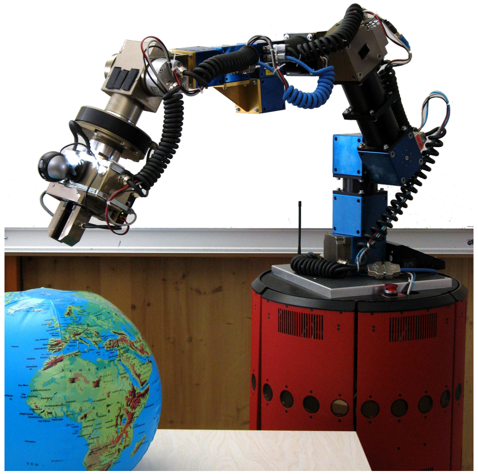
Techniques for 3D Mapping

Wolfram Burgard

Dept. of Computer Science, University of Freiburg, Germany

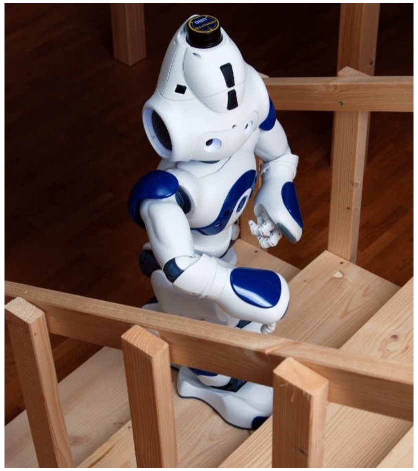
Joint work with: D. Dolgov, B. Frank, G. Grisetti, D. Haehnel, K. Kersting, R. Kuemmerle, P. Pfaff, C. Plagemann, C. Stachnis, R. Triebel, K. Wurm, …

Robots in 3D Environments

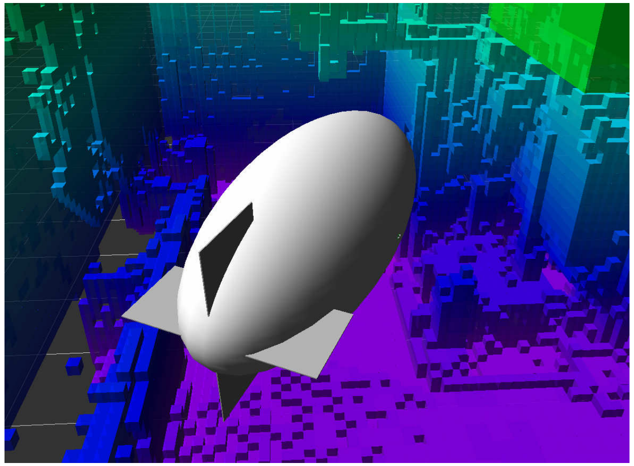


Mobile manipulation

Outdoor navigation



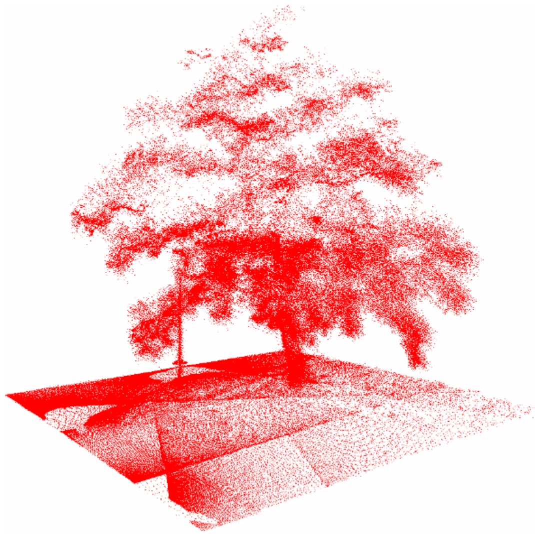
Humanoid robots



Flying robots

Map Representations

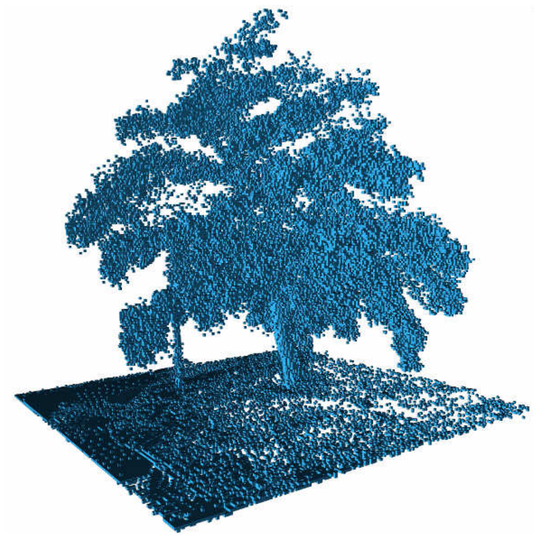
Pointclouds



- 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

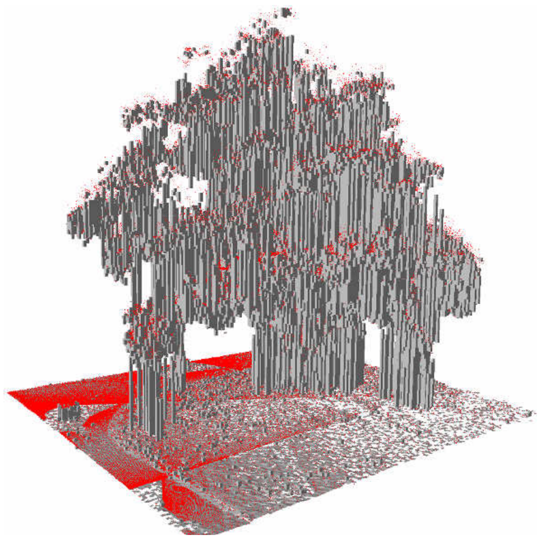


- Pro:
  - Constant access time
  - Probabilistic update
- Contra:
  - Memory requirement
    - Complete map is allocated in memory
  - Extent of map has to be known/guessed

Map Representations

2.5D Maps

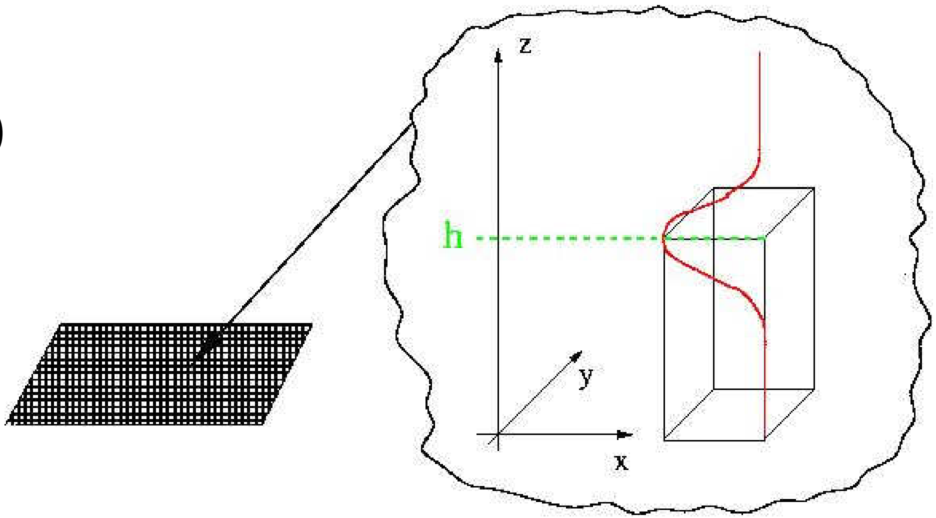
- 2D grid
- Height value(s) in each cell



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

Elevation Maps

- 2D grid which additionally stores a height (elevation) for each cell
- Use a Kalman Filter to estimate the elevation.
- Elevation h = μ.



