\sum 

Courtesy: Cyrill Stachni
EKF SLAM: Measurement Prediction

\[
\begin{pmatrix}
x_R \\
m_1 \\
\vdots \\
m_n
\end{pmatrix} \quad \begin{pmatrix}
\sum x_R x_R & \sum x_R m_1 & \cdots & \sum x_R m_n \\
\sum m_1 x_R & \sum m_1 m_1 & \cdots & \sum m_1 m_n \\
\vdots & \vdots & \ddots & \vdots \\
\sum m_n x_R & \sum m_n m_1 & \cdots & \sum m_n m_n
\end{pmatrix}

\]

Courtesy: Cyrill Stachniss
EKF SLAM: Obtained Measurement

\[
\begin{pmatrix}
    x_R \\
    m_1 \\
    \vdots \\
    m_n
\end{pmatrix}
\begin{pmatrix}
    \sum x_R x_R & \sum x_R m_1 & \cdots & \sum x_R m_n \\
    \sum m_1 x_R & \sum m_1 m_1 & \cdots & \sum m_1 m_n \\
    \vdots & \vdots & \ddots & \vdots \\
    \sum m_n x_R & \sum m_n m_1 & \cdots & \sum m_n m_n
\end{pmatrix}
\]

\[
\mu \\
\mu
\]

Courtesy: Cyrill Stachniss
EKF SLAM: Data Association and Difference Between \( h(x) \) and \( z \)

\[
\begin{pmatrix}
    x_R \\
    m_1 \\
    \vdots \\
    m_n
\end{pmatrix}
\quad \begin{pmatrix}
    \sum x_R x_R \\
    \sum m_1 x_R \\
    \vdots \\
    \sum m_n x_R
\end{pmatrix} - \begin{pmatrix}
    \sum x_R m_1 \\
    \sum m_1 m_1 \\
    \vdots \\
    \sum m_n m_1
\end{pmatrix} = \sum \begin{pmatrix}
    \sum x_R m_n \\
    \sum m_1 m_n \\
    \vdots \\
    \sum m_n m_n
\end{pmatrix}

Courtesy: Cyrill Stachni
EKF SLAM: Update Step

\[
\begin{pmatrix}
x_R \\
m_1 \\
\vdots \\
m_n \\
\end{pmatrix}
\begin{pmatrix}
x_R x_R & \Sigma x_R m_1 & \cdots & \Sigma x_R m_n \\
\Sigma m_1 x_R & \Sigma m_1 m_1 & \cdots & \Sigma m_1 m_n \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma m_n x_R & \Sigma m_n m_1 & \cdots & \Sigma m_n m_n \\
\end{pmatrix}
\]

\[
\mu \\
\Sigma
\]

Courtesy: Cyrill Stachniss
Approximate the SLAM posterior with a high-dimensional Gaussian [Smith & Cheesman, 1986] ... 

Single hypothesis data association
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 Stachni
Online SLAM Example
Before the Loop-Closure

Courtesy: K. Arras
After the Loop-Closure
Example: Victoria Park Dataset

Courtesy: E. Nebc
Victoria Park: Data Acquisition

Courtesy: E. Nebc
Victoria Park: EKF Estimate

Courtesy: E. Nebot
Victoria Park: EKF Estimate

Courtesy: E. Nebc
Victoria Park: Landmarks

Courtesy: E. Nebc
Victoria Park: Landmark Covariance

![Covariance Graph](image_url)

Courtesy: E. Nebc
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.
EKF Algorithm

1. **Extended Kalman filter**$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:

2. **Prediction:**
   
3. $\bar{\mu}_t = g(u_t, \mu_{t-1})$ \quad $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$

4. $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$ \quad $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

5. **Correction:**

6. $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$ \quad $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$

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

8. $\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$ \quad $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

9. **Return** $\mu_t, \Sigma_t$

   $H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t}$ \quad $G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$
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 Stachni
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

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

\[ q_{ij} = D_{ij} + Q_{ij} \]
Loop Closure

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