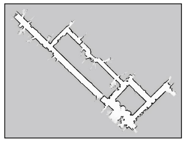
CSE-571
Robotics

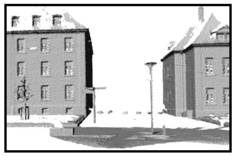
Mapping

1

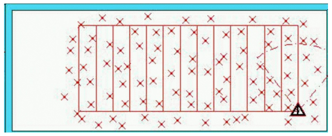
Types of SLAM-Problems

Grid maps or scans





Sparse landmarks



RGB / Depth Maps

2

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?

3

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!

4

Updating Occupancy Grid Maps

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

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

- Additional assumption: Map is static

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

- Log odds representation:

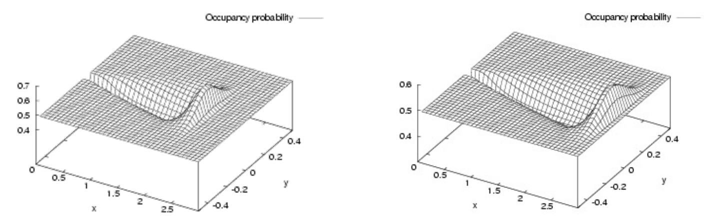
$$l_{t,i} = \log\left(\frac{p(m_i|z_{1:t}, x_{1:t})}{1 - p(m_i|z_{1:t}, x_{1:t})}\right)$$

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

5

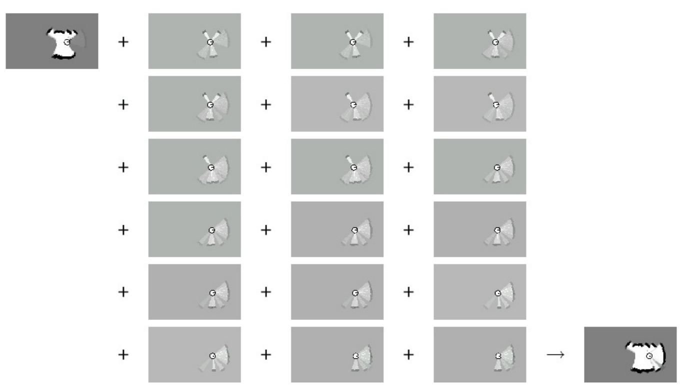
Inverse Sensor Model for Occupancy Grid Maps

Combination of linear function and Gaussian:



6

Incremental Updating of Occupancy Grids (Example)



7

Alternative: Simple Counting

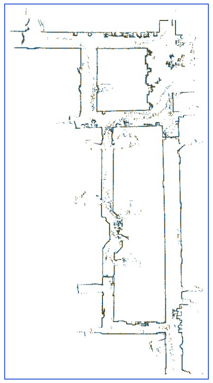
- For every cell count
  - hits(*x*,*y*): number of cases where a beam ended at <*x*,*y*>
  - misses(*x*,*y*): number of cases where a beam passed through <*x*,*y*>

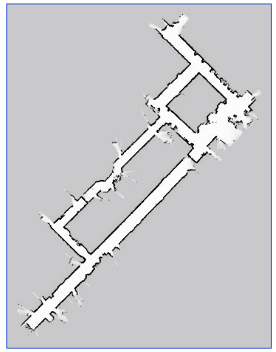
$$\boxed{Bel(m^{[xy]}) = \frac{\text{hits}(x, y)}{\text{hits}(x, y) + \text{misses}(x, y)}}$$

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

8

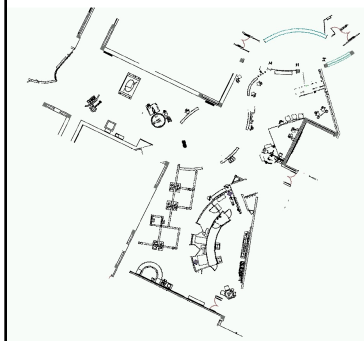
Occupancy Grids: From scans to maps



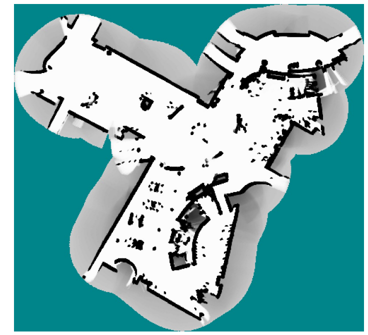


9

Tech Museum, San Jose



CAD map



occupancy grid map

10

OctoMap

A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems

UNI FREIBURG

AIS Autonomous Intelligent Systems

Humanoid Robots Lab

University of Freiburg

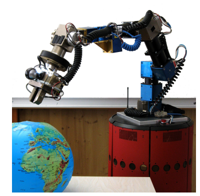
K.M. Wurm, *A. Hornung*,
M. Bennewitz, C. Stachniss, W. Burgard