Pros:

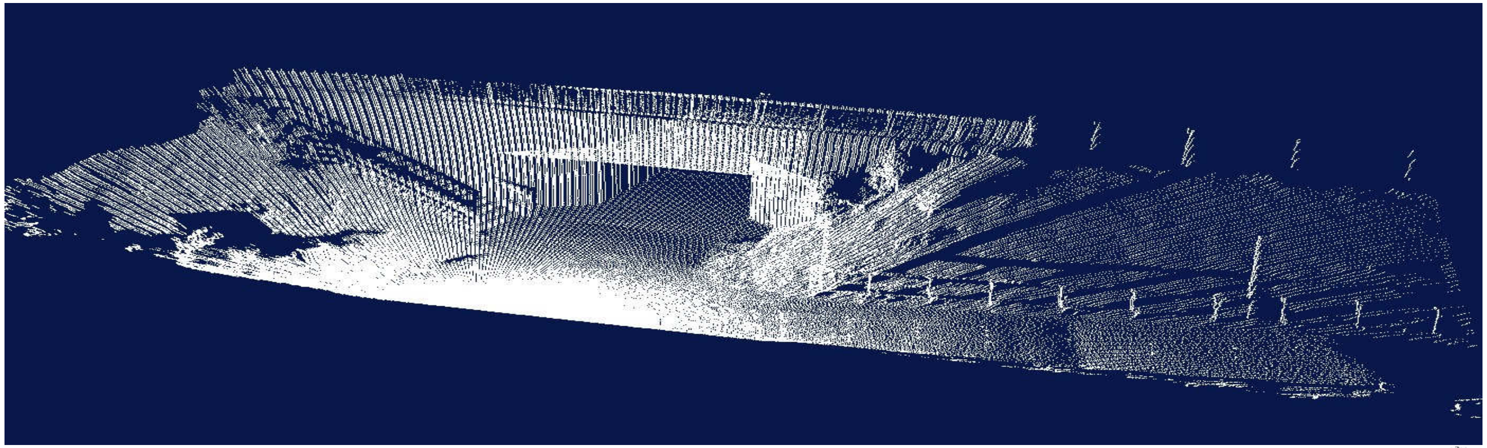
- 2½-D representation (vs. 3D for grids)
- Constant time access
- Straightforward computation of cell traversibility
- Path planning like in 2D

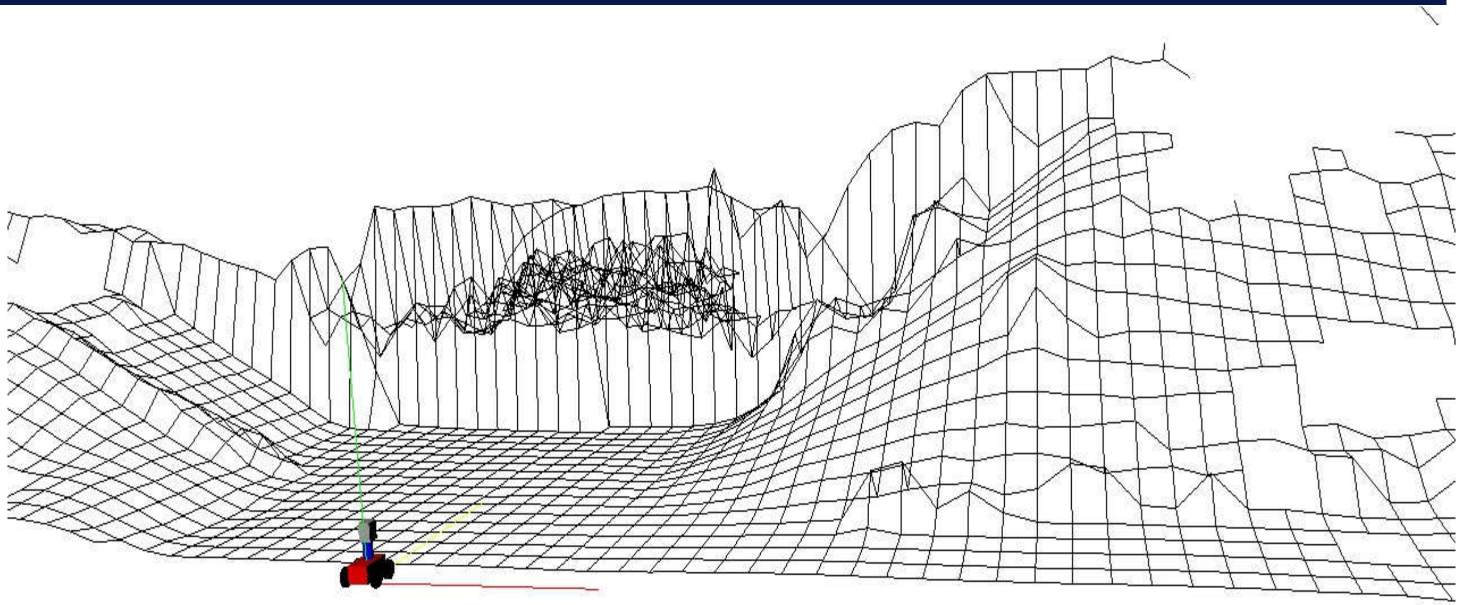
Cons:

- No vertical objects
- Only one level
- μ depends on viewpoint

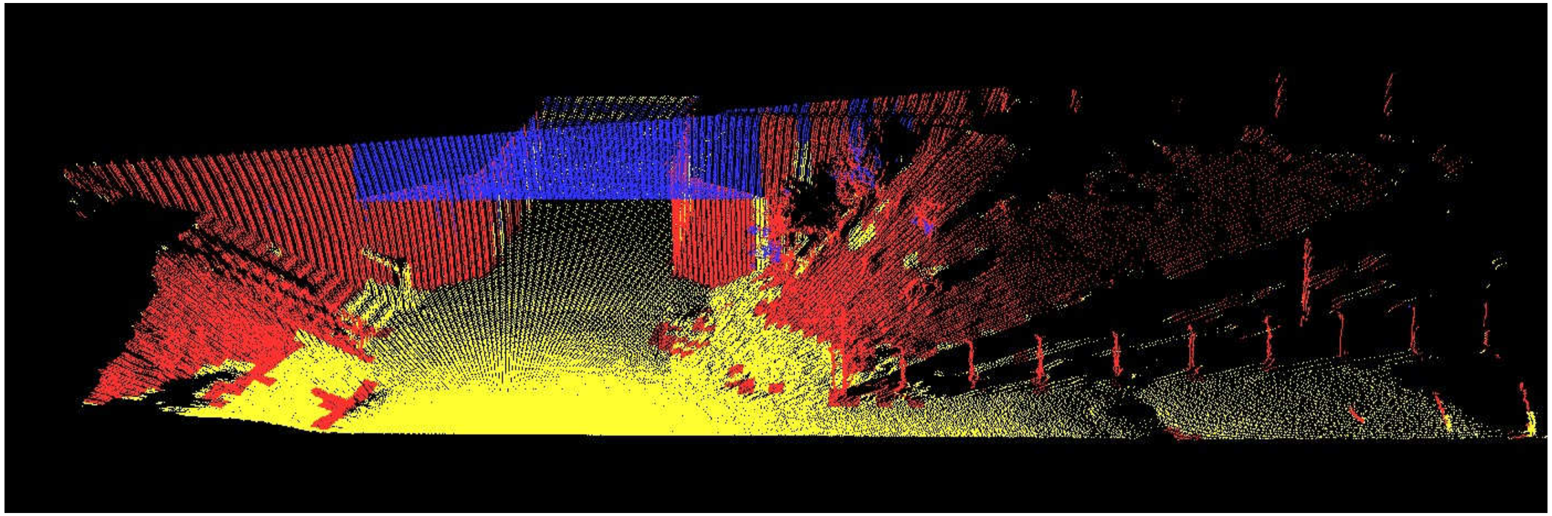
→ Extended Elevation Maps

Typical Elevation Map



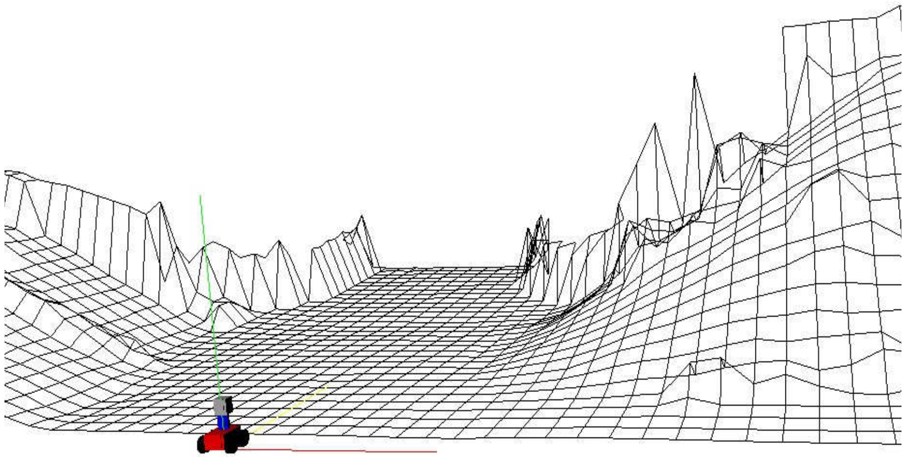


Extended Elevation Map

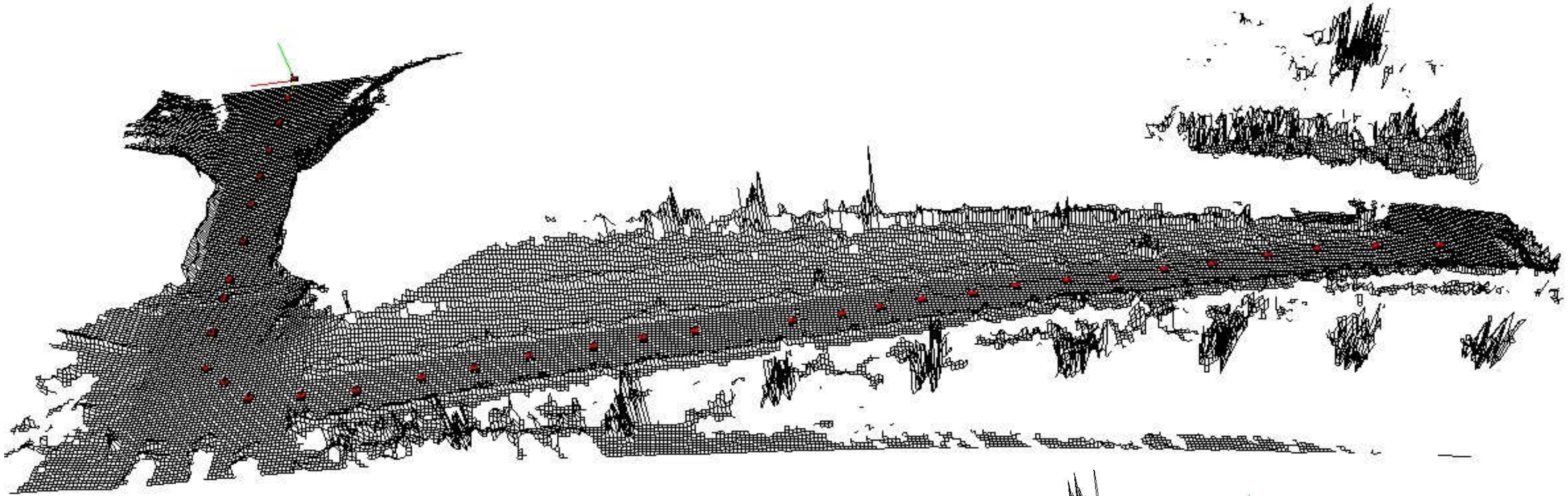


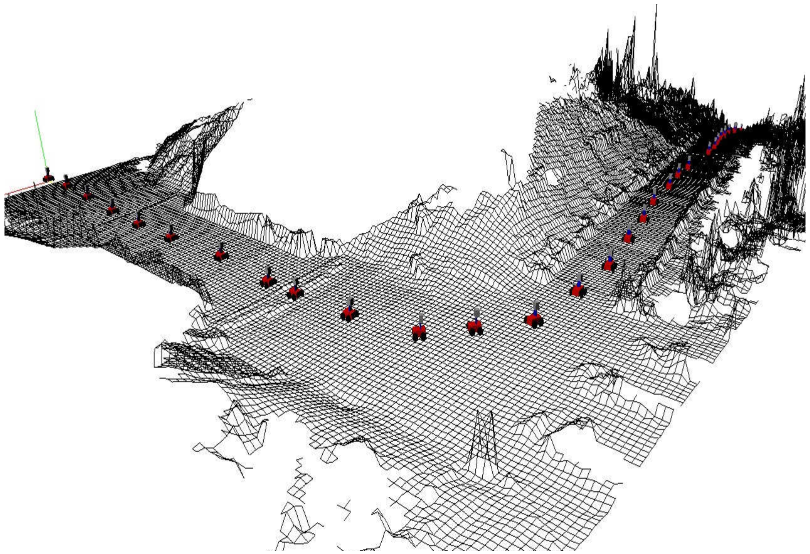
- Cells with vertical objects **(red)**
- Cells with a big vertical gap e.g. windows, bridges, door frames **(blue)**
- Cells, seen from above **(yellow)**

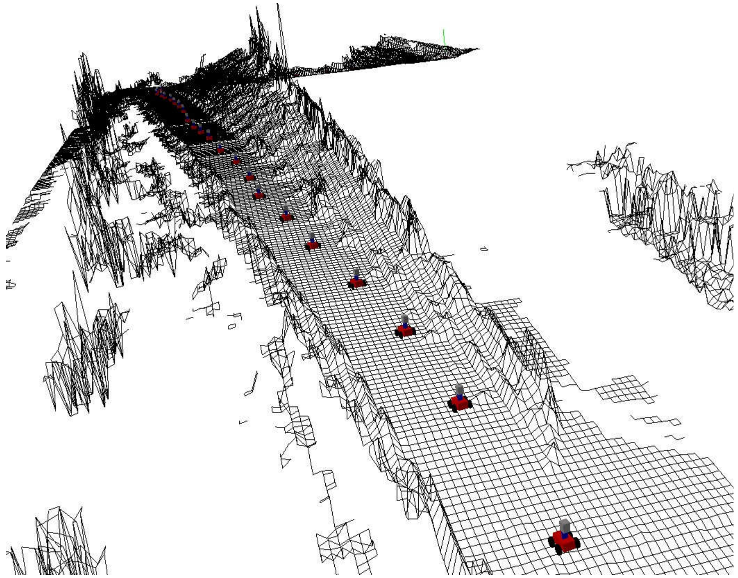
→ store gaps in cells to determine traversibility



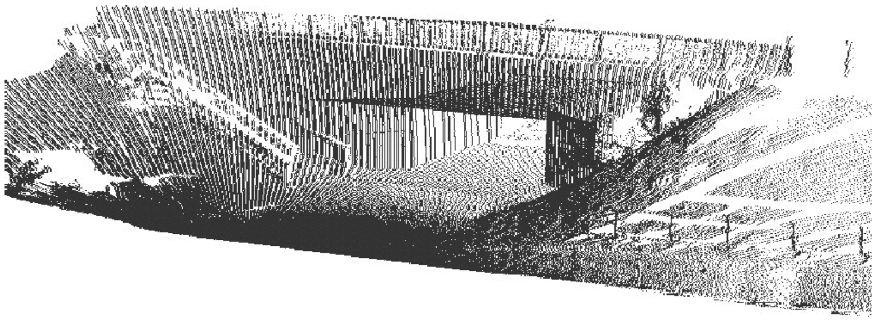
Multiple Elevation Maps



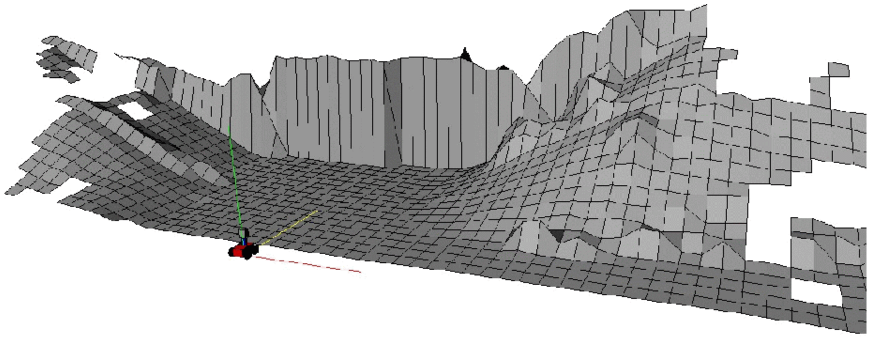




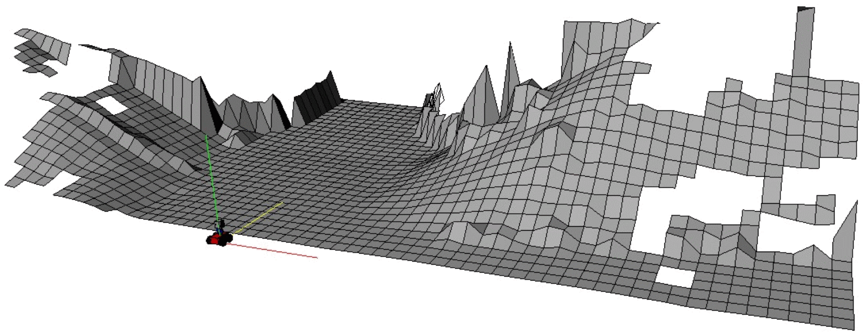
Terrain Maps



Point Cloud



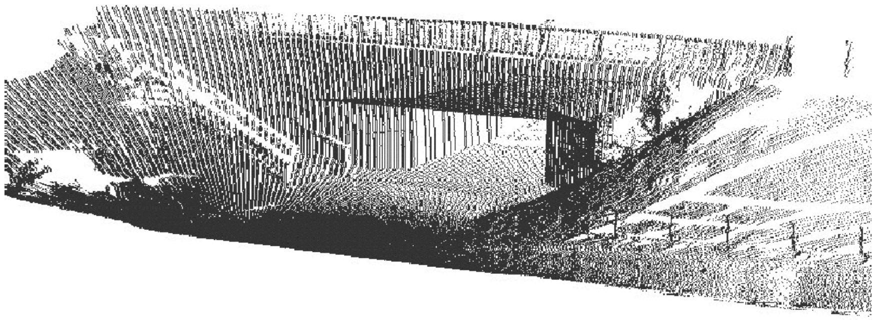
Standard Elevation Map



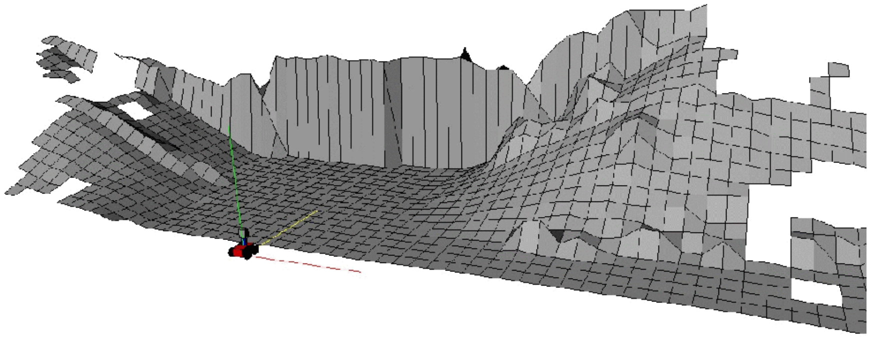
Extended elevation map

+ Planning with underpasses possible (cells with vertical gaps)
− No paths *passing under* and *crossing over* bridges possible (only one level per grid cell)

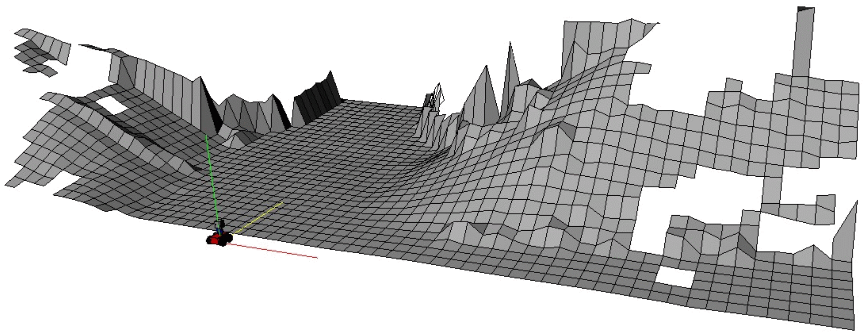
Terrain Maps



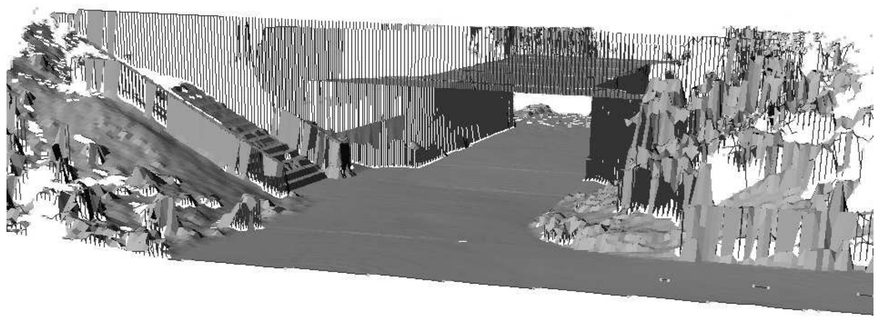
Point cloud



Standard elevation map

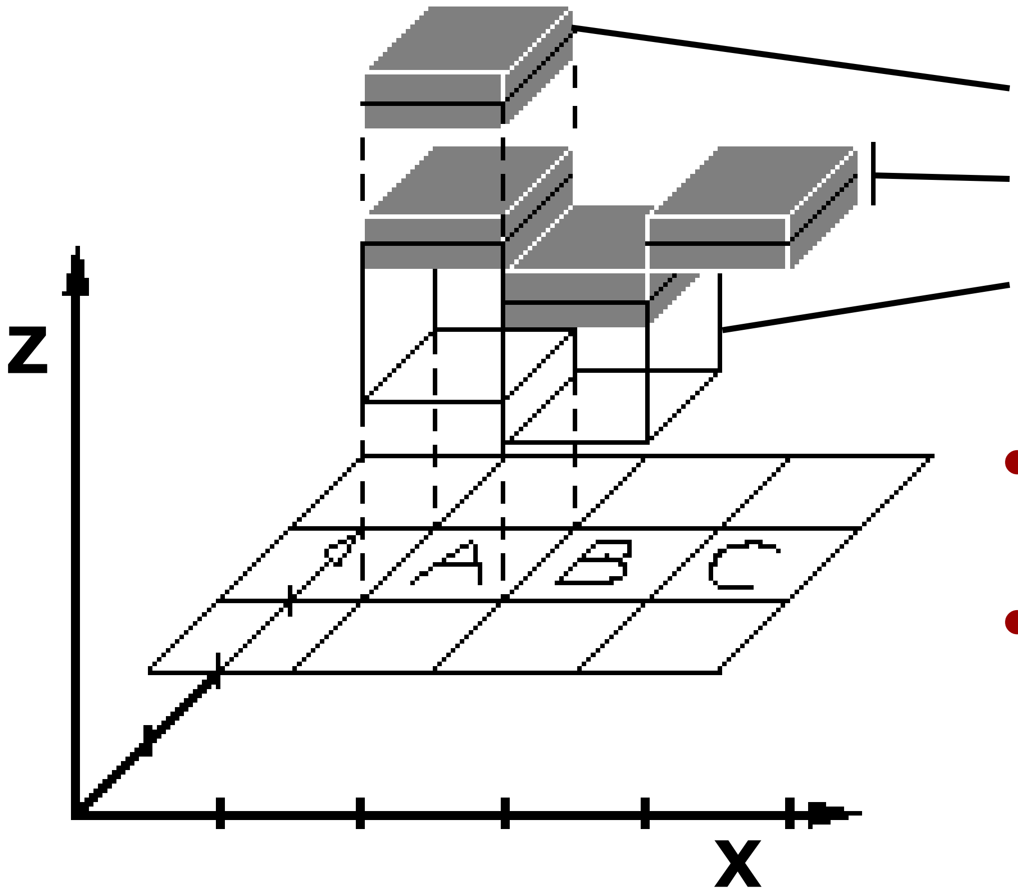


Extended elevation map



Multi-level surface map

MLS Map Representation



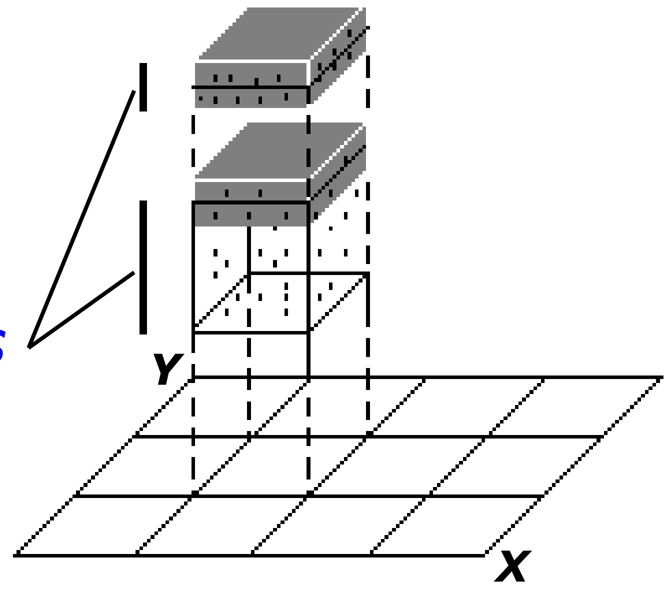
Each 2D *cell* stores various *patches* consisting of:

