Robotics and Autonomous Systems, Volume 88, Pages 104-115, Elsevier, February 2017.
Available online 27 October 2016. http://dx.doi.org/10.1016/j.robot.2016.10.017

# Continuous Mapping and Localization for Autonomous Navigation in Rough Terrain using a 3D Laser Scanner

David Droeschel, Max Schwarz, Sven Behnke

*Autonomous Intelligent Systems Group, Computer Science Institute VI*
*University of Bonn, Friedrich-Ebert-Allee 144, 53113 Bonn, Germany*

**Abstract**

For autonomous navigation in difficult terrain, such as degraded environments in disaster response scenarios, robots are required to create a map of an unknown environment and to localize within this map. In this paper, we describe our approach to simultaneous localization and mapping that is based on the measurements of a 3D laser-range finder. We aggregate laser-range measurements by registering sparse 3D scans with a local multiresolution surfel map that has high resolution in the vicinity of the robot and coarser resolutions with increasing distance, which corresponds well to measurement density and accuracy of our sensor. By modeling measurements by surface elements, our approach allows for efficient and accurate registration and leverages online mapping and localization. The incrementally built local dense 3D maps of nearby key poses are registered against each other. Graph optimization yields a globally consistent dense 3D map of the environment. Continuous registration of local maps with the global map allows for tracking the 6D robot pose in real time. We assess the drivability of the terrain by analyzing height differences in an allocentric height map and plan cost-optimal paths. The system has been successfully demonstrated during the DARPA Robotics Challenge and the DLR SpaceBot Camp. In experiments, we evaluate accuracy and efficiency of our approach.

*Keywords:*
Mapping, Localization, Rough Terrain

## 1. Introduction

In order to enable robot systems to enter areas inaccessible to humans, e.g., in disaster scenarios or for planetary exploration, autonomous navigation is key. It necessitates the capability to simultaneously build maps of unknown environments and to localize within. These environments can be cluttered or degraded and pose a challenge for perception algorithms. To enable autonomous navigation, the perceived map of the environment has to be accurate enough to allow for analyzing whether a particular region is drivable or not. Besides that, the efficiency of the perception system is important since the operation in these environments often requires online mapping and localization in real time with limited onboard computers.

In this paper we describe our system for mapping and localization on our mobile manipulation robot Momaro. The robot has been developed according to the requirements of the DARPA Robotics Challenge[1] (DRC). The goal of the DRC was to foster research for robots that are able assist humans in responding to catastrophic situations, such as the nuclear

*Email addresses:* droeschel@ais.uni-bonn.de (David Droeschel), max.schwarz@uni-bonn.de (Max Schwarz), behnke@cs.uni-bonn.de (Sven Behnke)

[1] https://web.archive.org/web/20160402011550/http://www.theroboticschallenge.org

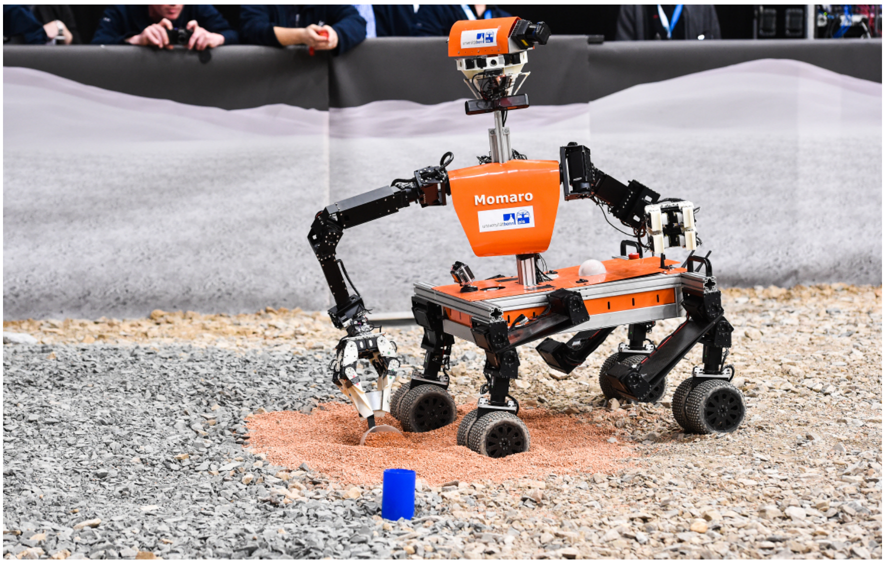


Figure 1: The mobile manipulation robot Momaro taking a soil sample during the DLR SpaceBot Camp. Without intervention of an operator, the robot learned a map of the previously unknown environment, localized within this map, and autonomously navigated to the goal pose that has been specified in a coarse environment map beforehand.

disaster at Fukushima in 2011. Being teleoperated over a limited network connection, the robots had to solve eight tasks relevant to disaster response. While the DRC showed the potential of robots for tasks found in disaster response scenarios, it also showed that fully autonomous navigation and manipulation in unstructured environments—also due to the lack of applicable perception methods—is still beyond the state of the art.

In contrast to the DRC, where robots could be teleoperated for navigation, the DLR SpaceBot Camp 2015 focused on autonomy. Based on a coarse map of the environment, the robot had to explore a previously unknown planetary-like environment and to perform a set of mobile manipulation tasks. Figure 1 shows our robot Momaro taking a soil sample. By means of a 3D continuously rotating laser scanner, Momaro acquires range measurements in all spatial directions. The 3D scans of the environment are aggregated in a robot-centric local multiresolution map. The 6D sensor motion is estimated by registering the 3D scan to the map using our efficient surfel-based registration method [1]. In order to obtain an allocentric map of the environment—and to localize in it—individual local maps are aligned to each other using the same surfel-based registration method. A pose graph that connects the maps of neighboring key poses is constructed and optimized globally. By localizing the robot with respect to the optimized pose graph, we gain an accurate estimate also in larger environments with big loops, where filter-based approaches would obtain an inaccurate estimate. The graph-based formulation allows to globally minimize accumulated errors, resulting in an accurate map of the environment and localization pose.

The remainder of the paper describes our laser perception system that was used during the DRC Finals and the DLR SpaceBot Camp. During the DRC, only the local mapping components where used to build a egocentric map of the robot's direct vicinity. This map was used by the manipulation operator when planning motions and to correct odometry drift of the robot, when aligning to a previously acquired local map. This part of the system is described in Section 4 and Section 5 and is based on our previous work in [1]. Apart from the local mapping, our allocentric mapping component [2] was used to allow for fully autonomous navigation during DLR SpaceBot Camp and is described in Section 6.

In this article, we present a complete system for continuous mapping and localization, fully integrated in our navigation system and extensively tested. Building a fully integrated system with the given requirements led to the following advances over our previous work:

1. We extended our local multiresolution map to address for dynamics in the environment. By efficiently maintaining occupancy information we increase the quality of the maps and the robustness of the registration.
2. We extended our allocentric mapping system to allow for fully continuous mapping and localization during mission, without the necessity to map the environment beforehand or to stop for acquiring new 3D scans and to process them.
3. In the evaluation section, we show data acquired during the DARPA Robotics Challenge Finals and the DLR SpaceBot Camp 2015.

Our mapping pipeline is published open-source[2], making it available to other researchers in order to facilitate developing robotic applications, contributing to the system, and for comparing and reproducing results.

## 2. Related Work

For mobile ground robots that operate in cluttered and degraded environments, 3D laser scanners are the preferred sensor for mapping and localization. They provide accurate distance measurements, are almost independent on lighting conditions, and have a large field-of-view.

Mapping with 3D laser scanners has been investigated by many groups [3, 4, 5, 6]. A common research topic in laser-based simultaneous localization and mapping (SLAM) is efficiency and scalability, i.e. maintaining high run-time performance and low memory consumption. To gain both memory and runtime efficiency, we build local multiresolution surfel grid maps with a high resolution close to the sensor and a coarser resolution farther away. Local multiresolution corresponds well to the sensor measurement characteristics. Measurements are aggregated in grid cells and summarized in surface elements (surfels) that are used for registration. Our registration method matches 3D scans on all resolutions concurrently, utilizing the finest common resolution available between both maps, which also makes registration efficient. In previous own work [7, 8], we used this concept within an octree voxel representation.

For aligning newly acquired 3D scans with the so far aggregated map, we use our surfel-based registration method [1]. In contrast to many methods for point set registration—mostly based on the Iterative Closest Point (ICP) algorithm [9]—our method recovers the transformation between two points sets through probabilistic assignments of surfels. Probabilistic methods for point set registration are becoming more and more popular recently and show promising results [10, 11, 12].

Hornung et al. [13] implement a multiresolution map based on octrees (OctoMap). Ryde et al. [14] use voxel lists for efficient neighbor queries. Both of these approaches consider mapping in 3D with a voxel being the smallest map element. The 3D-NDT [15] discretizes point clouds in 3D grids and aligns Gaussian statistics within grid cells to perform scan registration.