University of Freiburg, Germany

http://octomap.sf.net

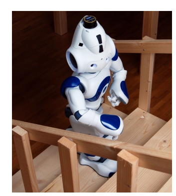
11

Robots in 3D Environments

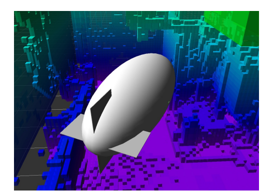


Mobile manipulation

Outdoor navigation



Humanoid robots



Flying robots

12

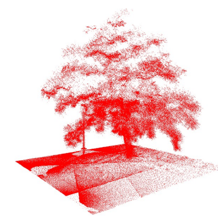
3D Map Requirements

- 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

13

Map Representations

Pointclouds

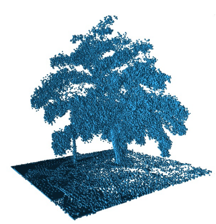


- Pro:
  - No discretization of data
  - Mapped area not limited
- Contra:
  - Unbounded memory usage
  - No direct representation of free or unknown space

14

Map Representations

3D voxel grids

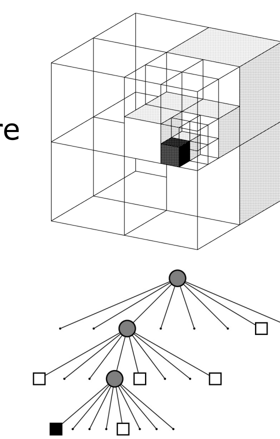


- Pro:
  - Probabilistic update
  - Constant access time
- Contra:
  - Memory requirement
    - Extent of map has to be known
    - Complete map is allocated in memory

15

Map Representations

Octrees



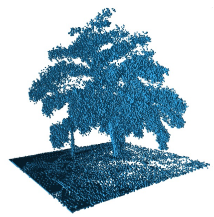
- Tree-based data structure
- Recursive subdivision of space into octants
- Volumes allocated as needed
- Multi-resolution

16

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

17

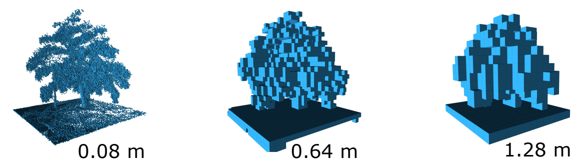
Probabilistic Map Update

- Clamping policy ensures updatability [Yguel '07]

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

- Update of inner nodes enables multi-resolution queries

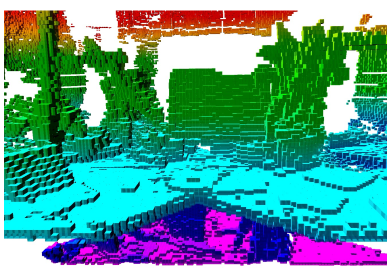
$$L(n) = \max_{i=1..8} L(n_i)$$



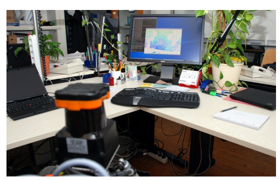
18

Examples

- Cluttered office environment



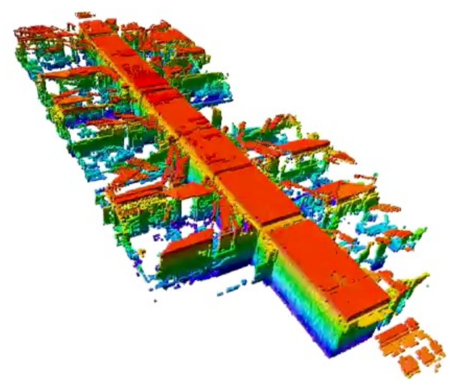
Map resolution: 2 cm



19

Examples: Office Building

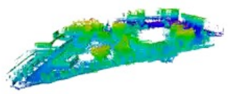
- Freiburg, building 079



20

Examples: Large Outdoor Areas

- Freiburg computer science campus
  (292 x 167 x 28 m³, 20 cm resolution)

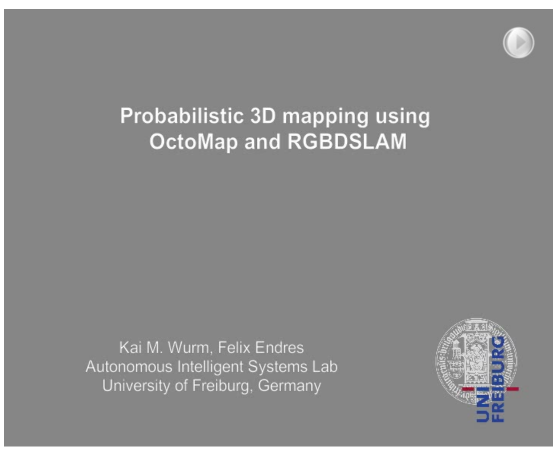


21

Examples: Tabletop

22

Adding Color



23

Memory Usage

Map dataset	Mapped area [m^3]	Resolution [m]	Memory consumption [MB]			File size [MB]	
			Full grid	No compr.	Lossless compr.	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

24