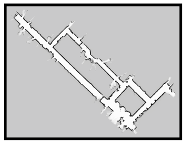
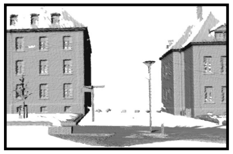
CSE-571
Robotics

Mapping

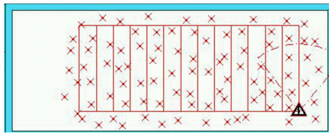
Types of SLAM-Problems

Grid maps or scans

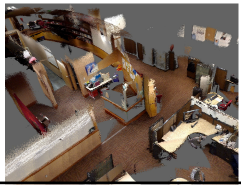




Sparse landmarks



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

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

  - Robot positions are known!

Updating Occupancy Grid Maps

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

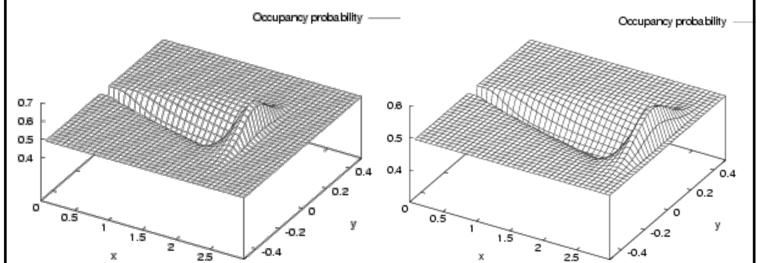
$$Bel(m_t^{[xy]}) = \eta\ p(z_t \mid m_t^{[xy]}) \sum_{m_{t-1}^{[xy]}} p(m_t^{[xy]} \mid m_{t-1}^{[xy]}, u_{t-1}) Bel(m_{t-1}^{[xy]})$$

- Additional assumption: Map is static.

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

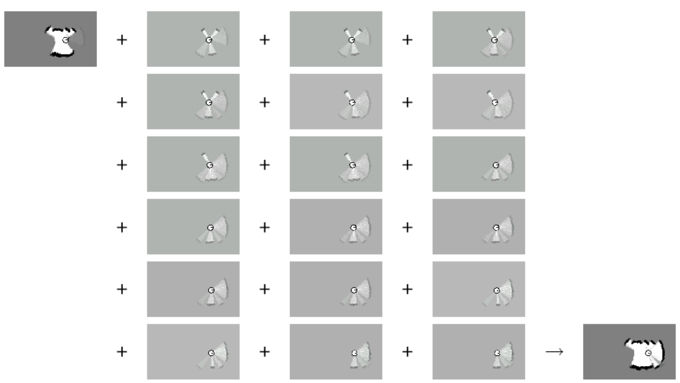
Inverse Sensor Model for Occupancy Grid Maps

Combination of linear function and Gaussian:



$$\bar{B}\left(m_t^{[xy]}\right) = \log odds\left(m_t^{[xy]} \mid z_t, x_t\right) - \log odds\left(m_t^{[xy]}\right) + \bar{B}\left(m_{t-1}^{[xy]}\right)$$

Incremental Updating of Occupancy Grids (Example)



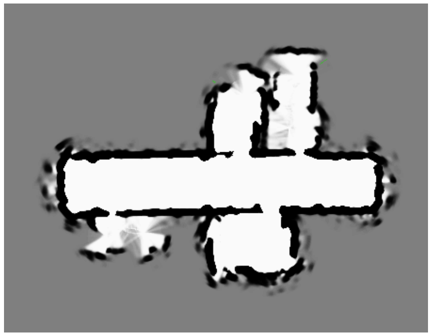
Alternative: Simple Counting

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

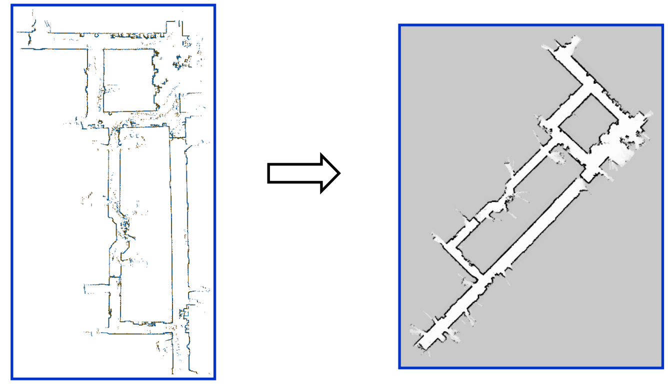
$$Bel(m^{[xy]}) = \frac{\text{hits}(x,y)}{\text{hits}(x,y) + \text{misses}(x,y)}$$

- Assumption: *P*(*occupied*(*x*,*y*)) = *P*(*reflects*(*x*,*y*))