- A height mean μ
- A height variance σ
- A depth value d

- A *patch* can have no depth (flat objects, e.g., floor)
- A *cell* can have one or many patches (vertical gap cells, e.g., bridges)

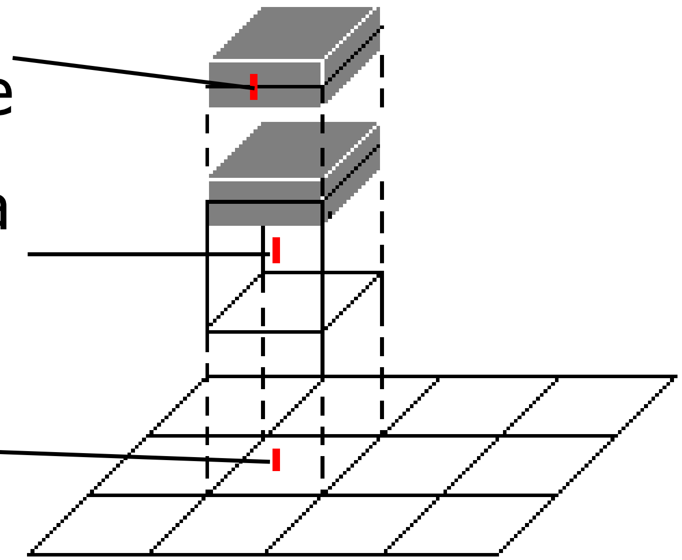
From Point Clouds to MLS Maps

- Map creation:
  - Determine xy cell for each point
  - Compute vertical *intervals*
  - Classify into *vertical* and *horizontal* intervals
  - Apply Kalman update rule to all measurements in horizontal intervals (patches)
  - Use highest measurement as mean in vertical intervals

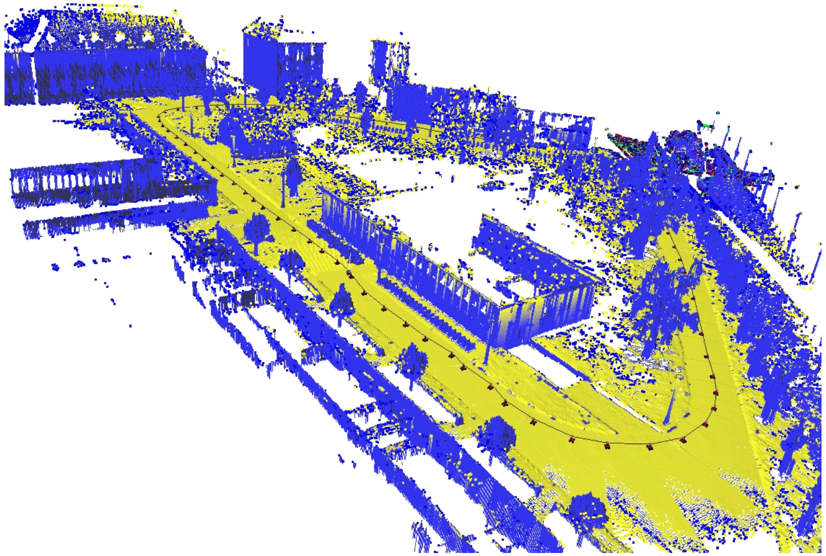


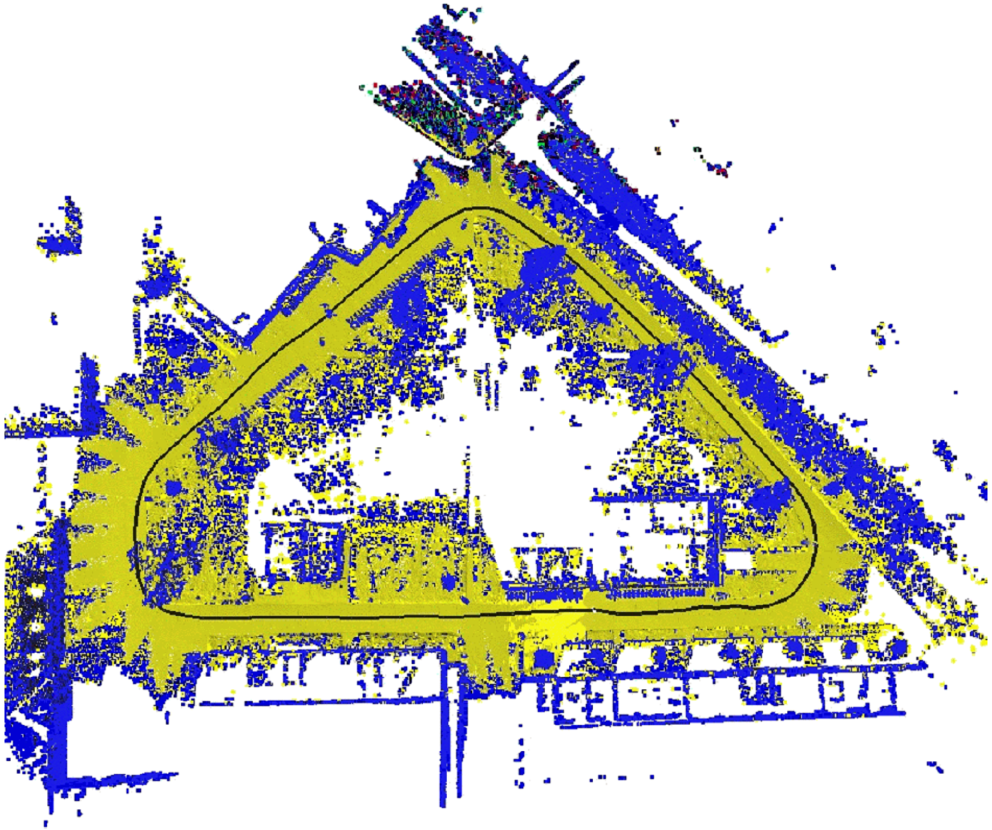
Map Update

- Given: new measurement $z=(\boldsymbol{p},\sigma)$ with variance
- Determine the corresponding cell for *z*
- Find closest surface patch in the cell
- If *z* is inside 3 variances of the patch, do Kalman update
- If *z* is in occupied region of a surface patch, disregard it
- Otherwise, create a new surface patch from *z*



