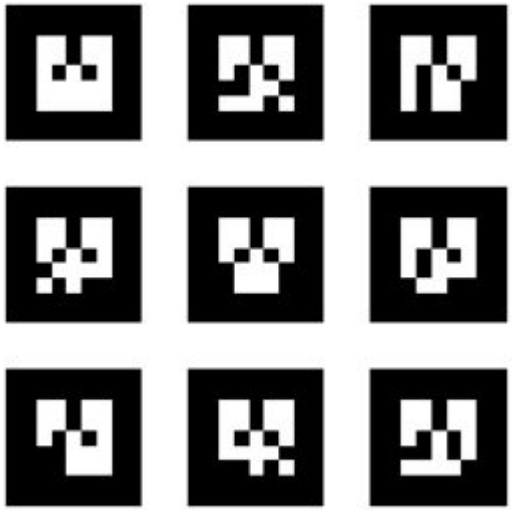
CSE-571
Robotics

Mapping

Some slides adapted from Dieter Fox, Cyrill Stachniss, and probabilistic-robotics.org

Fiducials: AR Tags



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

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

  - Robot positions are known!

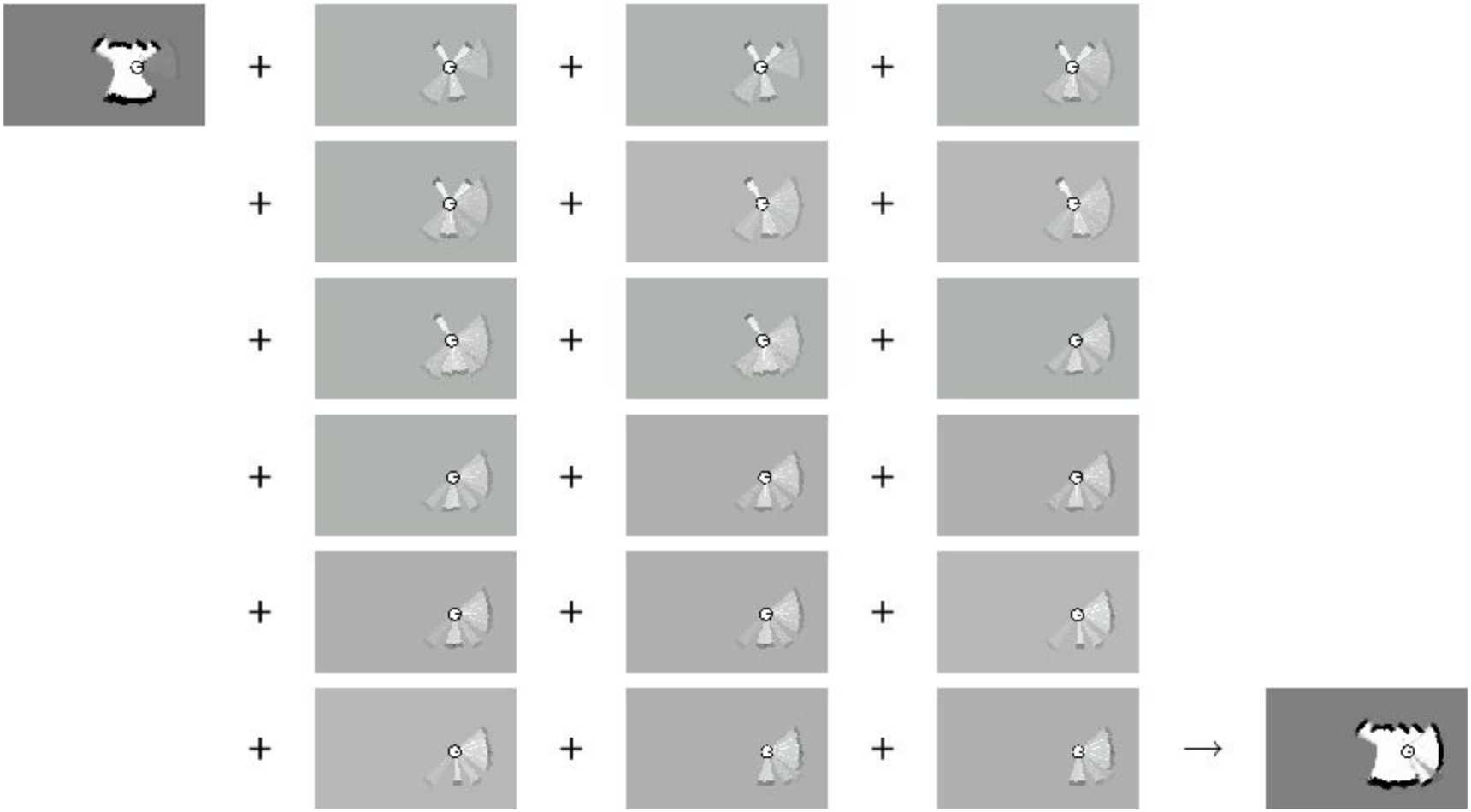
Using Bayes:

- Log odds representation:

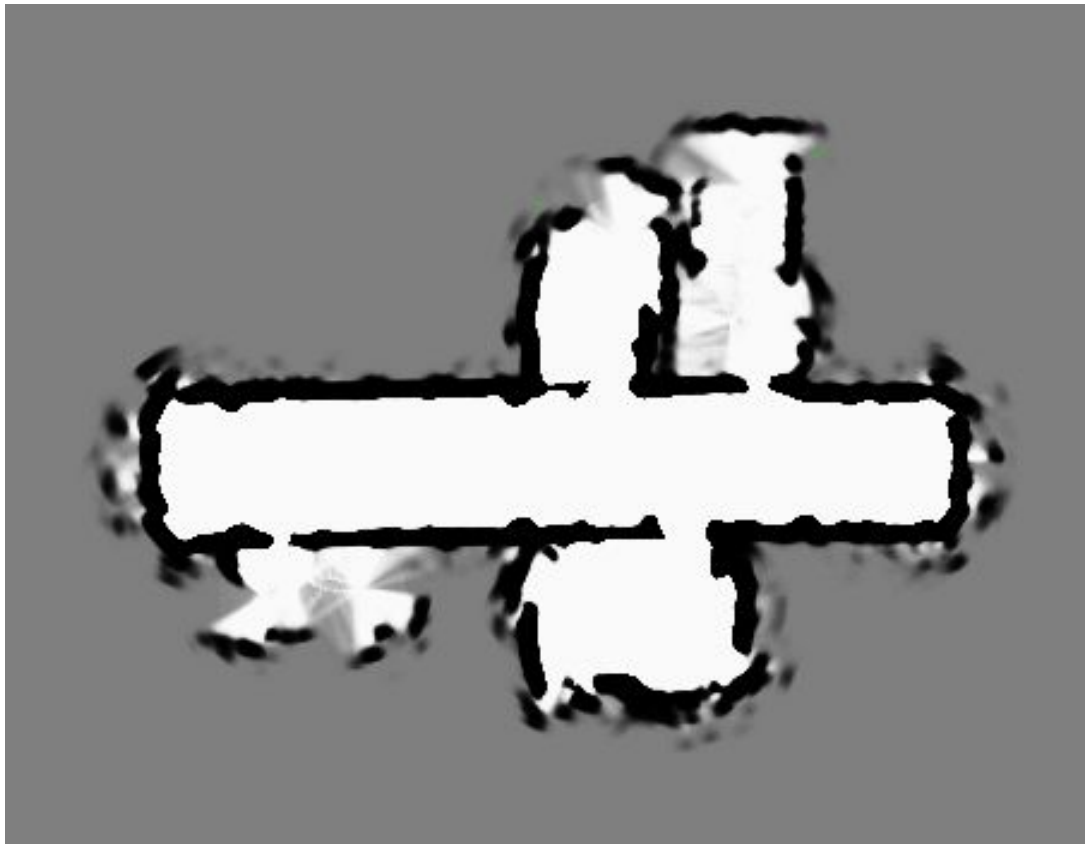
$$l_{t,i} = \log\left(\frac{p(m_i|z_{1:t}, x_{1:t})}{1 - p(m_i|z_{1:t}, x_{1:t})}\right)$$

$$l_{t,i} = l_{t-1,i} + log\left(\frac{p(m_i|z_t, x_t)}{1 - p(m_i|z_t, x_t)}\right) - log\left(\frac{p(m_i)}{1 - p(m_i)}\right)$$

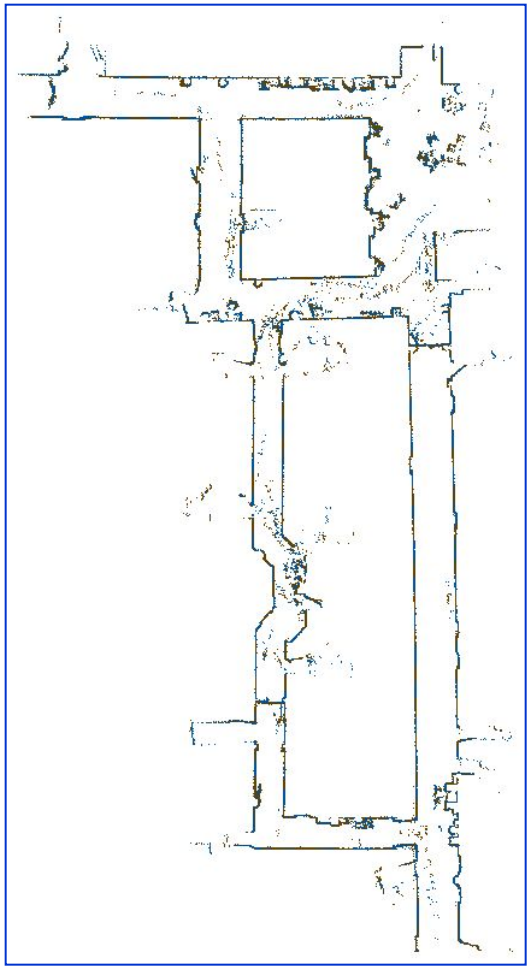
Incremental Updating of Occupancy Grids (Example)

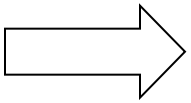


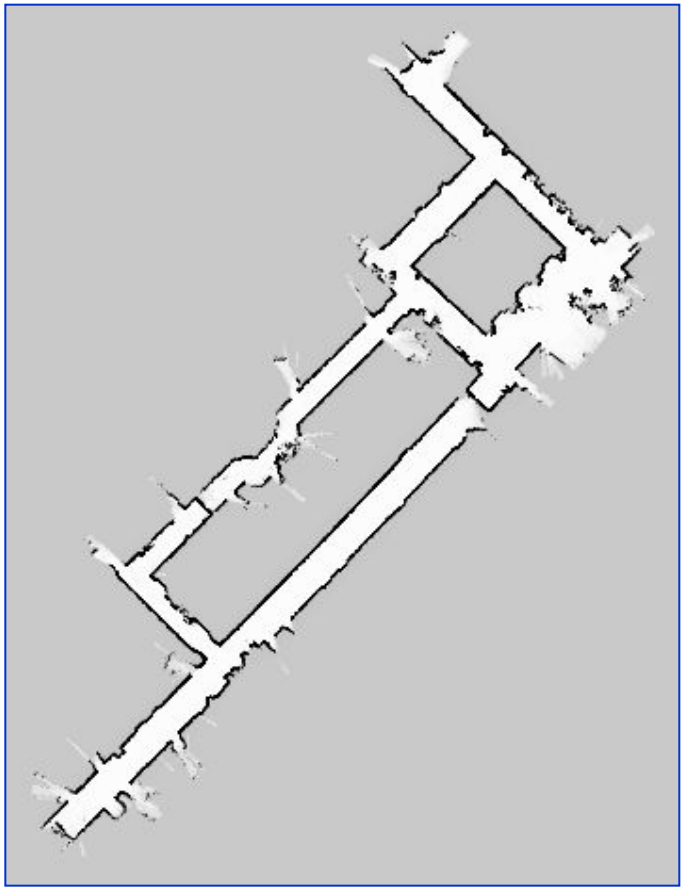
Resulting Map Obtained with Ultrasound Sensors



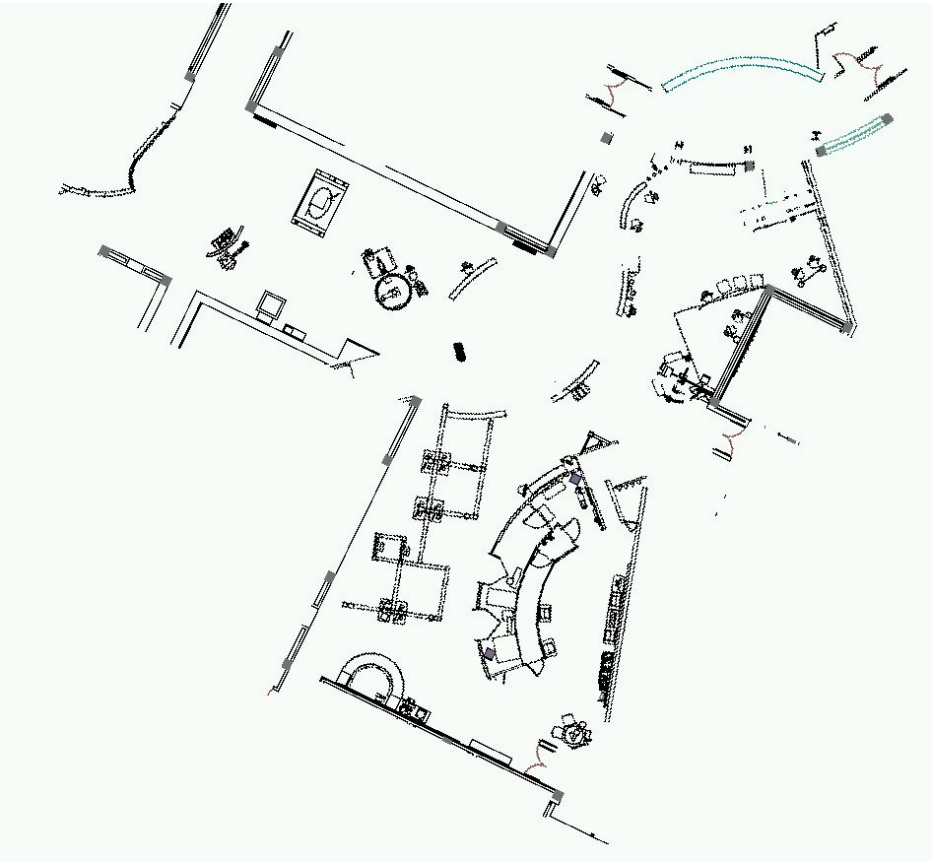
Occupancy Grids: From scans to maps



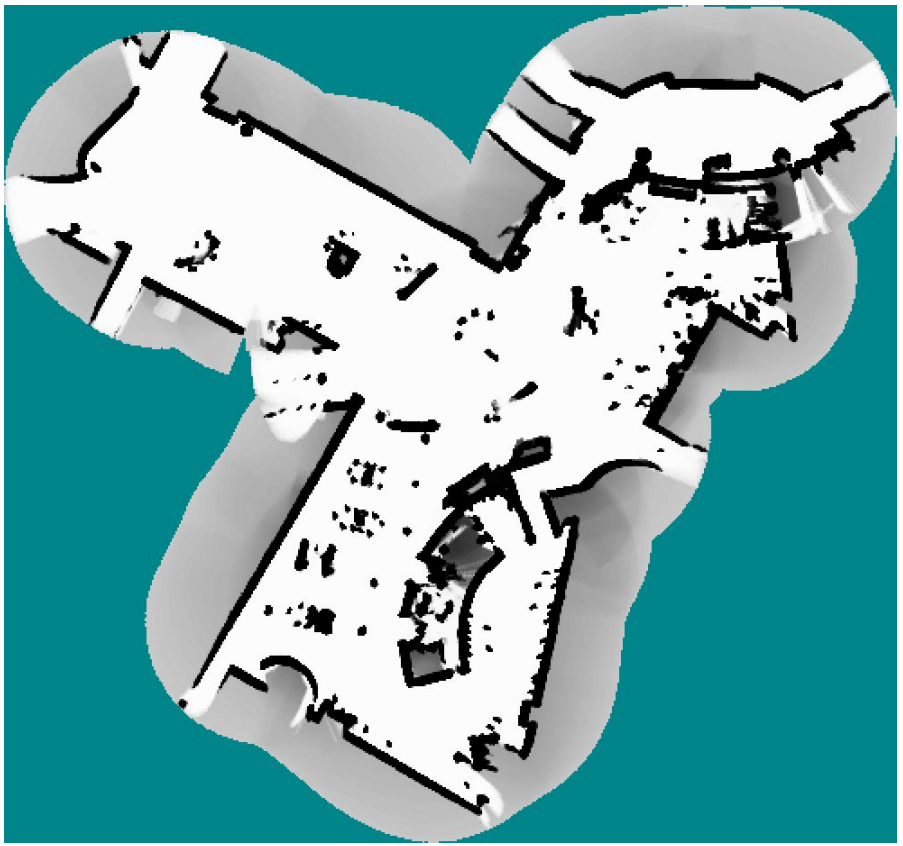




Tech Museum, San Jose



CAD map



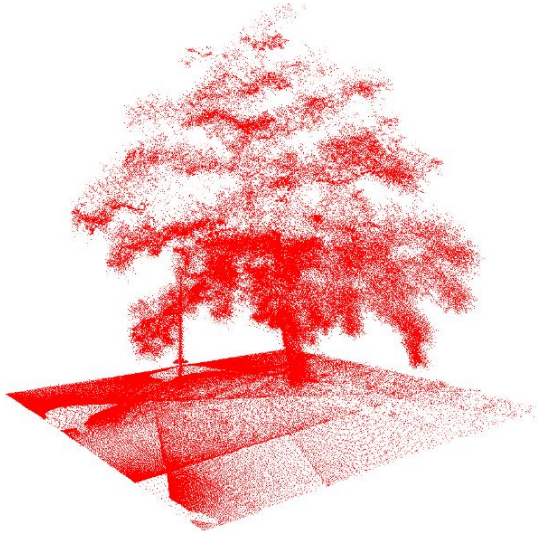
occupancy grid map

3D Map Requirements

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

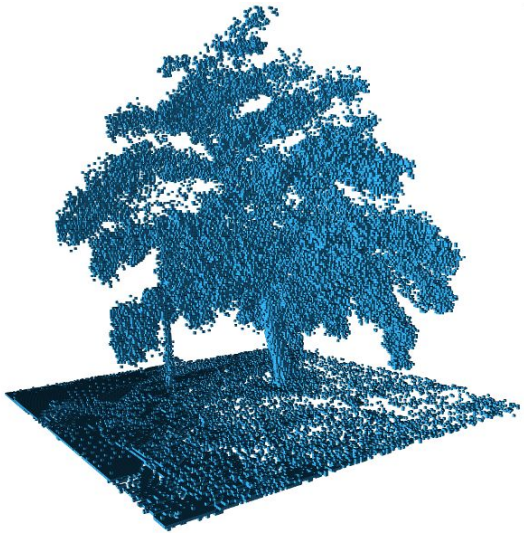
Pointclouds



- 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

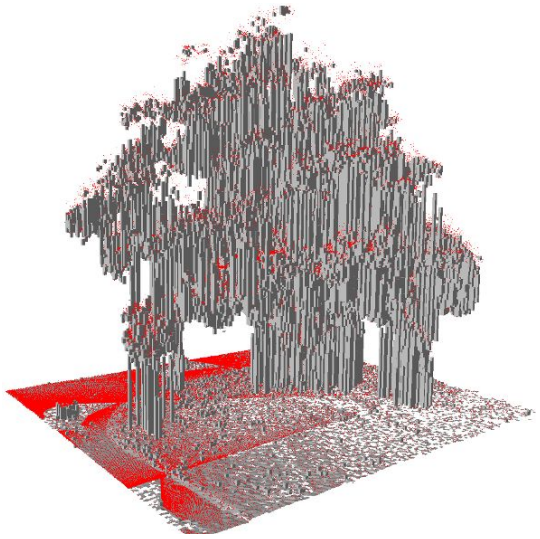


- Pro:
  - Probabilistic update
  - Constant access time
- Contra:
  - Memory requirement
    - Extent of map has to be known
    - Complete map is allocated in memory

Map Representations

2.5D Maps

- 2D grid
- Height value(s) in each cell

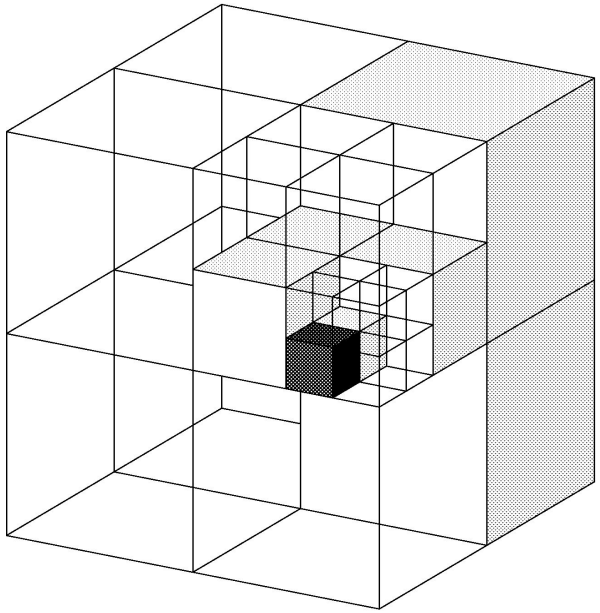


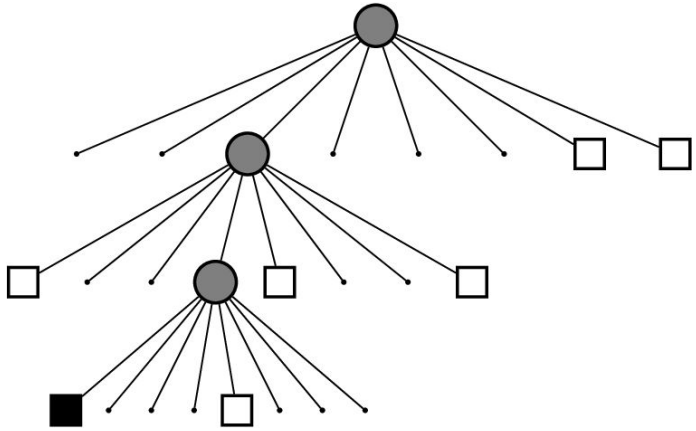
- Pro:
  - Memory efficient
- Contra:
  - Not completely probabilistic
  - No distinction between free and unknown space

Map Representations

Octrees

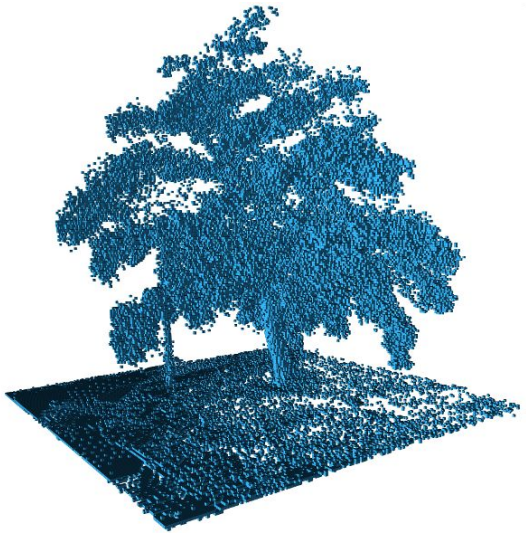
- Tree-based data structure
- Recursive subdivision of space into octants
- Volumes allocated as needed
- Multi-resolution





Map Representations

Octrees



- Pro:
  - Full 3D model
  - Probabilistic
  - Flexible, multi-resolution
  - Memory efficient
- Contra:
  - Implementation can be tricky

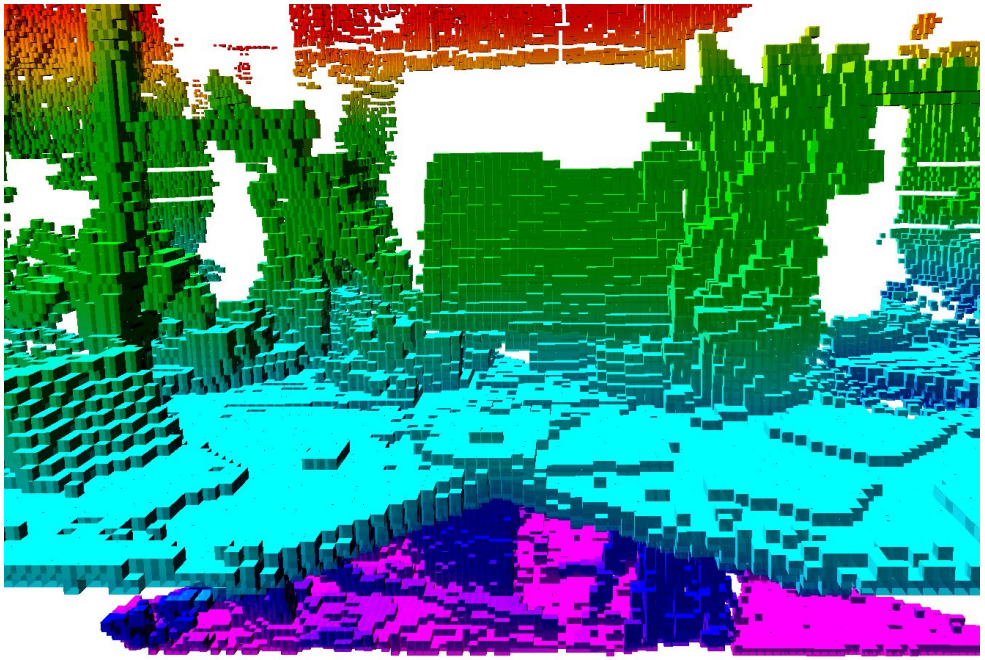
OctoMap Framework

- Based on octrees
- Probabilistic representation of occupancy including unknown
- Supports multi-resolution map queries
- Lossless compression

- Open source implementation as C++ library available at http://octomap.sf.net

Examples

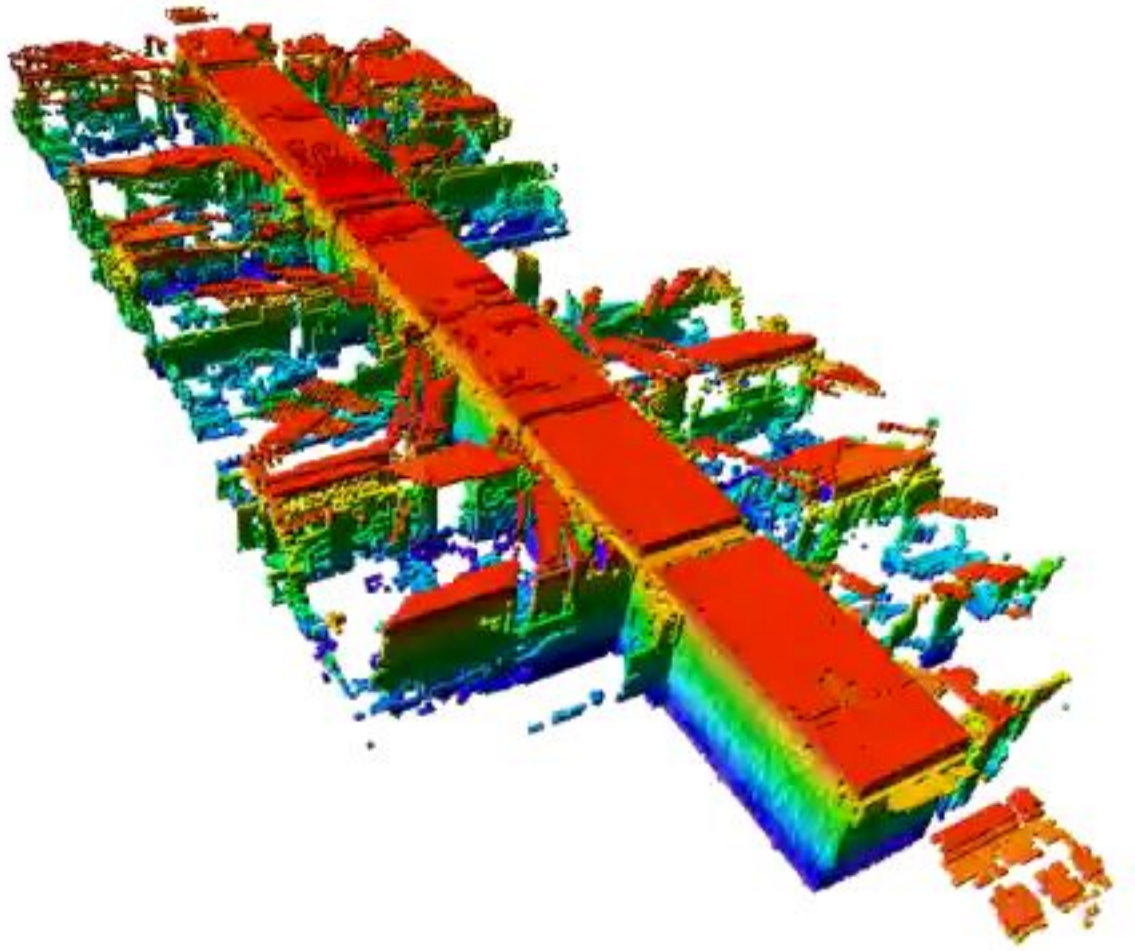
- Cluttered office environment



Map resolution: 2 cm

Examples: Office Building

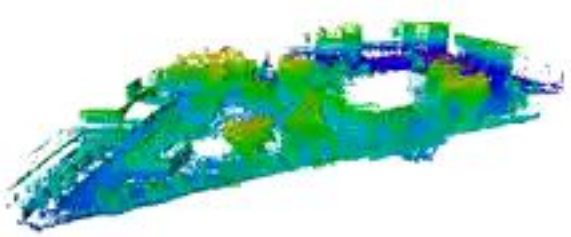
- Freiburg, building 079



Examples: Large Outdoor Areas

- Freiburg computer science campus

  (292 x 167 x 28 m³, 20 cm resolution)



OctoMap Implementation

- Open source C++ library
- Fully documented
- Can be easily adapted to your projects
- ROS integration
- Includes OpenGL viewer
- Already used by several other researchers

http://octomap.sf.net