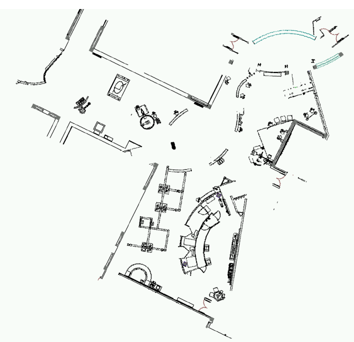
Resulting Map Obtained with Ultrasound Sensors



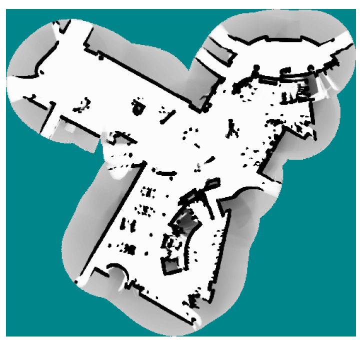
Occupancy Grids: From scans to maps



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

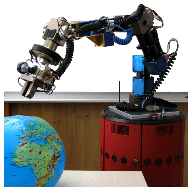
UNI FREIBURG

AIS Autonomous Intelligent Systems

Humanoid Robots Lab University of Freiburg

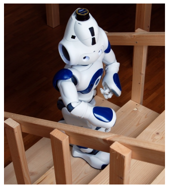
http://octomap.sf.net

Robots in 3D Environments

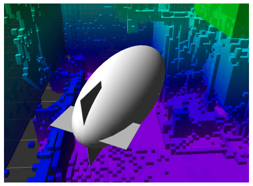


Mobile manipulation

Outdoor navigation



Humanoid robots



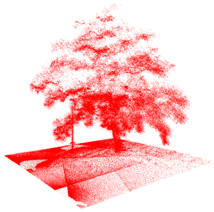
Flying robots

3D Map Requirements

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

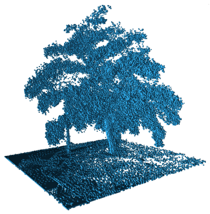
Pointclouds



- 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

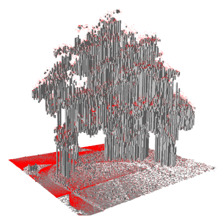


- Pro:
  - Probabilistic update
  - Constant access time
- Contra:
  - Memory requirement
    - Extent of map has to be known
    - Complete map is allocated in memory

Map Representations

2.5D Maps

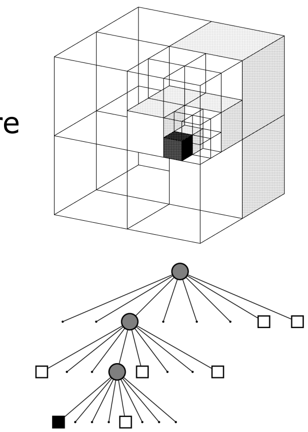
- 2D grid
- Height value(s) in each cell



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

Map Representations

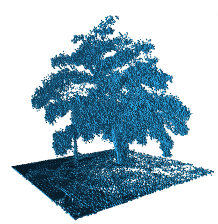
Octrees



- 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 (memory, update, map files, …)

OctoMap Framework

- Based on octrees
- Probabilistic representation of occupancy including unknown
- Supports multi-resolution map queries
- Lossless compression
- Compact map files
- Open source implementation as C++ library available at http://octomap.sf.net

Probabilistic Map Update

- Occupancy modeled as recursive binary Bayes filter [Moravec '85]

$$P(n \mid z_{1:t}) = \left[1 + \frac{1 - P(n \mid z_t)}{P(n \mid z_t)} \frac{1 - P(n \mid z_{1:t-1})}{P(n \mid z_{1:t-1})} \frac{P(n)}{1 - P(n)}\right]^{-1}$$

- Efficient update using log-odds notation

$$L(n \mid z_{1:t}) \;=\; L(n \mid z_{1:t-1}) + L(n \mid z_t)$$

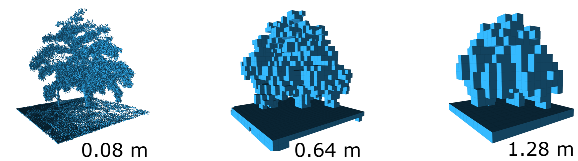
Probabilistic Map Update

- Clamping policy ensures updatability [Yguel '07]

$$L(n) \in [l_{\text{min}}, l_{\text{max}}]$$

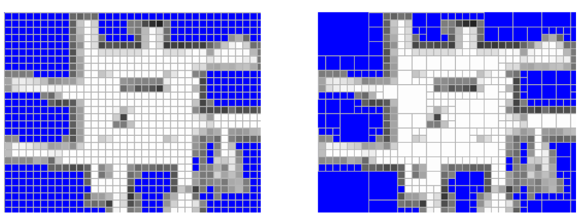
- Update of inner nodes enables multi-resolution queries

$$L(n) = \max_{i=1..8} L(n_i)$$



Lossless Map Compression

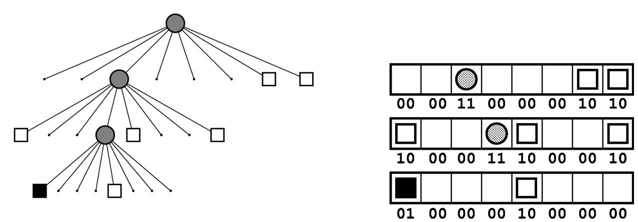
- Lossless pruning of nodes with identical children
- High compression ratios esp. in free space



