

# EXTRACTING DRIVABLE SURFACES IN OUTDOOR 6D SLAM

Andreas Nüchter, Kai Lingemann, and Joachim Hertzberg  
University of Osnabrück  
Institute for Computer Science  
Knowledge-Based Systems Research Group  
D-49069 Osnabrück, Germany  
`{nuechter,lingemann,hertzberg}@informatik.uni-osnabrueck.de`

**Speaker:** Joachim Hertzberg

**Topic:** Simultaneous Localization and Mapping (SLAM)

**Keywords:** Outdoor Driving, 6D SLAM, 3D Mapping, Kurt3D

**Abstract.** A basic issue of mobile robotics is generating environment maps automatically. Outdoor terrain is challenging since the ground is uneven and the surrounding is structured irregularly. In earlier work, we have introduced 6D SLAM (Simultaneous Localization and Mapping) as a method to taking all six DOF of robot poses ( $x$ ,  $y$ ,  $z$  translation; roll, pitch, yaw angles) into account. This paper adds to 6D SLAM a method for extracting drivable surfaces in the 3D maps while they are being generated. Experiments have been made in a Botanical Garden, with drivable surfaces consisting of gravel paths or lawn, both involving significant slope.

## Introduction

Digital 3D models of the environment are needed in rescue, exploration and inspection robotics, industrial automation, facility management, agriculture and architecture. Many robotic tasks require highly precise environment maps as well. Building them manually is tedious: Thrun et al. report a time of about one week hard work for creating a 2D map of the museum in Bonn for RHINO [26]. It is even harder when 3D maps are needed, and is getting nearly impossible when mapping general outdoor environments. Therefore, automatic mapping is crucial in robotics. Autonomous mobile robots equipped with 3D laser scanners are well suited for the gaging task [18]. Contrary to indoor applications, robot poses in natural outdoor environments also involve pitch, roll and elevation, turning pose estimation into a problem in six mathematical dimensions. 6D SLAM [18] considers all of these six dimensions for the robot pose while generating 3D maps. In our experiments we use the mobile robot Kurt3D (Fig. 1, and Fig. 5, left). It is equipped with a 3D laser scanner, which is built on the basis of a SICK 2D range finder by extension with a mount and a small servo motor [17, 23]).



Figure 1: The mobile robot Kurt3D.

Figure 1, and Fig. 5, left). It is equipped with a 3D laser scanner, which is built on the basis of a SICK 2D range finder by extension with a mount and a small servo motor [17, 23]).

Mapping autonomously outdoor terrain adds another complication that is not present in typical indoor SLAM: Finding of drivable surfaces ahead. Indoor SLAM can typically assume that the robot may get along and extend its map wherever no obstacle is visible. This strategy has always required some care in the vicinity of staircases, but is good enough in many buildings. In outdoor terrain, actively looking for navigable surface is mandatory, as ragged ground, potholes, or steps are to be expected at any place.

This paper adds to our previous work on 6D SLAM the functionality of detecting geometrically navigable surfaces in the 3D map that is being built. This information may then be used to determine the next pose or poses to steer to. To make the paper self-sufficient, we include a brief review of the complete 6D SLAM process in which the surface interpretation process is embedded; for details of 6D SLAM, please refer to previous publications, i.e., [17,18].

## Related Work

**SLAM.** SLAM algorithms differ depending on the map type. State of the art for metric maps are probabilistic methods, where the robot has probabilistic motion and uncertain perception models. By integrating these two distributions with a Bayes filter, e.g., Kalman or particle filter, it is possible to localize the robot. Mapping is often regarded as an extension to this estimation problem. Beside the robot pose, positions of landmarks are estimated. Closed loops, i.e., a second encounter of a previously visited area of the environment, play a special role here. Once detected, they enable the algorithms to bound the error by deforming the already mapped area such that a topologically consistent model is created. However, there is no guarantee for a correct model. Several strategies exist for solving SLAM. Thrun reviews in [27] existing techniques, i.e., maximum likelihood estimation [9], expectation maximization [8,28], extended Kalman filter [6] or (sparse extended) information filter [30]. In addition to these methods, FastSLAM [29] that approximates the posterior probabilities, i.e., robot poses, by particles, and the method of Lu and Milios on the basis of IDC scan matching [15] exist.

In principle, these probabilistic methods are extendable to 6D. However, no reliable feature extraction nor a strategy for reducing the computational costs of multi hypothesis tracking, e.g., FastSLAM, that grows exponentially with the degrees of freedom, has been published to our knowledge.

**3D Mapping.** Instead of using 3D scanners, which yield consistent 3D scans in the first place, some groups have attempted to build 3D volumetric representations of environments with 2D laser range finders. Thrun et al. [29], Früh et al. [10] and Zhao et al. [33] use two 2D laser scanners finders for acquiring 3D data. One laser scanner is mounted horizontally, the other vertically. The latter one grabs a vertical scan line which is transformed into 3D points based on the current robot pose. Since the vertical scanner is not able to scan sides of objects, Zhao et al. use two additional, vertically mounted 2D scanners, shifted by  $45^\circ$  to reduce occlusions [33]. The horizontal scanner is used to compute the robot pose. The precision of 3D data points depends on that pose and on the precision of the scanner.

