EFFICIENT METHODS FOR LIDAR-BASED MAPPING AND LOCALIZATION

DISSERTATION

zur Erlangung des Doktorgrades (Dr. rer. nat.)

der Mathematisch-Naturwissenschaftlichen Fakultät

der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

DAVID MARCEL DROESCHEL

aus

Bonn

Bonn, 2019

Angefertigt mit Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

Erster Gutachter:	Prof. Dr. Sven Behnke
Zweiter Gutachter:	Prof. Dr. Cyrill Stachniss
Tag der Promotion:	31. August 2020
Erscheinungsjahr:	2020

Abstract

An expedient aim in robotics research is to enable robot systems to enter areas which are inaccessible or too dangerous to humans, such as disaster scenarios. For autonomous navigation in these environments, robust and reliable perception is key. One fundamental perception problem is to build maps of unknown environments and to localize within them simultaneously.

This thesis presents an approach to simultaneous localization and mapping (SLAM) using local multiresolution. Local multiresolution corresponds well to the measurement density and accuracy of most range sensors and allows for efficient and concise map representations. The proposed map data structure is designed for memory-efficient aggregation of measurements and enables online mapping and localization. To align acquired sensor data, a probabilistic registration method is proposed, exploiting the properties of the map data structure.

Local multiresolution maps from different view poses are aligned with each other to create an allocentric map of the environment. Optimization of the view poses yields a globally consistent dense 3D map of the environment. Continuous registration of local maps with the global map allows for tracking the robot pose in real time.

Furthermore, a method for reassessing previously aggregated measurements, to account for registration errors due to missing information or erroneous sensor data is proposed. In order to incorporate corrections when refining the alignment, the individual sensor poses of the measurements in the local map are modeled as a sub-graph in a hierarchical graph structure. Sensor poses in the sub-graphs are optimized to account for drift and misalignments in the local maps. Each sub-graph maintains a continuous-time representation of the sensor trajectory to interpolate measurements between discrete sensor poses.

The proposed methods are evaluated on different datasets and compared to state-of-the-art methods, with the results indicating superior accuracy and efficiency. Particular applications demonstrate that they have been successfully employed on different robotic platforms, such as micro aerial vehicles and ground robots, in different research projects and robot competitions.

Zusammenfassung

Ein Ziel in der Robotikforschung ist es, Robotersystemen den Eintritt in Bereiche zu ermöglichen, die für Menschen unzugänglich oder zu gefährlich sind, wie beispielsweise Katastrophenszenarien. Für autonome Navigation in diesen Umgebungen, ist eine robuste und zuverlässige Wahrnehmung entscheidend. Ein grundlegendes Wahrnehmungsproblem in der Robotik besteht darin, Karten von unbekannten Umgebungen zu erstellen und sich gleichzeitig in ihnen zu lokalisieren.

Kern dieser Arbeit ist ein Ansatz zur simultanen Lokalisierung und Kartierung (Simultaneous Localization and Mapping, SLAM) basierend auf lokaler Multiresolution. Lokale Multiresolution entspricht der Charakteristik, in Bezug auf Messdichte und Genauigkeit, der meisten Entfernungssensoren und ermöglicht eine effiziente Repräsentation der Umgebung. Die vorgeschlagene Kartendatenstruktur, gennant Lokale Multi-Resolutions-Karten, ist ausgelegt für Speicher-effiziente Aggregation von Messungen und ermöglicht Online-Kartierung und Lokalisierung. Um erfasste Sensordaten zu registrieren, wurde ein probabilistisches Verfahren entwickelt, welches die Eigenschaften der Kartendatenstruktur nutzt.

Lokale Multi-Resolutions-Karten aus verschiedenen Schlüsselposen werden miteinander registriert um eine allozentrische Repräsentation der Umgebung zu erstellen. Optimierung der Schlüsselposen ergibt eine global konsistente, dichte 3D-Karte. Die fortlaufende Registrierung von lokalen Karten mit der globalen Karte ermöglicht die Lokalisierung des Roboters in Echtzeit.

Weiterhin wird eine Methode zum Korrigieren zuvor aggregierter Messungen und Aussbessern von Registrierungsfehler, die aufgrund fehlender Informationen oder fehlerhafter Sensordaten entstehen, vorgestellt. Um diese Korrekturen in der Repräsentation zu berücksichtigen, werden die Sichtposen einzelner Messungen in der lokalen Karte als Teilgraph in einer hierarchischen Graphenstruktur modelliert. Die Sichtposen der Messungen eines Teilgraphen werden optimiert, um Drift und Fehlausrichtungen in den lokalen Karten zu berücksichtigen. Um Posen von Messungen zwischen diskreten Sensorpositionen zu interpolieren, verwaltet jeder Untergraph eine zeitkontinuierliche Repräsentation der Sensortrajektorie.

Die vorgeschlagenen Methoden werden anhand verschiedener Datensätze evaluiert und mit Methoden die Stand der Forschung sind verglichen, wobei die Ergebnisse der Evaluation höhere Genauigkeit und Effizienz aufweisen. Auërdem wird

der Einsatz der Methode auf unterschiedlichen Roboterplattformen, wie Flug- oder Bodenroboter gezeigt.

Acknowledgements

First of all, my deepest gratitude goes to my advisor Professor Sven Behnke, for helping me with his enormous experience, pushing me in the right direction, extending my horizon when I got stuck, the many opportunities to participate in interesting challenges, and finally giving me the chance to write this thesis. Furthermore, I'd like to thank Professor Cyrill Stachniss, Professor Reinhard Klein, and Professor Horst Bleckmann, for serving on my committee.

Next, special thanks to colleagues, students, and friends from the Autonomous Intelligent Systems group in Bonn. In order of appearance, Nils, Thomas, Micha, Jörg, Marcel, Kathrin, Matthias, Hannes, Dirk, Max, Marion, Marius, Tobi, Tobi, Nicola, Sebastian, Jan, and Nadine. Thank you for the great atmosphere, inspiring and fruitful discussions, short breaks, after-work calibration sessions, and having a great time during competitions.

Above all, I'd like to thank my family—especially my parents Marlene and Wolfgang—and friends for the endless support and understanding. Shout-out to Tim Büthe, Simon Grützner, Stefan Gundelach, Schrotti, Hauke Müller and Jens Kessler, for support, encouragement, shelter, ride, criticism, and welcome distraction. Without you guys, this project would have been either impossible or much faster. Furthermore, I wanna express my gratitude to Kerstin I, Linda, Caro, Kerstin II, and Lin for their relentless support.

Contents

List of Figures

List of Tables

1. Introduction

Recent advancements in robotics and artificial intelligence led to a number of variations in our every-day live, making our work easier but also inducing an economic revolution by reducing physical labor. While superseding human labor by intelligent machines is contentious, the increase of computing power and advancements in artificial intelligence and robotics facilitates an expedient aim in robotics: making our lives safer. For instance, enabling robot systems to enter areas inaccessible or too dangerous for humans—such as disaster scenarios—or taking up on activities where humans tend to fail—such as driving—will safe a tremendous amount of lives. However, while today's intelligent machines can reliably operate in well specified narrow situations, our aspired live-saving rescue robot needs the ability to autonomously navigate in unknown, diverse environments.

Autonomous navigation requires perceptual capabilities that are self-evident for humans. These capabilities include the ability to build an internal representation of the environment from perceptual inputs. This internal representation, called the map, is used to infer about location from new perceptual inputs and to plan paths. When entering an unknown environment, we simultaneously build this representation and localize within. In robotics this problem is known as simultaneous localization and mapping (SLAM). SLAM has been studied in the robotics community for decades, aiming to allow robots to navigate autonomously in scenarios where a map is not available beforehand. Consequently, a number of approaches addressing the SLAM problem exist (Cadena et al., 2016). However, SLAM is an extensive problem and it depends on a number of aspects, such as the type of the robot, the configuration of the environment, and the desired performance requirements. Thus, depending on the specific aspects, research in this field deserves more or attention or is more mature. For instance, for a wheeled robot platform, equipped with a 2D laser range finder and wheel encoders, moving on a planar indoor environment the SLAM problem is sometimes considered as solved and reliable approaches exist. In contrast, SLAM on a highly dynamic platform—such as a micro aerial vehicle (MAV)— with limit computing resources in a cluttered 3D environment is still actively discussed in the research community. Here, especially considering the vast amount of data today's sensors produce, efficiency, scalability and robustness are major limiting factors. For example, a modern lidar sensor pro-

duces up to 1 million 3D measurements per second that need to be processed with a certain performance, to allow for consistent mapping and robust localization.

Often the huge amount of data and the aforementioned limitations are addressed by relaxing the problem. For example, by modeling the environment in lower dimensionality, such as projecting 3D measurements in a 2D representation. Furthermore, various simplification to the geometry of data and environment, by considering only significant parts of the data or modeling the environment by compressing geometry. On the contrary, these relaxations require assumptions about the environment, for example supposing that geometry in the environment is either planar or has a certain structure, which can not be assumed in general. Similarly, modeling the environment in lower dimensionality makes certain assumptions about the environment which are not feasible in many situations, such as representing the environment in a 2D map for overhanging structures, or robots that move on non-planar trajectories.

This thesis presents an approach to SLAM that differs from the previously mentioned approaches. Based on the idea, that a proper representation of the geometry in the environment is key, a map representation is proposed to facilitate online mapping and localization.

Central to our approach are so-called local multiresolution grid maps, which is a concise map representation, allowing for efficient aggregation of sensor measurements in a dense map, modeling the geometry of the environment. Local multiresolution grid maps partition the environment in 3D volume pixels (voxels) with varying resolution. Local multiresolution hereby refers to the property of having a fine resolution close to the robot and a coarser resolution farther away. This inherent discretization correlates with the characteristics in relative distance accuracy and measurement density of modern 3D lidars and reduces memory and computational requirements without loosing information. Furthermore, local multiresolution grid maps are robot-centric maps that *move* with the robot resulting in a constant number of grid cells necessary to model the robots surrounding, independent of the traveled distance. The latter allows to accomplish constant memory consumption and computation time. Chapter 3 details local multiresolution grid maps as the central data structure in this thesis.

However, to aggregate sensor measurements in local multiresolution grid maps—and thereby building a dense representation of the geometry in the environment—registration of newly acquired sensor data is key. Registration means to estimate the sensor motion between the current sensor measurement and the map by aligning them. For this alignment, a probabilistic registration method is presented in Chapter 4, leveraging the local multiresolution property and allowing for efficient and accurate registration. It allows to align new sensor data to a local multireso-

lution map, but also for the alignment of two maps with each other.

The latter allows to generate spatial constraints between two maps from different view poses. These spatial constraints and the local maps from different view poses are used to model the environment in a graph-based structure. When the robot revisits a part of the environment, a spatial constraint between the current local map and the previously acquired map allows to compensate for accumulated inaccuracies in the motion estimate (called drift). The graph is optimized in order to minimize the accumulated error. The resulting pose graph maps larger environments and provides a allocentric frame of reference for localization. Registration of scans to the local map—and thereby estimating the sensor's motion with respect to a local frame—is called the *front-end*. In contrast, construction of the pose graph, its optimization and the alignment of the current local map to it, is called the *back-end*. The proposed back-end using local multiresolution maps is described in Chapter 5.

Inaccuracies, caused by wrong data associations or missing information, may remain. For example, incrementally mapping the environment necessitates bootstrapping from sparse sensor data at the beginning—resulting in relatively poor registration accuracy, compared to aligning with a dense and accurate map. Consequently, map quality degrades due to misaligned sensor data. To overcome degradation of the map quality, Chapter 6 proposes an approach for reassessing the registration of previously added 3D scans. By modeling individual 3D scans of a local map as a sub-graph, we build a hierarchical graph structure, enabling refinement of the map in case misaligned measurements when more information is available. Furthermore, the sensor trajectory is modeled by a continuous-time representation, allowing to incorporate refinement results on on the finest-possible granularity.

1.1. Scientific Contributions

This thesis contains the following contributions:

A robot-centric 3D map representation for efficient aggregation of lidar measurements coined local multiresolution grid map is proposed in Chapter 3. Local multiresolution grid maps discretize the environment in voxels, having a fine resolution close to the robot and coarser resolution farther away, which correlates with the sensor's characteristics in relative distance accuracy and measurement density. Furthermore, it's an egocentric representation centered around the robot. Aggregated measurements in local multiresolution

grid maps *move* according to the robot's motion, preventing costly reprocessing of the data. This allows efficient local mapping with constant computational and memory requirements. Compared to other grid-based map representations, local multiresolution allows to considerably reduce the number of grid cells without losing information.

Surfel-based registration for local multiresolution grid maps using an efficient, probabilistic approach, is proposed in Chapter 4. To allow for accurate aggregation of measurements and precise estimation of the sensor's motion, an efficient probabilistic registration approach, leveraging the local multiresolution property, is presented. Surface-elements (surfels) that summarize the statistics of the measurements in a grid cells are used for registration of new sensor measurements to the map in an Expectation Maximization (EM) framework. The local multiresolution property allows a coarse-to-fine strategy allowing for robust and accurate registration. Evaluation of the approach shows superior over state-of-the-art methods.

A graph-based allocentric mapping approach using local multiresolution grid maps is proposed in Chapter 5. To facilitate allocentric mapping and localization, while taking advantage of the properties of local multiresolution, a pose graph consisting of local maps from different view poses is presented. By surfel-based registration, spatial constraints between view poses are used to construct to pose graph. Accumulated drift is minimized by spatial constraints between nearby view poses. Graph optimization globally minimizes the accumulated error and find the most likely configuration of constraints in the graph.

A hierarchical refinement method allowing for reassessing the registration of previously added measurements. By modeling individual 3D scans of a local map as a sub-graph, we build a hierarchical graph structure, enabling refinement of the map in case misaligned measurements when more information is available. Furthermore, the approach preserves efficient local and allocentric mapping, as with our previous method. In summary, the contribution of our work is a novel combination of a hierarchical graph structure—allowing for scalability and efficiency—with local multiresolution maps to overcome alignment problems due to sparsity in laser measurements, and a continuous-time trajectory representation.

1.2. Open-Source Software

To facilitate developing robotic applications, contributing to the system, and for comparing and reproducing results, parts of the described methods are published open-source[1], making it available to other researchers. The current release includes an implementation of local multiresolution grid maps, surfel-based registration and the allocentric mapping back-end. Furthermore, most data sets used in the experiments were made publicly available[2].

1.3. Publications

Parts of this thesis have been published in conference proceedings and journals. The publications are provided in chronological order:

- D. Droeschel and S. Behnke (2018). "Efficient Continuous-time SLAM for 3D Lidar-based Online Mapping". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*

- D. Droeschel et al. (2017). "Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner". In: *Robotics and Autonomous Systems* 88, pp. 104 –115

- D. Droeschel and S. Behnke (2017). "MRSLaserMap: Local Multiresolution Grids for Efficient 3D Laser Mapping and Localization". In: *RoboCup 2016: Robot World Cup XX*. ed. by S. Behnke et al. Cham: Springer International Publishing, pp. 319–326

- D. Droeschel et al. (2015). "Multilayered Mapping and Navigation for Autonomous Micro Aerial Vehicles". In: *Journal of Field Robotics (JFR)*

- D. Droeschel et al. (Oct. 2014c). "Omnidirectional Perception for Lightweight MAVs using a Continuously Rotating 3D Laser". In: *PFG Photogrammetrie, Fernerkundung, Geoinformation* 2014.5, pp. 451–464. URL: `http://dx.doi.org/10.1127/1432-8364/2014/0236`

- D. Droeschel et al. (2014b). "Local Multi-Resolution Surfel Grids for MAV Motion Estimation and 3D Mapping". In: *Proc. of the Int. Conference on Intelligent Autonomous Systems (IAS)*

[1] `https://github.com/AIS-Bonn/mrs_laser_map`
[2] `http://www.ais.uni-bonn.de/laser_mapping`

- D. Droeschel and S. Behnke (2014). "3D Local Multiresolution Grid for Aggregating Omnidirectional Laser Measurements on a Micro Aerial Vehicle". In: *ISR/Robotik 2014; 41st International Symposium on Robotics*, pp. 1–7

- D. Droeschel et al. (2014a). "Local Multi-Resolution Representation for 6D Motion Estimation and Mapping with a Continuously Rotating 3D Laser Scanner". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*

2. Sensors and Robotic Platforms

This chapter gives an overview over the robotic platforms, especially the sensor setup, used for data collection and experimental evaluation in this thesis. Furthermore, efforts in building custom light detection and ranging (LIDAR) sensor and necessary methods for data acquisition and preprocessing are briefly summarized.

The following chapters introduce methods that have been developed for omnidirectional distance sensor, such as 3D LIDAR sensors. Although the approach is not limited to LIDAR sensors in general, it leverages certain characteristics of them, such as the distance-related measurement density and noise characteristics.

This chapter gives an overview of the sensors and the robotic platforms that have been used for experimental evaluation of the proposed methods. Depending on the application and the employed robotic platform, different requirements and constraints are imposed. These include measurement range, measurement rate, size and weight of the sensors. After introducing three different MAV platforms with different sensor setups, a mobile ground robot with a combination of wheeled and legged limbs is described. Compared to the mobile ground robot, MAVs are highly dynamic and have strong size and weight requirements. Furthermore, depending on the measurement rate of the employed LIDARs, additional sensors might be required to capture the motion during acquisition. The additional motion estimate is used to compensate distortion of the measurements during one revolution of the LIDAR sensor.

The methods developed in this thesis are intended to enable autonomous navigation of these robots. Thus, experiments assessing the applicability of the sensor setup for obstacle detection complete this chapter.

2.1. Micro Aerial Vehicles

MAVs such as quadrotors have attracted attention in the field of robotics. Their size and weight limitations pose a challenge in designing sensory systems. Most of today's MAVs are equipped with ultra sound sensors and camera systems due to their minimal size and weight. While these small and lightweight sensors provide valuable information, they suffer from a limited field-of-view (FoV) and are sensitive to illumination conditions. Only few systems Bachrach et al., 2009; Grzonka et al., 2009; Shen et al., 2011; Tomić et al., 2012 are equipped with 2D laser range finders (LRFs) that are used for navigation.

In contrast, the following subsections describe two custom-built continuously rotating laser scanners that are minimalistic in terms of size and weight and thus particularly well suited for obstacle perception and localization on MAVs, allowing for environment perception in all directions.

2.1.1. Lightweight Continuously Rotating Laser Scanners

Figure 2.1 shows the first MAV used in this thesis. It is equipped with a custom-built continuously rotating laser scanner. The sensor consists of a Hokuyo UTM-30LX-EW 2D LRF, which is rotated by a Robotis Dynamixel MX-28 servo actuator to gain a three-dimensional FoV. As shown in Figure 2.1, the scanning plane is parallel to the axis of rotation, but the heading direction of the scanner is twisted slightly away from the direction of the axis—in order to enlarge its FoV.

The 2D LRF is electrically connected by a slip ring, allowing for continuous rotation of the sensor. The axis of rotation is pitched downward by 45° in forward direction, which places the core of the MAV upwards behind the sensor as depicted in Figure 2.1. Hence, the sensor can measure in all directions, except for a conical blind spot pointing upwards behind the robot.

The 2D laser scanner has a size of 62 mm×62 mm×87.5 mm and a weight of 210 g. Together with the actuator (72 g) and the slip ring, the total weight of the 3D scanner is approximately 400 g. It 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 measurement accuracy is specified by the manufacturer with $\pm 30\,mm$ at $0.1 - 10\,m$ distance ($\pm 50\,mm$ at $10 - 30\,m$). The Dynamixel actuator rotates the 2D LRF at one rotation per second, producing 40 scan lines and 43,200 distance measurements per full rotation. Slower rotation is possible if a higher angular resolution is desired. For our setup, a half rotation leads to a full 3D scan of most of the environment. Hence, we can acquire 3D scans with up to 21,600 points with 2 Hz. Figure 2.2 shows a resulting 3D scan from a parking

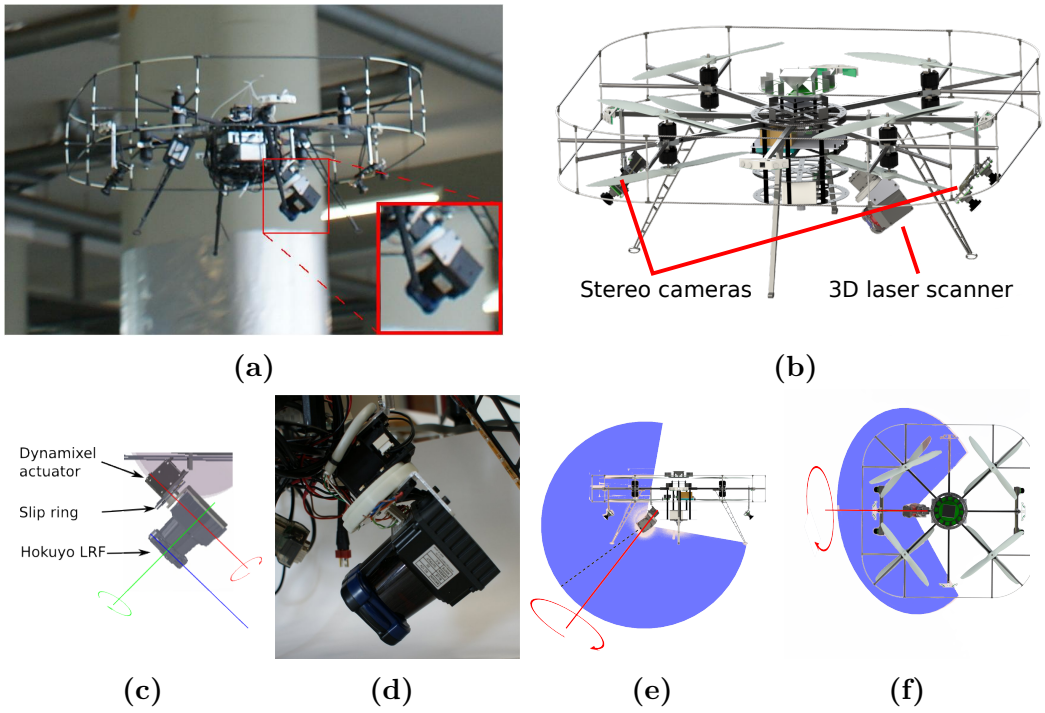


Figure 2.1.: The MAV equipped with a 3D laser scanner and a stereo camera system (a). CAD drawings of (b) our MAV and (c) the continuously rotating laser scanner. The Hokuyo 2D LRF is mounted on a bearing and rotated around the red axis. Its mirror is rotated around the green axis, resulting in a 2D measurement plane (blue). (d) Photo of the sensor. (e + f) CAD drawings illustrating the FoV of individual scans of the laser scanner (blue) from side and top view. The black dashed line illustrates the center of the measurement plane. The 2D LRF is rotated around the red axis.

garage. The offset between the scanned plane section and the rotation axis results in a different FoV for two different half rotations which is shown in Figure 2.3a. In this way, occlusion from small parts of the MAV is reduced significantly.

The Hokuyo UTM-30LX-EW is able to measure up to three echoes of a single emitted light pulse. The number of echoes for a light pulse depends on the surface of the measured objects, i.e., shape and reflectivity. For example, transparent materials, vegetation or edges of buildings often yield more than one echo. Often, the second echo comes from a structure in the original pulse direction, behind a partial occlusion, which means that it can be treated as an additional distance measurement. Measurements from the first and the second echo are shown in Figure 2.3b.

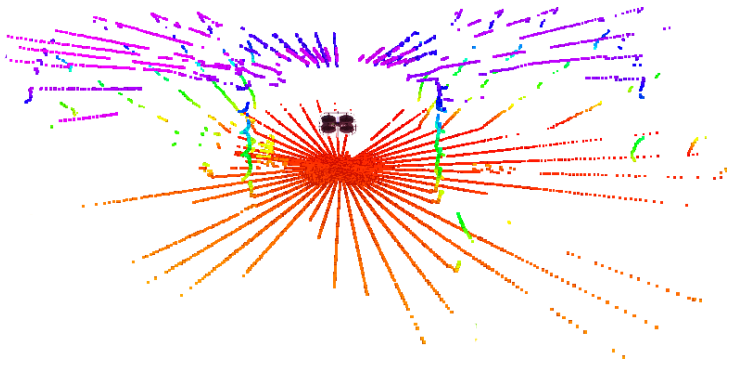


Figure 2.2.: Indoor 3D scan acquired with our continuously rotating laser scanner. Color encodes height.

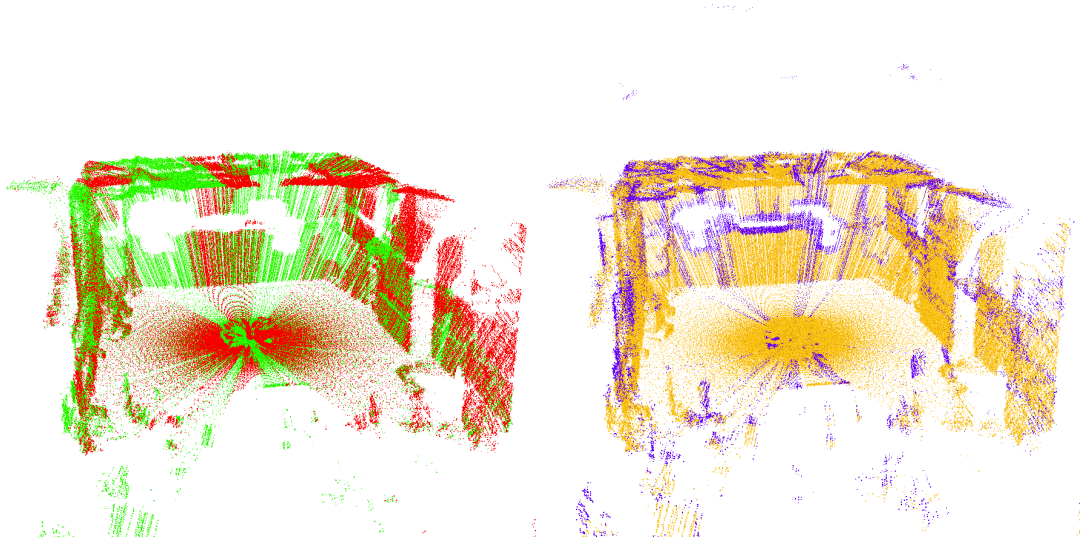
2.1.2. Lightweight Continuously Rotating Dual Laser Scanner

The second MAV platform used in this thesis is shown in Figure 2.4. The actuators for rotation are similar to the previous setup, resulting in the same horizontal angular resolution and rotational velocity. Furthermore, the update rate of the sensor for acquiring full 3D scans is also 2 Hz. In contrast to the first rotating laser scanner, the sensor combines two Hokuyo UST-20LX LRFs mounted on a bearing plate. Figure 2.4 illustrates the scanner arrangement, showing that one scanning plane is parallel to the axis of rotation while the other is twisted by 45°. This arrangement results in a cylindrical blind spot of the first scanner and a conical, upward pointing blind spot for the twisted scanner. Hence, this setup maximizes the FoV and results in a minimal blind spot. Since the blind spot is pointing upwards, it does not degrade mapping or obstacle detection in our scenario. Furthermore, due to the scanner arrangement, a half rotation of the bearing plate results in a complete 3D scan.

Each 2D laser range finder has a scanning frequency of 40 Hz with 1,080 measurements per scan plane resulting in 43,200 measurements per second. Figure 2.4 shows resulting point clouds of the environment perceived by each laser and the combined point cloud. Each scanner weights 143 g (without cables). The whole sensor assembly weights 420 g including motor, a network switch, and a slip ring allowing for continuous rotation.

Preprocessing

The wide FoV of the laser scanner inherently leads to many measurements lying on the MAV itself. Considering the complex structure of the MAV, with moving parts like propellers, measurements that belong to the robot's body are removed in a preprocessing step. This so-called *self filter* approximates the model of the MAV



(a) (b)

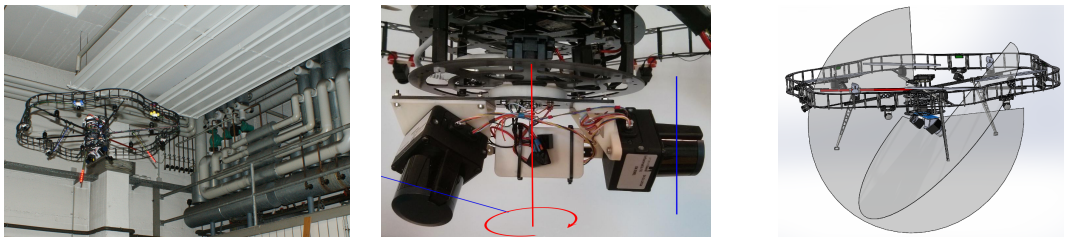
Figure 2.3.: Accumulated 3D scans of an indoor environment. (a) The color encodes the different half rotations of the scanner. Moving the optical center of the 2D LRF away from the rotation axis of the actuator results in different self-occlusions of the scans from the first (green) and the second (red) half rotation. (b) Measurements from first echo (yellow) and the second echo (purple). In case of partial occlusions, e.g., by the MAV itself, multi-echo detection leads to an increase of distance measurements.

by a cylinder with the diameter and height of the MAV. Furthermore, we use a modified shadow filter to remove not only incorrect measurements at the edges of the geometry, but also erroneous measurements caused by the fast rotating propellers. Filtering results are shown in Figure 2.5.

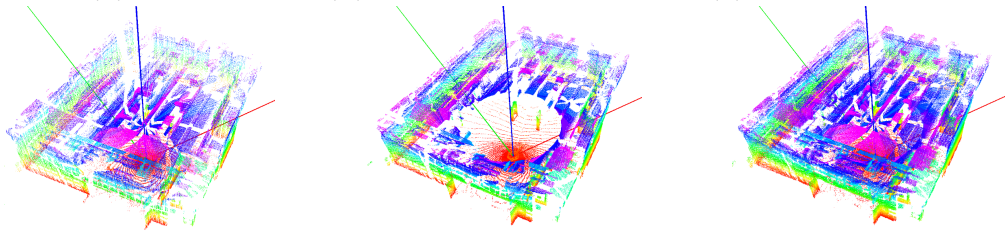
3D Scan Assembly

Due to the relative slow rotation of the 2D laser scanner, compared to the motion of the robot, the first two MAV platforms are equipped with additional sensors, allowing to recover the sensor motion during one half rotation of the sensor. Thus, when assembling 3D scans from raw laser scans, for the rotation of the sensor w.r.t. the MAV and for the motion of the robot during acquisition, is accounted.

First, measurements of individual scan lines are undistorted with regards to the rotation of the 2D laser scanner around the servo rotation axis (red axis in Figures 2.1 and 2.4). Here, the rotation between the acquisition of two scan lines is distributed over the measurements by using spherical linear interpolation. Second, the motion of the MAV during acquisition of a full 3D scan is compensated. To this end, a motion estimate from the low-level filters running on the Pixhawk

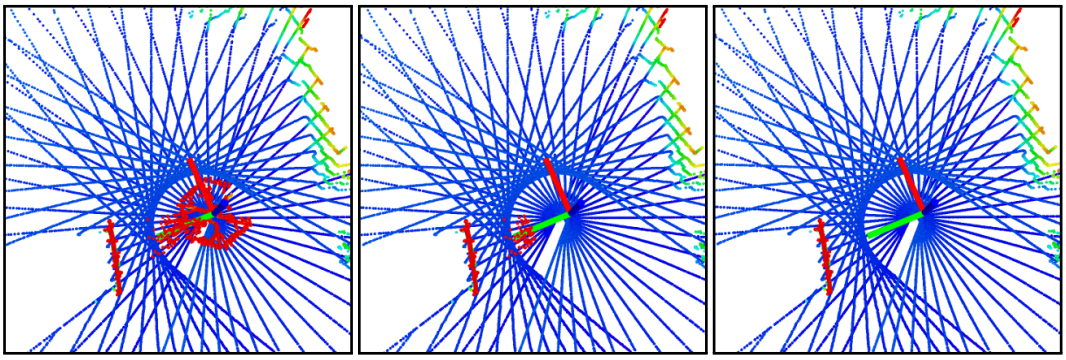


(a) **(b)** Laserscanners Photo **(c)** Laserscanners CAD



(d) Laser scanner 1 **(e)** Laser scanner 2 **(f)** Combined scans

Figure 2.4.: The MAV (a) has been designed for inventory and short-range inspection tasks in indoor environments. Photo (b) and CAD (c) drawing of our 3D laser scanner with the FoV of the individual 2D laser scanners (blue). (b) Scanner 1 (right), and scanner 2 (left) have diverse FoVs. They are mounted on a bearing and rotated around the red axis. (c) Scanner 1 is rotated to the back of the image plane to show the 270° opening angle of the scanner. Scanner 2 is in the front, showing the twisted scan plane. (d-f) Point clouds from the rotating 3D laser scanner. While the individual scanner show substantial blind spots, nearly no occlusions occur in the combined scan.



(a) **(b)** **(c)**

Figure 2.5.: Demonstration of the employed scan filters. A 3D scan assembled from one half rotation of the 3D laser scanner is shown from a top-view. Color encodes height. The MAV (depicted by the axes) passes the obstacle on the left. The red points close to the MAV are spurious measurements caused by the MAV itself and the occluded transition between the obstacle and the MAV. (a) Unfiltered 3D scan. (b) Filtered 3D scan using the self filter only. Spurious measurements remain. (c) Filtered 3D scan using self filter and modified shadowing filter. Spurious measurements are removed.

incorporating inertial measurement unit (IMU) and visual odometry measurements are incorporate. The 6D motion estimate is used to assemble the individual 2D scan lines of each half rotation to a 3D scan as shown in Figure 2.6.

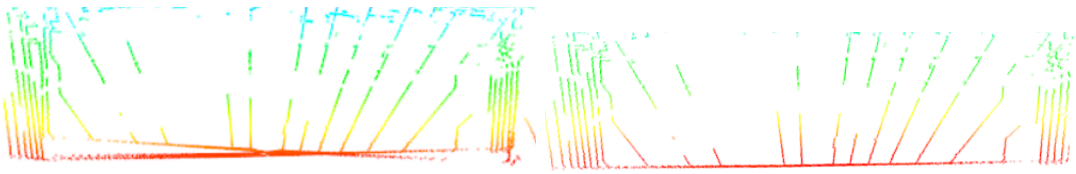
2.1.3. MAV with Static Mounted 3D Laser Scanner

The third MAV used in this thesis is shown in Figure 2.7. It is based on the DJI Matrice 600 platform with a diameter of $d \approx 170\,\text{cm}$ and is equipped with ample onboard computer, a Intel NUC6i7KYK with an Intel® Core™ i7-6770HQ quad-core CPU running at 2.6/3.5 GHz and 32 GB of RAM. As primary environment perception sensor, a Velodyne Puck LITE™ lidar is deployed. It features a low weight of 590 g and yields 300,000 range measurements per second in 16 horizontal scan lines at a vertical angle of 30° in up to 20 Hz. Its maximum range is 100 m. Due to the larger FoV, compared to the Hokuyo 2D sensor, rotation of the sensor is not necessary.

2.2. The Mobile Manipulation Robot Momaro

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 (Rodehutskors et al., 2015), 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 2.8). 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 gain a 3D FoV. Hence, the sensor can measure



(a) **(b)**

Figure 2.6.: Side view on an indoor 3D scan with flat ground. (a) Assembled 3D scan without considering sensor movement during the scan acquisition. (b) We incorporate visual odometry to correct for the sensor movement.

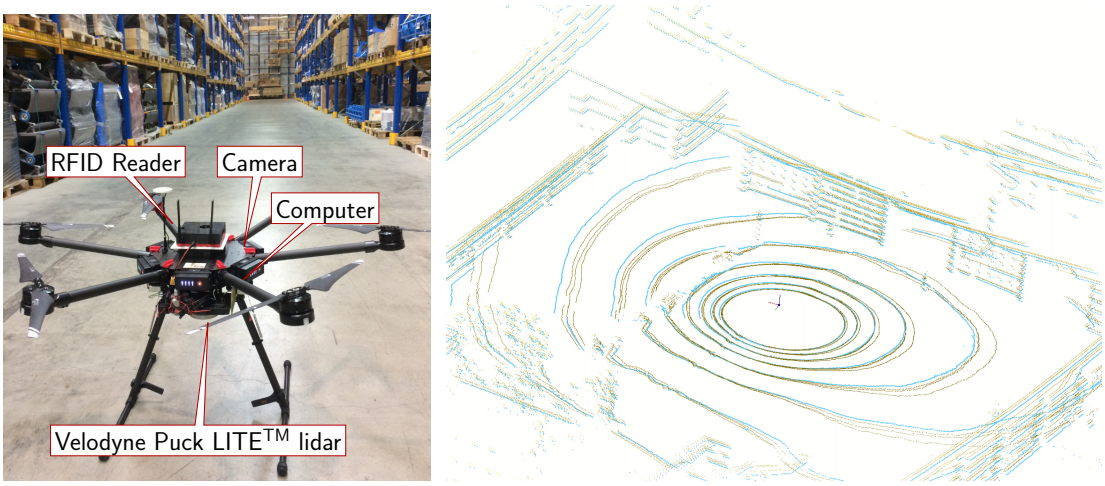


Figure 2.7.: Left: the DJI Matrice 600 MAV platform equipped with a Velodyne Puck LITE™ lidar, fast onboard computer, two synchronized global shutter color cameras and an RFID reader. The landing feet are retractable to allow for true 360° perception. Right: four consecutive scans from the Velodyne Puck LITE™ lidar during flight. Last scan is light blue.

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 3.8a).

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

2.3. Experiments

Due to varying shape and reflectance properties of objects, not all obstacles are perceived in every 3D laser scan (one half rotation of the scanner). Especially farther away from the robot, multiple scans may be necessary in order to adequately detect an obstacle.

The intuitions behind these experiments are the following: if a certain object can only be perceived (at least once) in n 3D laser scans, it is sufficient for reliable collision avoidance if our local mapping approach can reliably aggregate n 3D

Figure 2.8.: Left: the mobile manipulation robot Momaro taking a soil sample during the DLR SpaceBot Camp. Right: Momaro's sensor head, with continuously rotated Hokuyo (top) and Velodyne Puck LITE™ (bottom). The 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

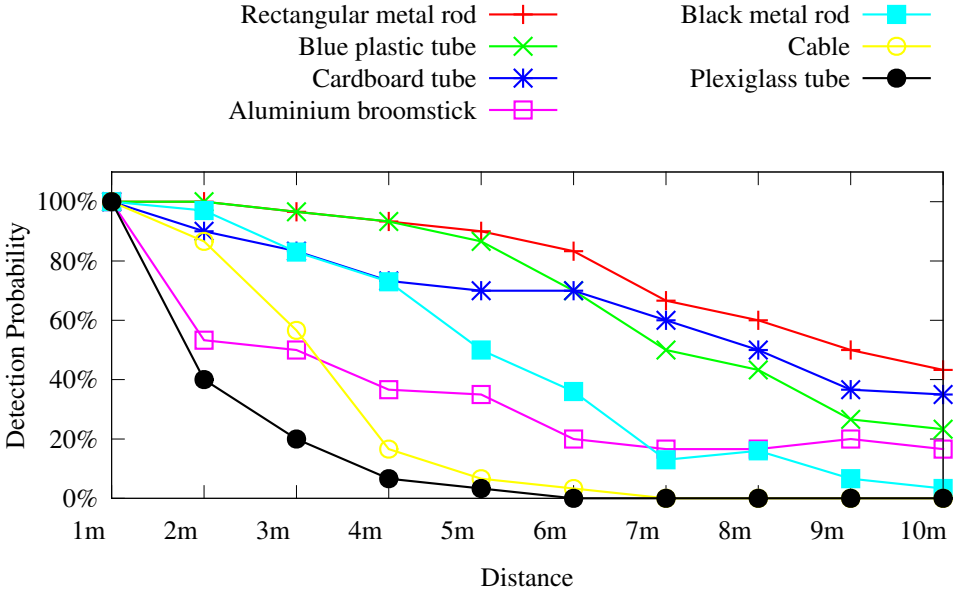
laser scans without inducing inconsistencies in the egocentric obstacle map (e.g. blurring effects due to drifts in the pose estimates). Obviously, whether or not an obstacle can be avoided also depends on the distance to the obstacle and movement direction and speed of the MAV. Both can be neglected if the sensor is able to detect all types of obstacles in the immediate vicinity of the MAV (and the MAV is not flying too fast).

For assessing the probability of detecting objects in the vicinity of the robot, we have chosen seven test obstacles differing, amongst other characteristics, in size (diameter), color and material (reflectivity), and transparency. Referring to the experiment setup in Figure 2.9a, the objects are mounted on a tripod holder. The MAV is positioned with distances to the holder of 1 m to 10 m. For each distance, a total of 30 3D scans are captured. We visually inspect the acquired data and count the 3D scans in which at least a part of the object is visible in the distance measurements and estimate the average detection probability. For the estimation of the detection probabilities, we considered roughly the same lengths for all obstacles (1 m, which is also the minimum safety distance during navigation). That is, the probabilities primarily depend on diameter and reflection properties rather than object length.

As can be seen in the plots (Figure 2.9b) and the detailed results (Figure 2.9c), all types of obstacles are detected at 1 m distance, and with an aggregation period of 10 s up to 3 m (for the transparent plexiglass tube). Obstacles with better visibility like the metal rod, the cardboard and plastic tubes, are reliably detected up to 10 m away from the robot.

Based on the achievable results, we distinguish, respectively, different obstacle

(a) Left: Different object types, from left to right: cable (∅0.75 cm), aluminium broomstick (∅2.7 cm), plexiglass tube (∅5 cm), cardboard tube (∅7.5 cm), rectangular metal rod (∅4 cm), blue plastic tube (∅11 cm). Middle and right: experiment setup with object holder, measuring tape and flying multicopter.



(b) Detection probabilities over increasing distance by object (standing).

Object	**Distance**									
	1m	2m	3m	4m	5m	6m	7m	8m	9m	10m
Rectangular metal rod	100.0	100.0	96.6	93.3	90.0	83.3	66.6	60.0	50.0	43.3
Blue plastic tube	100.0	100.0	96.6	93.3	86.6	70.0	50.0	43.3	26.6	23.3
Cardboard tube	100.0	90.0	83.3	73.3	70.0	70.0	60.0	50.0	36.6	35.0
Aluminium broomstick	100.0	53.3	50.0	36.6	35.0	20.0	16.6	16.6	20.0	16.6
Black metal rod	100.0	97.0	83.0	73.0	50.0	36.0	13.0	16.0	6.6	3.3
Cable	100.0	86.6	56.6	16.6	06.6	03.3	00.0	00.0	00.0	00.0
Plexiglass tube	100.0	40.0	20.0	06.6	03.3	00.0	00.0	00.0	00.0	00.0

(c) Detailed detection probabilities in percent (standing MAV).

Figure 2.9.: Assessing the probabilities of detecting different types of objects in different distances, measured over 30 3D scans for each obstacle and distance. We count the 3D scans in which at least a part of the object is visible in the measurements and estimate the average detection probability.

types (and distances) and detection probabilities: objects that can be reliably detected when aggregating over 2.5 s (5 scans) are considered safe and easy to detect (green in table Figure 2.9c), obstacles that cannot be detected at least once in 10 s (20 scans) are considered especially dangerous and very hard to detect (red in Figure 2.9c), and obstacles of moderate detection probability (yellow in table Figure 2.9c) can be reliably handled by scan aggregation when not flying too fast.

2.4. Summary

This chapter briefly introduced the robot platforms and sensor setups used in the experiments of the following chapters. While most of the experiments has been carried out on the MAV platforms, additional evaluation has been conducted on data sets acquired with the mobile ground robot. Evaluation on different robot platforms allows for assessing the versatility of the approach. Furthermore, this chapter shows experiments assessing the detection probability of different objects measured with the custom-built continuously rotating 3D LIDAR scanner. The experiments indicate the necessity of sensor data aggregation over multiple measurements in a data structure presented in Chapter 3, in combination with an efficient registration approach as described in Chapter 4.

3. Local Multiresolution Maps

In this chapter, we present an efficient 3D multiresolution map that we use to aggregate sensor measurements. 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's characteristics in relative distance accuracy and measurement density. Compared to uniform grids, local multiresolution leads to the use of fewer grid cells—without losing relevant information—and consequently results in lower computational costs.

For fully autonomous operation, robots need to map the environment they operate in. Mapping means building a spatial model that reflects the geometry of the world. Such a model is typically used for navigation—to localize the robot—or for manipulation to avoid collisions with objects. We refer to this spatial model as *map*. While the mapping process itself consists of multiple steps which will be addressed in the following chapters, this chapter focuses on the underlying data structure—the map. Maps are built from sensory data, for example by aggregating measurements from a laser scanner. These sensors provide an enormous amount of data that has to be stored and processed efficiently.

There is a multitude of different map types (Thrun, 2002) and properties to distinguish them. First of all, they can be categorized in metric and topological maps. Metric maps model geometry and and reflect scale, distances and direction. In contrast, topological maps simplify the environment to a graph that relates between subparts. Another important property of a map is the ability to model the occupancy of the environment. For example, in an occupancy grid map (Elfes, 1987), each cell is assigned with a probability which reflects the occupancy believe

of the cell, based on the measurements and a model of the sensor. This occupancy information allows to distinguish between occupied, free, but also unknown parts of the environment. For certain applications—such as navigation or manipulation—this information is vital. Furthermore, maps can be distinguished in egocentric and allocentric maps. In egocentric maps, the robot is in the origin and the map is centered around it. In other words, if the robot is moving, an egocentric map needs to be updated to preserve the structure that reflects the geometry of the world. In contrast, the origin of an allocentric map is independent of the robot position. Thus, the robot moves inside an allocentric map, whereas an egocentric map moves with the robot.

An example of a metric map—which has been prevalent in robotics in the last decades—are occupancy grid maps (Elfes, 1987). Occupancy grid maps discretize the environment in equally sized cells, which aggregate measurements of the spatial area they correspond to. While this discretization of measurements in uniform gird maps is feasible in 2D for robots moving on a planar surface, the aggregation of measurements in a fine-grained 3D grid becomes intractable due to the number of cells. Consequently, using uniform grid maps to store 3D point clouds from laser scanners often results in a trade-off between memory efficiency and accuracy. Especially for beam-based distance sensors—where the measurement density decreases with the distance to the sensor—uniform grid maps do not reflect the measurement properties correctly and result in large number of unused cells, increasing memory consumption and general computation requirements.

In this chapter, we present a data structure for mapping that is used to aggregate sensory data, such as measurements from a 3D laser scanner. The presented data structure—coined *local multiresolution grid map*—is designed for memory efficiency and has two important properties. First, it is a *local* map, according to its egocentric property. Second, the data structure has the multiresolution property. For map data structures, multiresolution means having varying resolution in different areas of the map. Combining the two properties results in an robot-centric multiresolution map, which means a high resolution in the close proximity to the sensor and a lower resolution with increasing distance. Compared to uniform grid maps, local multiresolution leads to the use of fewer grid cells without loosing information and consequently results in lower computational costs. Since the local multiresolution property correlates with a rough approximation of the sensor's characteristics in relative distance accuracy and measurement density, the loss of information compared to a regular grid can be neglected. Each cell in the map summarizes its attributed measurements by Gaussian statistics in a surface element (surfel).

The presented data structure is used to map the environment by aggregating

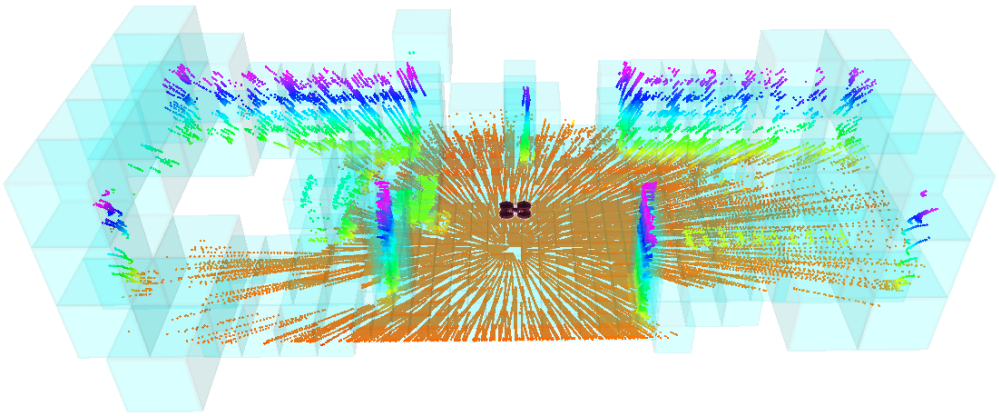


Figure 3.1.: Local multiresolution grid map with a higher resolution in the close proximity to the sensor and a lower resolution with increasing distance. Aggregated measurements from a continuously rotating laser scanner are depicted by colored points. Color encodes height from ground. The data has been acquired during flight with a MAV (center of the map).

measurements from consecutive time steps, e.g., scans from a 3D laser scanner. Aggregating measurements from consecutive time steps necessitates a robust and reliable estimate of the sensor's motion. Therefore, newly acquired scans need to be aligned with the so far aggregated map by means of scan registration. The aggregated map is then used for different tasks of autonomous navigation, such as collision detection, mapping, and localization. Figure 3.1 shows an example of a local multiresolution grid map and measurements acquired with continuously rotating laser scanner mounted on a MAV during flight. Furthermore, the figure illustrates the mentioned property of beam-based distance sensors in regards of measurement density. Close to robot, measurements have a higher density than farther away.

3.1. Related Work

Mapping has been heavily researched in the robotics community in the last decades and a variety of approaches exist.

Storing and processing of 3D data are important problems in many research fields, such as computer graphics or robotics. Thus, a variety of approaches exist for different purposes or applications. In the following, we discuss data structures that are used in robotics. We focus on metric map representations—modeling the geometry of the environment—compared to topological representations, which will be addressed later.

Although directly storing 3D measurements in point clouds has been used in robotics for SLAM, e.g., by Cole and Newman (2006) and Nüchter et al. (2007b), it inferiors in terms of memory efficiency. Furthermore, it does not model unmapped areas—called free space.

The first important map representation in robotics modeling free space are occupancy grid maps, which where pioneered by Elfes (1987) and Moravec (1988). Occupancy grid maps discretize the environment in uniform grids. Each grid cell stores information about its occupancy—being free, occupied or unknown. While occupancy grid maps are still popular in robotics for 2D mapping, the use of fine-grained grids to represent the environment in 3D—especially in larger environments—bears the problem of memory efficiency and scalability. Thus, a trade-off between efficiency and accuracy has to be made when using standard grid maps. To this end, a number of approaches try to tackle the problem of memory efficiency by extending or compressing the grid map data structure, allowing for 3D mapping.

Hebert et al. (1989) propose elevation maps, extending 2D grid maps by adding a height for every grid cell. While elevation maps only model a single surface, multi-level surface maps (Triebel et al., 2006) store multiple heights in each grid cell, allowing to model environments with more than on surface, such as bridges for example. Another approach in extending 2D grid maps by attributing information about the height to grid cells is presented in (Wolcott and Eustice, 2015). The authors propose using Gaussian mixture models to characterize the height distribution of measurements in a grid cell. The main drawback of these methods is—while being more memory efficient than plain grid maps—that they do not represent the real environment and therefore cannot be used for localization. Ryde and Hu (2010) overcome this problem by storing lists of occupied voxels. However, compared to standard occupancy grid maps, these approaches do not model free space. In contrast, multi-volume occupancy grids (MVOGs) proposed by Dryanovski et al. (2010) model free space by additionally storing lists of free voxels. An alternative representation to occupancy grid maps are so-called *coverage maps* Rocha et al., 2005; Stachniss and Burgard, 2003a,b. In contrast to occupancy grid maps—where a cell's occupancy is either occupied or free—coverage maps store information about the amount a cell is covered by an object.

A more complex representation that is often used for 3D mapping are octrees. Octrees model the environment in a tree-based structure where every node in the tree represents a cubic volume. Children of a node recursively subdivide its volume into eight sub-volumes. Compared to standard grid maps, octrees allow to model the environment without storing unused grid cells since the recursive subdivision

of nodes is based on individual measurements. Octrees allow to represent the environment in different resolutions, depending on the depth to which the tree is traversed. Thus, octrees also have the multiresolution property, but compared to our approach, resolution does not depend on the location in the map, but on the distribution of measurements. In contrast, resolution in the presented map data structure depends on the distance to the sensor, called *local* multiresolution, while octrees are better described by *global* multiresolution. While octrees for mapping has been proposed decades ago (Meagher, 1982; Payeur et al., 1997), they gained their real popularity recently with the increase of achievable 3D sensors. For example Hornung et al. (2013) propose *OctoMap*, a complete framework for octree-based multiresolution mapping, which also models free space and provides a compression method.

Another way in reducing the number of cells in occupancy grid maps was proposed by Khan et al. (2015). They adapt the resolution by fusing neighboring cells to rectangular cuboids based on occupancy probabilities. The results reported by the authors, show that this approach leads to the use of fewer grid cells and faster access time for occupied cells, but slower insertion due to the fusion process, compared to OctoMap. Einhorn et al. (2011) generalize the idea of octrees to N^d-trees where, that subdivide a d-dimensional volume recursively into Nd children. Both of these approaches consider mapping in 3D with a voxel being the smallest map element.

Belter et al. (2012) 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.

While occupancy grids and octrees model the occupancy for each cell, Biber and Strasser (2003) introduce the Normal-Distribution Transform (NDT), which assigns a Gaussian probability distribution to each cell, summarizing attributed measurements. Similar to occupancy grids, the environment is subdivided into cells or voxels, but instead of a single probability for each cell reflecting the occupancy, NDT reflects the distribution of measurements in the cell. The normal distributions in the cells are used to register a new scan to the model. The extension to three dimensions (3D Normal-Distribution Transform (3D-NDT)) for point set registration is presented in Magnusson et al. (2007), subdividing the environment in 3D voxels. Similar to 3D-NDT, multi-resolution surfel maps (Stückler and Behnke, 2014) match Gaussian statistics to register RGB-D images or laser scans (Schadler et al., 2013). The approach uses octrees to discretize the environment to allow for multi-resolution queries.

Another alternative representation to occupancy grid maps is the signed distance function (SDF) introduced in Curless and Levoy (1996). While being more popular

in graphics and 3D reconstruction, they have been used for SLAM with RGB-D cameras (Newcombe et al., 2011).

The presented data structure bears similarities to 3D-NDT and multi-resolution surfel maps by summarizing measurements by Gaussian statistics, but compared to the previous approaches efficiently enforces subdivision of the environment based on the distance to the sensor.

3.2. Local Multiresolution Grid Maps

Local multiresolution grid maps are intended to better reflect the properties of beam-based, omnidirectional distance sensors by having a high resolution in the close proximity to the sensor and a lower resolution with increasing distance. They consist of multiple robot-centered uniform grid maps with different resolutions, called *levels*. Each grid map is concentrically embedded in the next level with coarser resolution and doubled cell length. Figure 3.2 illustrates the structure of a local multiresolution map with four levels. Each level has $8 \times 8 \times 8$ cells, each of them corresponding to a spatial area. The total number of cells sums up to 512 cells per level and 2048 cells for the local multiresolution map. In contrast, modeling the environment in an uniform grid map with the same resolution as the finest resolution in the multiresolution map would result in 262144 grid cells.

The total area that can be modeled by a local multiresolution map is defined by the size of the coarsest level. This size, the number of levels, and the resolution are chosen based on characteristics of the sensor. Since the origin of the map, and thus the map itself, moves with the robot, measurements vanish when being farther away from the robot than the size of the map. Consequently, this results in a constant upper-bound for memory consumption, independent of the size and the structure of the environment.

Each level consists of a uniform grid map with gird cells storing 3D point measurements along with the occupancy information of the cell and cell statistics. The map's cells are stored in a circular buffer allowing for efficient translation of the map (Section 3.2.1). Multiple circular buffers are interlaced to obtain a map with three dimensions. The length of the circular buffers depends on the resolution and the size of the map.

The occupancy information represents the cell's probability of being free or occupied and is described in Section 3.2.3. It is updated when new measurements are added to the map and can be used to distinguish free from unknown space. Furthermore, it helps to address spurious measurements in the map, for example caused by dynamic in the environment. The statistics of the cell summarize the

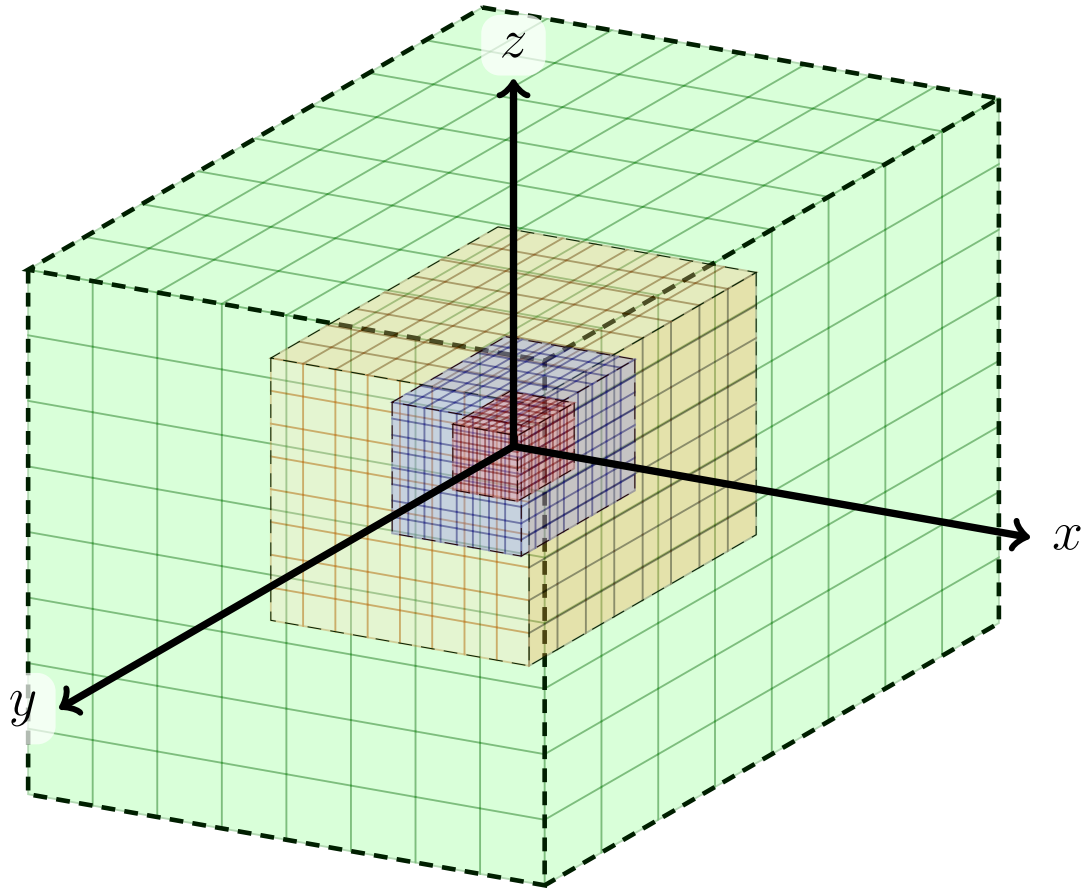


Figure 3.2.: An illustration of a local multiresolution map with four levels. Each level (indicated by color) is a uniform grid map with $8 \times 8 \times 8$ cells, centered around the robot. While the number of cells for each level is constant, resolution doubles.

3D measurements in a surfel as described in Section 3.2.2. Furthermore, a surface normal $\mathbf{n} \in \mathbb{R}^3$—i.e., the (unit-length) vector pointing from the surfel to the sensor origin—is stored for each surfel.

When using a local multiresolution map to aggregate measurements from a 3D laser scanner, the individual distance measurements from the sensor can be referred to as a point in a 3D coordinate system. A multitude of such measurements, for example from one sensor sweep or rotation is called a *3D scan* in the following. Every cell in the map stores multiple measurements from different 3D scans. For storing these measurements, every cell has a measurement buffer that is implemented as a fixed-size circular data structure. Besides memory allocation, circular buffers allow for constant time insertion, which is important for efficiency. If the capacity of the circular buffer is exceeded, old measurements are discarded and replaced by new measurements. The capacity of the buffer corresponds to the

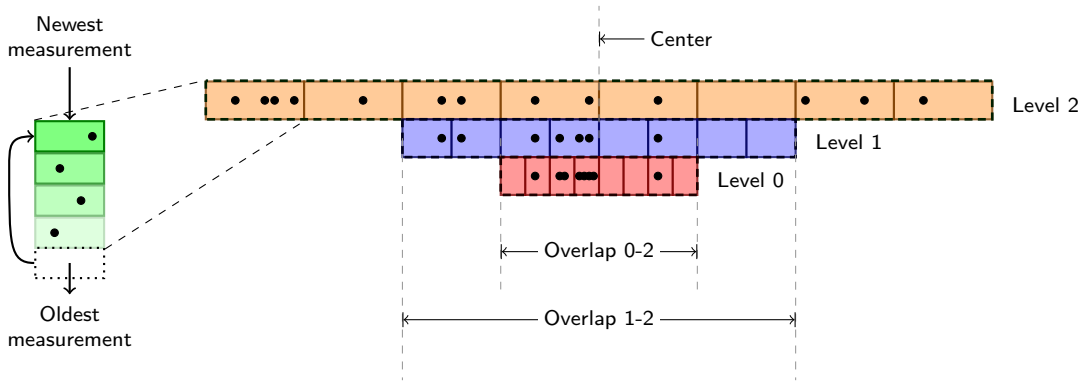


Figure 3.3.: One-dimensional illustration of a local multiresolution map with three levels (orange, blue, red) and 8 cells per level. Along with the occupancy information, every grid cell maintains a circular buffer (green) with its associated measurement points (black). If the capacity of the buffer is exceeded, old measurements are discarded and replaced by new measurements. Local multiresolution results in an overlap between the coarser levels in the center of the map.

maximum number of measurements a cell can store and is equal for all cells in the map. Thus, a constant maximum number of measurements is stored in every grid cell. The capacity of the measurement buffers needs to be chosen according to the measurement density and the resolution, size and number of levels of the map.

Stacking the finer level maps in the center results in an overlapping region of the maps. These overlapping regions store redundant information, since measurements in the coarser level are also included in the finer levels. While this redundancy could be easily neglected, it is stored to initialize grid cells on the finer levels, as described in next section. Figure 3.3 shows a 1D schematic illustration of the map organization and the cell's measurement buffer.

When adding 3D measurements to a cell, the point coordinates are transformed in the cell's local coordinate frame, and back to the map's coordinate frame when accessing. The transformation into the cell's coordinate frame allows for efficient shifting of cells without the necessity to transform all measurements. Furthermore, it facilitates generation of the cell's surface statistics.

3.2.1. Retaining the Local Multiresolution Property

To gain the advantage of local multiresolution maps the egocentric property of the map has to be adhered. Thus, if the robot moves, the map has to be rearranged, according to the movement of the robot, such that the sensor is centered in the map again. Since reconstructing the map on every sensor update is computationally demanding, we aim for an efficient rearrangement of the map. A key requirement

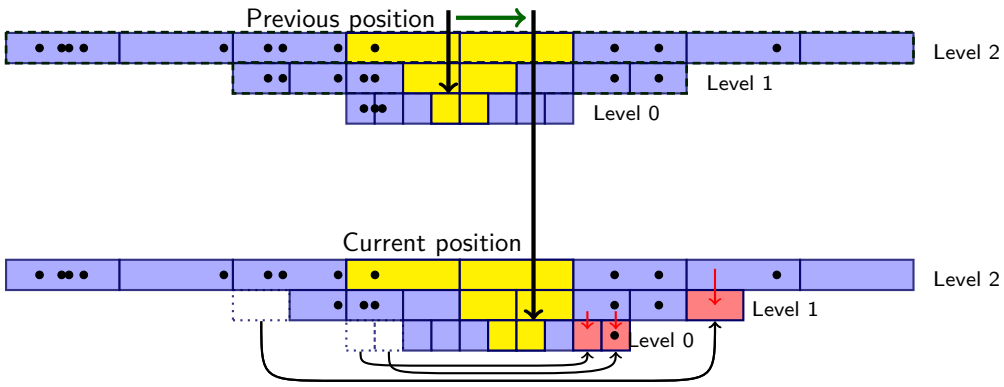


Figure 3.4.: One-dimensional illustration of the hybrid local multiresolution map. Along with the occupancy information, every grid-cell (blue) maintains a circular buffer with its associated measurement points (black). The map is centered around the robot and in case of a robot motion, ring buffers 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).

for an efficient rearrangement is to store the map in a coordinate system which is independent from the robot orientation. Therefore, rotation of the robot can be taken in to account by maintaining the orientation of the robot in the map's coordinate system. Thus, the robot motion can be simplified to the translational parts of the movement.

Rearranging the map according to the translation of the robot in the map coordinate frame can be easily accomplished by shifting cells. To allow for shifting grid cells in constant time, cells are stored in a circular buffer. 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.

Due to the individual sizes of the different map layers, each level is shifted independent of the other levels. When a map layer is shifted, new cells are added to the circular buffers of the respective dimension. Thus, grid cells at the end of the level *fall* out of the map at the end. Furthermore, empty cells are added at the other end of the map. In order gaps int , measurement from the coarser level are retained to initialize empty cells. Figure 3.4 illustrates shifting of cells the process.

Since rotating the map would necessitate to shuffle all cells, local multiresolution maps are oriented independent of the robot's orientation. Therefore the orientation

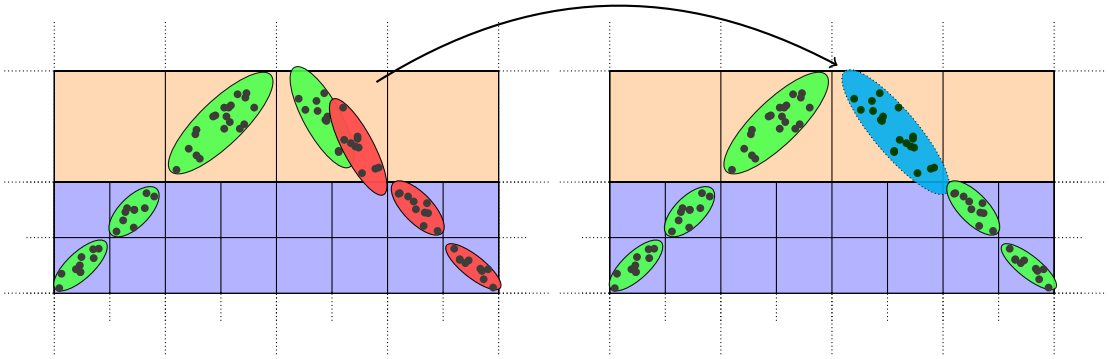


Figure 3.5.: Surfels (colored ellipses) are estimated in each grid cell from the attributed points (black circles). New measurements are added to a grid cell by combining surfels from old measurements (green ellipse) and new measurements (red ellipse) to a single surfel (blue ellipses).

between the map and the robot is maintained and used to rotate measurements when accessing the map.

3.2.2. Local Surface Representation

Due to the large amount of data generated by today's laser range sensors, each grid summarizes the measurements covered by its volume in a surfel. The surfel represents the shape of the measured surface by a local probability density function (PDF). Thus, for each cell, the sample mean $\mu \in \mathbb{R}^3$ and covariance $\Sigma \in \mathbb{R}^3$ is computed from its attributed points $\mathcal{P} = \{p_1, \ldots, p_m\}$. In other words, we now have a piecewise continuous description of the geometry measured in the cell's volume. Subdividing the environment and summarizing points by estimating parameters of a normal distribution has been introduced by Biber and Strasser (2003) and extended to 3D by Magnusson et al. (2007).

When adding new measurements to a cell, surfels are combined by updating mean and covariance as depicted in Figure 3.5. Calculating the mean vector and covariance matrix from a point distribution necessitates careful considerations about the numerical stability and efficiency. Following Stueckler (2014), an efficient one-pass update scheme is used to update mean and covariance when adding points to grid cells. Inspired by Chan et al. (1979), the algorithm calculates sample mean and covariance by storing the sum $S(\mathcal{P}) := \sum_{p \in \mathcal{P}} p$ and sum of squares $SS(\mathcal{P}) := \sum_{p \in \mathcal{P}} pp^T$ of the points in $\mathcal{P}$. From this, mean $\mu(\mathcal{P}) = \frac{1}{|\mathcal{P}|} S(\mathcal{P})$ and covariance $\Sigma(\mathcal{P}) = \frac{1}{|\mathcal{P}|-1} SS(\mathcal{P}) - \mu\mu^T$ are calculated. For two point sets $\mathcal{P}^A$

and $\mathcal{P}^B$, S and SS are updated by

$$\begin{aligned} S(\mathcal{P}^A \cup \mathcal{P}^B) &\leftarrow S(\mathcal{P}^A) + S(\mathcal{P}^B), \\ SS(\mathcal{P}^A \cup \mathcal{P}^B) &\leftarrow SS(\mathcal{P}^A) + SS(\mathcal{P}^B) + \frac{\delta\delta^T}{N_A N_B (N_A + N_B)}, \end{aligned} \tag{3.1}$$

where

$$\begin{aligned} N_{(\cdot)} &\leftarrow |\mathcal{P}^{(\cdot)}|, \\ \delta &\leftarrow N_B S(\mathcal{P}^A) - N_A S(\mathcal{P}^B). \end{aligned} \tag{3.2}$$

By storing S and SS for each grid cell, new measurements can be added without iterating over all aggregated measurements in the cell. Similar to Stueckler (2014), numeric stability is enforced by requiring a minimum sample size of $|\mathcal{P}| \geqslant 10$ and stop adding measurements when $|\mathcal{P}| \geqslant 10,000$. Algorithm 1 describes how points are added to a grid cell and μ and Σ are updated.

3.2.3. Occupancy Representation

Each cell in the local multiresolution map stores information about its occupancy, i.e., the probability of the cell being occupied. This probability helps to asses if a cell is part of a measured object, free space or unknown. Using a sensor model, which reflects the characteristics of the measurements, it is maintained when adding measurements to the map. In the following we refer to free space as the parts of the map that have been traversed by measurements, i.e., cells that are on the ray between the sensor and a measured object. In contrast unknown space are cells that have not been traversed by the other measurements, e.g., when occluded by another object or outside the measurement range of the sensor. Lastly, cells are occupied if measurements from an object are inside the cell.

This information is useful for a number of reasons. First, it helps to address and cancel out spurious measurements that are caused by dynamic objects in the scene. Second, depending on the application—for instance if the map is used for autonomous navigation—knowledge if parts of the map are free or unknown is important for motion planning. Furthermore, it helps to cancel out abandoned measurements which are caused by wrong motion estimates or registration failures.

To estimate the occupancy of a cell based on sensor measurements, a beam-based inverse sensor model is used to reflect the measurement characteristics of the sensor (Thrun et al., 2005). Integrating measurements into the map thereby consists of determining the affected cells for a each measurement and updating

Algorithm 1: Add points to surfel

Data: Points $\mathcal{P} = \{p_1, \ldots, p_{N_P}\}$, aggregated sum S, aggregated sum of squares SS, and number of aggregated points N.
Result: Mean μ and covariance Σ.

sum and squared sum of $\mathcal{P}$
$S(\mathcal{P}), SS(\mathcal{P}) \leftarrow 0$
for $p_k \in \mathcal{P}$ **do**
  $\delta \leftarrow S(\mathcal{P}) - N_P p_k$
  $S(\mathcal{P}) \leftarrow S(\mathcal{P}) + p_k$
  $SS(\mathcal{P}) \leftarrow SS(\mathcal{P}) + \frac{\delta\delta^T}{N_P(N_P+1)}$

update aggregated sum and sum of squares
if $N_P > 0$ **and** $N_P < 10,000$ **then**
  if $N = 0$ **then**
    $S \leftarrow S(\mathcal{P})$
    $SS \leftarrow SS(\mathcal{P})$
    $N \leftarrow N_P$
  else
    $\delta \leftarrow N_P S - N S(\mathcal{P})$
    $S \leftarrow S + S(\mathcal{P})$
    $SS \leftarrow SS + SS(\mathcal{P}) + \frac{\delta\delta^T}{N} N_P(N + N_P)$
    $N \leftarrow N + N_P$

update mean and covariance
$\mu \leftarrow \frac{1}{N} S$
$\Sigma \leftarrow \frac{1}{N-1} SS - \mu\mu^T$

the probability of the cells based on the inverse sensor model. For measurement $\mathbf{z}_i$, the affected cells $\mathcal{C} = \{c_1, \ldots, c_m\}$ are all cells on the line of sight between the sensor origin and the measurement z_i. To determine $\mathcal{C}$, ray casting between sensor origin and measurement is performed with an approximated 3D Bresenham algorithm (Amanatides and Woo, 1987). When the measurement falls inside the map, the last cell c_m on the ray contains the measurement z_i and is called the endpoint cell. Figure 3.6 illustrates the integration of measurements. According to the sensor model the probability of the endpoint cell c_m being occupied is increased, whereas the probability of cells traversed by the ray $\{c_1, \ldots, c_m\}$ are decreased.

Based on the measurements $z_{1:t}$, the probability $p(c|z_{1:t})$ of a cell c being occupied is calculated using a binary Bayes filter and log-odds notation (Moravec, 1988).

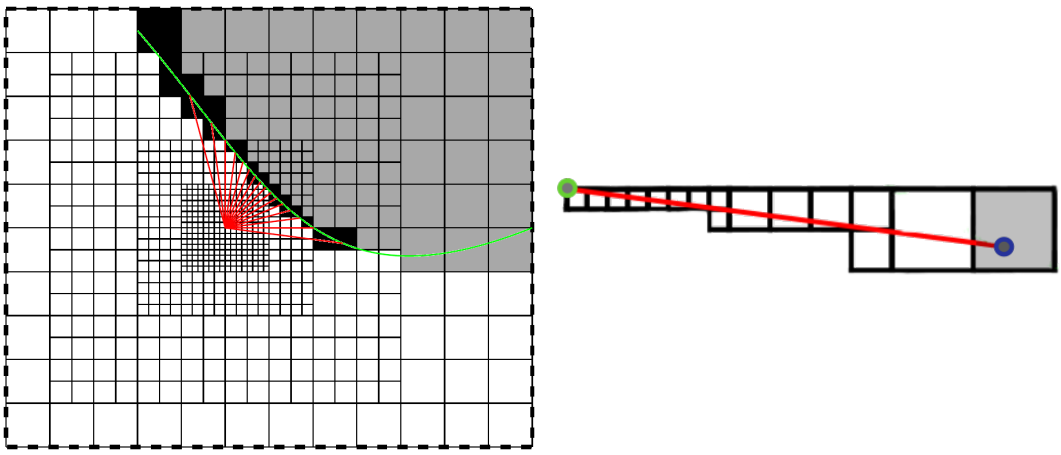


Figure 3.6.: Two-dimensional schematic illustration of the occupancy mapping. Measurements are integrated into the local multiresolution grid map by ray-casting from the origin (center of the map) to the endpoints of the measurements. Cells along the ray (red) are updated. The color of the grid cells depicts the state of the cell, being occupied (black), free (white) or unknown (gray). The measured surface is outlined by the green curve.

The odds of cell c being occupied is defined as $\frac{p(c|z_{1:t})}{1-p(c|z_{1:t})}$ and can be calculated by

$$\frac{p(c|z_{1:t})}{1-p(c|z_{1:t})} = \frac{p(c|z_t)}{1-p(c|z_t)} \frac{1-p(c)}{p(c)} \frac{p(c|z_{1:t-1})}{1-p(c|z_{1:t-1})}, \tag{3.3}$$

from the new measurement z_t, the previous measurements $z_{1:t-1}$ and the prior $p(c)$. With this recursive formulation $p(c|z_{1:t})$ can be updated based on the previous estimate $p(c|z_{1:t-1})$ and a uniform prior $p(c)$. It can be further transformed to a logarithmic representation

$$log\frac{p(c|z_{1:t})}{1-p(c|z_{1:t})} = log\frac{p(c|z_t)}{1-p(c|z_t)} + log\frac{1-p(c)}{p(c)} + log\frac{p(c|z_{1:t-1})}{1-p(c|z_{1:t-1})}. \tag{3.4}$$

The advantage of the logarithmic representation is twofold: it allows to substitute multiplications by computational less expensive additions and is numerical more stable at probabilities close to zero or one. Since $p(c)$ is a uniform prior, Equation (3.4) can be simplified to

$$l_{1:t} = l_t + l_{1:t-1}, \tag{3.5}$$

with $l_{1:t}$ being the log-odds at t. By precomputing l_t the recursive formulation can be solved efficiently by incrementally updating the log-odds value of the cells. For the beam-based inverse sensor model, two specific log-odds values l_{hit}, l_{miss} are

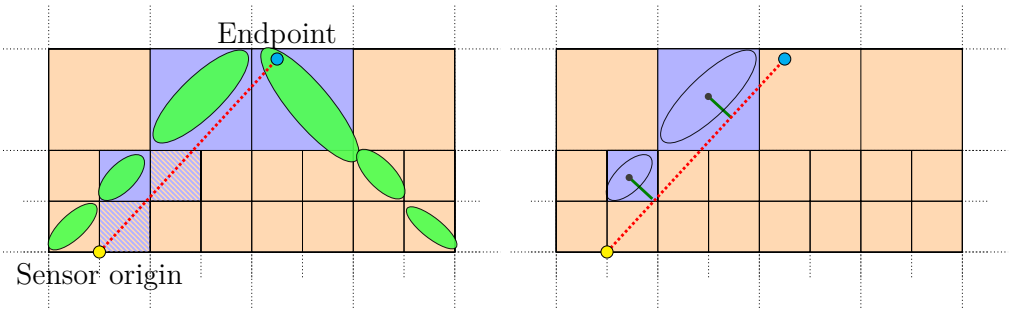


Figure 3.7.: Left: When integrating measurements by ray-casting from the sensor origin (yellow point) to the endpoint (cyan point) the ray (dashed red) traverses free cells (blue striped) and occupied cells (blue solid). Grid discretization may lead to wrongly decreased occupancy probability for cells that are occupied. Right: to avoid wrongly decreasing occupied cells, the Mahalanobis distance between the surfel and the closest point on the ray is considered (green).

precomputed. For cells that are traversed by the beam, not being the endpoint, $l_t = l_{miss}$ and for cells where the beam is reflected in the cell $l_t = l_{hit}$. To asses occupancy, for example for navigation planning, the probability $p(c|z_{1:t})$ can be recovered from the stored log-odds value by

$$p(c|z_{1:t}) = 1 - \frac{1}{1 + exp(l_{1:t})}. \tag{3.6}$$

Thresholding the probability $p(c|z_{1:t})$ leads to discrete states for the cell c.

In contrast to static environments, where the log-oods values of a cell usually converge to a single value, dynamics in the environment necessitate changing of the cell's occupancy during mapping. Thus, a *clamping* update policy is used to limit the stored log-odds value (Yguel et al., 2008). Thus, $l_{1:t}$ is clamped based on the two thresholds l_{max}, l_{min} by

$$l_{1:t} = max(min(l_t + l_{1:t-1}, l_{max}), l_{min}). \tag{3.7}$$

Based on the precomputed log-odds values l_{hit}, l_{miss}, the thresholds control the number of measurements necessary for the transition between a cell's state being occupied to free. The inherent discretization of grid maps results in an undesirable effect when the ray traverses cells, that are partially occupied by a different multiple surfaces. For example when measuring a flat surface in a shallow angle, as described by Hornung et al. (2013). The discretization results in canceling out occupied cells during traversal. The authors suggest to only decrease occupancy of cells that have not been covered by any endpoint of the current scan. In other words, after ray-casting, traversed cells are grouped into endpoint cells and non-

endpoint cells and only the occupancy of non-endpoint cells is decreased. While this method helps to avoid canceling out occupied cells during one scan, it does not prevent canceling out cells when the sensor moves.

For local multiresolution grid maps, the distribution of measurements inside the cells is known and modeled by the surfel. In the aforementioned case, when the ray traverses occupied cells due to discretization, the surfel can be used to decide if the ray measures through the point distribution in the cell. As depicted in Figure 3.7, the vector $\mathbf{v^r}$ being the vector from the surfel mean to the closest point on the ray is calculated. $\mathbf{v^r}$ can be easily calculated by projecting the surfel mean on the sensor ray.

The Mahalanobis distance

$$D_M(\mathbf{v^r}) = \sqrt{(\mathbf{v^r} - \mu_\mathbf{c})^T \Sigma_c^{-1} (\mathbf{v^r} - \mu_\mathbf{c})}, \tag{3.8}$$

measures the distance between the surfel mean taking into account the distribution of measurements in the cell. For non-empty cells that are traversed during ray-casting, $D_M(\mathbf{v^r})$ is calculated to decide if the ray is likely to measure through the surfel.

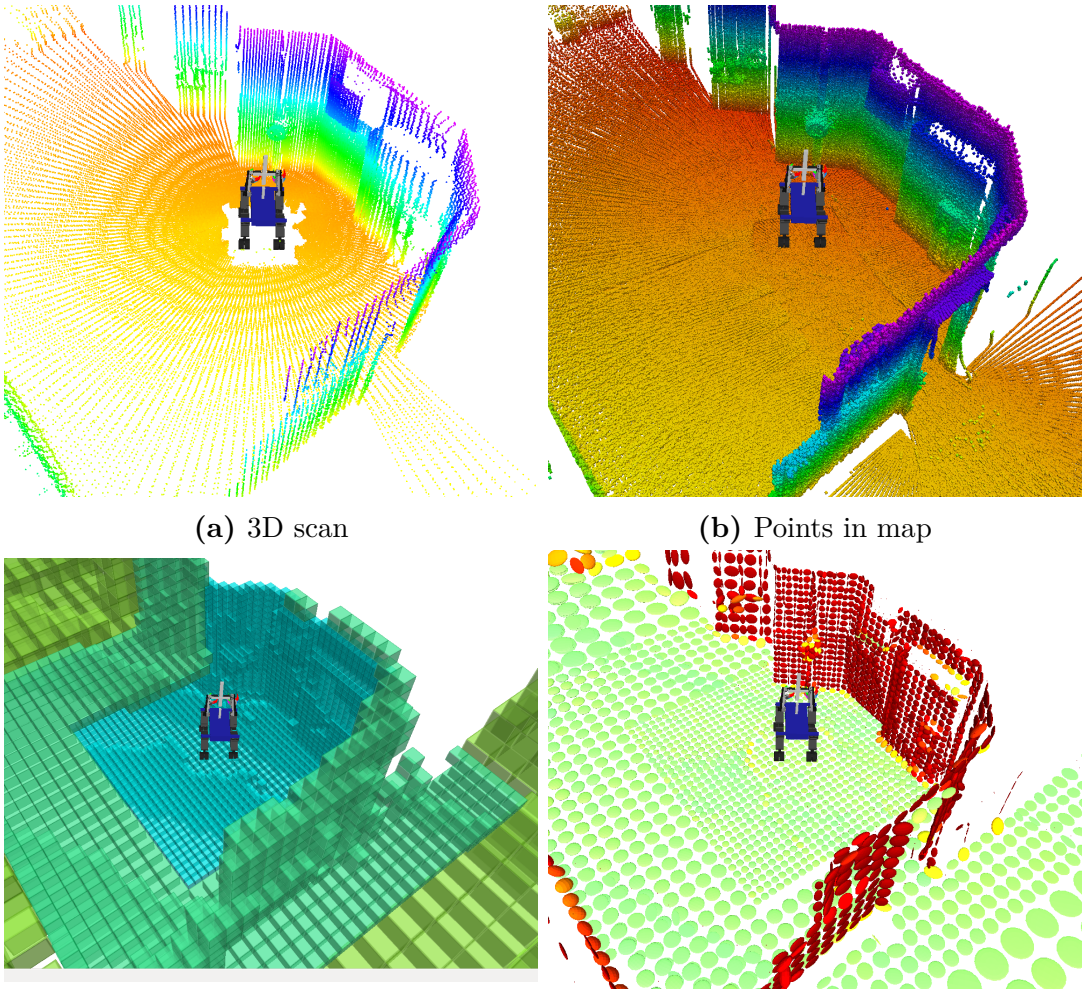
3.3. Experiments and Results

This section describes the evaluation of the presented data structure. After assessing qualitative results from data sets recorded during the the DARPA Robotics Challenge (DRC) [1] memory consumption is evaluated. For example Figure 3.8 shows resulting grid cells, surfels, and aggregated points, recorded by a continuously rotating laser scanner on the mobile manipulation robot Section 2.2. For the same sensor setup, the occupancy mapping described in the previous section is demonstrated in a dynamic environment as shown in Figure 3.11. The impact of the different map parameters are not subject to this evaluation, but will be assessed in the following chapter together with the presented registration approach.

3.3.1. Memory Consumption

In the following experiments, memory usage of the presented data structure is evaluated. Compelling evaluation of memory utilization depends on a variety of factors, such as parameters—in case of local multiresolution maps, size, number of levels, resolution and the number of points per cell—but, even more important, on the characteristics of the data set used for the evaluation. Therefore, a publicly

[1] `http://www.theroboticschallenge.org/`



(a) 3D scan

(b) Points in map

(c) Grid cells in map

(d) Surfels in map

Figure 3.8.: 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.

available data set is used in the following experiment. Following Hornung et al. (2013), an indoor data set[2], recorded on the computer science campus at University of Freiburg, is used to evaluate memory consumption with respect to map parameters. The data set consists of 66 scans from a SICK LMS laser scanner. The laser scanner was mounted on a pan-tilt unit to gain a 3D FoV.

[2]The FR-079 corridor dataset is available at: `http://ais.informatik.uni-freiburg.de/projects/datasets/octomap/`

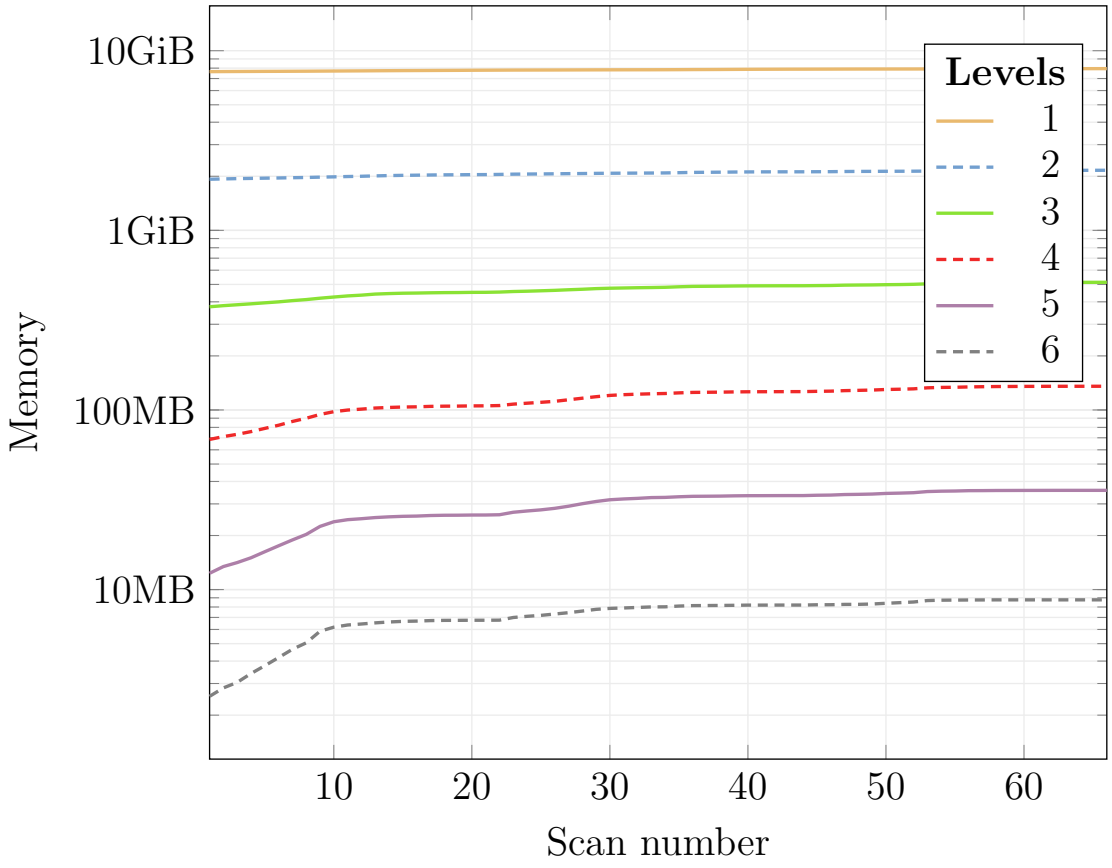


Figure 3.9.: Memory usage (logarithmic scale) for different number of levels in the map. 66 scans from the FR-079 have been added to the map. The map size was chosen 50m, and the number of points per cell is fixed to 50. The cell size of the finest resolution is always 5cm.

In terms of registration accuracy, the most crucial parameters are the number of levels and the resolution, since they directly affect the size of the grid cell and therefore, the size of the surfels. While small surfels allow fine and accurate registration, wider surfels are necessary to recover larger transformations between scans. Thus, the right tradeoff between fine grid cells for accuracy and coarse surfels for robustness has to be found. Furthermore, both parameters affect memory consumption heavily, as shown in Figure 3.9. In this experiment, all 66 scans from the dataset are added to local multiresolution maps with different parameters for the number of levels and resolution. This comparison is only meaningful if the same order of fidelity, in terms of registration accuracy is given between the different parameter combinations. Therefore, the cell size of the finest level in a map is required to equal between the parameter combinations, while the number of levels is alternated. Hence, the resolution of the map is chosen based on the number of

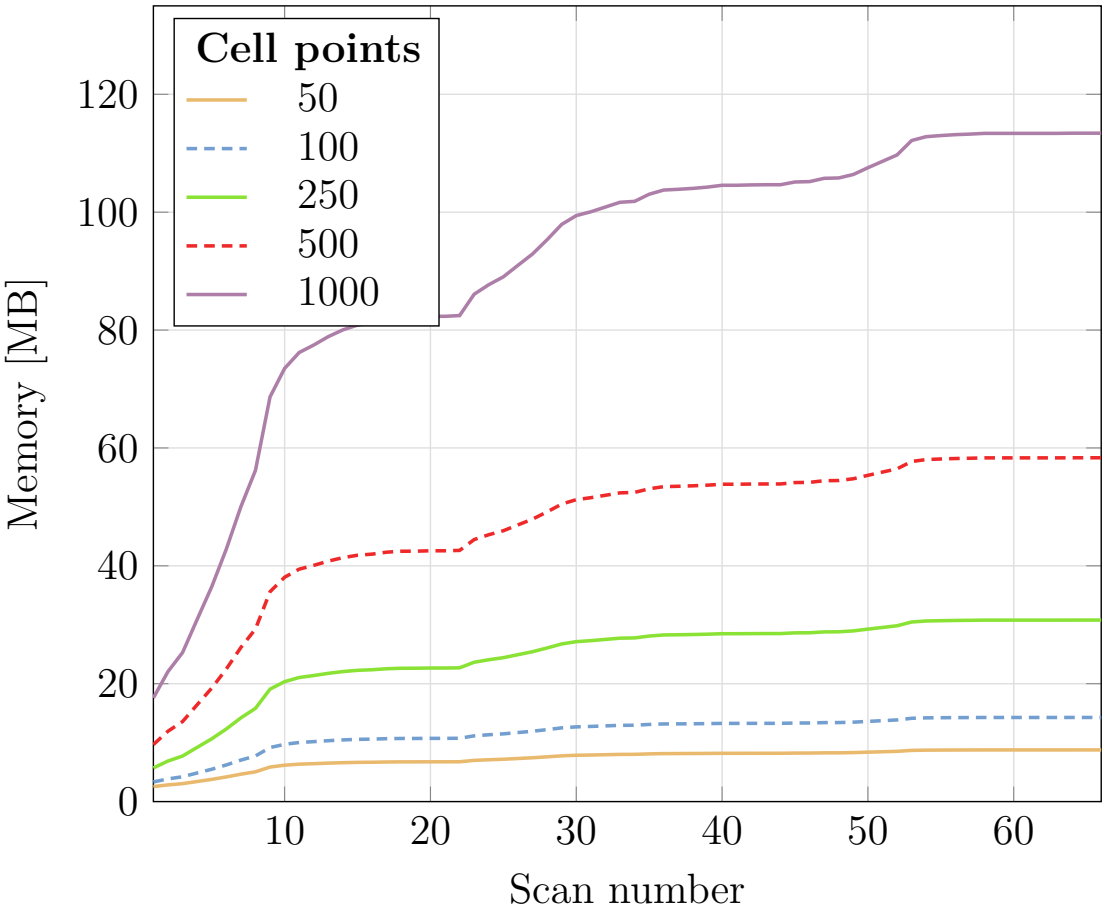


Figure 3.10.: Memory usage for a map with 6 levels and different number of points per cell. 66 scans from the FR-079 have been added to the map. The map size was chosen 50m and the cell size of the finest resolution is 5cm.

map levels.

Based on the experiment in Hornung et al. (2013), a cell size of the finest level of 5cm is used. It has to be mentioned, that the OctoMap data structure as it is presented in Hornung et al. (2013) is not intended as map representation for accurate registration, but for compact and efficient modeling of the geometry and free space in the environment. In contrast to OctoMaps, where the resolution limits the granularity of the geometry which can be represented, the resolution in local multiresolution maps limits the size of the surfels, which itself represent the distribution of measurements Based on the experiment in Hornung et al. (2013), a cell size of the finest level of 5cm is used. in a volume. In other words, since a surfel represents finer granularity as the an Octree cell—being either unknown, free, or occupied—resolution of the in a local multiresolution map can be much coarser, while still capturing similar details of the geometry. Obviously, an Octree

can also store surfels in the leaf notes and capture similar details (Stückler and Behnke, 2014). The severe effect of local multiresolution, in terms of difference in memory consumption in orders of magnitude, is shown in Figure 3.9.

The second crucial parameter affecting memory consumption of local multiresolution maps, is the number of points stored in each cell. Figure 3.10 shows the memory consumption in relation to the number of points that can be stored in a cell. The results confirm the assumption, showing increasing memory utilization in relation to the number of points that can be stored in the cell. However, due to the ability to incrementally update surfels without the need to store the individual measurements it summarizes. However, the current implementation to retain the local multiresolution property when the robot moves requires at least a minimal amount of points to initialize empty grid cells, but for this, a minimum amount of 50 points per cell are sufficient and show same results in terms of registration quality as storing e.g., 1000 points per cell. Furthermore, the current implementation uses the cell points for visualization.

3.4. Summary

This chapter introduced local multiresolution grid maps as a concise environment representation. Besides introducing local multiresolution grid maps, efficient aggregation of measurements in surfels and occupancy mapping with local multiresolution grid maps are described. The data structure is the foundation for the methods presented in the following chapters. The main intuition behind local multiresolution grid maps is the advantage of having a constant upper-bound for memory consumption, independent of the size and the structure of the environment, which is an important property when developing robotics applications.

The evaluation shows that local multiresolution reduces memory consumption and runtime. However, evaluating the properties of the map representation in terms of registration quality is assessed in the following chapter which introduces a registration method exploiting properties of local multiresolution maps.

As presented, local multiresolution grid maps capture the geometry of the scene, disregarding reflectance, color or texture information. While capturing geometry solely achieved sufficient results in this thesis, future work could exploit incorporating other modalities. Here, the surfels of a local multiresolution grid map could be used to accommodate texture, color or reflectance information.

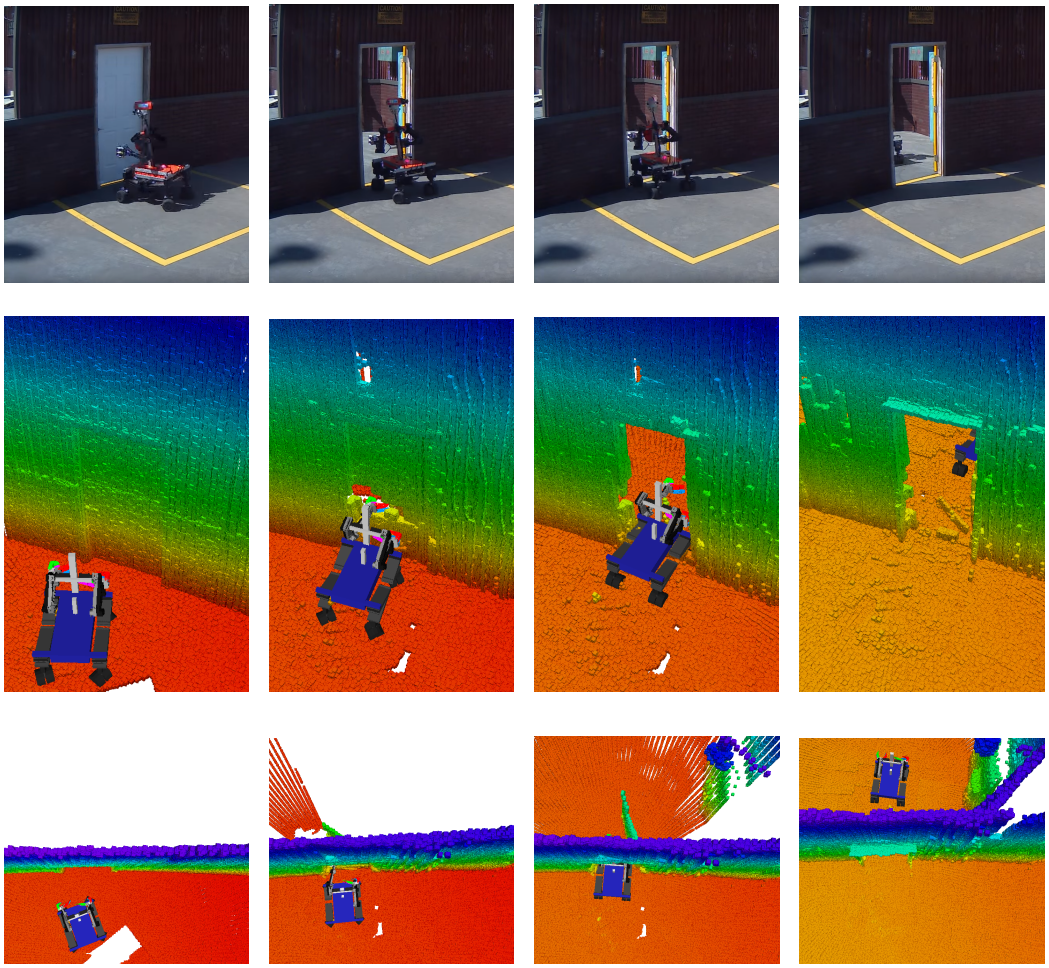


Figure 3.11.: Filtering dynamic objects as the door during the DRC Finals. After opening the door, abandoned measurements are filtered from the local multiresolution map. Camera image and the point-based representation of the map at 4 different time steps (columns). Color encodes height from the ground.

4. Efficient Registration of Local Multiresolution Maps

In this chapter, an efficient and accurate registration method is presented, leveraging the properties of the map data structure presented in the previous chapter. Gaussian point statistics in the grid cells are matched in a probabilistic registration approach.

The presented method allows to align new 3D scans with the map, utilizing the multi-resolution property in a coarse-to-fine approach. Experiments demonstrate superior accuracy and efficiency of the registration approach compared to state-of-the-art methods.

The previous chapter presented a concise data structure designed for efficient aggregation of sensor measurements. In order to aggregate sensor data, measurements need to be aligned to other measurements. Aligning measurements is considered as *registration* and has been researched since decades. Consequently, a variety of different registration methods exist, all having different strengths and weaknesses or being designed for different sensor modalities or applications.

Registration usually considers two sensor measurements—for instance measured at different times—and aims to find a rigid transformation that aligns the two measurements, minimizing the displacement between them. The displacement is commonly modeled as cost function that represents the quality of the alignment. Minimizing the cost function, and thus finding the best alignment, is usually formulated as optimization problem. When registering consecutive sensor measurements, the transformations to align the measurements can be considered to describe the motion of the sensor. Thus, the sensor trajectory can be reconstructed from the alignments. Thus, incremental registration of consecutive measurements from a

laser scanner, is often referred to as *Lidar Odometry*.

In case of local multiresolution grid maps, where surfels summarize sensor measurements, the objective is to find the transformation that aligns the new measurements with the surfels in the aggregated map. The inherent structure of local multiresolution maps thereby implicitly enables a coarse-to-fine approach, since cells farther away from the sensor result in larger surfels for coarse registration—allowing to recover larger displacements between the sensor measurements—and closer cells, resulting in smaller surfels allow a fine and accurate registration. Thus, the coarse-to-fine approach increases robustness while still being accurate.

Summarizing the measurements by surfels thereby has the advantage of having several orders of magnitudes less map elements to be considered for registration. Thus, for registering 3D scans with a map, we also represent the scans in local multiresolution grid maps. By that, registration of new measurements to the map consists of finding a transformation that minimizes the distance between the surfels in the so-called *scene* map—being the map built from the new sensor data—with the aggregated map, the *model* map. The registration of two local multiresolution maps with each other is considered as an optimization problem, minimizing an error function that models this surfel distance. In order to define this error function, the question of data association—i.e., determining correspondence between surfels of the scene and model map—is crucial and heavily influences registration performance. We define a probabilistic surfel association, assigning surfels within a Gaussian mixture model. For every scene surfel a number of model surfels is considered and the transformation, minimizing the displacement between the associations is searched. Sometimes, reconsidering these associations during registration, might become necessary. In this case, new associations are searched based on the last estimated transformation. The result is an iterative method, that finds associations and minimizes the displacement between the associations until either the displacement between the surfels satisfies a criteria, or has not changed during a number of iterations.

In the this chapter, after reviewing related work and the state-of-the art, the surfel-based registration method is described, followed by an experimental evaluation of the proposed method, comparing it to state-of-the-art registration methods.

4.1. Related Work

One of the first algorithms for scan registration and also the origin of most of today's algorithms, is the iterative closest points (ICP) algorithm (Besl and McKay, 1992). The method finds a rigid transformation between the scene and the model,

by iteratively minimizing point distances between the point clouds. Each iteration of the ICP algorithm consist of searching corresponding points and calculation of the transformation between the correspondences, usually by a closed-form solution (Arun et al., 1987). Minimizing these point-to-point distances bears a number of problems. First of all, finding point correspondences is computationally demanding—especially with the huge amount of measurements provided by today's sensors. Thus, research many works in this area focuses on improving the computational efficiency (Rusinkiewicz and Levoy, 2001), e.g., by accelerating the correspondence search (Nüchter et al., 2005, 2007a).

Furthermore, when minimizing point-to-point distances, the assumption is made that each point in the scene scan matches another point in the model scan, which is unrealistic for real-world data. Thus, a number of approaches improve data association by incorporating more suitable cost functions. A well-known example is the Generalised-ICP (GICP) algorithm, proposed by Segal et al. (2009). The algorithm unifies the ICP formulation for various error metrics such as point-to-point, point-to-plane, and plane-to-plane. Extensions to overcome the sparseness an non-uniform point density have been proposed by Holz and Behnke (2015). In general, minimizing the point-to-plane distance outperform point-to-point approaches (Pomerleau et al., 2013a). The disadvantage when minimizing point-to-plane distance, are the computational requirements to estimate the surface normals and mesh reconstruction. For an extensive survey on these and other registration algorithms for 3D point clouds we refer to the recent works of Pomerleau et al. (Pomerleau et al., 2013a; Pomerleau et al., 2013b).

Another group of registration algorithms are so-called feature-based methods. The idea is to extract salient features from the measurements that are used for data association, instead of the point-to-point associations. Features include geometric (Johnson and Hebert, 1999; Rusu et al., 2009; Tombari et al., 2010) or image features from the reflectance information (Böhm and Becker, 2007). Instead of single 3D points, features are matched and the number of elements to consider during data association is reduced. However, feature extraction usually requires expensive preprocessing of the sensor data and registration performance mainly depends on the properties of the extracted features.

Instead of considering the individual scan points, algorithms such as the NDT algorithm by Biber and Strasser (2003) for 2D, or the 3D-NDT (Magnusson et al., 2007; Stoyanov et al., 2012) for 3D, discretize point clouds in a grid structure and align Gaussian statistics within grid cells to perform scan registration. Instead of a uniform grid map for estimating the Gaussian statistics, Ulaş and Temeltaş (2013) propose to use multiple layers of grids, with different cell sizes. Recently, multi-resolution surfel maps have been proposed that match Gaussian statistics in

multi-resolution voxel representations to efficiently and accurately register RGB-D images (Stückler and Behnke, 2014) and 3D laser scans (Schadler et al., 2013).

Recently, probabilistic methods for point set registration became popular and show promising results (Evangelidis et al., 2014; Horaud et al., 2011; Jian and Vemuri, 2011; Mateo et al., 2014; Myronenko and Song, 2010; Tamaki et al., 2010). Instead of assigning binary correspondences, multiple *soft* assignments are are established and a probability is associated to each assignment. Most approaches are based on the EM (Dempster et al., 1977) for exploiting soft assignments and optimizing for the transformation. Probabilistic methods tend to be more accurate, by mitigating the assumption that each point in the scene corresponds to a model point, but are usually slower due to the increased number of associations.

This chapter presents a registration approach that combines multiple ideas that have been proposed in other works. Similar to Magnusson et al. (2007) point clouds are discretized into voxels and Gaussian statistics summarize points in a volume. While Magnusson et al. (2007) discretizes in a uniform grid, the presented approach uses local multiresolution having different voxel sizes depending on the distance to the sensor and thus better reflects the properties of beam-based, omnidirectional distance sensors. Furthermore, instead of using binary associations, soft associations as in (Tamaki et al., 2010) are used. Data asscociation is considered as latent variable as in (Mateo et al., 2014).

4.2. Surfel-Based Registration

We register a 3D scan $\mathcal{P} = \{p_1, \ldots, p_P\}$ with the points $\mathcal{Q} = \{q_1, \ldots, q_Q\}$ in the local grid map of the environment. We formulate the registration of the 3D scan with the local environment map as optimizing the joint data-likelihood

$$p(\mathcal{P} \mid \theta, \mathcal{Q}) = \prod_{k=1}^{P} p(p_k \mid \theta, \mathcal{Q}). \tag{4.1}$$

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

$$p(\mathcal{P} \mid \theta, \mathcal{Q}) \approx \prod_{i=1}^{N} p(x_i \mid \theta, Y)^{P_{x,i}}. \tag{4.2}$$

By this, several orders of magnitudes less map elements are used for registration. We denote the set of surfels in the scene (the 3D scan) by $X = \{x_1, \ldots, x_N\}$ and write $Y = \{y_1, \ldots, y_M\}$ for the set of model surfels in the environment map. E.g.,

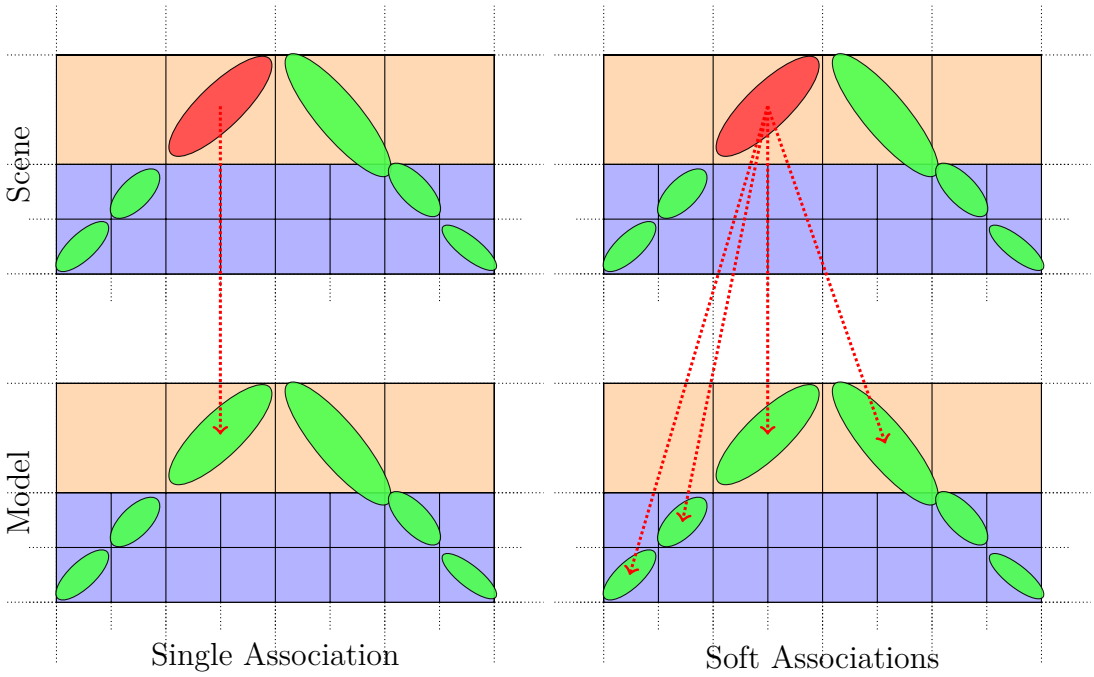


Figure 4.1.: Probabilistic surfel association: Instead of associating each scene surfel to a single model surfel (left), a scene surfel (red ellipses) is assigned to multiple model surfels, so-called soft associations.

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 θ of the scene towards the model. An exemplary surfel map together with its originating points is shown in Figure 4.2.

4.2.1. 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 Myronenko and Song, 2010. A surfel x_i is observed under the mixture defined by the model surfels and an additional uniform component that explains outliers, i.e.,

$$p(x_i \mid \theta, Y) = \sum_{j=1}^{M+1} p(c_{i,j})\; p(x_i \mid c_{i,j}, \theta, Y), \tag{4.3}$$

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

$$p(x_i \mid c_{i,j}, \theta, Y) := \mathcal{N}\left[T(\theta)\mu_{x,i}; \mu_{y,j}, \Sigma_{y,j} + R(\theta)\Sigma_{x,i}R(\theta)^T + \sigma_j^2 I\right], \quad (4.4)$$

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

$$p(x_i \mid c_{i,M+1}, \theta) = \frac{P_{x,i}}{P}\, \mathcal{N}(0; 0, R(\theta)\Sigma_{x,i}R(\theta)^T + \sigma_j^2 I). \quad (4.5)$$

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 $j \in \{1, \ldots, M\}$ 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.

4.2.2. Registration through Expectation-Maximization

The alignment pose θ is estimated through maximization of the logarithm of the joint data-likelihood

$$\ln p(\mathcal{P} \mid \theta, \mathcal{Q}) \approx \sum_{i=1}^{N} P_{x,i} \ln \sum_{j=1}^{M+1} p(c_{i,j})\; p(x_i \mid c_{i,j}, \theta, Y). \quad (4.6)$$

We optimize this objective function through EM (Bishop, 2006; Dempster et al., 1977). The component associations $c = \{c_1, \ldots, c_N\}$ are treated as latent variables to yield the EM objective

$$L(q, \theta) := \sum_{i=1}^{N} P_{x,i} \sum_{j=1}^{M+1} q(c_{i,j}) \ln \frac{p(c_{i,j})\; p(x_i \mid c_{i,j}, \theta, Y)}{q(c_{i,j})}, \quad (4.7)$$

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 θ

$$\widehat{\theta} = \underset{\theta}{\mathrm{argmax}}\;\; L(\overline{q}, \theta) \quad (4.8)$$

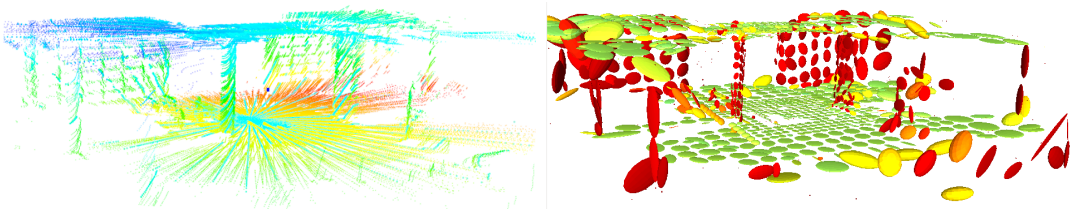


Figure 4.2.: The point-based representation (left) of our local environment map and corresponding surfels (right).

with

$$L(\overline{q}, \theta) := const. + \sum_{i=1}^{N} P_{x,i} \sum_{j=1}^{M+1} \overline{q}(c_{i,j}) \ln p(x_i \mid c_{i,j}, \theta, Y). \tag{4.9}$$

This optimization is efficiently performed using the Levenberg-Marquardt method as in Stückler and Behnke, 2014. 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 $\overline{\theta}$

$$\widehat{q}(c_{i,j}) = \frac{p(c_{i,j})\ p(x_i \mid c_{i,j}, \overline{\theta}, Y)}{\sum_{j'=1}^{M+1} p(c_{i,j'})\ p(x_i \mid c_{i,j'}, \overline{\theta}, Y)}. \tag{4.10}$$

In order to evaluate these soft assignments, we perform a local search in the local multi-resolution 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.

4.3. Experiments and Evaluation

Several works address the benchmarking of registration methods. A comparative evaluation of ICP and NDT in terms of registration accuracy was presented by Magnusson et al. (2009). Wulf et al. (2007) compare ICP-based pairwise and incremental registration (Nüchter et al., 2007b) and a 3D-variant (Borrmann et al., 2008) of Lu-Milios-style graph SLAM (Lu and Milios, 1997). A recent effort for benchmarking SLAM algorithms for RGB-D cameras including datasets and performance metrics for pose accuracy is the RGB-D SLAM Dataset and Benchmark[1] by Sturm et al. (2012). In order to compare constructed maps based on their quality, Schwertfeger et al. proposed a fiducial map metric (Schwertfeger et al., 2010).

[1] `http://vision.in.tum.de/data/datasets/rgbd-dataset`

The data sets used in the evaluation are captured on different robot platforms, with different sensors, and in different environments. The first dataset provides ground-truth pose information from an indoor motion capture (MoCap) system. Thus, the mapping accuracy can be quantified by comparing the estimated trajectory with the one provided by the MoCap system. In particular, metrics such as the absolute trajectory error (ATE) from Sturm et al. (2012) to measure pose accuracy are used. For data sets where ground-truth data is not available, an entropy-based metric measuring the map quality is used. The metric is described in the next section.

4.3.1. Measuring Map Quality

Evaluating registration quality by means of measuring the deviation from ground truth data is the preferred method when evaluating mapping approaches. However, ground truth data is not always available, furthermore, it usually limits the scenarios for evaluation, by being restricted to a certain volume or environment. Furthermore, gathering ground truth data usually involves changing the environment and the robot, e.g., by installing a motion capture system and markers. Thus, a measure to assess the quality of registration results is necessary.

An additional measure, that is independent from external devices and ground truth data is proposed and used during the evaluation. The overall idea is to evaluate registration quality based on the resulting aggregated map quality. As quantitative measure for the quality of a map, the mean map entropy (MME) is proposed. It coincides with the noise in the map and thus allows to measure *sharpness* or *crispness*. In other words, lower MME indicates sharpness or less noise in the map. It has to be mentioned, that the MME measure and absolute value depends on the data set and therefore does not allow to compare maps from generated on different data sets.

The entropy h for a map point q_k is calculated by

$$h(q_k) = \frac{1}{2} \ln \, |2\pi e \, \Sigma(q_k)|, \tag{4.11}$$

where $\Sigma(q_k)$ is the sample covariance of mapped points in a local radius r around q_k. We select $r = 0.3\,\mathrm{m}$ in our evaluation. The mean map entropy $H(\mathcal{Q})$ is averaged over all map points

$$H(\mathcal{Q}) = \frac{1}{Q} \sum_{k=1}^{Q} h(q_k). \tag{4.12}$$

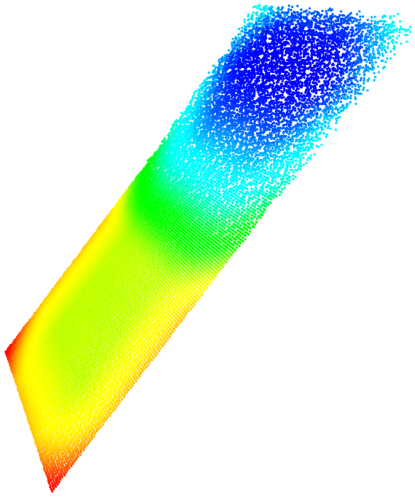


Figure 4.3.: Visualization of the mean map entropy measure. The resulting entropy on an synthetic, wedge-shaped point cloud. The color of a point indicates the value of the metric, from red/yellow (low) to green/blue (high).

Figure 4.3 illustrates the resulting measure based on a synthetic generated point cloud with increasing noise level. It shows that the MME increases with the noise in the data. Additionally, the experiment shows that the MME decreases at the borders of the point cloud, resulting from the lower number of points in the radius around the points at the border.

In a second experiments, the MME is calculated on a data set where ground truth data from a motion capture system is available. To compare the measure to the ground truth data and to assess the reliability, the data set has been registered in a variety of runs with different parameters for the registration method, resulting in different MMEs. Figure 4.4 shows the resulting correlation between the deviation of the ground truth sensor trajectory by means of the ATE and the calculated MME. The figure shows that for runs where the ATE is low, also the MME is low. Thus, it confirms the significance of the proposed measure.

A particular shortcoming of the metrics, however, is that they can only be applied in case of globally consistent trajectory estimates. Large registration errors can cause the map of all aligned laser scans to contain regions where points are scattered. In these regions, both the neighborhood searches for the computation of the metrics and our assumption of roughly aligned environmental structures fail. Consequently, both metrics may yield smaller values than those obtained from correct trajectory estimates. Moreover, in case of similar ATEs, the MME can in the aforementioned cases suggest that a solution actually being worse is better.

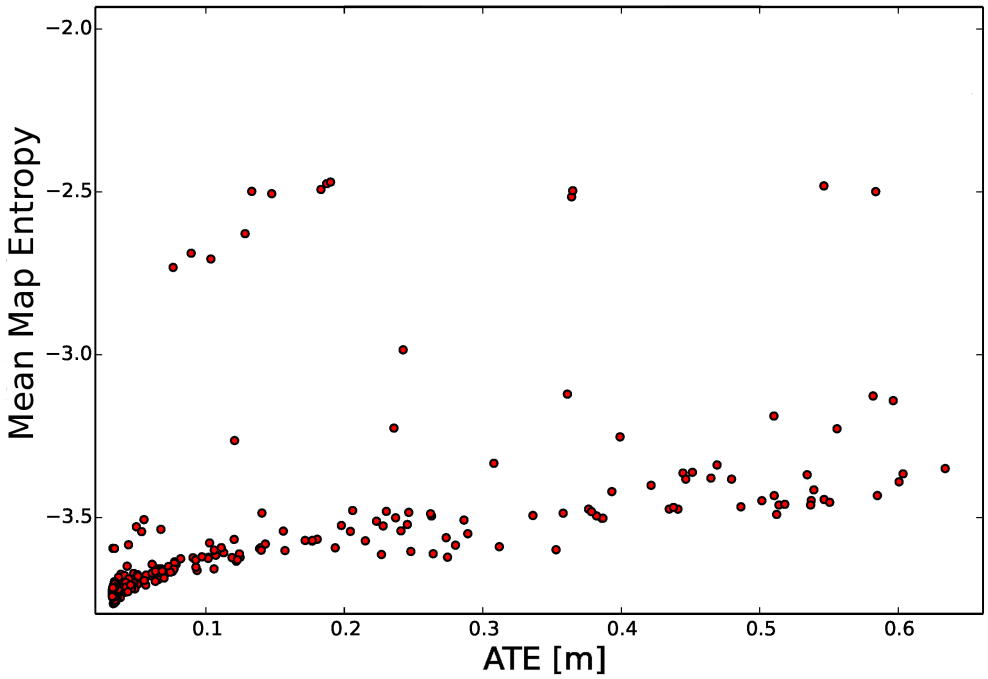


Figure 4.4.: Correlation between the proposed map quality measure MME and the ATE of the poses over multiple runs on the same data set with different parameters.

4.3.2. Motion Capture Dataset

The data set has been recorded in a lab environment equipped with an *OptiTrack* motion capture system. The system consists of 12 cameras (*Flex:V100*). It allows to measure objects that are equipped with special markers, providing accurate pose information at high frame rates (100 Hz). However, the system is restricted by the capture volume of approximately 2×2×3 m. Measurements have been acquired with an MAV equipped with a continuously rotating laser scanner, described in Section 2.1.1. Furthermore, it is equipped with a camera system, providing visual odometry information. During the 46 s flight, visual odometry and laser data for 92 3D scans have been recorded. Figure 4.5 shows the resulting map from the presented method.

To assess the registration accuracy, the results of the registration method are compared to the ground truth poses of the motion capture system. Prior to the experiment, the system has been calibrated and the clock of the system has been synchronized to the MAV computer, ensuring calibrated time stamps between the measurements. In order to compare results, the registration results between consecutive measurements is aggregated resulting in an incremental trajectory. To quantify the deviation between the motion capture system and the estimated trajectory, the ATE is calculated. Furthermore, the acquired data is also processed

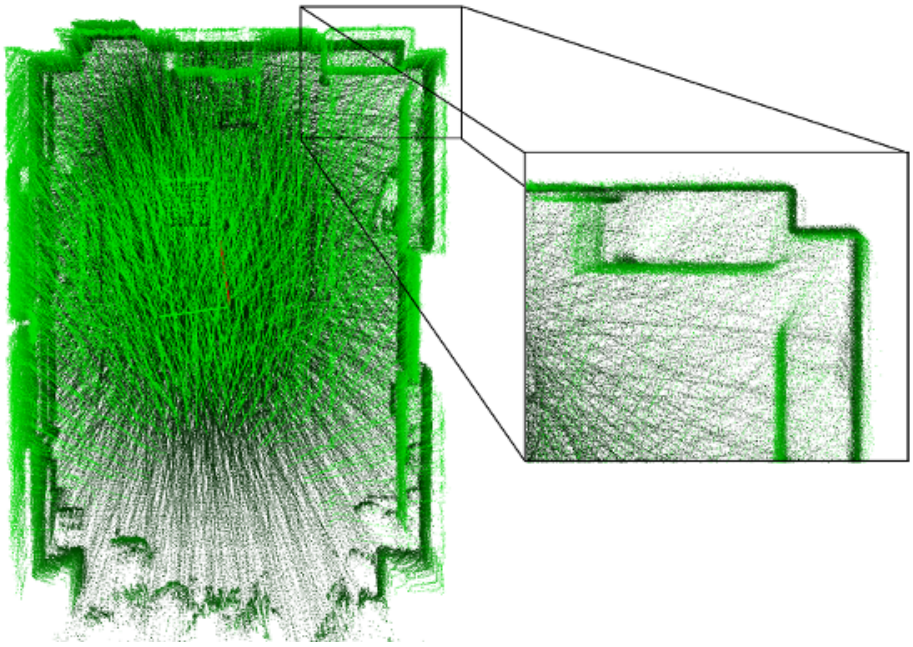


Figure 4.5.: Top-down view of the resulting map of the lab environment. The map has been registered with the presented surfel-based registration method. The thin walls indicate accurate registration.

with two other registration methods, ICP and GICP (Segal et al., 2009).

For this comparison, all three methods aggregate the registered sensor data in a local multiresolution grid map. While the presented method aligns the data utilizing the surfels, ICP and GICP make use of the 3D point measurements stored in the map. Due to the later, map parameters for the experiment have been adapted to store all points, ensuring comparison on the same input data. Throughout the experiments, four resolution levels are used for the map with a cell length of 0.25 m at the finest level, which yields a cell length of 2 m at the coarsest level. Note that the parameters for the local multiresolution grid map are not significant for the ICP and GICP, since the map is configured to aggregate all measurements. On the other hand, the map parameters impact the cell size and by that the size and distribution of surfels used for registration with by the presented method. While the parameters for this experiment were determined empirically, a more method for parameter optimization is shown in the later experiments.

The visual odometry that has been calculated from the camera information on the MAV is used as motion prior for all three registration methods. As base line, the visual odometry is also treated as individual method in the following evaluation. The resulting trajectories of the presented method, GICP, and the ground truth trajectory are shown in Figure 4.6. The figure visually shows that

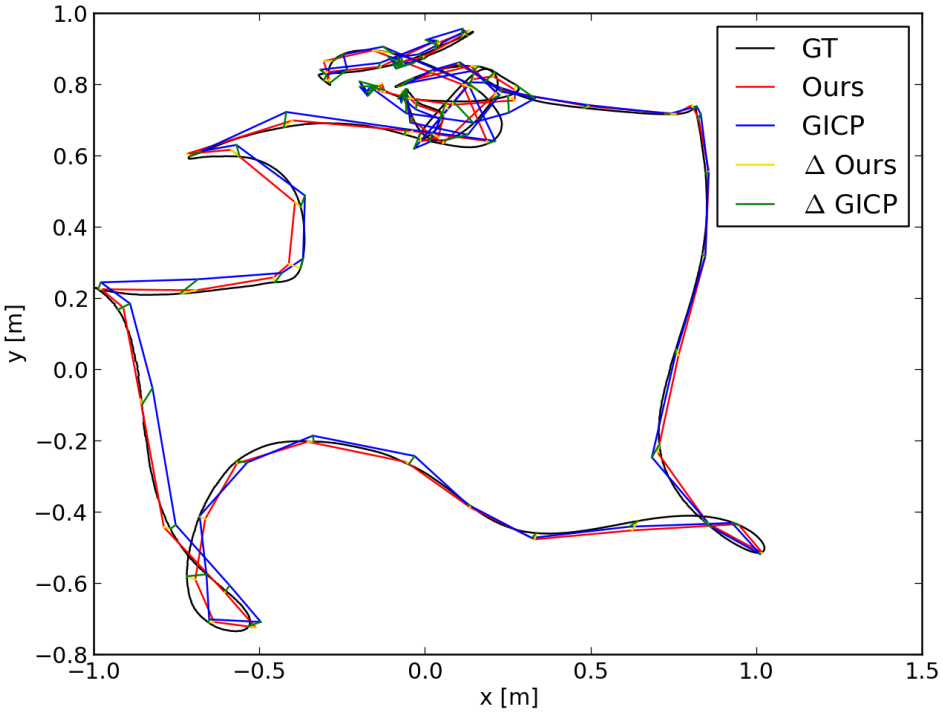


Figure 4.6.: Absolute trajectory error ATE of the scan registration using the multiresolution map. Points of the trajectory are projected on the xy-plane.

the trajectory resulting from our method is closer to the ground truth. Table 4.1 summarizes the ATE for different methods, confirming that the trajectory of the presented method has a lower deviation from the ground truth. Besides the ATE, the MME is of the resulting maps is evaluated. Therefore, a point cloud is assembled from the sensor measurements using the poses from the estimated trajectory. This assembled point cloud is considered as map for evaluating the MME and an example is shown in Figure 4.5. Similar to the ATE, the resulting MME indicates the accuracy of the presented method.

In summary, the results indicate that all scan registration methods improve the motion estimate produced by visual odometry. The surfel-based method results in a lower ATE compared to ICP and GICP. The results in mean map entropy confirm improved accuracy by the presented registration method.

4.3.3. Parameter Optimization

The registration methods that are included in the evaluation have different parameters that affect the behavior of the method. Depending on the scenario or the data set, different parameter values sometimes lead to a significantly different

registration result. In order to have a fair comparison between the different registration methods, we optimize these parameters using Hyperopt (Bergstra et al., 2013) and the Tree-of-Parzen-Estimators (Bergstra et al., 2011). The parameters of the registration methods are described in the following.

As parameters of the ICP algorithm, we use $\mathtt{d}_{\mathtt{max}}$ as general distance threshold and $\mathtt{ICP}_{\mathtt{recp}}$ to assure symmetric correspondences. Correspondences are rejected if the point-to-point distance exceeds this threshold $\mathtt{d}_{\mathtt{max}}$.

The second algorithm in our evaluation is the GICP. We optimize the parameters $\mathtt{d}_{\mathtt{max}}$, ε , and $\mathtt{GICP}_{\mathtt{inner}}$. As for the ICP algorithm, $\mathtt{d}_{\mathtt{max}}$ is a correspondence threshold. The parameter ε models the noise in the data and the $\mathtt{GICP}_{\mathtt{inner}}$ determines the maximum number of inner iterations to optimize the transformation.

The parameters of the 3D-NDT (Magnusson et al., 2007) are $\mathtt{d}_{\mathtt{max}}$, ε , $\mathtt{NDT}_{\mathtt{res}}$, and $\mathtt{NDT}_{\mathtt{step}}$. While the first two parameters are similar to the parameters of GICP, $\mathtt{NDT}_{\mathtt{res}}$ controls the cell size in which the normal distribution is computed and $\mathtt{NDT}_{\mathtt{step}}$ is the step size used for transformation optimization.

For the presented surfel-based soft assignment registration method (Surfel) we also use a parameter $\mathtt{S}_{\mathtt{prior}}$ to model the noise in the data. Besides that, we use three parameters to weight the soft assignments. First, $\mathtt{S}_{\mathtt{size}}$ weights assignments by the surfel size. Second, $\mathtt{S}_{\mathtt{points}}$ weights soft assignments by the number of points they encompass. And lastly, $\mathtt{S}_{\mathtt{neighbor}}$ weights by the number of surfels in the vicinity.

In addition to parameters of the registration methods, we also optimize for the parameters of the underlying multiresolution grid map: the resolution of the map, the number of map levels, and the maximum number of points that are stored in each grid cell (cell capacity).

We use one dataset from the MoCap system to optimize the parameters of the different registration methods (training dataset) and another dataset of the MoCap

	ATE [m]						**MME**
	RMSE	mean	median	std	min	max	mean
VO	0.151	0.134	0.129	0.059	0.024	0.324	-3.112
ICP	0.040	0.035	0.034	0.019	0.006	0.117	-3.411
GICP	0.034	0.031	0.030	0.014	**0.005**	0.088	-3.363
ours	**0.021**	**0.019**	**0.016**	**0.010**	**0.005**	**0.061**	**-3.572**

Visual Odometry (VO), Iterative Closest Point (ICP) and Generalized-ICP (GICP)

Table 4.1.: ATE, map entropy, and run-time of our surfel registration method, in comparison to visual odometry (VO), ICP, and GICP.

	Parameter	Value	Range
ICP	d_{max} :	2.5206	[2.5 - 30]
	ICP_{recp}	False	{True, False}
GICP	d_{max}	12.3845	[2.5 - 30]
	ε	0.4169	[0 - 1]
	$GICP_{inner}$	13	[5 - 30]
NDT	d_{max}	9.7048	[2.5 - 30]
	ε	0.8568	[0.1 - 0.9]
	NDT_{res}	0.1431	[0.05 - 1]
	NDT_{step}	0.6596	[0.05 - 0.95]
Surfel	S_{prior}	0.25	[0.05 - 0.95]
	S_{size}	0.45	[0.05 - 1]
	S_{points}	0.9	[0.1 - 1]
	$S_{neighbor}$	10	[1 - 10]

Table 4.2.: Parameter values of the evaluated registration methods after optimization.

system to evaluate the different methods (test dataset).

Before comparing the different registration methods, we search, for each method individually, for the best parameters that minimize the ATE in the training dataset. Parameter search is performed using Hyperopt as previously described. We report the found parameter sets in Table 4.2.

To illustrate the applied parameter estimation, we first optimize a single parameter for a single method (the S_{size} factor of the surfel registration). We show the results of this optimization in Figure 4.7. The minimum ATE can be achieved for a S_{size} factor of roughly 0.375. In the actual parameter optimization, it is not only a single parameter being optimized but the set of all parameters the registration method has. In comparison, in this particular example, the best found parameter set includes $S_{size} = 0.45$. This clearly shows how the different parameters influence each other and the achievable result.

After optimizing the parameters on the training dataset, we evaluate the different registration methods on two test datasets. The first test dataset was also acquired in our MoCap volume. We report the results obtained using the different metrics in Table 4.3.

In terms of pose accuracy, both ICP and GICP could not improve the initial pose estimates from visual odometry. We note that in the smaller MoCap volume, the visual odometry produces already very accurate relative pose estimates and a

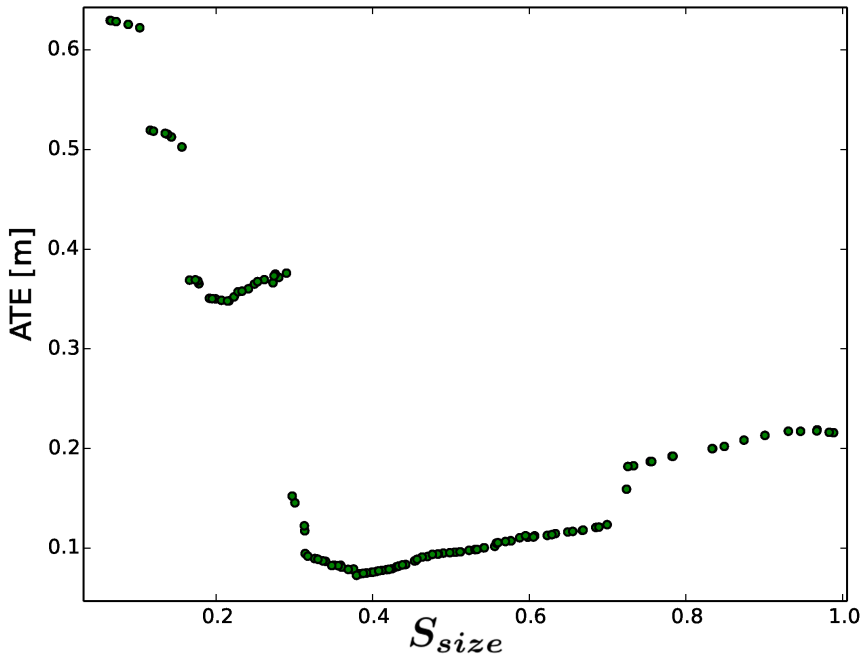


Figure 4.7.: Optimized $\mathtt{S}_{\mathtt{size}}$ parameter of the surfel registration method. The plot shows the characteristics of the parameter in relation to the ATE (other parameters fixed).

	Method	**ATE-RMSE** [m]	**GT-RMSE** [m]	**MME** []
Motion capture	VO	0.029977	0.0058409	-2.52013
	ICP	0.033751	0.0036989	-3.65248
	GICP	0.039058	0.0031005	-3.41114
	NDT	0.025246	0.0021494	-3.74142
	Mesh	0.024969	**0.0019273**	-3.80589
	Surfel	**0.024224**	0.0020391	**-3.80874**
Frankenforst	VO	—	0.2722540	-2.33082
	ICP	—	0.0660561	-2.65460
	GICP	—	0.0640025	-2.55505
	NDT	—	0.0697013	-2.63396
	Mesh	—	**0.0421434**	-2.72391
	Surfel	—	0.0937490	**-2.81387**

Table 4.3.: Results for the two datasets (pose accuracy, and map quality).

globally consistent trajectory. Hence, it is not surprising that plain registration methods could not considerably improve the ATE. However, by accurately aligning the acquired laser scans, they achieve better values in the MME and scores, compared to visual odometry. This is caused by the fact that the acquired laser scans are locally better aligned than in case of the visual odometry while the global trajectory is slightly more inaccurate, e.g., due to small drifts.

The surfel registration could achieve very good values in almost all metrics. It produces a considerably more accurate trajectory estimate, compared to visual odometry solely. Furthermore, the laser scans are very well aligned. The surfel registration is also the fastest of the compared algorithms. Note, however, that especially the parameter optimization focused only on the accuracies of trajectory and map rather than runtime. The map obtained from this best trajectory estimate is shown in Figure 4.5. Only in the GT-RMSE map quality metric, the mesh-based registration achieves a better score than the surfel registration. It follows a very similar mechanism of aligning Gaussian statistics to perform robust scan registration, but on approximate surface reconstructions of the 3D scans.

The mesh-based registration is considerably slower than the surfel registration, but achieves both a comparable ATE and comparable map quality metrics. Overall, both approaches achieve very similar results although they follow two completely different approaches. An interesting fact is that the local window alignment applied in the mesh-based registration achieves a considerably better trajectory estimate (without any drifts) compared to the pairwise registration as reported in Holz and Behnke, 2014. In fact, the obtained scores in both absolute trajectory estimate and MME do not rank behind the globally optimized trajectory in Holz and Behnke, 2014. A likely cause is the aforementioned simplicity of the environment allowing for robustly optimizing the trajectory although it is only optimized in local windows.

The second test dataset—the Frankenforst outdoor dataset—has been recorded during a flight along the facade of a building of *Gut Frankenforst*—a research station operated by the Institute for Veterinary Research at the University of Bonn (see Figure 5.10). In the acquired laser scans the building, surrounding vegetation with trees and the ground are visible. This dataset is far more challenging since the surroundings of the building are cluttered and the scene contains fewer visible distinct environmental structures compared to the small indoor motion capture volume. Due to missing ground truth pose estimates in this dataset, we cannot compute the ATE and only report the other performance measures in Table 4.3.

For this dataset, all registration methods improved the initial visual odometry estimates. Here, the trajectory estimate by the visual odometry shows a significant drift. The map obtained from the mesh-based registration achieves the best GT-

Figure 4.8.: Resulting map of the Frankenforst outdoor dataset obtained by the surfel registration (lowest MME).

RMSE. Furthermore, by using a constant number of edges between scans being aligned and a constant size of the local window, it achieves nearly constant-time updates and the best runtime for this dataset. In terms of MME the surfel-based registration shows the best results. The map obtained by the surfel-based registration is shown in Figure 5.10.

4.3.4. Runtime Evaluation

Table 4.4 reports the measured runtimes, i.e., the average processing time of one 3D laser scan for the different approaches and the optimized parameter sets. The runtime was not used in the parameter optimization. Instead, we focus on the accuracy of the estimated trajectory and optimize the parameters w.r.t. the ATE. Consequently, some approaches show larger processing times than others simply because of using parameters that achieve (slightly) better ATE scores at the cost of considerably longer runtimes. However, Table 4.4 shows that the presented method is not only accurate but also efficient.

4.3.5. Robustness Evaluation

Besides registration accuracy, the robustness of the registration method is assessed. In order to evaluate the robustness, two consecutive 3D scans are aligned with each other, similar to Magnusson et al. (2009). The result of an initial registration, with manual verification of the alignment, is considered as reference for the experiment. After systematic translation and rotation of the of the scene scan, the registration result is compared to the reference alignment and registration errors are assessed

Method	MoCap1	MoCap2	Frankenforst
ICP	0.2903 ± 0.1087	1.6659 ± 0.3866	1.0264 ± 0.3828
GICP	1.7695 ± 0.8139	1.3534 ± 1.1927	0.3196 ± 0.2355
NDT	—	5.2497 ± 1.7360	2.0627 ± 0.7125
Mesh	—	0.1097 ± 0.0382	**0.1314 ± 0.0415**
Surfel	**0.0511 ± 0.0273**	**0.0466 ± 0.0140**	0.3508 ± 0.1522

Table 4.4.: Resulting run times for the datasets. Runtimes are measured per 3D scan being registered and given with mean and standard deviation.

by rotational and translational differences. The scans were acquired with the MAV and lidar sensor described in Section 2.1.3. Figure 4.9 shows the scans and a sample instance with a rotation around the z-axis of 80°. Similar to Magnusson et al. (2009) we translate the scene scan along the x- and y-axis, and rotate around z-axis. Choosing only two axes for translation and one for rotation is motivated by the assumption that most displacements between consecutive scans are in the horizontal plane. For instance, the MAV used in the experiment is equipped with an IMU measuring rotations around the x- and y-axis, but fails to reliably measure rotations around z-axis. Furthermore, limiting the the systematic displacements to three parameters in total, allows concise visualization of the result. For the following experiment, the registration of the scene scan to the model scan is considered *Successful*, if the translational error ϵ_t and rotational error ϵ_r is not larger than 0.1 m and less than 5° respectively. Furthermore, a registration result is considered as *Weak Success* if the translational error is less than 1.0 m with the same rotational threshold as before. Registration results exceeding these thresholds are considered as *Failure*.

Since the aim of the experiment is to assess the robustness and limitations of the registration method, the range in which scans are translated and rotated is chosen based on the ratio between the number of successful and failed registrations. For the scans shown in Figure 4.9 translations between -4 m and 4 m with a step size of 1 m and rotations between -80° and 80° with a step size of 20° are evaluated. Results are shown in Figure 4.10. From the 729 different combinations of rotation and translations, 553 are were considered as successful (75%) and the others are considered as failure.

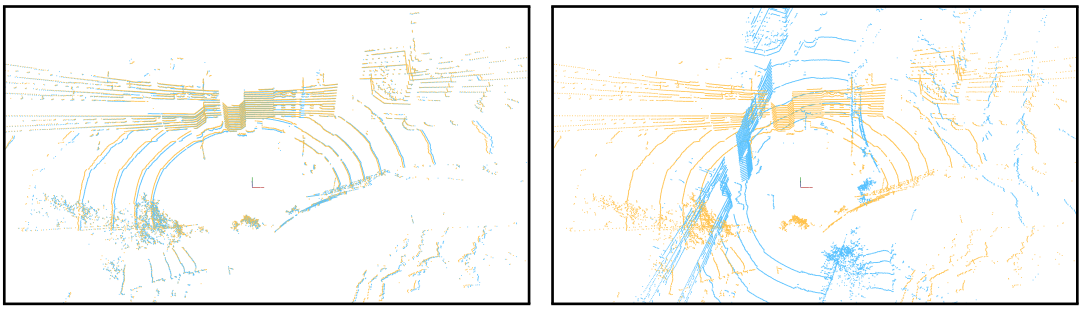


Figure 4.9.: Two consecutive scans (orange and blue) for evaluating registration robustness. Left: after initial registration of the scene scan (blue) to the model scan (orange) and manual verification of the alignment, the scene scan is systematically translated. Right: The scene scan (blue) rotated around the z-axis by 80°.

4.4. Summary

The chapter presented an efficient method for registering 3D scans using local multiresolution grid maps. The method does not match individual scan points, but utilizes the condensed representation of the measurements as surfels. These surface elements are aligned efficiently and at high accuracy in a registration framework which overcomes the discretization in a grid through probabilistic assignments.

In experiments, results of the proposed method are compared to ground-truth data from a motion capture system and GICP, a state-of-the-art registration algorithm, as well as standard point-based ICP. Overall, the local multiresolution surfel approach is more accurate and results in sharper maps as indicated by the lower ATE and map entropy. Besides that, it is computational more efficient, allowing to register scans and to build local 3D maps in-flight in real-time.

The method is evaluated on data of a light-weight 3D laser scanner mounted on a micro aerial vehicle. The registration of this data is particularly challenging due to the sparsity of the data. Besides a comparison of the surfel-based registration method to state-of-the-art registration methods, a metric for assessing the quality of the resulting maps is proposed. In addition, we determined optimal sets of parameters for all algorithms—except for the mesh-based registration—in a hyper parameter optimization.

Besides an accurate alignment, robustness is an important property of a registration algorithm. Robustness is evaluated by systematically translating and rotating the scene scan to assess limitations of the proposed registration method. The experiment shows, besides extreme translations and rotations, the method is able to recover the alignment. In most cases where the scan was rotated up to 80°the method was not able to find the correct alignment. However, these cases are not really realistic when incrementally matching consecutive scans with a sensor

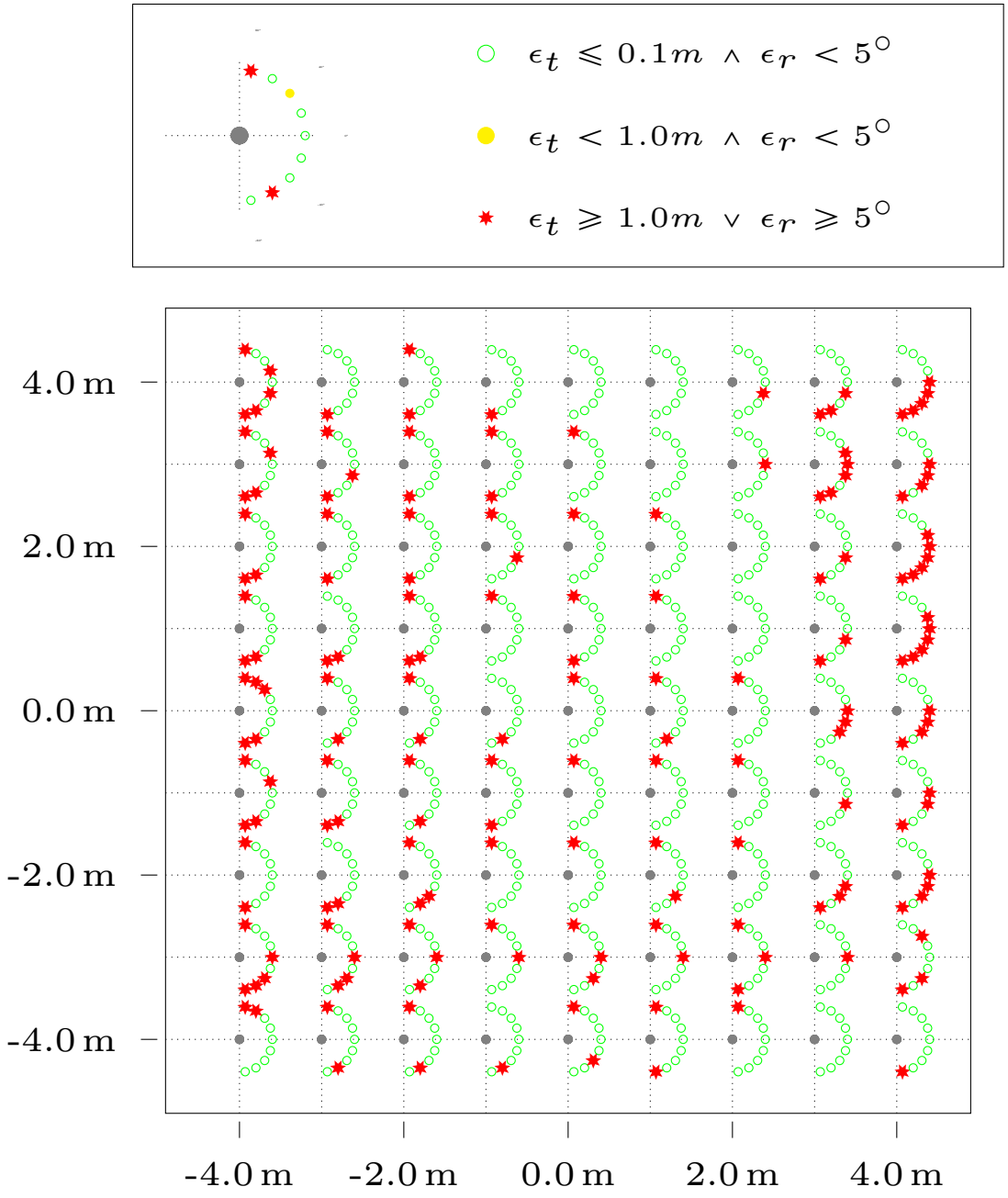


Figure 4.10.: Results of the robustness evaluation. The scene scan is systematically translated along the x- and y-axis between -4 m and 4 m (same axis in the plot) and rotated around the z-axis between -80° and 80°. The rotational displacements are shown by a half circle with each rotation angle on the corresponding angle in the circle (0° is horizontal) with a step size of 20°. The registration result is depicted by either a green circle (Success), a yellow filled circle (Weak Success), or a red star (Failure).

setup as described in Section 2.1.3.

5. Allocentric Mapping with Local Multiresolution Maps

The previous chapters presented an efficient egocentric map data structure and the registration method to aggregate sensor measurements. The presented local multiresolution grid maps are egocentric, allowing to track the robot only in a local frame and modeling the geometry of the robot's direct vicinity. While this property is beneficial in terms of memory and computational resources, it does not allow to map larger environments.

This chapter presents an approach to build allocentric maps from local multiresolution grid maps, by modeling local maps in a graph-based structure. By taking advantage of the incrementally built local maps, it allows to combine efficient local mapping with globally consistent allocentric mapping of larger environments.

As shown in the previous chapters, local multiresolution allows for efficient local mapping and registration in terms of memory and computational requirements. Even more important than efficiency is the ability to roll the map according to the motion of the robot. By that, the number of grid cells stays constant during operation, independent of the environment. This enables to define constant upper bounds for memory consumption and computation. The main reasoning behind this is, that the local multiresolution map captures enough of the vicinity of the robot, to allow robust motion estimation by means of scan registration and reliable collision avoidance. In limited environments, e.g., as the motion cap-

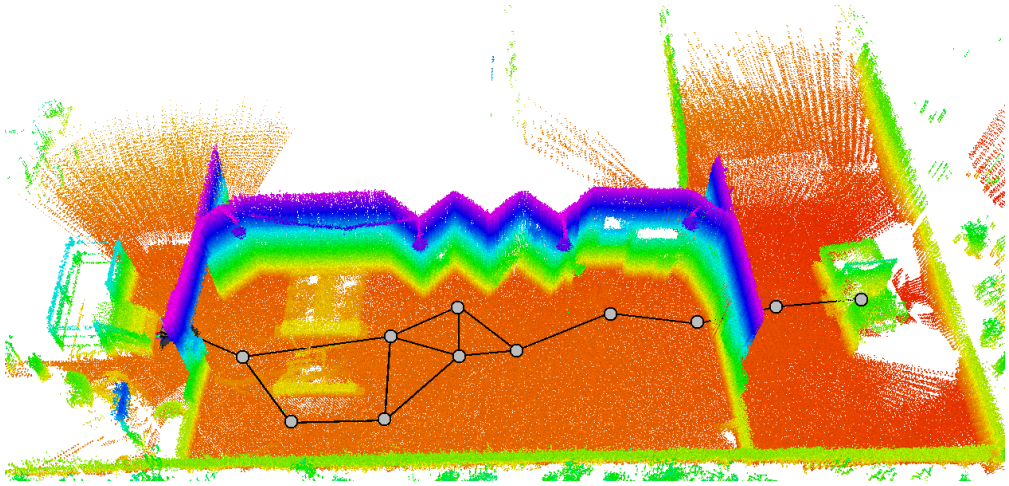


Figure 5.1.: An allocentric map of the mock-up disaster scenario during the DRC. Local multiresolution maps from different view poses (grey discs) are aligned to each other resulting in a graph structure.

ture volume, where the local map captures the entire environment, this suffice for navigation. Furthermore, in open environments where an accurate external pose tracking system, such as a motion capture system or global positioning system (GPS) is available, local perception might be adequate.

However, in most realistic scenarios, the size of the environment significantly exceeds the dimension of the volume shown in the previous chapter, and limiting the workspace to one that can be covered by a MoCap system is also not an option. While MoCap systems limit the workspace of the robot to the capture volume, global navigation satellite systems (GNSS) such as GPS, provide pose information as geolocation anywhere on earth, where there is a free line of sight to at least four satellites. However, the requirement of having a certain number of unobstructed connections to satellites is not met in most urban environments. Furthermore, atmospheric effects and multipath reflections of the signal result in degraded accuracy of the pose. In summary, assuming a reliable localization from an external pose tracking system is negligent.

Here, the idea of SLAM comes into play, by providing a localization pose with respect to a simultaneously built map of the environment. If a global reference, such as the geolocation from the GNSS is not vital, the localization pose with respect to an allocentric map can be used instead. In contrast, if global reference is necessary, SLAM might be used to overcome degraded pose accuracy of the GNSS system.

The key idea is, to build a global consistent representation of an unknown environment while simultaneously localizing relative to this map, from noisy sensor

measurements. In contrast to sole local mapping, as described in the previous chapter, global consistent mapping allows to compensate for accumulated drift and provides an allocentric reference frame. An example for an allocentric map is shown in Figure 5.1.

This chapter describes an approach to SLAM utilizing the efficient local mapping approach described in the previous chapter by globally registering local dense 3D maps with 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.

By using local multiresolution maps the environment can be modeled as sparse graph of key poses, gaining scalability and robustness. Furthermore, the approach allows online operation and is designed for mapping and localization on limited resources, during operation.

5.1. Related Work

The SLAM problem is one of the fundamental problems in robotics, and thus has been studied in the robotics community since decades (Stachniss et al., 2016; Thrun et al., 2005). The current *de-facto* standard in robotics, is to formulate the SLAM problem as maximum a posteriori estimation in a factor graph (Kschischang et al., 2006), going back to the work of Lu and Milios (1997).

Building maps with 3D laser scanners in the field of autonomous navigation has been studied for mobile ground robots (Anderson and Barfoot, 2013; Bosse and Zlot, 2009; Elseberg et al., 2012; Maddern et al., 2012; Magnusson et al., 2007; Nüchter et al., 2005; Stoyanov and Lilienthal, 2009) and also for MAVs in 2D (Bachrach et al., 2009; Grzonka et al., 2009, 2012; Huh et al., 2013; Shen et al., 2011; Tomić et al., 2012) and recently in 3D (Cover et al., 2013; Scherer et al., 2012; Takahashi et al., 2008; Thrun et al., 2003). While the 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 (Anderson and Barfoot, 2013; Bosse and Zlot, 2009; Elseberg et al., 2012; Maddern et al., 2012; Stoyanov and Lilienthal, 2009).

A common research topic in SLAM with 3D laser scanners is how to maintain high run-time performance and low memory consumption simultaneously. Ensuring scalability, when processing the vast amount of data produced by modern sensors, remains challenging. Approaches addressing this problem reduce the complexity of the graph, e.g., by sparsification or pruning of unnecessary parts in the graph (Stachniss and Kretzschmar, 2017; Teniente et al., 2011) For instance, Kret-

zschmar et al. (2011) propose an information-theoretic approach to select nodes that can be removed without significant loss of information and thereby compress the pose graph. Similarly, the method of Carlevaris-Bianco and Eustice (2013) aims at reducing the number of nodes in a graph. The approach produces a sparser approximation of the constraints in the graph and thereby reduces inference complexity. Konolige et al. (2010) use a combination of sparse bundle adjustment and graph-SLAM.

Another way addressing computational and scalability challenges is to partition the pose graph in *submaps* (Bosse et al., 2004; Hess et al., 2016; Lenac et al., 2017; Ni and Dellaert, 2010; Zhao et al., 2013). The presented approach in this chapter also partitions the graph in submaps, where each submap is a local multiresolution map.

Belter et al. (2012) 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 (Hebert et al., 1989), extending 2D grid maps by adding a height for every grid cell. While elevation maps only model a single surface, multi-level surface maps (Triebel et al., 2006) store multiple heights in each grid cell, allowing to model environments with more than on surface, such as bridges for example. Pfaff et al. (2007) propose a method for detecting loop closures in elevation maps. Fankhauser et al. (2014) use local elevation maps and handle drift by propagating uncertainties of the robot pose through the map.

5.2. Graph-SLAM with Local Multiresolution Maps

We model the allocentric map as a graph of robot poses, also called *view poses*. Each view pose v_i captures the 6D robot pose at step i and is augmented with the local multiresolution map at step i. The allocentric map is build by aligning multiple local multiresolution maps, acquired from different view poses. The different view poses are modeled as nodes $\mathcal{V} = \{v_1, \ldots, v_M\}$ in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{Z})$ that are connected by edges $\mathcal{Z}$. The graph representation is shown in Figure 5.2. When the robot traverses the environment, new nodes are generated based on the distance to the last node. The new node is aligned with the last node by surfel-based registration. The registration result $T(i, j)$ between a new node v_i and the previous node v_j imposes a spatial constraint modeling an edge $z_{ij} \in \mathcal{Z}$. The uncertainty of the $T(i, j)$ is captured by the covariance matrix $\mathbf{\Omega}_{i,j}$ that is calculated from the assigned surfels from registration, as described in Section 5.2.1.

From the graph of spatial constraints, we infer the probability of the trajectory

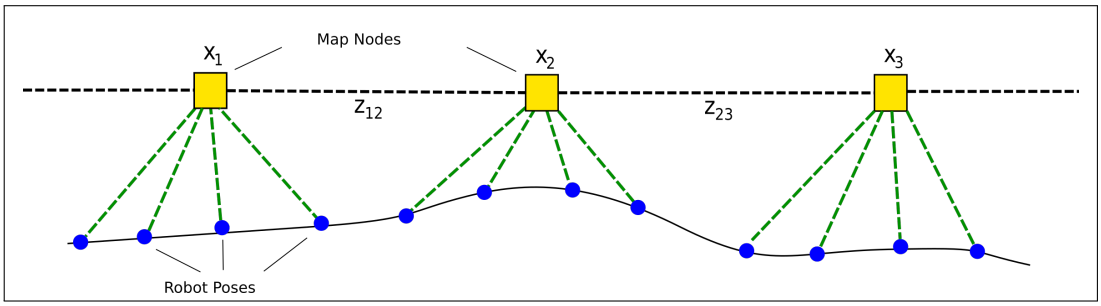


Figure 5.2.: Graph representation of the optimization problem at hand: The vertex set $\mathcal{V}$, correspond to the estimation variables, i.e., the poses of the local multiresolution maps. The edge sets $\mathcal{Z}$ represent spatial constraints from aligning two local maps to each other.

estimate given all relative pose observations

$$p(\mathcal{V} \mid \mathcal{Z}) \propto \prod_{z_{i,j} \in \mathcal{Z}} p(v_i^j \mid v_i, v_j). \tag{5.1}$$

Each spatial constraint is a normally distributed estimate with mean and covariance. Optimization of the pose graph is efficiently solved using the g^{2}o framework by (Kuemmerle et al., 2011), yielding maximum likelihood estimates of the view poses $\mathcal{V}$.

Modeling the robot trajectory as a discrete set of poses and the relative transformations between the robot poses as constraints allows to formulate SLAM as an optimization problem. This formalism is considered as *factor graph* and estimating the robot poses consists of finding a set of variables that align the nodes as close as possible. This is defined by the error term $\mathbf{e}_{i,j}$ between node v_i and node v_j. $\mathbf{e}_{i,j}$ becomes zero when $z_{i,j} = (v_i^{-1} v_j)$. Consequently, the optimization problem can be formulated as

$$\mathbf{F}(\mathbf{x}) = \sum_{i,j} \underbrace{\mathbf{e}_{i,j}^T(x_i, x_j)\mathbf{\Omega}_{i,j}\mathbf{e}_{i,j}(x_i, x_j)}_{\mathbf{F}_{i,j}} \tag{5.2}$$

$$\mathbf{x}^* = \underset{\mathbf{x}}{argmin}\, \mathbf{F}(\mathbf{x}) \tag{5.3}$$

$$= \underset{\mathbf{x}}{argmin} \sum_{k:=i,j} \mathbf{e}_k^T(x)\mathbf{\Omega}_k\mathbf{e}_k(x). \tag{5.4}$$

and the optimal configuration of the nodes can be found by minimizing the sum of squares of Equation (5.4). Note that two things are crucial in order to allow this approach to work. First of all, the system has to be overdetermined, meaning that

minimizing the sum of squares of Equation (5.4) requires to have more observations than states. In the context of slam, so-called loop closures are key to achieve this overdeterminism. Secondly, finding a good estimate for covariance matrix $\mathbf{\Omega}_{i,j}$ is important.

In the absence of noise and registration inaccuracies, this process would lead to an allocentric representation of the environment by view poses and local maps. However, noise in the sensor measurements and even slight misalignments by the registration accumulate over the entire robot trajectory and result in a displacement of the estimated robot pose. The accumulated error is called *drift*.

To accommodate drift, spatial constraints between close-by view poses that are not in temporal sequence, are added. The later is usually called *loop closure* and occurs e.g., when the robot re-enters parts of the environment already visited. A spatial constraint between the current view pose and an old view pose of the same part of the environment, captures the relative error between the accumulated pose and the registration result. The relative error allows for optimizing the view poses to minimize the accumulated error and yields global consistency of the view poses.

5.2.1. Spatial Constraints Between Local Multiresolution Maps

We denote edges $e = ((V_e, V_e'), T_e, I_e) \in \mathcal{E}$ as spatial constraint between the local maps V_e and V_e' with the relative pose T_e and the information matrix I_e, which represents the uncertainty of the relative pose T_e.

Besides the relative pose between two local multiresolution maps, a measure of uncertainty, called the information matrix, is necessary in order to model an edge in the pose graph. In other words, the quality of the registration has to be determined from the two maps and the estimated transformation between them. A common approximation for the pose uncertainty $\Sigma_{\mathbf{T}_e}$ is introduced by Censi (2007) and can be computed by

$$\Sigma_{\mathbf{T}_e} \approx \left(\frac{\partial^2 L}{\partial x^2}\right)^{-1} \frac{\partial^2 L}{\partial z \partial x} \Sigma(z) \frac{\partial^2 L}{\partial z \partial x}^T \left(\frac{\partial^2 L}{\partial x^2}\right)^{-1}, \tag{5.5}$$

where x is the pose estimate $\Sigma_{\mathbf{T}_e}$ in parametrized form $x = (\mathbf{t}, \mathbf{q})^T$ with translation $\mathbf{t}$ and quaternion $\mathbf{q}$, L is the objective function from Equation (4.7), z the associated surfels between the two maps, and $\Sigma(z)$ the covariance of the associated surfels. Thus, I_e is the (symmetric and positive semi-defined) inverse of the covariance matrix $\Sigma_{\mathbf{T}_e}$ from registration, capturing the certainty of a spatial constraint along the observable dimensions.

The graph is initialized from a local multiresolution map and extended while the robot traverses the environment by adding new view poses. New view poses

are added when the robot moved sufficient far and a spatial constraint between the generated view pose and the previous is added to the graph. In other words, new view poses are generated based on the distance between the current robot pose and the closest view pose in the graph. Consequently, the distance threshold to generate new view poses controls the density of the graph. A reasonable choice for this parameter is based on the sensor range.

In addition to edges between the previous node and the current node, we add spatial constraints between close-by view poses that are not in temporal sequence. These constraints enable loop closure when the robot revisits parts of the environment. Since checking all nodes for possible loop closures is computationally intractable, we determine a probability

$$p_{\text{chk}}(v_{\text{cmp}}) = \mathcal{N}\left(d(x_{\text{ref}}, x_{\text{cmp}}); 0, \sigma_d^2\right) \tag{5.6}$$

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

5.2.2. Pose Graph Optimization

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

$$p(\mathcal{V} \mid \mathcal{E}) \propto \prod_{e_{ij} \in \mathcal{E}} p(x_i^j \mid x_i, x_j). \tag{5.7}$$

Each spatial constraint is a normally distributed estimate with mean and covariance. Optimization of the pose graph is efficiently solved using the g^{2}o framework by (Kuemmerle et al., 2011), yielding maximum likelihood estimates of the view poses $\mathcal{V}$.

Note that this formulation—and consequently the the optimization of the factor graph—only depends on an initial odometry estimate and constraints between local maps from registration. Typically the formulation as factor graph includes modeling landmarks that the robot perceives from different view poses. However, the presented approach allows to model the problem without the formulation of landmarks in the factor graph.

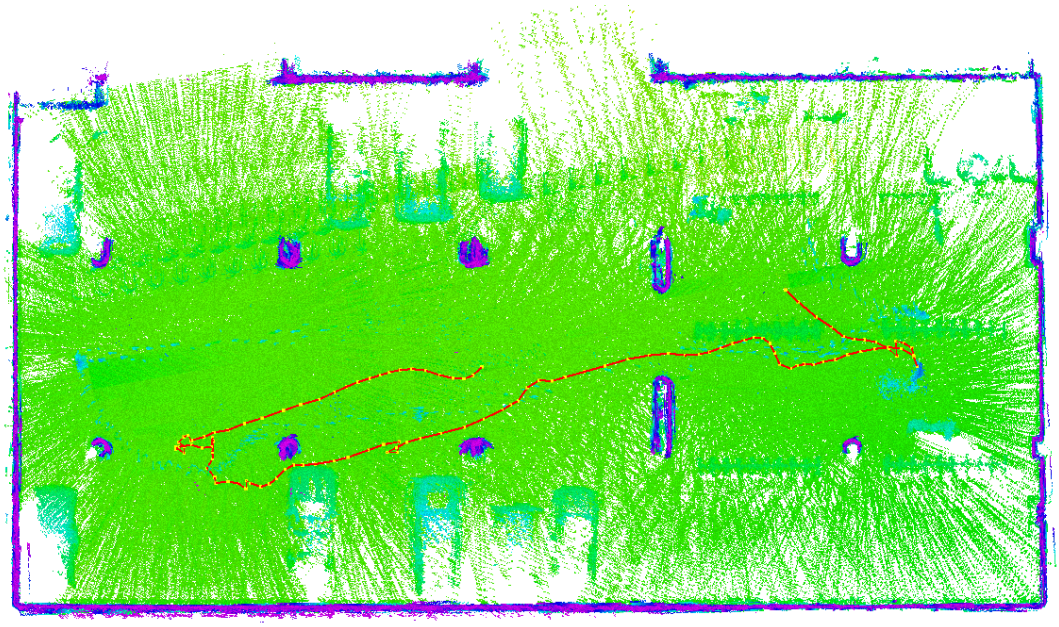


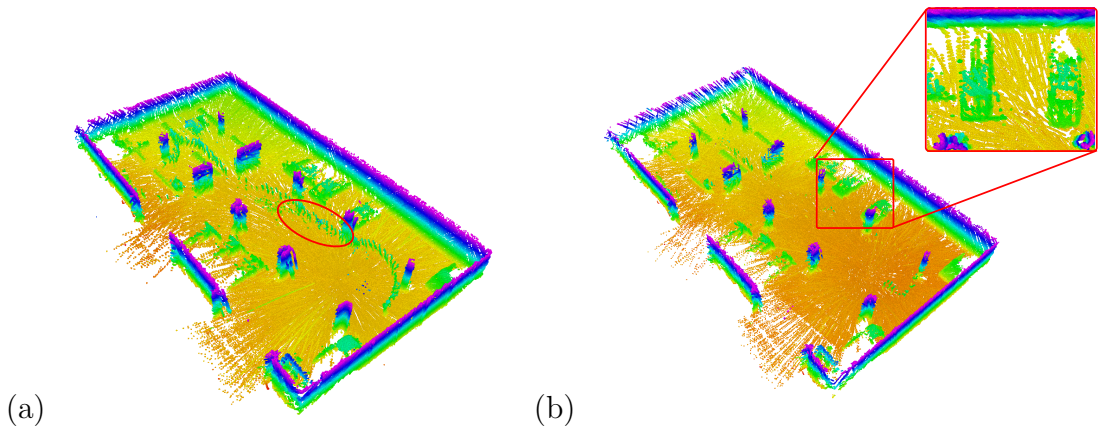
Figure 5.3.: A top-down view of the resulting map. The point color (green to purple) encodes the distance from the ground. The yellow points, connected by red lines, show the trajectory of the pose graph optimization.

5.3. Experiments

This section describes the experiments that have been conducted to evaluate the presented approach. The experiments are carried out on different robotic platforms, such as MAVs and mobile ground robots, and different laser scanners, to gain variety in the sensor data. Besides a qualitative assessment of the resulting map data, experiments involving an integrated system are shown. The latter are performed to demonstrate robustness, efficiency and versatility of the approach. However, a quantitative evaluation e.g., based on data with ground-truth information is difficult due to a missing reference systems, such as a MoCap system or highly accurate GPS.

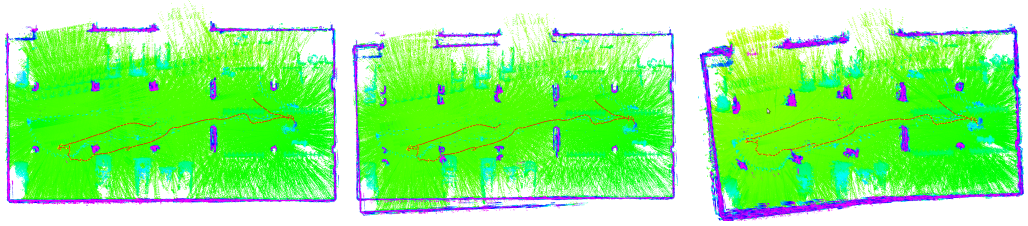
In the first experiment, a data set has been acquired with an MAV during flight in a parking garage. The MAV was controlled by a human operator following it. The data sequence contains 200 3D scans and the overall trajectory length is 73 m, covering the complete parking garage. The sensor used in this experiment is a continuously rotating lidar described in Section 2.1.1. Throughout the experiments, four levels are used for the map with a cell length of 0.25 m at the finest level, which yields a cell length of 2 m at the coarsest level.

Figure 5.3 shows the resulting allocentric map and trajectory after pose graph optimization. For this figure, we chose an orthogonal top-down perspective to get



(a) (b)

Figure 5.4.: The resulting map of the parking garage after pose graph optimization from different views. Points corresponding to the ceiling of the parking garage have been cut out for visualization. