CSE-571
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!
Updating Occupancy Grid Maps

• **Idea**: Update each individual cell using a binary Bayes filter.

\[
Bel(m_t^{[xy]}) = \eta \ p(z_t \mid m_t^{[xy]}) \sum_{m_{t-1}^{[xy]}} p(m_t^{[xy]} \mid m_{t-1}^{[xy]}, u_{t-1}) Bel(m_{t-1}^{[xy]})
\]

• **Additional assumption**: Map is static

\[
Bel(m_t^{[xy]}) = \eta \ p(z_t \mid m_t^{[xy]}) Bel(m_{t-1}^{[xy]})
\]

• **Log odds representation**:

\[
l_{t,i} = \log \left( \frac{p(m_i \mid z_{1:t}, x_{1:t})}{1 - p(m_i \mid z_{1:t}, x_{1:t})} \right)
\]

\[
l_{t,i} = l_{t-1,i} + \log \left( \frac{p(m_i \mid z_t, x_t)}{1 - p(m_i \mid z_t, x_t)} \right) - \log \left( \frac{p(m_i)}{1 - p(m_i)} \right)
\]
Inverse Sensor Model for Occupancy Grid Maps

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

- For every cell count
  - \textbf{hits}(x,y): number of cases where a beam ended at \( <x,y> \)
  - \textbf{misses}(x,y): number of cases where a beam passed through \( <x,y> \)

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

- 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

Outdoor navigation

Humanoid robots

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**
  \[ L(n) = \max_{i=1..8} L(n_i) \]

![Images of trees at different resolutions: 0.08 m, 0.64 m, 1.28 m]
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
Adding Color

Probabilistic 3D mapping using OctoMap and RGBDSLAM

Kai M. Wurm, Felix Endres
Autonomous Intelligent Systems Lab
University of Freiburg, Germany
## 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>Full grid</td>
<td>No compr.</td>
<td>Lossless 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>