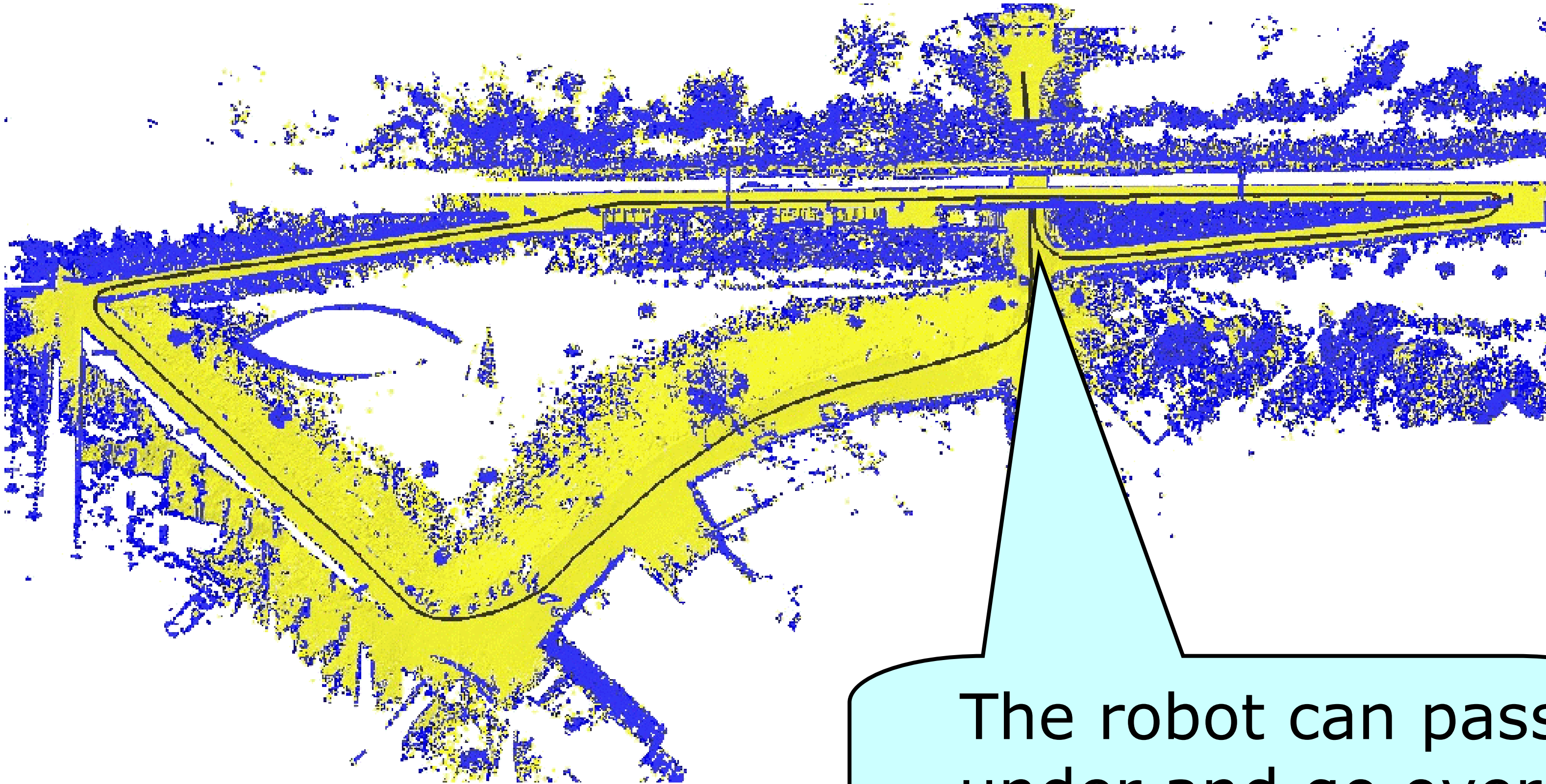
Results





- Map size: 195 by 146 *m*
- Cell resolution: 10 *cm*
- Number of data points: 20,207,000

Results

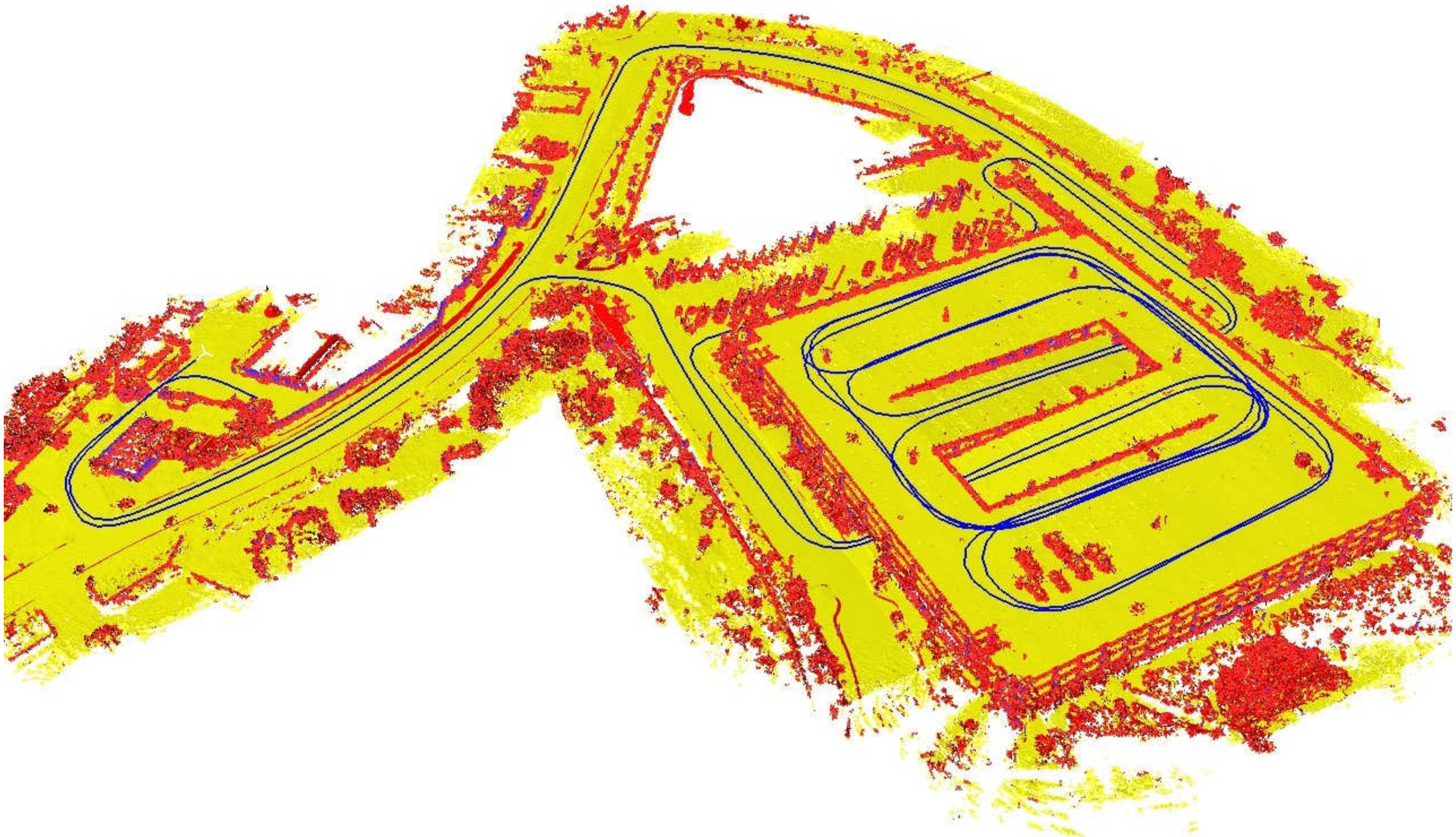


- Map size: 299 by 147 *m*
- Cell resolution: 10 *cm*
- Number of data points: 45,000,000

Experiments with a Car

- Task: Reach a parking spot on the upper level.

MLS Map of the Parking Garage

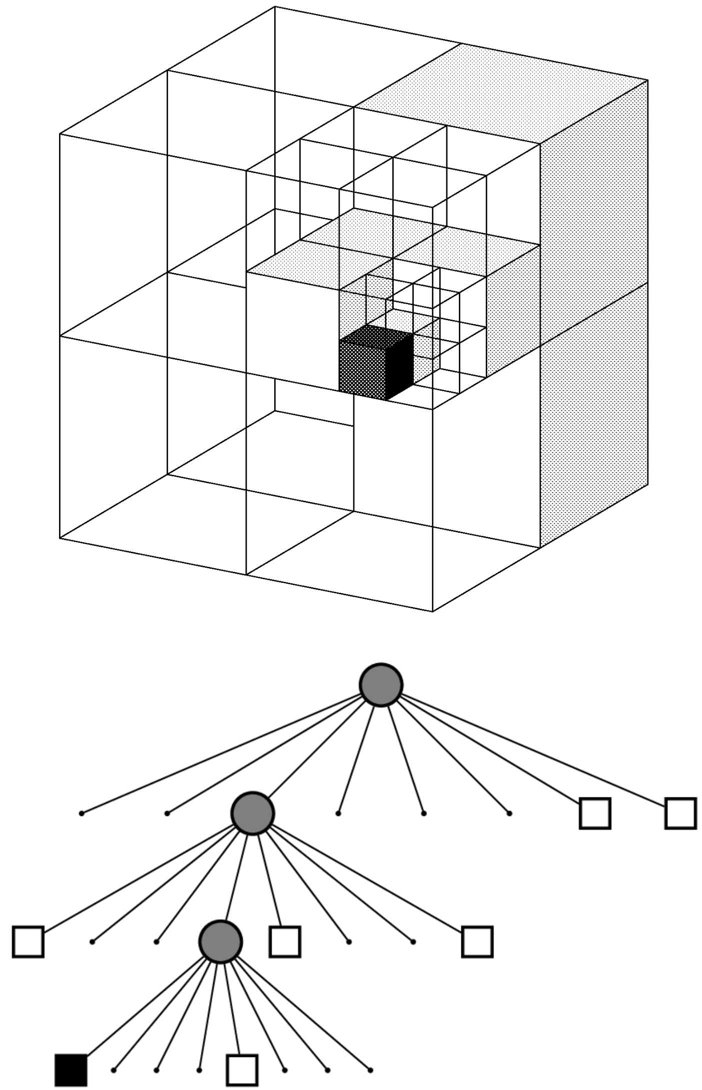


Map Representations

Octrees

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

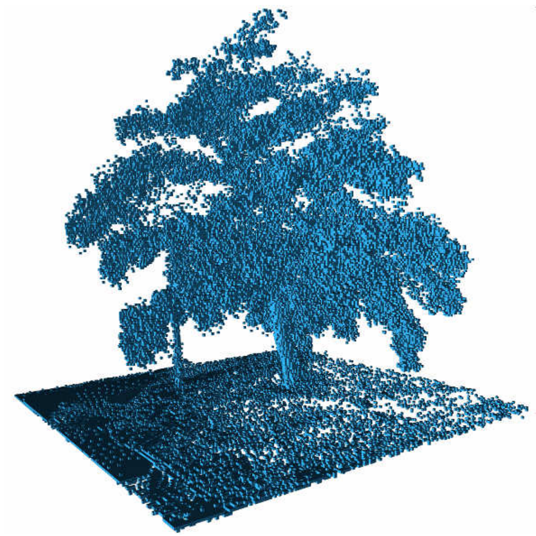
➡ Smart 3D grid



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
- Memory efficient
- Compact map files
- Optimized for runtime
- 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

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

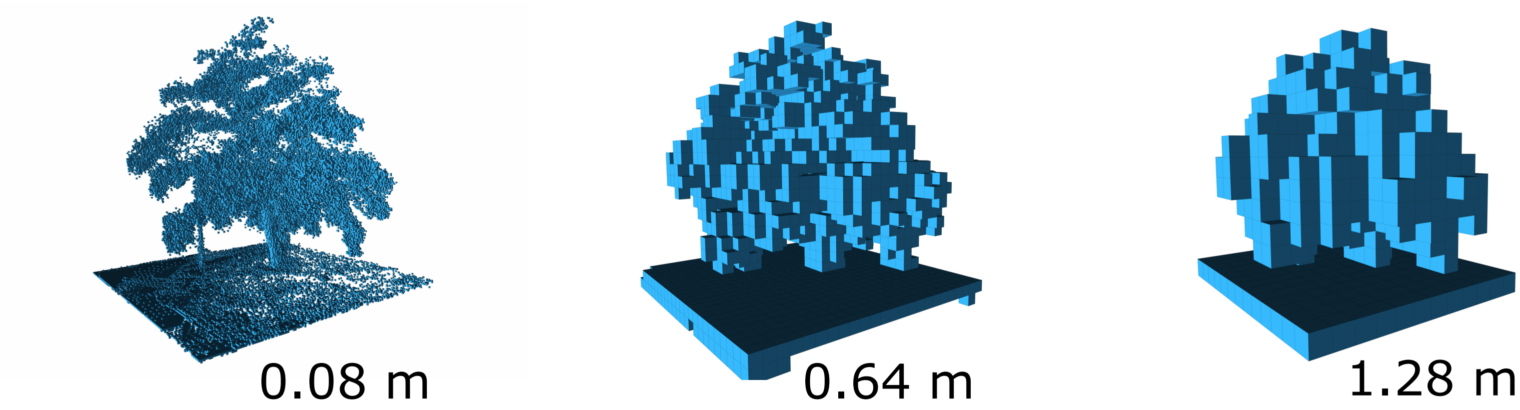
Probabilistic Map Update

- Clamping policy ensures updatability [Ygues '07]

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

- Update of inner nodes enables multi-resolution queries

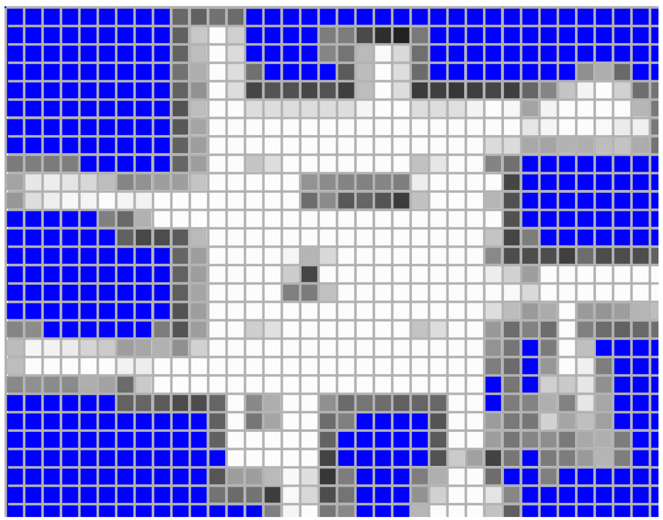
$$L(n) = \max_{i=1..8} L(n_i)$$

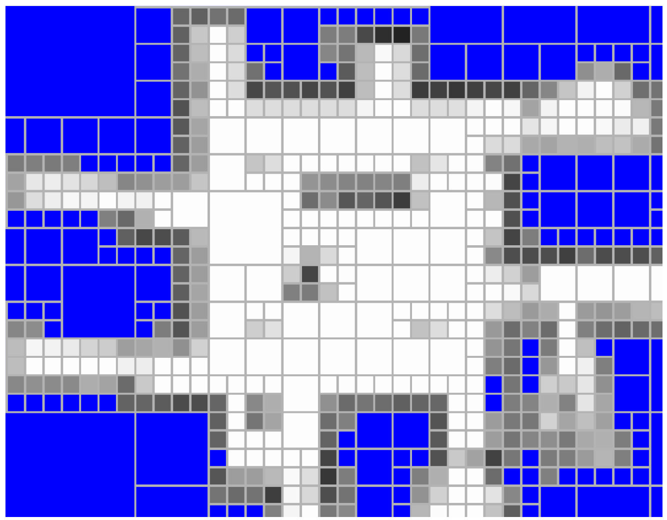


Lossless Map Compression

- Lossless pruning of nodes with identical children

- High compression ratios esp. in free space

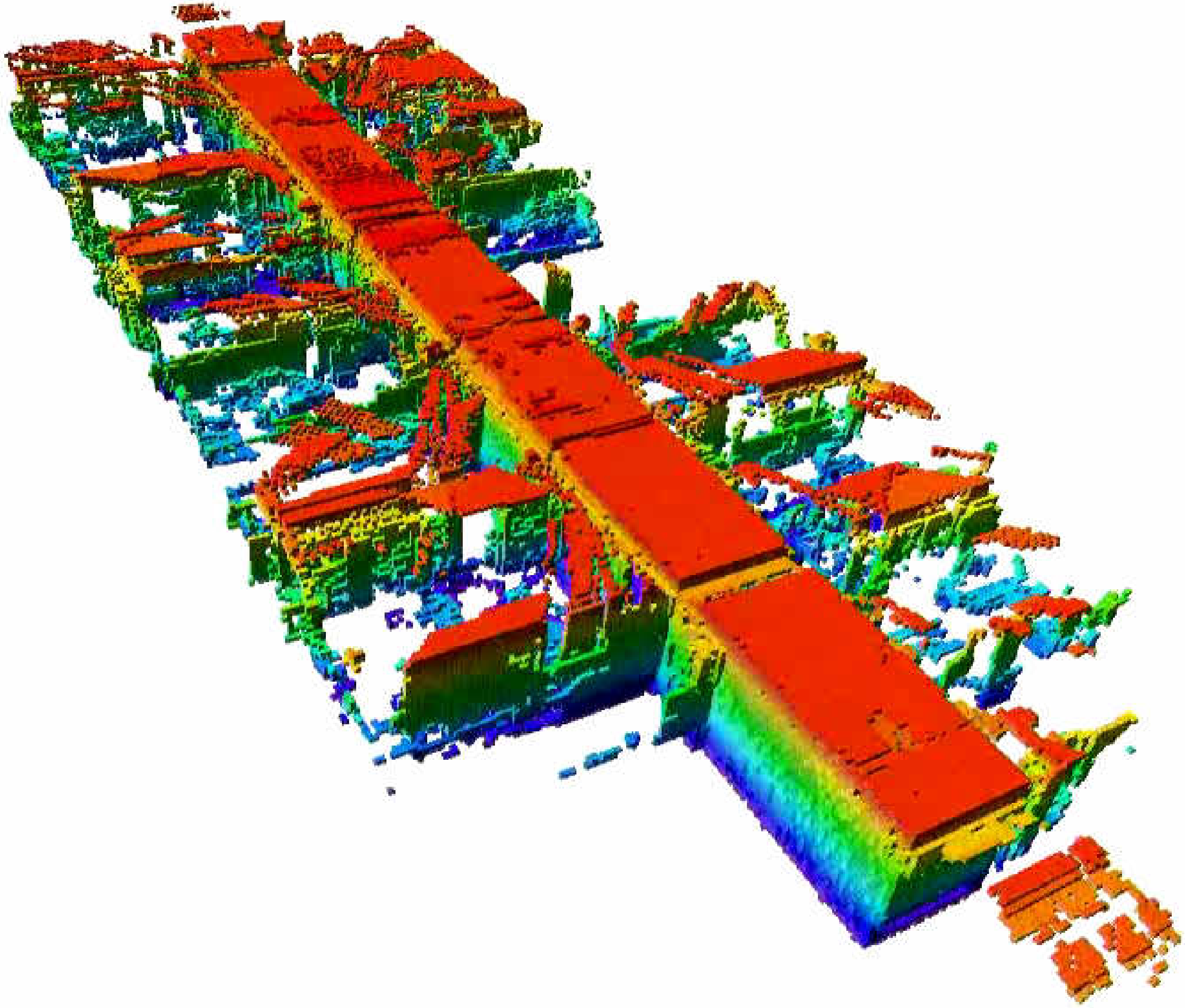




[Kraetzschmar 04]

Video: Office Building

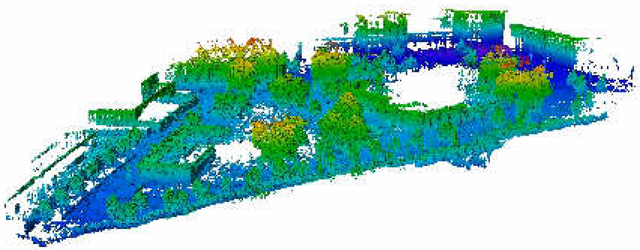
- Freiburg, building 079



Video: Large Outdoor Areas

- Freiburg computer science campus

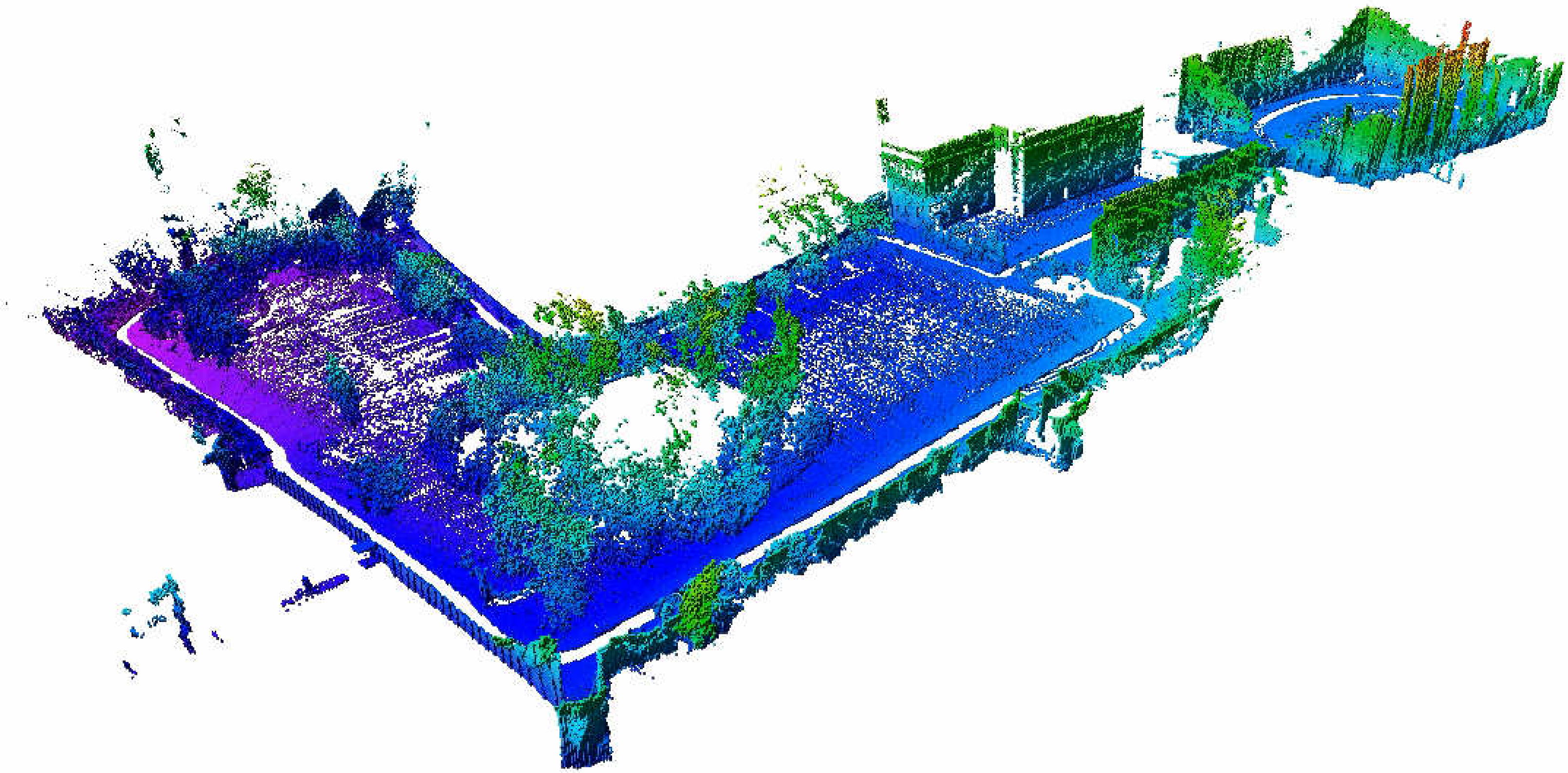
  (292 x 167 x 28 m³, 20 cm resolution)



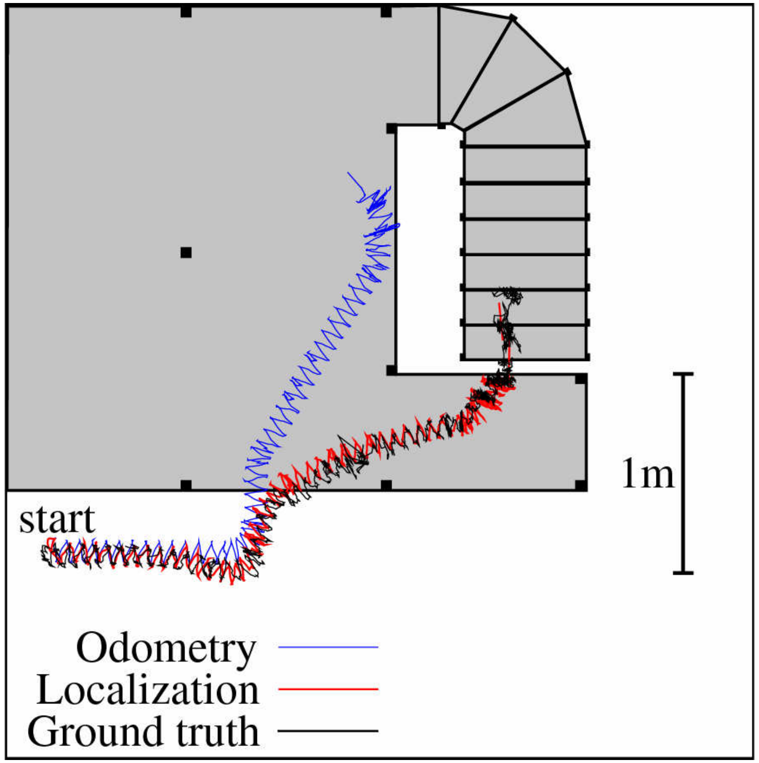
Video: Large Outdoor Areas

- Oxford *New College* dataset (Epoch C)

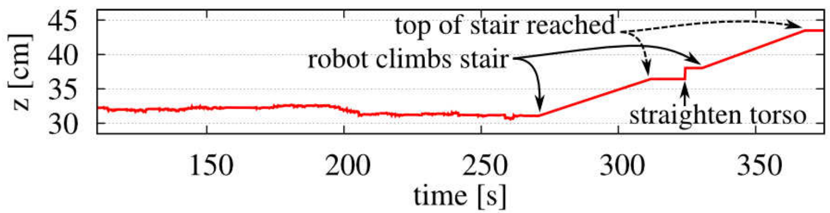
  (250 x 161 x 33 m³, 20 cm resolution)



6D Localization with Humanoid Robot



Goal: Accurate pose tracking while walking and climbing stairs



Localization (video)