Belter et al. [16] also propose to use local grid maps with different resolutions. In contrast to our approach, different map resolutions are used for different sensors, resulting in an uniform grid map for each sensor. Herbert et al. propose elevation maps [17], extending 2D grid maps by adding a height for every grid cell. While elevation maps only model a single surface, multi-level surface maps [18] store multiple heights in each grid cell, allowing to model environments with more than on surface, such as bridges for example. Pfaff et al. [19]

[2] `https://github.com/AIS-Bonn/mrs_laser_map`

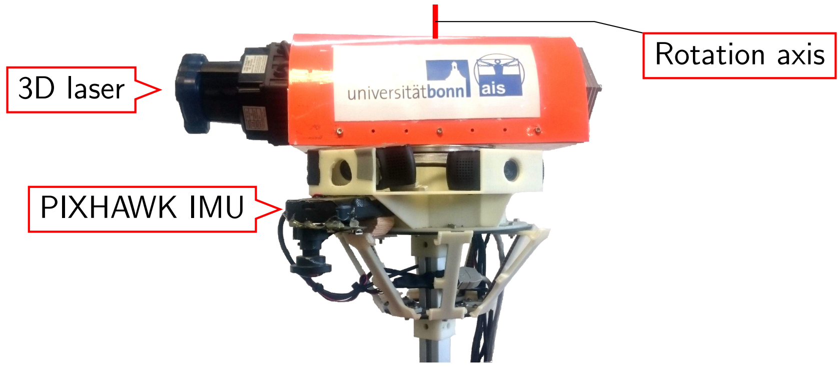



Figure 2: Momaro's sensor head. The Hokuyo laser scanner is rotated by an actuator around the red axis to allow for an omnidirectional field-of-view. The IMU is used to compensate for motion during scan acquisition and for estimating the attitude.

propose a method for detecting loop closures in elevation maps. Frankhauser et al. [20] use local elevation maps and handle drift by propagating uncertainties of the robot pose through the map.

Our mapping system has been successfully applied on micro aerial vehicles (MAV) to allow for fully autonomous navigation [21]. In contrast to this work, we do not rely on accurate visual odometry anymore, but on imprecise wheel odometry in combination with measurements from an inertial measurement unit (IMU). Compared to other mapping approaches, we efficiently build robot-centric maps that are locally consistent—with constant computation and memory requirements. We construct an allocentric graph of local maps from different view poses, resulting in a sparse pose graph that can be optimized efficiently. Compared to the mapping system used in our previous work [22], the system presented here is more efficient and maps the environment in a continuous manner—without the requirement to stop for acquisition and processing of new 3D scans. Besides that, the robustness of the localization improved since the current system aligns dense local maps to the allocentric map, in contrast to single 2D scans. While many methods assume the robot to stand still during 3D scan acquisition, some approaches also integrate scan lines of a continuously rotating laser scanner into 3D maps while the robot is moving [23, 24, 25, 26, 27].

Path planning for navigating in 3D indoor environments with flat floors is well-studied [28, 29]. For navigation on non-flat terrain, several approaches generate 2D cost maps from sensor readings and plan paths in these [30, 31, 32, 33]. Rusu et al. [34] model 3D maps by a set of convex polygons and adapt existing 2D planners to operate in 3D terrain. Chhaniyara et al. [35] and Papadakis [36] compiled surveys on judging traversability of terrain and avoiding obstacles with robots. In many environments, color or texture do not provide sufficient traversability information, so 3D geometry is needed. We present an integrated system for efficient laser-based 3D SLAM, traversability analysis, and cost-optimal path planning.

## 3. System Overview

Momaro is equipped with four articulated compliant legs that end in pairs of directly driven, steerable wheels. The combination of legs and steerable wheels allows for omnidirectional driving and stepping locomotion. To perform a wide range of manipulation tasks [37], Momaro is equipped with an anthropomorphic upper body with two 7 degrees of freedom manipulators that end in dexterous grippers.

Momaro's main sensor for environmental perception is a continuously rotating laser scanner on its sensor head (see Figure 2). It consists of a Hokuyo UTM-30LX-EW 2D laser scanner which is rotated around the vertical axis by a Robotis Dynamixel MX-64 servo actuator to

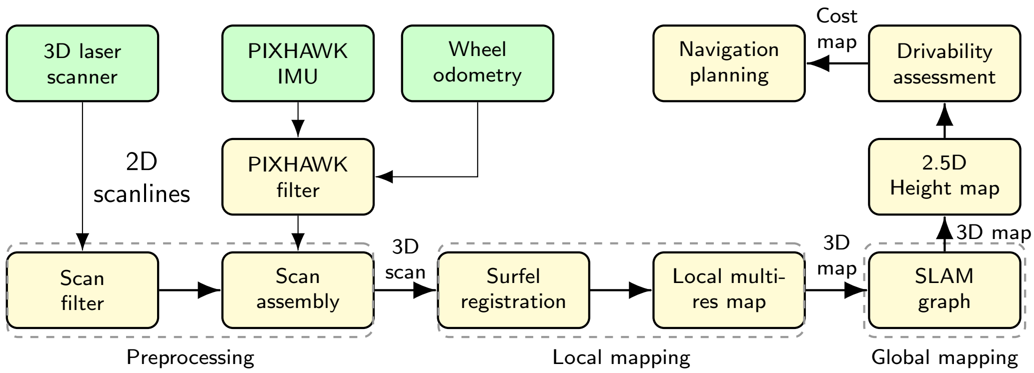



Figure 3: Overview of our mapping, localization and navigation system. The measurements are processed in preprocessing steps described in Section 5.1. The resulting 3D point cloud is used to estimate the transformation between the current scan and the map (Section 5). Registered scans are stored in a local multiresolution map (Section 4). Keyframe views of local maps are registered against each other in a SLAM graph (Section 6). A 2.5D height map is used to assess drivability. A standard 2D grid-based approach is used for planning (Section 7).

gain a 3D FoV. Hence, the sensor can measure in all directions, except for a cylindrical blind spot around the vertical axis centered on the robot. The 2D LRF is electrically connected by a slip ring, allowing for continuous rotation of the sensor.

The Hokuyo 2D laser scanner has an apex angle of 270° and an angular resolution of 0.25°, resulting in 1080 distance measurements per 2D scan, called a *scan line*. The Dynamixel actuator rotates the 2D laser scanner at 0.2 rotations per second, resulting in 200 scan lines per full rotation. Slower rotation is possible if a higher angular resolution is desired. For our current setup, we acquire one full 3D scan with up to 216,000 points per rotation every 5 seconds (shown in Figure 4a).

A PIXHAWK IMU is mounted close to the laser scanner, which is used for motion compensation during scan aggregation and attitude estimation.

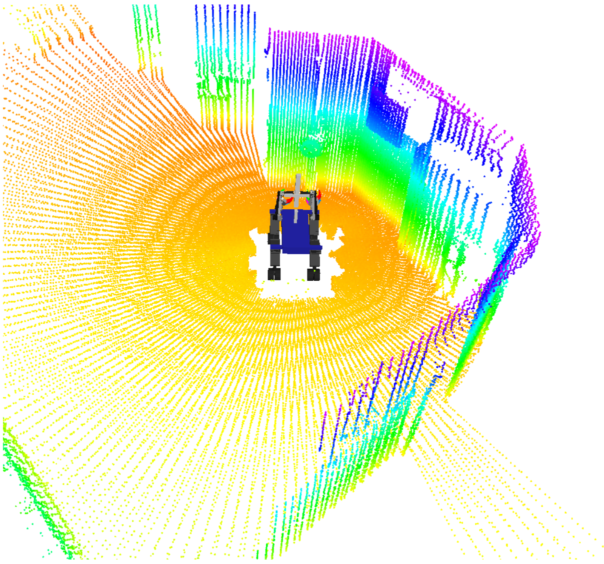
An overview of our software system is shown in Figure 3. It consists of preprocessing steps to assemble 3D scans (Section 5.1), local mapping (Sections 4 and 5), global mapping (Section 6), and navigation planning (Section 7).

## 4. Local Multiresolution Map

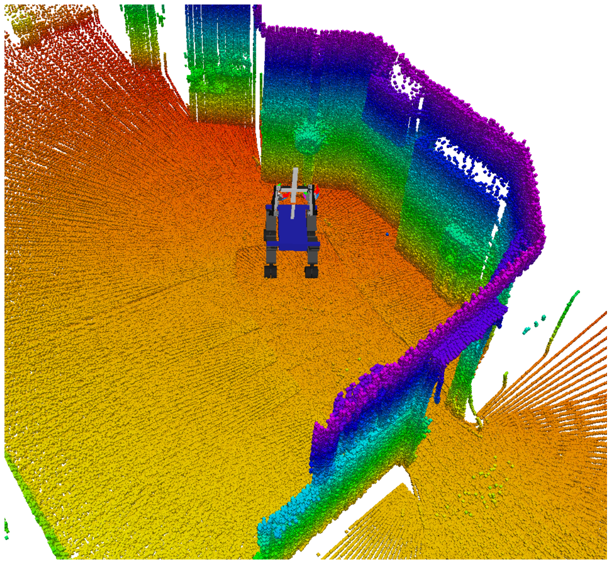
Distance measurements from the laser-range sensor are accumulated in a 3D multiresolution map with increasing cell sizes from the robot center. The representation consists of multiple robot-centered 3D grid-maps with different resolutions. On the finest resolution, we use a cell length of 0.25 m. Each grid-map is embedded in the next level with coarser resolution and doubled cell length. The stored points and grid structure is shown in Figure 4.

We use a hybrid representation, storing 3D point measurements along with occupancy information in each cell. Point measurements of consecutive 3D scans are stored in fixed-sized circular buffers, allowing for point-based data processing and facilitating efficient nearest-neighbor queries. Figure 5 shows the point-based representation of the local multiresolution map during the SpaceBot Camp. It even shows relatively small objects—like a battery pack that the robot shall grasp.

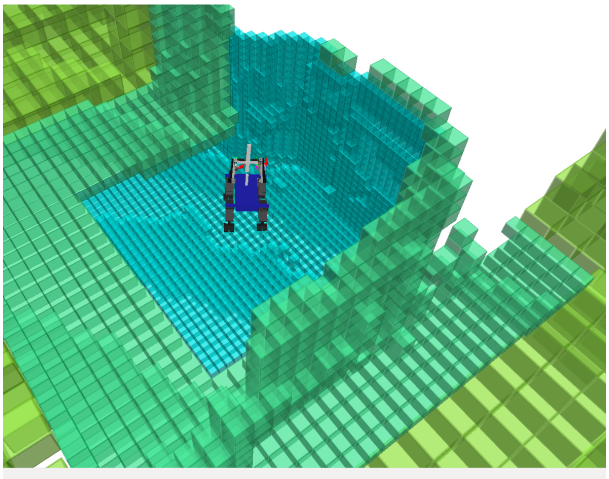
Figure 6 shows a 1D schematic illustration of the map organization. We aim for efficient map management for translation and rotation. Individual grid cells are stored in a circular buffer to allow for shifting elements in constant time. We interlace multiple circular buffers to obtain a map with three dimensions. The length of the circular buffers depends on the



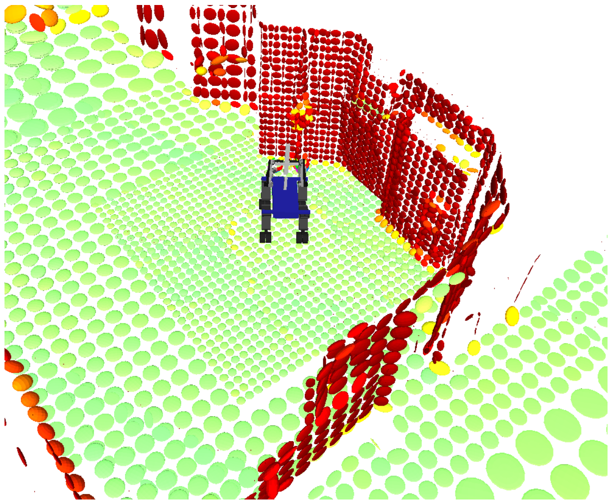
(a) 3D scan



(b) Points in map



(c) Grid cells in map



(d) Surfels in map

Figure 4: The local multiresolution grid map during the first DRC competition run. (a): The 3D scan acquired with our continuously rotating laser scanner. (b): 3D points stored in the local multiresolution map. Color encodes height from ground. (c): The multiresolution grid structure of the map. Cell size (indicated by color) increases with the distance from the robot. (d): For every grid cell a surfel es summarizes the 3D points in the cell. Color encodes the orientation of the surfel.

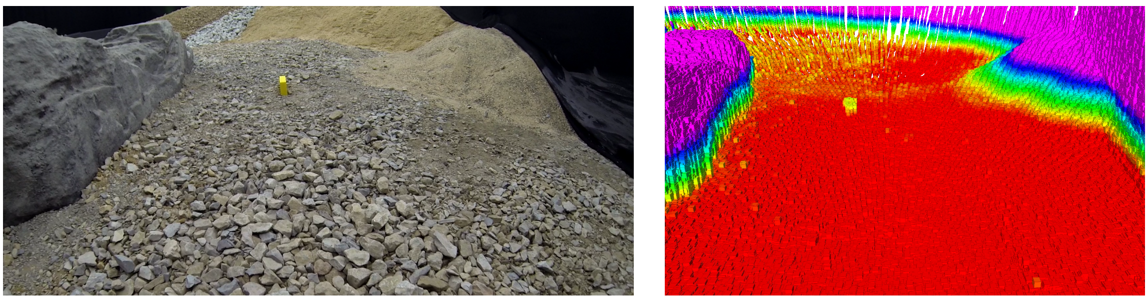


Figure 5: Photo and the corresponding local map of the battery pack—one of the objects to manipulate during the SpaceBot Camp. Color encodes distance from ground.

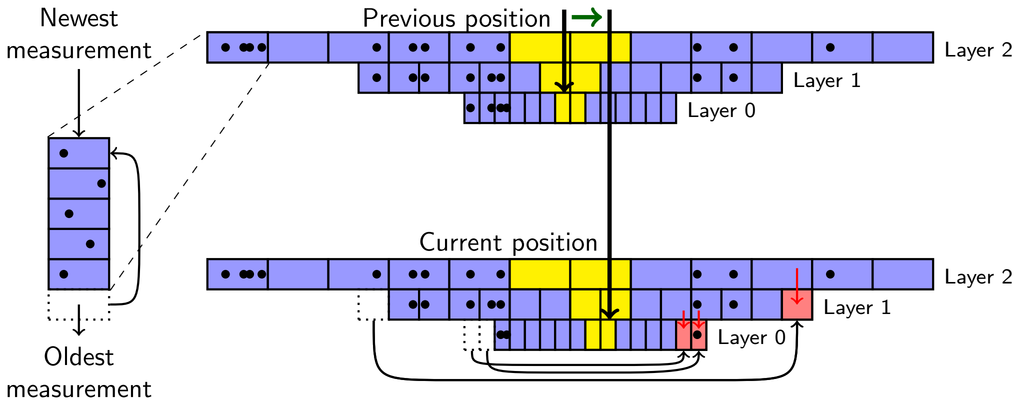



Figure 6: One-dimensional illustration of the hybrid local multiresolution map. Along with the occupancy information, every grid-cell (blue) maintains a circular buffer with its associated measurement points (green). The map is centered around the robot and in case of a robot motion, ring buffers are shifted according to the translational parts of the movement, maintaining the egocentric property of the map. Cells at coarser levels are used to retain points from vanishing cells at finer levels and to initialize newly added cells (red arrows).

resolution and the size of the map. In case of a translation of the robot, the circular buffers are shifted whenever necessary to maintain the egocentric property of the map. In case of a translation equal or larger than the cell size, the circular buffers for respective dimensions are shifted. For sub-cell-length translations, the translational parts are accumulated and shifted if they exceed the length of a cell.

Since we store 3D points for every cell for point-based processing, single points are transformed in the local coordinate frame of a cell when adding, and back to the map coordinate frame when accessing. Every cell in the map stores a list of 3D points from the current and previous 3D scans. This list is also implemented by a fixed-sized circular buffer. If the capacity of the circular buffer is exceeded, old measurements are discarded and replaced by new measurements.

Rotating the map would necessitate to shuffle all cells. Consequently, our map is oriented independent to the robot orientation. We maintain the orientation between the map and the robot and use it to rotate measurements when accessing the map.

## 5. Scan Registration

We register consecutive 3D laser scans with our local multiresolution grid map to estimate the motion of the robot. Since the scans are taken while the robot is driving, the motion of the robot needs to be compensated for when assembling the scan measurements into 3D scans. We register 3D scans with the so far accumulated map of the environment and update it with the registered 3D scan.

### *5.1. Preprocessing and 3D Scan Assembly*

The raw measurements from the laser scanner are subject to spurious measurements at foreground-background transitions between two objects. These so-called *jump edges* are filtered by comparing the angle of neighboring measurements. After filtering for jump edges, we assemble a 3D scan from the 2D scans of a complete rotation of the scanner. Since the sensor is moving during acquisition, we undistort the individual 2D scans in two steps.

First, measurements of individual 2D scans are undistorted with regards to the rotation of the 2D laser scanner around the sensor rotation axis. Using spherical linear interpolation, the rotation between the acquisition of two scan lines is distributed over the measurements.

Second, the motion of the robot during acquisition of a full 3D scan is compensated. Due to Momaro's flexible legs, it is not sufficient to simply use wheel odometry to compensate for the robot motion. Instead, we estimate the full 6D state with the PIXHAWK IMU attached to Momaro's sensor head. Here we calculate a 3D attitude estimate from accelerometers and gyroscopes to compensate for rotational motions of the robot. Afterwards, we filter the wheel odometry with measured linear acceleration to compensate for linear motions. The resulting 6D state estimate includes otherwise unobservable motions due to external forces like rough terrain, contacts with the environment, wind, etc. It is used to assemble the individual 2D scans of each rotation to a 3D scan.

### *5.2. Scan To Map Registration*

We register the points $\mathcal{P} = \{p_1, \dots, p_P\}$ in a 3D scan with the points $\mathcal{Q} = \{q_1, \dots, q_Q\}$ in the local grid map of the environment [1]. Similarly, the registration of two local maps is treated as the registration of their point sets. We formulate the registration of the 3D scan with the local environment map as optimizing the joint data-likelihood

$$p(\mathcal{P} \mid \theta, \mathcal{Q}) = \prod_{k=1}^{P} p(p_k \mid \theta, \mathcal{Q}). \tag{1}$$

Instead of considering each point individually, we map the 3D scan into a local multiresolution grid and match surfels, i.e.,

$$p(\mathcal{P} \mid \theta, \mathcal{Q}) \approx \prod_{i=1}^{N} p(x_i \mid \theta, Y)^{P_{x,i}}. \tag{2}$$

By this, several orders of magnitudes less map elements are used for registration. Figure 4d shows the surfels of an exemplary multiresolution map. We denote the set of surfels in the scene (the 3D scan) by $X = \{x_1, \dots, x_N\}$ and write $Y = \{y_1, \dots, y_M\}$ for the set of model surfels in the environment map. E.g., a surfel $x_i$ summarizes its attributed $P_{x,i}$ points by their sample mean $\mu_{x,i}$ and covariance $\Sigma_{x,i}$. We assume that scene and model can be aligned by a rigid 6 degree-of-freedom (DoF) transformation $T(\theta)$ from scene to model. Our aim is to recover the relative pose $\theta$ of the scene towards the model.

### *5.3. Gaussian Mixture Observation Model*

We explain each transformed scene surfel as an observation from a mixture model, similar as in the coherent point drift (CPD) method [10]. A surfel $x_i$ is observed under the mixture defined by the model surfels and an additional uniform component that explains outliers, i.e.,

$$p(x_i \mid \theta, Y) = \sum_{j=1}^{M+1} p(c_{i,j})\; p(x_i \mid c_{i,j}, \theta, Y), \tag{3}$$

where $c_{i,j}$ is a shorthand for the 1-of-(M+1) encoding binary variable $c_i \in \mathbb{B}^{M+1}$ with $j$-th entry set to 1. Naturally, $c_i$ indicates the association of $x_i$ to exactly one of the mixture components. The model is a mixture on Gaussian components for the $M$ model surfels through

$$p(x_i \mid c_{i,j}, \theta, Y) := \mathcal{N}\left[T(\theta)\mu_{x,i}; \mu_{y,j}, \Sigma_{y,j} + R(\theta)\Sigma_{x,i}R(\theta)^T + \sigma_j^2 I\right], \tag{4}$$

where $\sigma_j = \frac{1}{2}\rho_{y,j}^{-1}$ is a standard deviation that we adapt to the resolution $\rho_{y,j}$ of the model surfel. We set the likelihood of the uniform mixture component to $p(c_{i,M+1}) = w$. For this uniform component, the data likelihood of a surfel $x_i$ is

$$p(x_i \mid c_{i,M+1}, \theta) = \frac{P_{x,i}}{P}\, \mathcal{N}(0; 0, R(\theta)\Sigma_{x,i}R(\theta)^T + \sigma_j^2 I). \tag{5}$$

For the prior association likelihood, we assume the likelihood of $x_i$ to be associated to one of the points in the model map to be equal. Hence, for each Gaussian mixture component $j \in \{1, \dots, M\}$ we have $p(c_{i,j}) = (1-w)\frac{Q_{y,j}}{Q}$. By modeling the scene surfels as samples from a mixture on the model surfels, we do not make a hard association decision between the surfels sets, but a scene surfel is associated to many model surfels.

*5.4. Registration through Expectation-Maximization*

The alignment pose $\theta$ is estimated through maximization of the logarithm of the joint data-likelihood

$$\ln p(\mathcal{P} \mid \theta, \mathcal{Q}) \approx \sum_{i=1}^{N} P_{x,i} \ln \sum_{j=1}^{M+1} p(c_{i,j})\; p(x_i \mid c_{i,j}, \theta, Y). \tag{6}$$

We optimize this objective function through expectation-maximization (EM) [38]. The component associations $c = \{c_1, \dots, c_N\}$ are treated as latent variables to yield the EM objective

$$L(q, \theta) := \sum_{i=1}^{N} P_{x,i} \sum_{j=1}^{M+1} q(c_{i,j}) \ln \frac{p(c_{i,j})\; p(x_i \mid c_{i,j}, \theta, Y)}{q(c_{i,j})}, \tag{7}$$

for which we exploit $q(c) = \prod_{i=1}^{N}\prod_{j=1}^{M+1} q(c_{i,j})$. In the M-step, the latest estimate $\overline{q}$ for the distribution over component associations is held fixed to optimize for the pose $\theta$

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmax}}\;\; L(\overline{q}, \theta) \tag{8}$$

with

$$L(\overline{q}, \theta) := const. + \sum_{i=1}^{N} P_{x,i} \sum_{j=1}^{M+1} \overline{q}(c_{i,j}) \ln p(x_i \mid c_{i,j}, \theta, Y). \tag{9}$$

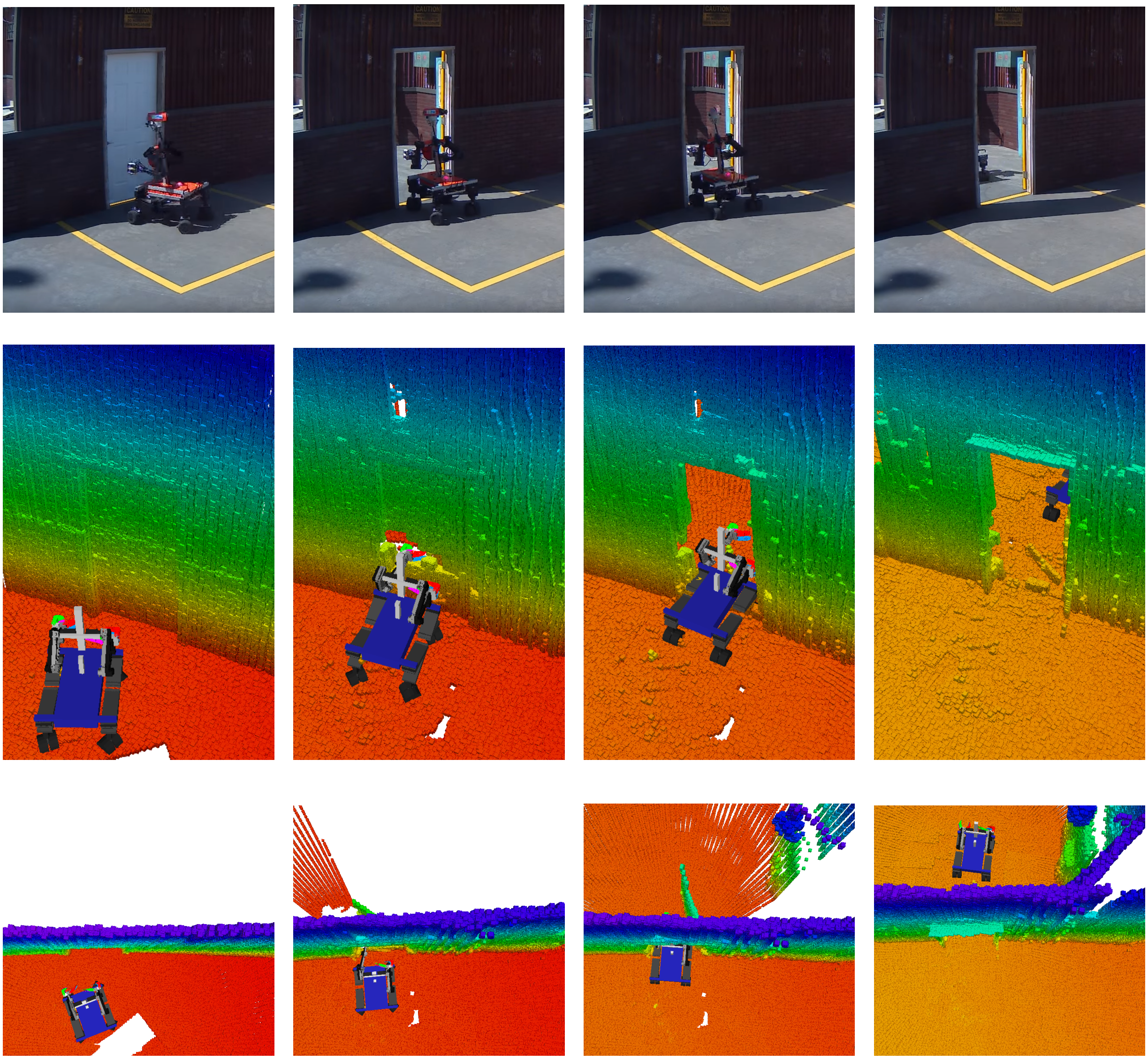
This optimization is efficiently performed using the Levenberg-Marquardt method as in [7]. The E-step obtains a new optimum $\widehat{q}$ for the distribution $q$ by the conditional likelihood of the cluster associations given the latest pose estimate $\overline{\theta}$

$$\widehat{q}(c_{i,j}) = \frac{p(c_{i,j})\; p(x_i \mid c_{i,j}, \overline{\theta}, Y)}{\sum_{j'=1}^{M+1} p(c_{i,j'})\; p(x_i \mid c_{i,j'}, \overline{\theta}, Y)}. \tag{10}$$

In order to evaluate these soft assignments, we perform a local search in the local multiresolution surfel grid of the model. We first look-up the grid cell with a surfel available on the finest resolution in the model map at the transformed mean position of the scene surfel. We consider the surfels in this cell and its direct neighbors for soft association.