A few other groups use highly accurate, expensive 3D laser scanners [1,11,22]. The RESOLV project aimed at modeling interiors for virtual reality and tele-presence [22]. They used a RIEGL laser range finder on robots and the ICP algorithm for scan matching [4]. The AVENUE project develops a robot for modeling urban environments [1], using a CYRAX scanner and a feature-based scan matching approach for registering the 3D scans. Nevertheless, in their recent work they do not use data of the laser scanner in the robot control architecture for localization [11]. The group of M. Hebert has reconstructed environments using the Zoller+Fröhlich laser scanner and aims to build 3D models without initial position estimates, i.e., without odometry information [12].

Recently, different groups employ rotating SICK scanners for acquiring 3D data [13,23,31,32]. Wulf et al. let the scanner rotate around the vertical axis. They acquire 3D data while moving, thus the quality of the resulting map crucially depends on the pose estimate that is given by inertial sensors, i.e., gyros [32]. In addition, their SLAM algorithms do not consider all six degrees of freedom.

Other approaches use information of CCD-cameras that provide a view of the robot's environment [5,21]. Nevertheless, cameras are difficult to use in natural environments with changing light conditions. Camera-based approaches to 3D robot vision, e.g., stereo cameras and structure from motion, have difficulties providing reliable navigation and mapping information for a mobile robot in real-time. Thus some groups try to solve 3D modeling by using planar scanner based SLAM methods and cameras, e.g., in [5].

**Autonomous Outdoor Driving.** Much work has been done in the area of autonomous outdoor driving. Batavia and Singh [3] navigate their robot in locally smooth hilly terrain and use a yawing SICK laser range scanner in a fixed pitching angle towards the the ground. They use an object vs. freespace classification for driving. Similarly, Patel et al. [20] use also a yawable SICK scanner to classify drivable surfaces. Their work focuses on controlling the yawable scanner to acquire the necessary depth information while driving. They also use local gradients to classify drivable surfaces.

A good overview of the state of the art in military context for autonomous navigation in highly unstructured terrain is given in [16]. Two classes of algorithms are discussed: First, obstacle avoidance using ladar (for short range) or stereo camera (for long range). Second, terrain cover classification using a stereo camera for color analysis or ladar for range texture analysis. The advantages and disadvantages of each kind of sensor for different applications is discussed.

## The Context: Outdoor SPLAM

Our drivable surface extraction method is part of an overall robot control system for Simultaneous Planning, Localization, and Mapping (SPLAM) in outdoor environments, which is currently under development. Before we describe the method, we will sketch this context, into which it has to fit.

The general idea is to do SLAM indoor or outdoor or in mixed indoor/outdoor environments, and integrate into it a planning process for selecting the next pose to move to for making the map under development more complete. This planning process should generate the next view pose and plan the path to go there under reactive execution; alternatively, it should be able to respond to other requests that the robot receives, interrupting the SLAM process for some time. The main difference to state-of-the-art SLAM, as briefly recapitulated in the previous section, lies in two points: First, we are not restricted to 2D maps of indoor environments, but do full 6D SLAM generating 3D maps; and second, we integrate into it a pose planning process allowing the SLAM process to be optimized explicitly according to criteria like expected time or expected path length. In previous work [24], we have demonstrated an instance of that scenario, which was restricted to a flat and continuous floor (i.e., regular indoor environments), and which has used a specialized next-best-view planner for the planning part.

Our outdoor SPLAM procedure consists of the following subtasks:

1. Extrapolate the odometry readings to all six degrees of freedom using previous registration matrices [18].
2. Find a heuristic pose update using octrees [18].
3. Use the ICP algorithm [4] to match the 3D scans. Hereby use point reduction and approximate  $k$ D-tree search to accelerate the scan matching [17].
4. Extract the drivable surface from the current 3D model.
5. Plan the next scan pose and a path to go there from the current pose along a drivable surface.
6. Execute this path in closed-loop control mode.
7. Find “closed loops” using octrees and, if a closed loop is detected, distribute any occurring error over the 3D scans forming the loop [25].
8. Use a global relaxation method off-line to create highly precise 3D maps [24].

Of this list, items 1–6 are executed in a looping fashion, while the closed loop detection (item 7) is only executed if the current pose estimation is such that a closed loop is possible; the global relaxation (item 8) is only executed off-line on the map consisting of all scans.

As the citations in the item list suggests, we have presented solutions to many of these subtasks in previous work, drivable surface extraction being the first one open on the list. Before we present our solution to this problem in the next section, the following subsections will recapitulate our solutions to the preceding items 1–3, to set the context for the surface extraction task. Readers familiar with this previous work may safely skip the rest of this section. Our solutions to items 6 and 7 will not be recapitulated here as they are not required for understanding the contribution of this paper. Please refer to the cited literature.

**Item 1: Odometry Extrapolation.** We first extrapolate the odometry readings to all six degrees of freedom using previous registration matrices. The robot pose is the 6-vector  $\mathbf{P} = (x, y, z, \theta_x, \theta_y, \theta_z)$  or, equivalently the tuple containing the rotation matrix and translation vector, written as  $4 \times 4$  OpenGL-style matrix  $\mathbf{P}$  [7].<sup>1</sup> The change of the robot pose  $\Delta\mathbf{P}$  given the odometry information  $(x_n, z_n, \theta_{y,n}), (x_{n+1}, z_{n+1}, \theta_{y,n+1})$  and the registration matrix  $\mathbf{R}(\theta_{x,n}, \theta_{y,n}, \theta_{z,n})$  is calculated by solving:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \\ \theta_{x,n+1} \\ \theta_{y,n+1} \\ \theta_{z,n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \\ z_n \\ \theta_{x,n} \\ \theta_{y,n} \\ \theta_{z,n} \end{pmatrix} + \left( \begin{array}{c|ccc} \mathbf{R}(\theta_{x,n}, \theta_{y,n}, \theta_{z,n}) & & & \\ \hline & \mathbf{0} & & \\ & & 1 & 0 & 0 \\ & & 0 & 1 & 0 \\ & & 0 & 0 & 1 \end{array} \right) \cdot \underbrace{\begin{pmatrix} \Delta x_{n+1} \\ \Delta y_{n+1} \\ \Delta z_{n+1} \\ \Delta \theta_{x,n+1} \\ \Delta \theta_{y,n+1} \\ \Delta \theta_{z,n+1} \end{pmatrix}}_{\Delta\mathbf{P}}.$$

Therefore, calculating  $\Delta\mathbf{P}$  requires a matrix inversion. Finally, the 6D pose  $\mathbf{P}_{n+1}$  is calculated by

$$\mathbf{P}_{n+1} = \Delta\mathbf{P} \cdot \mathbf{P}_n$$

using the poses' matrix representations.

**Item 2: Heuristic Initial Estimations for ICP Scan Matching.** Initial estimations for ICP scan matching are computed with an octree based heuristic. The algorithm sets initially  $\Delta\mathbf{P}_{\text{best}}$  to the 6-vector  $(\mathbf{t}, \mathbf{R}(\theta_{x,n}, \theta_{y,n}, \theta_{z,n})) = (\mathbf{0}, \mathbf{R}(\mathbf{0}))$ . Then, an octree  $\mathfrak{O}_M$  for the  $n$ th 3D scan (model set  $M$ ) and an octree  $\mathfrak{O}_D$  for the  $(n+1)$ th 3D scan (data set  $D$ ) is generated (cf. Fig. 2). The estimation is done for search depth  $t \in [t_{\text{Start}}, \dots, t_{\text{End}}]$  in the octrees. Hereby a transformation  $\Delta\mathbf{P}_{\text{best}} = (\mathbf{t}, \mathbf{R})$  is computed as follows:

1. Calculate a maximal displacement and rotation  $\Delta\mathbf{P}_{\text{max}}$  depending on the search depth  $t$  and currently best transformation  $\Delta\mathbf{P}_{\text{best}}$ .
2. For all discrete 6-tuples  $\Delta\mathbf{P}_i \in [-\Delta\mathbf{P}_{\text{max}}, \Delta\mathbf{P}_{\text{max}}]$  in the domain  $\Delta\mathbf{P} = (x, y, z, \theta_x, \theta_y, \theta_z)$  displace  $\mathfrak{O}_D$  by  $\Delta\mathbf{P}_i \cdot \Delta\mathbf{P} \cdot \mathbf{P}_n$ . Evaluate the matching of the two octrees by counting the number of overlapping cubes and save the best transformation as  $\Delta\mathbf{P}_{\text{best}}$ .

Finally, the scan pose is updated using matrix multiplication, i.e.,

$$\mathbf{P}_{n+1} = \Delta\mathbf{P}_{\text{best}} \cdot \Delta\mathbf{P} \cdot \mathbf{P}_n.$$

*Note:* Step 2 requires 6 nested loops, but the computational requirements are bounded by the coarse-to-fine strategy inherited from the octree processing. The size of the octree cubes decreases exponentially with increasing  $t$ . We start the algorithm with a cube size of  $75 \text{ cm}^3$  and stop when the cube size falls below  $10 \text{ cm}^3$ . Fig. 2 shows two 3D scans and the corresponding octrees. Furthermore, note that this heuristic works best outdoors. Due to the diversity of the environment the match of octree cubes will show a significant maximum, while indoor environments with their many geometry symmetries and similarities, e.g., in a corridor, are in danger of producing many plausible matches.

**Item 3: ICP for 3D Scan Matching.** The 3D scans have to be merged into one coordinate system. This process is called registration. The following method registers point sets in a common coordinate system. It is called *Iterative Closest Points (ICP)* algorithm [4]. Given two independently acquired sets of 3D points,  $M$  (model set) and  $D$  (data set) which correspond to a single shape, we aim to find the transformation consisting of a rotation  $\mathbf{R}$  and a translation  $\mathbf{t}$  which minimizes the following cost function:

<sup>1</sup>Note the bold-italic (vectors) and bold (matrices) notation. The conversion between vector representations, i.e., Euler angles, and matrix representations is done by algorithms from [7].

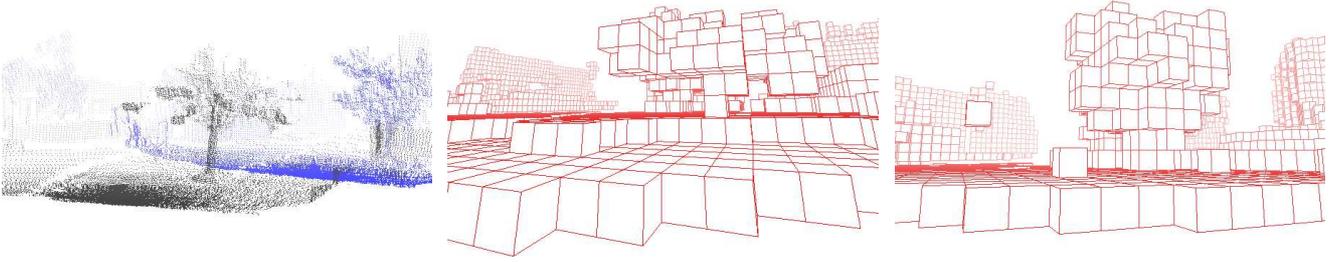


Figure 2: Left: Two 3D point clouds. Middle: Octree corresponding to the black point cloud. Right: Octree based on the blue points.

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{|M|} \sum_{j=1}^{|D|} w_{i,j} \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})\|^2. \quad (1)$$

$w_{i,j}$  is assigned 1 if the  $i$ -th point of  $M$  describes the same point in space as the  $j$ -th point of  $D$ . Otherwise  $w_{i,j}$  is 0. Two things have to be calculated: First, the corresponding points, and second, the transformation  $(\mathbf{R}, \mathbf{t})$  that minimizes  $E(\mathbf{R}, \mathbf{t})$  on the base of the corresponding points.

The ICP algorithm calculates iteratively the point correspondences. In each iteration step, the algorithm selects the closest points as correspondences and calculates the transformation  $(\mathbf{R}, \mathbf{t})$  for minimizing equation (1). The assumption is that in the last iteration step the point correspondences are correct. Besl et al. prove that the method terminates in a minimum [4]. However, this theorem does not hold in our case, since we use a maximum tolerable distance  $d_{\max}$  for associating the scan data. Such a threshold is required though, given that 3D scans overlap only partially.

In every iteration, the optimal transformation  $(\mathbf{R}, \mathbf{t})$  has to be computed. Eq. (1) can be reduced to

$$E(\mathbf{R}, \mathbf{t}) \propto \frac{1}{N} \sum_{i=1}^N \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_i + \mathbf{t})\|^2, \quad (2)$$

with  $N = \sum_{i=1}^{|M|} \sum_{j=1}^{|D|} w_{i,j}$ , since the correspondence matrix can be represented by a vector containing the point pairs.

Four direct methods are known to minimize Eq. (2) [14]. In earlier work [19, 24, 25] we used a quaternion based method [4], but the following one, based on singular value decomposition (SVD), is robust and easy to implement, thus we give a brief overview of the SVD-based algorithm. It was first published by Arun, Huang and Blostein [2]. The difficulty of this minimization problem is to enforce the orthonormality of the matrix  $\mathbf{R}$ . The first step of the computation is to decouple the calculation of the rotation  $\mathbf{R}$  from the translation  $\mathbf{t}$  using the centroids of the points belonging to the matching, i.e.,

$$\mathbf{c}_m = \frac{1}{N} \sum_{i=1}^N \mathbf{m}_i, \quad \mathbf{c}_d = \frac{1}{N} \sum_{i=1}^N \mathbf{d}_i \quad (3)$$

and

$$M' = \{\mathbf{m}'_i = \mathbf{m}_i - \mathbf{c}_m\}_{1,\dots,N}, \quad D' = \{\mathbf{d}'_i = \mathbf{d}_i - \mathbf{c}_d\}_{1,\dots,N}. \quad (4)$$

After substituting (3) and (4) into the error function,  $E(\mathbf{R}, \mathbf{t})$  Eq. (2) becomes:

$$E(\mathbf{R}, \mathbf{t}) \propto \sum_{i=1}^N \|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i\|^2 \quad \text{with} \quad \mathbf{t} = \mathbf{c}_m - \mathbf{R}\mathbf{c}_d. \quad (5)$$

The registration calculates the optimal rotation by  $\mathbf{R} = \mathbf{V}\mathbf{U}^T$ . Hereby, the matrices  $\mathbf{V}$  and  $\mathbf{U}$  are derived by the singular value decomposition  $\mathbf{H} = \mathbf{U}\mathbf{A}\mathbf{V}^T$  of a correlation matrix  $\mathbf{H}$ . This  $3 \times 3$  matrix  $\mathbf{H}$  is given by

$$\mathbf{H} = \sum_{i=1}^N \mathbf{d}'_i \mathbf{m}'_i{}^T = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{pmatrix}, \quad (6)$$

with  $S_{xx} = \sum_{i=1}^N m'_{ix} d'_{ix}$ ,  $S_{xy} = \sum_{i=1}^N m'_{ix} d'_{iy}$ ,  $\dots$  [2].

We have proposed and evaluated algorithms to accelerate ICP, namely point reduction and approximate  $k$ d-trees [19, 24, 25]. They are used here, too.

## Surface Extraction

We now turn to the issue of detecting drivable surface in the 3D point clouds and their registration into a global map, as delivered by the previously performed and described steps. We will first deal with single 3D scans, and then go into spreading this out to the complete 3D model.

**Labeling Surface Points in Single Scans.** Based on the idea by Wulf et al. [32] we have designed an algorithm for labeling floor points in 3D scans. This is done by computing the gradient between a point  $\mathbf{p}_{i,j} = (\varphi_i, r_{i,j}, z_{i,j})$ , given in a cylindrical coordinate system, and its  $k$ -th neighbor in measurement order within the vertical sweep plane, i.e., a search region around  $\varphi_i$ , according to the following equation (cf. Fig. 3 middle, vertical cylindrical coordinate system):

$$\alpha_{i,j} = \arctan \left( \frac{z_{i,j} - z_{i,j-k}}{r_{i,j} - r_{i,j-k}} \right)$$

with

$$-\frac{1}{2}\pi \leq \alpha_{i,j} < \frac{3}{2}\pi.$$

In comparison with a fixed threshold  $\tau$  (here:  $\tau = 20^\circ$ ), each 3D point is assigned to one of the following three groups, which has proved to be robust against uneven and non-horizontal ground:

1.  $\alpha_{i,j} < \tau$ :  $\mathbf{p}_{i,j}$  is a ground point
2.  $\tau \leq \alpha_{i,j} \leq \pi - \tau$ :  $\mathbf{p}_{i,j}$  is an object point
3.  $\pi - \tau < \alpha_{i,j}$ :  $\mathbf{p}_{i,j}$  is a ceiling point

A result of the ground segmentation is displayed in Fig. 4. The classification of a scan point as “ground” based on its neighborhood instead of performing a simple height comparison is essential due to potential inaccuracy in the scanner mount calibration and unknown starting pose, namely, the pitch angle of the robot, as well as the significant unevenness of the terrain.

Absolute height values of points do come into play when nearby drivable surface points are to be merged into one large drivable area. Our point classification scheme leaves the possibility open that two nearby points in a 3D model are both correctly labeled ground, but are of significant difference in absolute height, such as on the two horizontal neighborhoods of a vertical cliff. This needs to be checked when growing neighboring ground points to drivable surfaces, as described next.

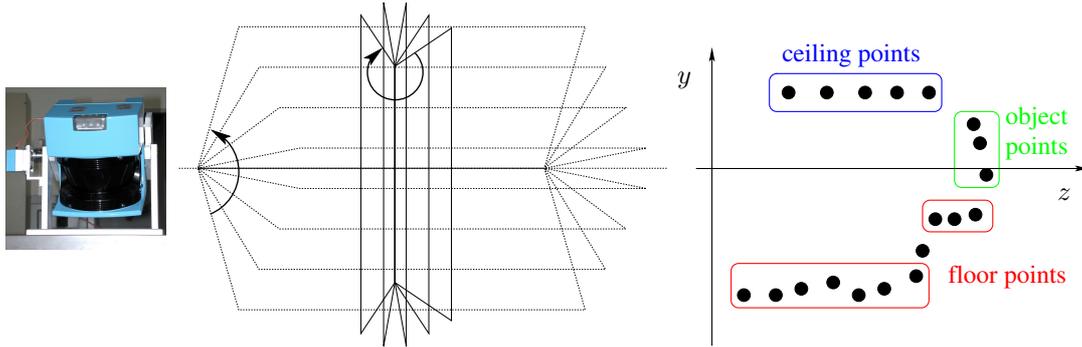


Figure 3: Left: 3D scan planes due to the rotation of the 2D laser range finder vs. 3D sweep planes. Right: Interpretation example: One vertical sweep plane.

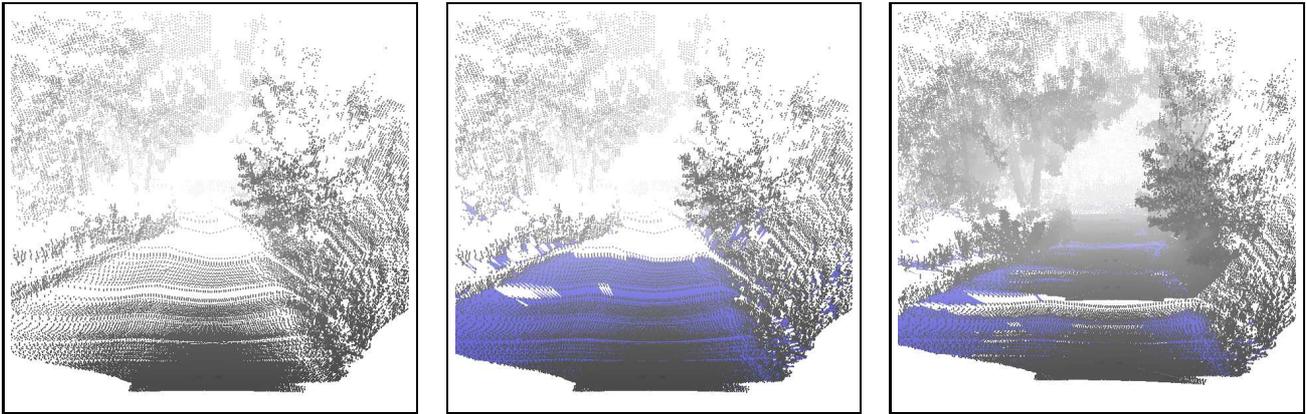


Figure 4: *Left*: A single outdoor 3D scan of a gravel path in the Botanical Garden. Note that the path is uneven. *Middle*: Areas (triangles) between neighboring surface points all labeled drivable are shaded in blue. Note first that the area in front is very dense with surface points, which are all labeled drivable. Note second that there are some disconnected patches of surface points in and behind the path shoulder. *Right*: View into the model from the same virtual view point as before, but with the next scans along the path registered. Sufficiently large areas sufficiently dense with drivable surface points are filled with blue. (Again, the area in front is completely drivable.) Note that the next scan has been taken too far away from the one in front to connect the drivable surface areas, so some of the objectively drivable path remains unlabeled here for lack of point density.

**Merging Ground Points into Drivable Surface.** Fig. 4, right, gives the idea how we connect ground points of one scan with the ones of its neighbors: The basic observation is that points labeled ground in individual neighboring scans and denoting the same patch of path in the environment would end up very close to each other after registration. So it has proved sound to grow regions of drivable surface around areas containing sufficiently many ground points in sufficient density, disregarding from which individual scans their labels were derived. Significant differences in absolute point height are checked, as mentioned previously.

This strategy assumes that neighboring scans overlap sufficiently well, to map the drivable surface sufficiently densely. While this is no completely new requirement (registration itself also needs some minimal overlap), we are still experimenting with suitable combinations of point density requirements and efficient scan distances.

## Results and Future Work

The experiments for this paper have been made with Kurt3D in the University of Osnabrück's Botanical Garden, with the Kurt3D robot being steered manually (joystick control) between scan points. 3D scans have been automatically acquired and matched, and drivable surface points extracted. Fig. 5 presents the mapping result as well as an impression of the maximal difference in elevation.

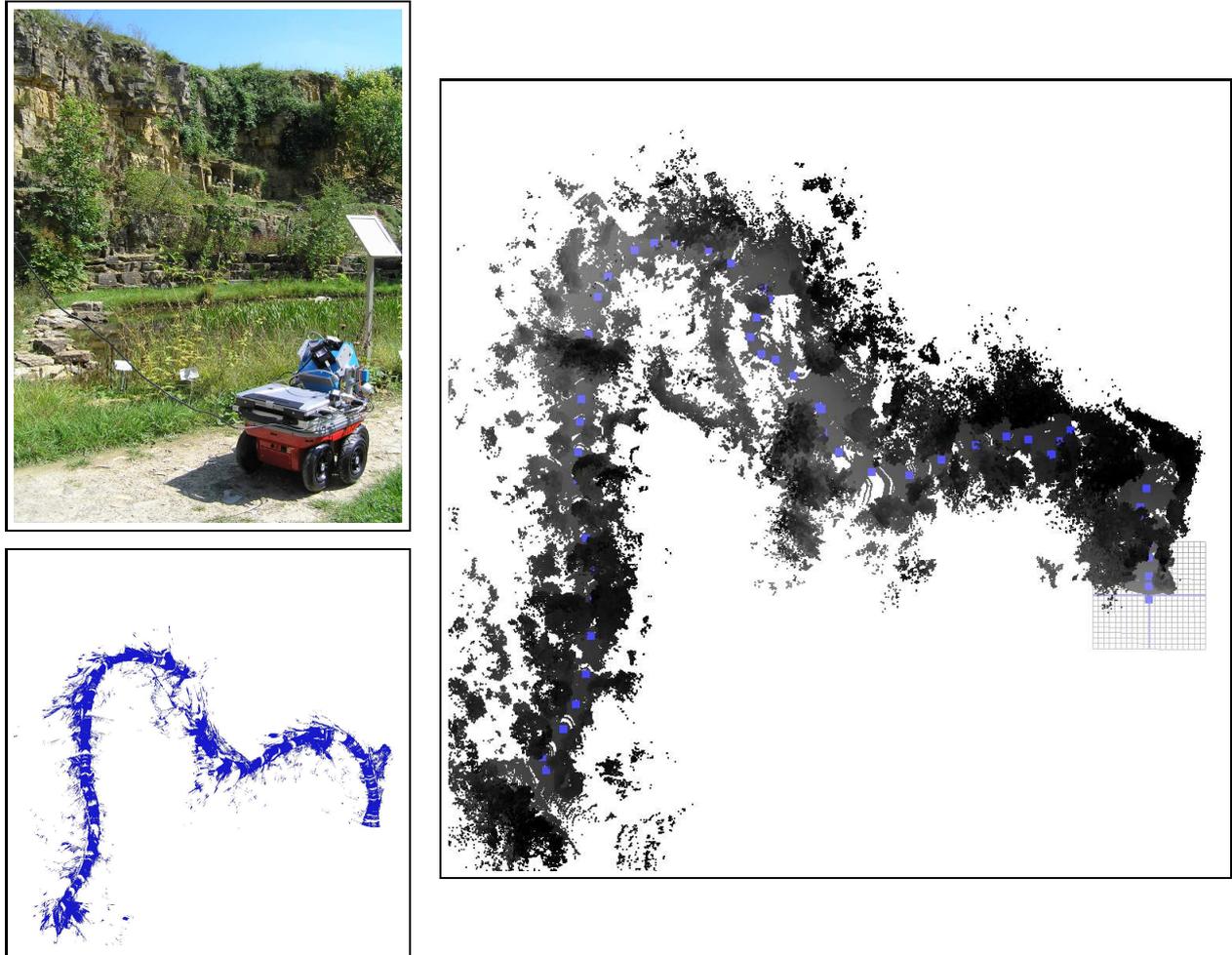


Figure 5: *Top left*: Kurt3D on a gravel path of the Botanical Garden of University of Osnabrück. In the background, you see the place of the path with maximal difference in elevation (about 6 meters, independently measured). *Right*: Top view of the automatically mapped area of the Botanical Garden. The grid on the right side denotes an area of  $20 \times 20 \text{ m}^2$ , blue boxes correspond to scan poses. Most scan points in the top view correspond to trees and bushes. The overall map has been generated of 55 scans of about 85,000 points each; the path length was about 150 m. *Bottom left*: Subset of the overall map, consisting of drivable surface only. The discontinuity in the middle part is due to very ragged ground.

The online extraction of drivable surface has yielded sufficiently much such area so that a subsequent planning process (recall item 5 in our sequence of SPLAM subtasks above!) would have enough opportunities for working.

We have encountered some problems. As mentioned for Fig. 4, right, we need to make sure that scan distances are sufficiently close for mapping a continuous path to a continuous surface. Very ragged surface like cobblestone with large and/or sharp gaps between the tiles or stones would not be classified drivable by our method; in fact, one may ask whether it objectively is drivable for our Kurt3D robot, or should rather be avoided. Finally, the elevation values of our mapped points were quite inaccurate at the end of the path, due to a medium error quite early in the drive. This angle error has then accumulated over the subsequent scans, which were locally correctly registered. Affordable inclinometers do not seem to solve the problem, as their expected measurement error seems to be in the same order of magnitude as the registration pitch angle error (some degrees). We expect this type of problem to be solved in a later stage of the overall SPLAM process by loop closing. Note that this error has not affected the locally correct classification of drivable surface areas.

Future work will continue to fill up the list of SPLAM subtasks named earlier. To help localization and provide waypoints in outdoor environments, we will integrate GPS information. To make drivable surface detection more flexible and more robust, we will add camera-based path detection.

## Acknowledgments

Special thanks to Oliver Wulf for the detailed discussions of his classification algorithm. Furthermore, we thank Sepp Kollmorgen for implementing the semantic interpretation. The development of Kurt3D started at the Fraunhofer Institute for Autonomous Intelligent Systems (AIS) and we also want to thank our former colleagues there.

## References

- [1] P. Allen, I. Stamos, A. Gueorguiev, E. Gold, and P. Blaer. AVENUE: Automated Site Modelling in Urban Environments. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling (3DIM '01)*, Quebec City, Canada, May 2001.
- [2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least square fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698 – 700, 1987.
- [3] P. H. Batavia and S. Singh. Obstacle detection in smooth high curvature terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '02)*, pages 3062 – 3067, 2002.
- [4] P. Besl and N. McKay. A method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239 – 256, February 1992.
- [5] P. Biber, H. Andreasson, T. Duckett, and A. Schilling. 3D Modeling of Indoor Environments by a Mobile Robot with a Laser Scanner and Panoramic Camera. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, Sendai, Japan, September 2004.
- [6] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, 17(3):229 – 241, June 2001.
- [7] Matrix FAQ. Version 2, <http://skal.planet-d.net/demo/matrixfaq.htm>, 1997.
- [8] J. Folkesson and H. Christensen. Outdoor Exploration and SLAM using a Compressed Filter. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '03)*, pages 419–426, Taipei, Taiwan, September 2003.
- [9] U. Frese and G. Hirzinger. Simultaneous Localization and Mapping – A Discussion. In *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, pages 17 – 26, Seattle, USA, August 2001.
- [10] C. Früh and A. Zakhor. 3D Model Generation for Cities Using Aerial Photographs and Ground Level Laser Scans. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR '01)*, Kauai, Hawaii, USA, December 2001.
- [11] A. Georgiev and P. K. Allen. Localization Methods for a Mobile Robot in Urban Environments. *IEEE Transaction on Robotics and Automation (TRO)*, 20(5):851 – 864, October 2004.
- [12] M. Hebert, M. Deans, D. Huber, B. Nabbe, and N. Vandapel. Progress in 3-D Mapping and Localization. In *Proceedings of the 9th International Symposium on Intelligent Robotic Systems, (SIRS '01)*, Toulouse, France, July 2001.
- [13] P. Kohlhepp, M. Walther, and P. Steinhaus. Schritthaltende 3D-Kartierung und Lokalisierung für mobile Inspektionsroboter. In R. Dillmann, H. Wörn, and T. Gockel, editors, *Proceedings of the Autonome Mobile Systeme 2003, 18. Fachgespräche*, December 2003.
- [14] A. Lorusso, D. Eggert, and R. Fisher. A Comparison of Four Algorithms for Estimating 3-D Rigid Transformations. In *Proceedings of the 4th British Machine Vision Conference (BMVC '95)*, pages 237 – 246, Birmingham, England, September 1995.
- [15] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333 – 349, October 1997.

- [16] R. Manduchi, A. Castano, A. Talukder, and L. Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous Robot*, 18, 2005.
- [17] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM with Approximate Data Association. In *Proceedings of the 12th International Conference on Advanced Robotics (ICAR '05)*, pages 242 – 249, July 2005.
- [18] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. Heuristic-Based Laser Scan Matching for Outdoor 6D SLAM. In *KI 2005: Advances in Artificial Intelligence. 28th Annual German Conference on AI, Proceedings Springer LNAI vol. 3698*, pages 304 – 319, Koblenz, Germany, September 2005.
- [19] A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6D SLAM with an Application in autonomous mine mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1998 – 2003, New Orleans, USA, April 2004.
- [20] K. Patel, W. Macklem, S. Thrun, and M. Montemerlo. Active sensing for high-speed offroad driving. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, Barcelona, Spain, April 2005.
- [21] S. Se, D. Lowe, and J. Little. Local and Global Localization for Mobile Robots using Visual Landmarks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '01)*, Hawaii, USA, October 2001.
- [22] V. Sequeira, K. Ng, E. Wolfart, J. Goncalves, and D. Hogg. Automated 3D reconstruction of interiors with multiple scan-views. In *Proceedings of SPIE, Electronic Imaging '99, The Society for Imaging Science and Technology /SPIE's 11th Annual Symposium*, San Jose, CA, USA, January 1999.
- [23] H. Surmann, K. Lingemann, A. Nüchter, and J. Hertzberg. A 3D laser range finder for autonomous mobile robots. In *Proceedings of the of the 32nd International Symposium on Robotics (ISR '01)*, pages 153 – 158, Seoul, Korea, April 2001.
- [24] H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Journal Robotics and Autonomous Systems*, 45(3 – 4):181 – 198, December 2003.
- [25] H. Surmann, A. Nüchter, K. Lingemann, and J. Hertzberg. 6D SLAM A Preliminary Report on Closing the Loop in Six Dimensions. In *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV '04)*, Lisabon, Portugal, July 2004.
- [26] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
- [27] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
- [28] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots*, 31(5):1 – 25, 1997.
- [29] S. Thrun, D. Fox, and W. Burgard. A real-time algorithm for mobile robot mapping with application to multi robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, San Francisco, CA, USA, April 2000.
- [30] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. F. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *Machine Learning and Autonomous Robots*, 23(7 – 8):693 – 716, July/August 2004.
- [31] A. Walthelm, R. Kluthe, and A.M. Mamlouk. Ein 3D Weltmodell zur teilaktiven Positionsverfolgung in komplexen dynamischen Umgebungen. In *Proc. AMS*, pages 272–281, 1999.
- [32] O. Wulf, K. O. Arras, H. I. Christensen, and B. A. Wagner. 2D Mapping of Cluttered Indoor Environments by Means of 3D Perception. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, pages 4204 – 4209, New Orleans, USA, April 2004.

- [33] H. Zhao and R. Shibasaki. Reconstructing Textured CAD Model of Urban Environment Using Vehicle-Borne Laser Range Scanners and Line Cameras. In *Second International Workshop on Computer Vision System (ICVS '01)*, pages 284 – 295, Vancouver, Canada, July 2001.

**Andreas Nüchter** Andreas Nüchter is a research associate at University of Osnabrück and a PhD student at University of Bonn. Past affiliations were with the Fraunhofer Institute for Autonomous Intelligent Systems (AIS, Sankt Augustin), University of Bonn and Washington State University. He received the diploma degree in computer science from University of Bonn in 2002 and was awarded with the best paper award by the German society of informatics (GI) for his thesis. His research interests include reliable robot control, 3D robotic mapping, 3D vision, and laser scanning technologies. He is a member of the GI and the IEEE.

**Kai Lingemann** Kai Lingemann is a research associate and PhD student at University of Osnabrück. Past affiliations were with the Fraunhofer Institute for Autonomous Intelligent Systems (AIS, Sankt Augustin) (Germany), University of Bonn (Germany) and Kyoto University (Japan). He received the diploma degree in computer science from University of Bonn in January 2004. His research interests include mobile robot localization, reliable robot control, 3D vision, and laser scanning technologies.

**Joachim Hertzberg** is a professor for Knowledge-Based Systems at the University of Osnabrück. Former affiliations were with GMD and with Fraunhofer AIS in Sankt Augustin. His research interests include mobile robotics, action planning, and the intersection of the two, such as plan-based robot control and symbolic interpretation of sensor data.