[Kraetzschmar 04]

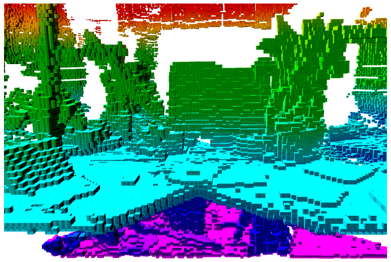
Map Files

- Maximum-likelihood map stored as compact bitstream
- Occupied, free, and unknown areas
- Very moderate space requirements (usually less than 2 MB)



Examples

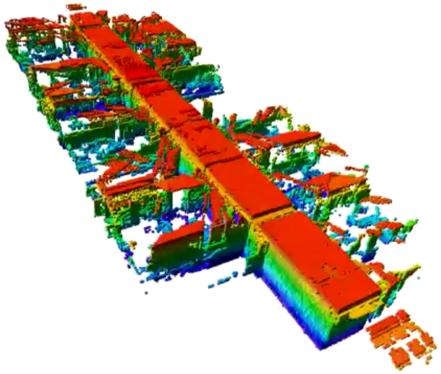
- Cluttered office environment




Map resolution: 2 cm

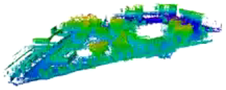
Examples: Office Building

- Freiburg, building 079



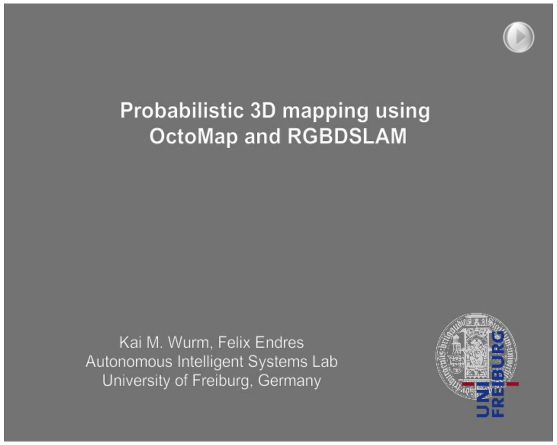
Examples: Large Outdoor Areas

- Freiburg computer science campus
  (292 x 167 x 28 m^3, 20 cm resolution)

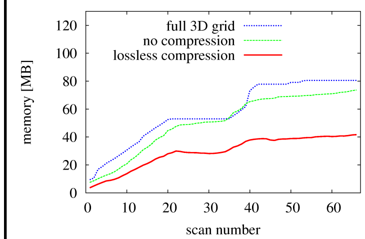


Examples: Tabletop

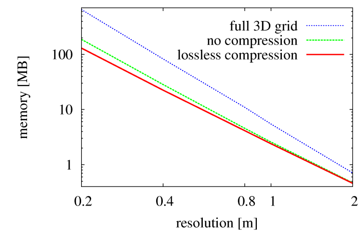
Adding Color



Memory Usage



FR-079 dataset

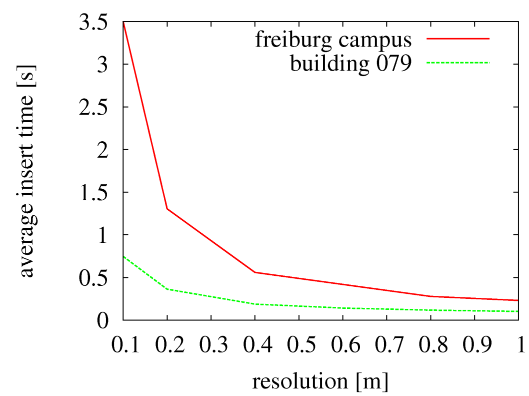


Freiburg campus outdoor dataset

Memory Usage

Map dataset	Mapped area [m^3]	Resolution [m]	Memory consumption [MB] Full grid	No compr.	Lossless compr.	File size [MB] All data	Binary
FR-079 corridor	43.8 × 18.2 × 3.3	0.05	80.54	73.64	41.70	15.80	0.67
		0.1	10.42	10.90	7.25	2.71	0.14
Freiburg outdoor	292 × 167 × 28	0.20	654.42	188.09	130.39	49.75	2.00
		0.80	10.96	4.56	4.13	1.53	0.08
New College (Epoch C)	250 × 161 × 33	0.20	637.48	91.43	50.70	18.71	0.99
		0.80	10.21	2.35	1.81	0.64	0.05

Insert time for 100,000 beams



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