*5.5. Filtering Dynamic Objects*

Dynamics in the environment—caused e.g., by moving doors or debris—results in spurious measurements during mapping. Also, registration failures or fast motion of the laser during acquisition of a 3D scan, that could not be compensated by the IMU, result in abandoned measurements in the map. These spurious measurements can affect registration or distract the operator when using the map to plan manipulation tasks. We account for these



1 scan (5 s) 2 scans (10 s) 5 scans (25 s)

Figure 7: Filtering dynamic objects such as the door during the DRC Finals. After opening the door, abandoned measurements are filtered from the local multiresolution map. Camera image and the point-based representation of the map at 4 different time steps. The columns shows the map before opening the door and after adding 1, 2, and 5 scans. Color encodes height from the ground.

by maintaining an occupancy probability—using log-odds notation to avoid multiplication when updating—for each cell in our multiresolution map. Similar to [13] we use a beam-based inverse sensor model and ray-casting to update the occupancy of a cell. For every measurement in the 3D scan, we update the occupancy information of cells on the ray between the sensor origin and the endpoint. Since this ray-casting operation is computationally expensive, we use an approximation to take advantage of the multiresolution structure of our map.

Before updating the occupancy information of the cells in question, we determine the endpoints of each beam—in our case the 3D point in the local coordinate system of the map—and the corresponding cell in every level. These cells are marked as occupied and are excluded from further occupancy updates of the 3D scan they belong to. We do this to prevent from artifacts caused by shallow angles between the line-of-sight of the sensor and the surface, as suggested by [13].

To update the occupancy information efficiently, we start with the coarsest level in our map and perform ray-casting with an approximated 3D Bresenham algorithm [39]. Information from the coarser level is used when updating the finer levels to quickly traverse empty spaces. In detail, we omit ray-casting points on the finer levels if the traversed cells on the coarsest level are observed free. An example is shown in Figure 7. One can observe that the opened door is quickly removed from the local map.

## 6. Allocentric Mapping and Localization

To estimate the motion of the robot, we incorporate IMU measurements, wheel odometry measurements and the the local registration results. While these estimates allow us to control the robot and to track its pose over a short period of time, they are prone to drift. Furthermore, they do not provide a fixed allocentric frame for the definition of mission-relevant poses. To overcome drift and to localize the robot with respect to a fixed frame, we build an allocentric map from local multiresolution maps acquired at different view poses [2].

Therefore, a pose graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed. Every node in the graph corresponds to a view pose and its local multiresolution map. Nearby nodes are connected by edges, modeling spatial constraints between two nodes. Each spatial constraint is a normally distributed estimate with mean and covariance. An edge $e_{ij} \in \mathcal{E}$ describes the relative position $x_i^j$ between two nodes $v_i$ and $v_j$, which arises from aligning two local multiresolution maps with each other. Similar to the alignment of a newly acquired 3D scan, two local multiresolution maps are aligned by our surfel-based registration method described in the previous section. Each edge models the uncertainty of the relative position by its information matrix, which is established by the covariance from registration.

During operation, the current local map is registered towards the closest node in the graph, the reference node $v_{\text{ref}}$. This allows us to track the current pose in the allocentric frame. A new node is generated for the current view pose, if the robot moved sufficiently far. In addition to edges between the previous node and the current node, we add spatial constraints between close-by nodes in the graph that are not in temporal sequence. Thus, we check for one new constraint between the current reference $v_{\text{ref}}$ and other nodes $v_{\text{cmp}}$. We determine a probability

$$p_{\text{chk}}(v_{\text{cmp}}) = \mathcal{N}\left(d(x_{\text{ref}}, x_{\text{cmp}}); 0, \sigma_d^2\right) \tag{11}$$

that depends on the linear distance $d(x_{\text{ref}}, x_{\text{cmp}})$ between the view poses $x_{\text{ref}}$ and $x_{\text{cmp}}$. According to $p_{\text{chk}}(v)$, we choose a node $v$ from the graph and determine a spatial constraint between the nodes.

By adding edges between close-by nodes in the graph, we detect loop closures. Loop closure allows us to minimize drift from accumulated registration errors. For example, if the

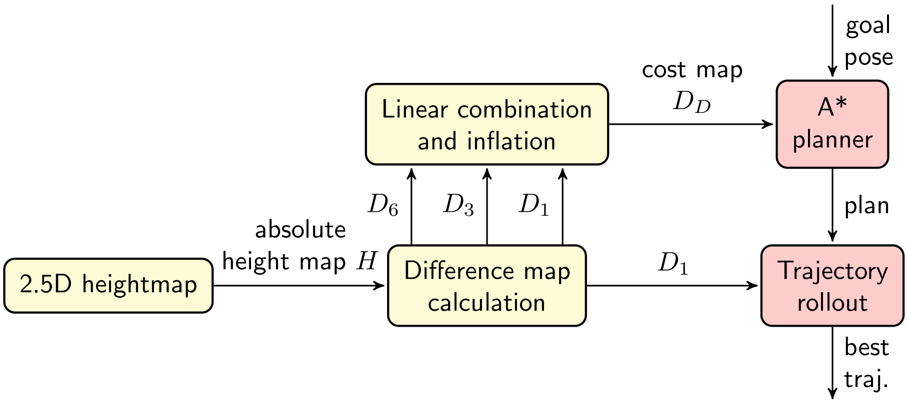



Figure 8: Data flow of our navigation method. Data filtering/processing modules are colored yellow and navigation components red.

robot traverses unknown terrain and reenters a known part of the environment.

From the graph of spatial constraints, we infer the probability of the trajectory estimate given all relative pose observations

$$p(\mathcal{V} \mid \mathcal{E}) \propto \prod_{e_{ij} \in \mathcal{E}} p(x_i^j \mid x_i, x_j). \tag{12}$$

This pose graph optimization is efficiently solved using the g$^2$o framework [40], yielding maximum likelihood estimates of the view poses $x_i$. Optimization is performed when a loop closure has been detected, allowing for on-line operation.

### *6.1. Localization*

While traversing the environment, the pose graph is extended whenever the robot explores previously unseen terrain and optimized when a loop closure has been detected. We localize towards this pose graph during mission to get the pose of the robot in an allocentric frame.

Since the laser scanner acquires complete 3D scans with a relatively low frame rate, we incorporate the egomotion estimate from the wheel odometry and measurements from the IMU. The egomotion estimate is used to track the pose of the robot w.r.t. the last localization result between two consecutive 3D scans. In detail, we track the pose hypothesis by alternating the prediction of the robot movement given the filter result and alignment of the current local multiresolution map towards the allocentric map of the environment.

The allocentric localization is triggered after acquiring a 3D scan and adding it to the local multiresolution map. Therefore, the updated local map is registered towards the closest node in the graph. By aligning the dense local map to the pose graph—instead of the relative sparse 3D scan—we gain robustness, since information from previous 3D scans is incorporated. We update the allocentric robot pose with the resulting registration transform. To achieve real-time performance of the localization module, we track only one pose hypothesis.

During the SpaceBot Camp, we assumed that the initial pose of the robot was known, either by starting from a predefined pose or by means of manually aligning our allocentric coordinate frame with a coarse height map of the environment. Thus, we could navigate to goal poses in the coarse height map by localizing towards our pose graph.

## 7. Navigation

One important application of the allocentric 3D maps and the localization approach is autonomous navigation. In order to demonstrate the suitability of our approach in this domain, we briefly discuss our navigation pipeline, even though it does not contain novel ideas. Figure 8 gives an overview of our the pipeline. Our approach is based on the RGB-D-based local navigation approach of [22], which is now used on the 3D laser measurements. Furthermore, we make no distinction between local and global navigation. For details on the approach, we refer to [22].

For most terrains, a 2.5D height map contains all information necessary for navigation. This reduction greatly reduces the amount of data to be processed and allows planning in real time. The allocentric 2.5D height map is represented as a 2D grid with a resolution of 5×5 cm. For each map cell $H(x, y)$, we calculate the median height of the points whose projections onto the horizontal plane lie in the map cell.

An absolute height map is not meaningful for planning local paths or for avoiding obstacles. To assess drivability, the allocentric height map is transformed into a height difference map. We calculate local height differences at multiple scales $l$. Let $D_l(x, y)$ be the maximum difference to the center pixel $(x, y)$ in a local $l$-window:

$$D_l(x,y) := \max_{\substack{|u-x|<l;u\neq x\\|v-y|<l;v\neq y}} |H(x,y) - H(u,v)|\,.$$

Missing $H(u, v)$ values indicated by NaN are ignored. If the center pixel $H(x, y)$ itself is not defined, or there are no other defined $l$-neighbors, we assign $D_l(x, y) :=$ NaN.

Small, but sharp obstacles show up on the $D_l$ maps with lower $l$ scales. Larger inclines, which might be better to avoid, can be seen on the maps with a higher $l$ value.

The height difference maps are transformed into cost maps as in [22]. In particular, the cost map for path planning is inflated by the robot radius (for an example, see Figure 13). We conduct a standard A* search on the graph defined by the 8-neighborhood in the inflated cost map.

To determine forward driving speed and rotational speed that follow the planned trajectory and avoid obstacles, we use the ROS trajectory rollout planner (*dwa_planner*). Replanning is done at least once every second to account for robot movement and novel terrain percepts.

## 8. Evaluation

This section describes the evaluation of our system during two public events, the DARPA Robotics Challenge Finals and the DLR SpaceBot Camp. The datasets used for the experiments are from our runs during the competition. Since quantitative measures are hard to generate—especially on rough terrain or disaster scenarios due to the absence of a reference measure—we focus on qualitative evaluation. We make the used data sets available on our website[3]. Parts of our system have been evaluated independently in our previous work. For example, our surfel-based registration method has been compared to state-of-the-art registration methods on data from a motion capture system [1], showing that it is more accurate and computationally more efficient. The data sets shown in the experiments are made publicly available[4].

### *8.1. DARPA Robotics Challenge*

Since the robots could be teleoperated during the competition, we did not use our allocentric mapping and localization at the DRC. The local mapping components where used

[3]Data sets captured during the DRC competition run and the DLR SpaceBot Camp demonstration `http://www.ais.uni-bonn.de/data/3D-Laser.html`

[4]`http://www.ais.uni-bonn.de/laser_mapping`

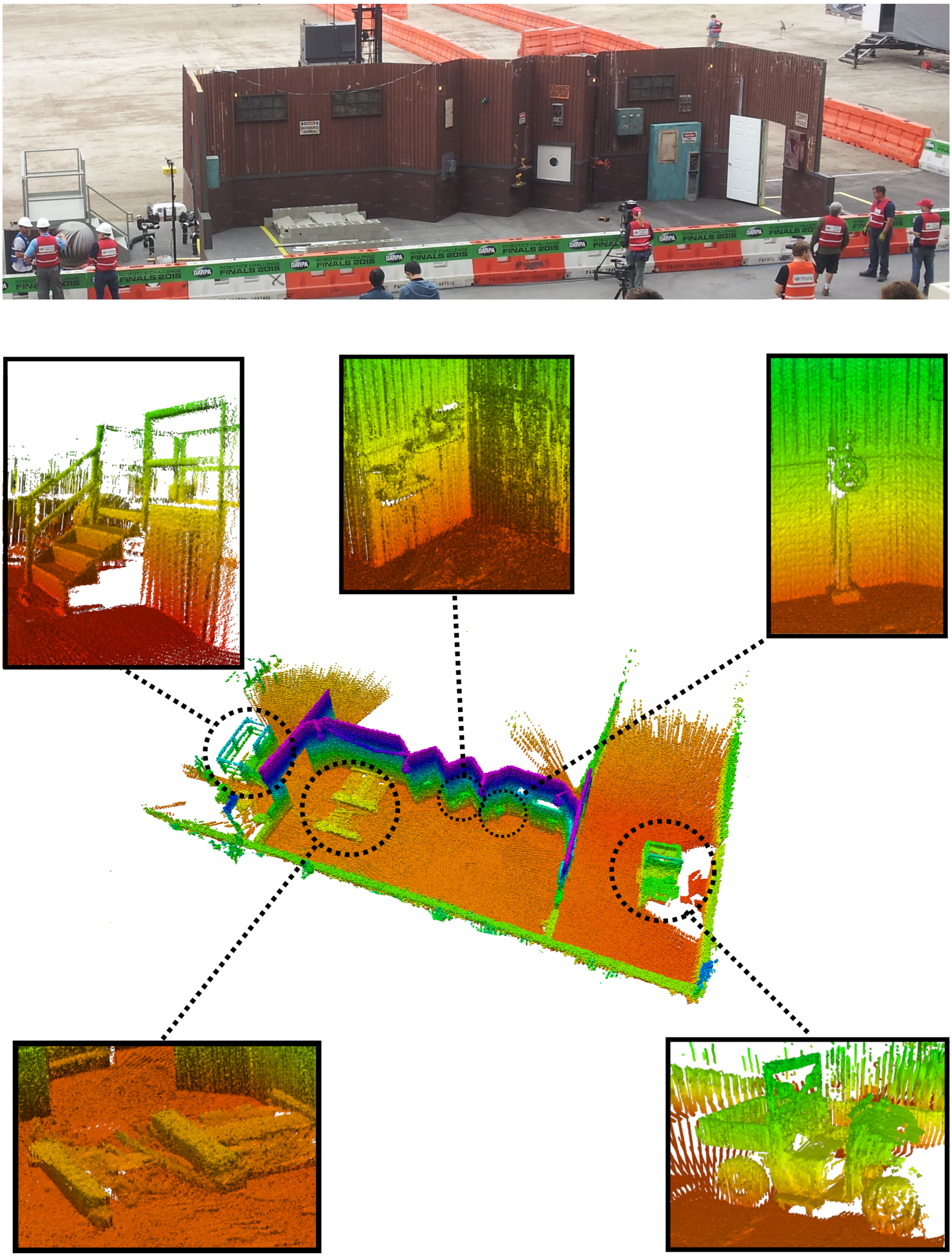


Figure 9: Top: The mock-up disaster scenario of the DRC. Bottom: The resulting allocentric map generated from the data of our first competition run. Color encodes the height from ground.

to build an egocentric map of the robot's direct vicinity. This map was used by the manipulation operator when planning motions. Besides that, the navigation operator used the resulting local maps and height images build from it to assess driveability. Also, the result of the registration corrects odometry drift of the robot when aligning to a previously acquired local map. Figure 9 shows the resulting allocentric map generated from the dataset of our first-day competition. Besides the allocentric map, selected local multiresolution maps of the pose graph are shown. Although reference data is not available, one can see that the resulting allocentric map is globally consistent and accurate, as indicated by the straight walls and plain floor. Also the local maps look clear and accurate.

Our team was able to solve seven out of eight tasks in the shortest time of all teams who solved seven tasks, which yielded a fourth place in the final ranking as the best European team. While we attribute part of our success to our flexible teleoperation solutions [37], the quality of the 3D environment perception and thus the situational awareness of the operator crew played a large part and was a necessary precondition for developing said teleoperation interfaces. Further information on our DRC competition entry is available of our website[5], including a video or our first day competition run[6].

*8.2. SpaceBot Camp 2015*

At the DLR SpaceBot Camp, robots had to conduct an exploration mission in a (simulated) extraterrestrial planetary environment. The mission was—based on a rough height image of the environment—to explore and map the environment and to manipulate objects in it. In contrast to the DRC, the robots did not have a permanent network connection that allowed for teleoperation. Consequently—in addition to local mapping—we used our allocentric mapping component and the described planning approach to allow for fully autonomous navigation.

The planetary-like environment was specially challenging due to the rough surface of the terrain, consisting of different types of stones and soil that caused slip in odometry and high-frequency motion of robot and sensor. Due to the relative small wheels of our robot, an accurate terrain map was necessary to assess driveability. The environment and the resulting allocentric map are shown in Figure 10. It was continuously built during autonomous navigation guided by waypoints specified on the rough height map. One can see, that although the robot was autonomously navigating in rough terrain the resulting allocentric map is accurate and precisely models the environment.

Figure 11 shows the allocentric map at different time steps. The figure shows how the map is extended during a mission. New nodes (i.e., local multiresolution maps) are added to the pose graph and new nodes are connected to existing nodes by edges. During a mission, the map is used for localization as shown in Figure 12 and to assess traversability for navigation as shown in Figure 13. Our system was able to solve all tasks with few interventions by the operator crew over the degraded communication link, such as stopping navigation before a scheduled communication blackout or re-triggering a failed manipulation task. Further information on our SpaceBot Camp entry is available of our website[7], including a video or our demonstration run[8].

[5] Website of our DRC entry `http://www.nimbro.net/Rescue`
[6] Video of first day DRC competition run `http://youtu.be/NJHSFelPsGc`
[7] Website of our DLR SpaceBot Camp entry `http://www.nimbro.net/Explorer`
[8] Video of SpaceBot Camp demonstration run `http://youtu.be/q_p5ZO-BKWM`

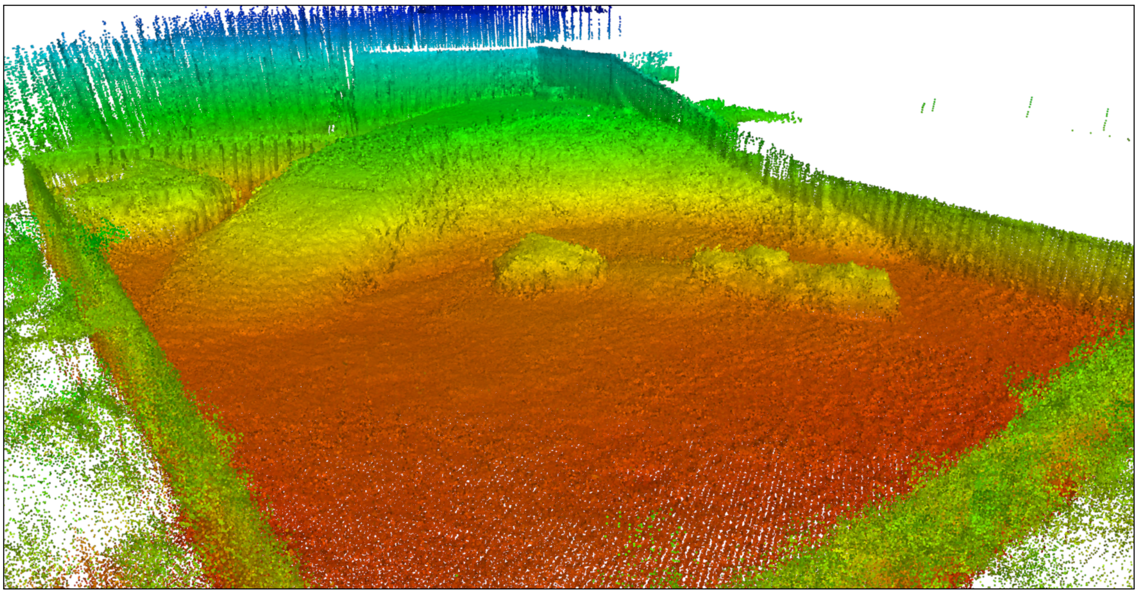


Figure 10: Top: Photo of the planetary-like environment at the DLR SpaceBot Camp consisting of different types of stones and soil. Bottom: The resulting 3D map built by our mapping component from data that has been collected during our run. Color encodes distance from ground.

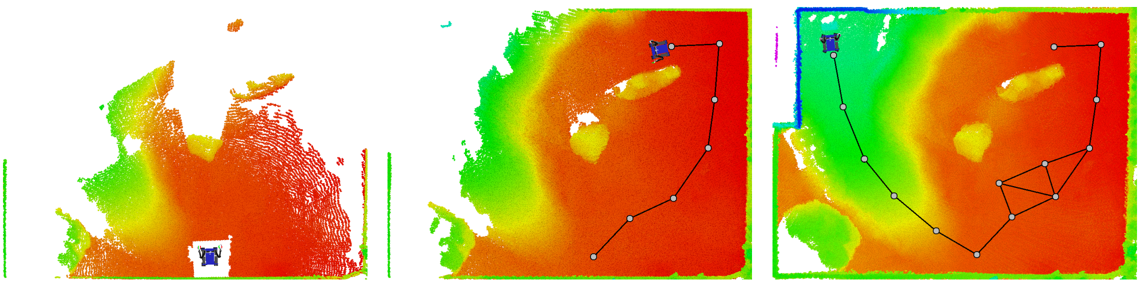


Figure 11: The allocentric map from a top view at different time steps, consisting of 1 (left), 7 (middle) and 14 (right) key frames. Color encodes height. The nodes in the pose graph (grey circles) are connected by spatial constraints (black lines). The robot model shows the current position of the robot.

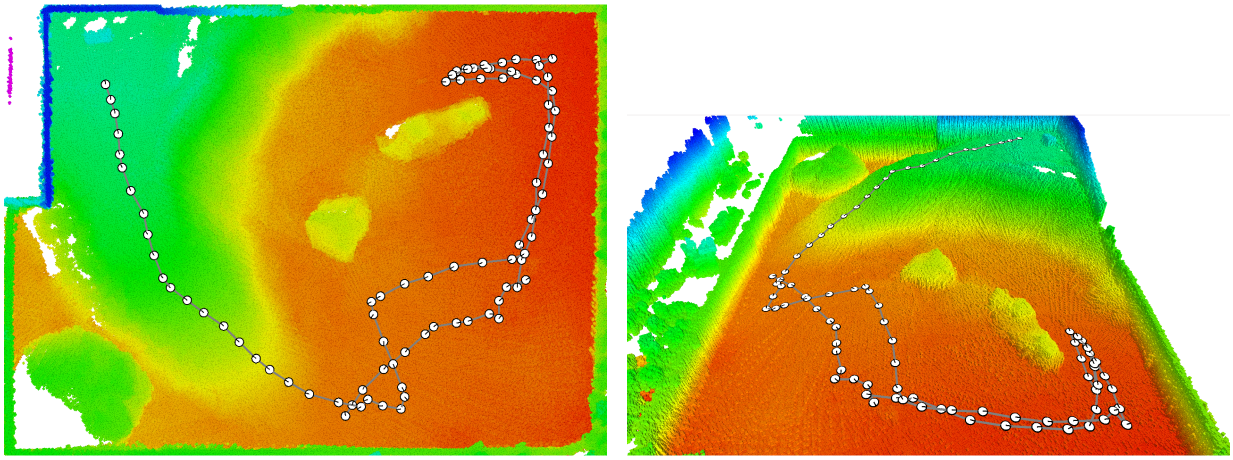


Figure 12: The resulting allocentric map from two different perspectives with the localization poses (black circle) from our run. Color encodes height from ground.

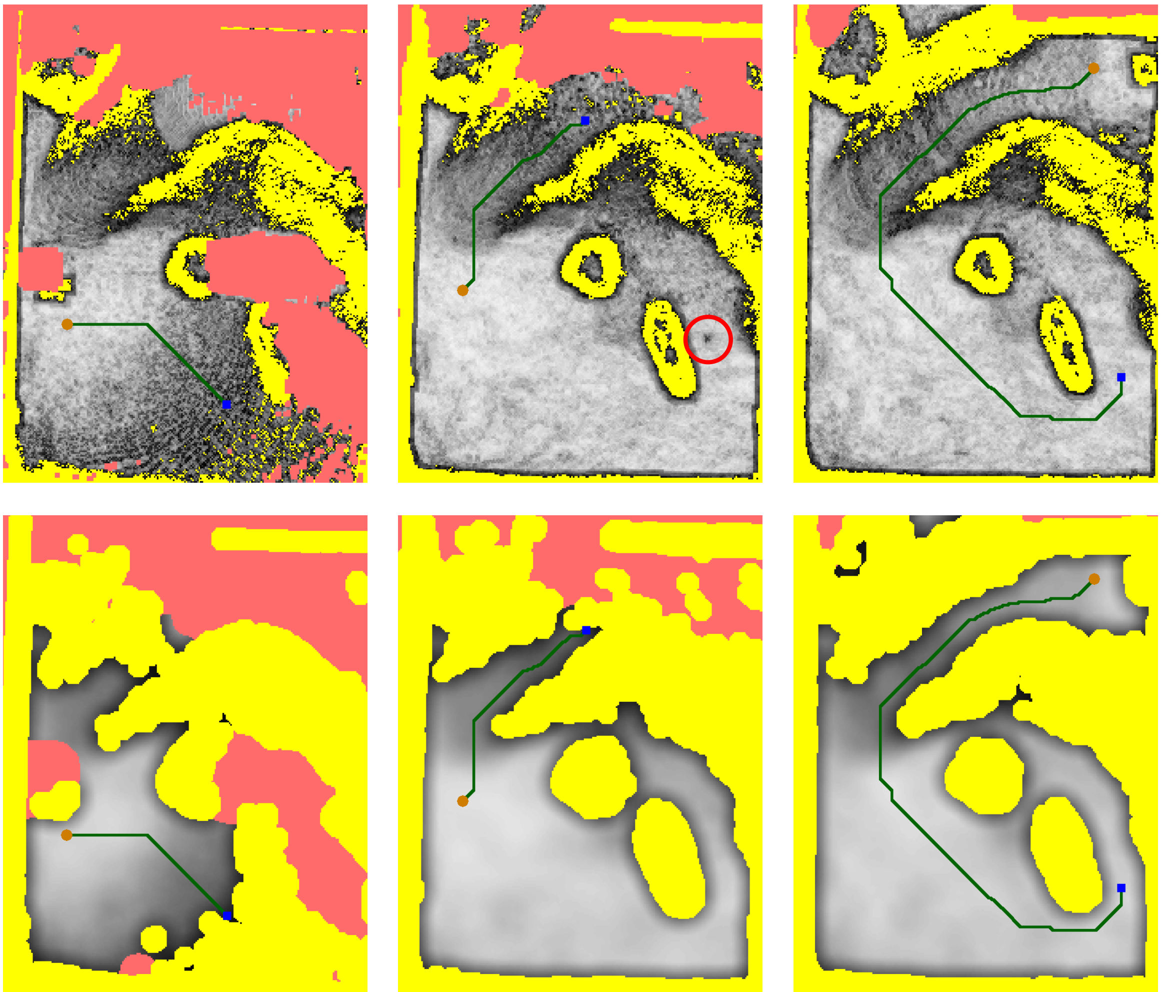


Figure 13: Navigation planning during (left, middle) and after exploration (right) of the SpaceBot Camp arena. The top row shows the calculated traversability costs for each cell. The bottom row shows inflated costs used for A* path planning. The orange dot represents the current robot position, the blue square the target position. The planned path is shown in green. Red areas indicate insufficient measurements for traversability analysis. Yellow areas correspond to absolute obstacles, which the robot may not traverse. In the middle situation, a small battery pack (20 cm × 10 cm × 4 cm) can be seen in the uninflated costs (marked with red circle, also shown in Figure 5).

## 9. Conclusions

We presented our local and allocentric mapping systems that uses an efficient 3D multiresolution map to aggregate measurements from a continuously rotating laser scanner and align acquired scans with it. By using local multiresolution, we gain computational efficiency by having a high resolution in the near vicinity of the robot and a lower resolution with increasing distance from the robot, which correlates with the sensor characteristics in relative distance accuracy and measurement density.

Scan registration is used to estimate the motion of the robot by aligning consecutive 3D scans to the map. We do not match individual scan points, but represent 3D scans also in local multiresolution grids and condense the points into surface elements for each grid cell. These surface elements are aligned efficiently and at high accuracy in a registration framework which overcomes the discretization in a grid through probabilistic assignments.

Modeling measurement distributions within voxels by surface elements allows for efficient and accurate registration of 3D scans with the local map. The incrementally built local dense 3D maps of nearby key poses are registered globally by graph optimization. This yields a globally consistent dense 3D map of the environment. Continuous registration of local maps with the global map allows for tracking the 6D robot pose in real time. We demonstrate accuracy and efficiency of our approach by showing consistent allocentric 3D maps in difficult environments with rough terrain.

The high-quality 3D environment representations were a key success factor for our team in the competitions. During the DRC Finals, our team NimbRo Rescue solved seven of the eight tasks in only 34 minutes, coming in 4th overall. Our team NimbRo Explorer was the only team to solve all tasks of the DLR SpaceBot Camp.

[1] D. Droeschel, J. Stückler, S. Behnke, Local multi-resolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner, in: Robotics and Automation (ICRA), IEEE International Conference on, 2014.

[2] D. Droeschel, J. Stückler, S. Behnke, Local multi-resolution surfel grids for mav motion estimation and 3d mapping, in: Proc. of the Int. Conf. on Intelligent Autonomous Systems, 2014.

[3] A. Nuechter, K. Lingemann, J. Hertzberg, H. Surmann, 6D SLAM with approximate data association, in: Int. Conf. on Advanced Robotics, 2005.

[4] M. Magnusson, T. Duckett, A. J. Lilienthal, Scan registration for autonomous mining vehicles using 3D-NDT, Journal of Field Robotics 24 (10) (2007) 803–827.

[5] S. Kohlbrecher, O. von Stryk, J. Meyer, U. Klingauf, A flexible and scalable slam system with full 3d motion estimation, in: Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on, 2011, pp. 155–160.

[6] D. Cole, P. Newman, Using laser range data for 3d slam in outdoor environments, in: Robotics and Automation (ICRA), IEEE International Conference on, 2006, pp. 1556–1563.

[7] J. Stückler, S. Behnke, Multi-resolution surfel maps for efficient dense 3D modeling and tracking, Journal of Visual Communication and Image Representation 25 (1) (2014) 137–147.

[8] M. Schadler, J. Stückler, S. Behnke, Multi-resolution surfel mapping and real-time pose tracking using a continuously rotating 2D laser scanner, in: Proc. of 11th IEEE Int. Symposium on Safety, Security, and Rescue Robotics (SSRR), 2013.

[9] P. J. Besl, N. D. McKay, A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 14 (2) (1992) 239–256.

[10] A. Myronenko, X. Song, Point set registration: Coherent point drift, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (12) (2010) 2262–2275.

[11] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, J. Zhang, Rigid and articulated point registration with expectation conditional maximization, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (3) (2011) 587–602.

[12] G. D. Evangelidis, D. Kounades-Bastian, R. Horaud, E. Z. Psarakis, A generative model for the joint registration of multiple point sets, in: European Conference on Computer Vision, Springer, 2014, pp. 109–122.

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, OctoMap: an efficient probabilistic 3D mapping framework based on octrees, Autonomous Robots 34 (2013) 189–206.

[14] J. Ryde, H. Hu, 3D mapping with multi-resolution occupied voxel lists, Autonomous Robots 28 (2010) 169 – 185.

[15] T. Stoyanov, M. Magnusson, H. Andreasson, A. J. Lilienthal, Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations, The International Journal of Robotics Research 31 (12) (2012) 1377–1393.

[16] D. Belter, P. Łabcki, P. Skrzypczyński, Estimating terrain elevation maps from sparse and uncertain multi-sensor data, in: Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on, IEEE, 2012, pp. 715–722.

[17] M. Hebert , C. Caillas, E. Krotkov, I. S. Kweon, T. Kanade, Terrain mapping for a roving planetary explorer, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '89), Vol. 2, 1989, pp. 997–1002.

[18] R. Triebel, P. Pfaff, W. Burgard, Multi-level surface maps for outdoor terrain mapping and loop closing, in: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, 2006, pp. 2276–2282.

[19] P. Pfaff, R. Triebel, W. Burgard, An efficient extension to elevation maps for outdoor terrain mapping and loop closing, The International Journal of Robotics Research 26 (2) (2007) 217–230.

[20] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, R. Siegwart, Robot-centric elevation mapping with uncertainty estimates, in: International Conference on Climbing and Walking Robots (CLAWAR), Poznań, Poland, 2014, pp. 433–440.

[21] D. Droeschel, M. Nieuwenhuisen, M. Beul, D. Holz, J. Stückler, S. Behnke, Multilayered mapping and navigation for autonomous micro aerial vehicles, Journal of Field Robotics (JFR).

[22] J. Stückler, M. Schwarz, M. Schadler, A. Topalidou-Kyniazopoulou, S. Behnke, NimbRo Explorer: Semiautonomous exploration and mobile manipulation in rough terrain, Journal of Field Robotics (JFR).

[23] M. Bosse, R. Zlot, Continuous 3D scan-matching with a spinning 2D laser, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2009.

[24] J. Elseberg, D. Borrmann, A. Nuechter, 6DOF semi-rigid SLAM for mobile scanning, in: Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on, 2012.

[25] T. Stoyanov, A. Lilienthal, Maximum likelihood point cloud acquisition from a mobile platform, in: Proc. of the Int. Conf. on Advanced Robotics (ICAR), 2009.

[26] W. Maddern, A. Harrison, P. Newman, Lost in translation (and rotation): Fast extrinsic calibration for 2D and 3D LIDARs, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2012.

[27] S. Anderson, T. D. Barfoot, Towards relative continuous-time SLAM, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2013.

[28] A. Hornung, M. Phillips, E. G. Jones, M. Bennewitz, M. Likhachev, S. Chitta, Navigation in three-dimensional cluttered environments for mobile manipulation, in: Robotics and Automation (ICRA), IEEE International Conference on, 2012.

[29] J. Klaess, J. Stueckler, S. Behnke, Efficient mobile robot navigation using 3D surfel grid maps, in: Proc. German Conf. on Robotics (ROBOTIK), 2012.

[30] J. Lee, C. Pippin, T. Balch, Cost based planning with rrt in outdoor environments, in: Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on, 2008.

[31] B. P. Gerkey, K. Konolige, Planning and control in unstructured terrain, in: Proc. of the ICRA Workshop on Path Planning on Costmaps, 2008.

[32] D. Ferguson, M. Likhachev, Efficiently using cost maps for planning complex maneuvers, in: Proc. of the ICRA Workshop on Planning with Cost Maps, 2008.

[33] T. Stoyanov, M. Magnusson, H. Andreasson, A. J. Lilienthal, Path planning in 3D environments using the normal distributions transform, in: Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on, 2010.

[34] R. Bogdan Rusu, A. Sundaresan, B. Morisset, K. Hauser, M. Agrawal, J.-C. Latombe, M. Beetz, Leaving flatland: Efficient real-time three-dimensional perception and motion planning, Journal of Field Robotics 26 (10) (2009) 841–862.

[35] S. Chhaniyara, C. Brunskill, B. Yeomans, M. Matthews, C. Saaj, S. Ransom, L. Richter, Terrain trafficability analysis and soil mechanical property identification for planetary rovers: A survey, J. Terramechanics 49 (2) (2012) 115–128.

[36] P. Papadakis, Terrain traversability analysis methods for unmanned ground vehicles: A survey, Engineering Applications of Artificial Intelligence 26 (4) (2013) 1373–1385.

[37] T. Rodehutskors, M. Schwarz, S. Behnke, Intuitive bimanual telemanipulation under communication restrictions by immersive 3d visualization and motion tracking, in: Proc. of the IEEE-RAS Int. Conference on Humanoid Robots (Humanoids), 2015.

[38] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[39] J. Amanatides, A. Woo, A fast voxel traversal algorithm for ray tracing, in: In Eurographics 87, 1987, pp. 3–10.

[40] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, G2o: A general framework for graph optimization, in: Robotics and Automation (ICRA), IEEE International Conference on, 2011.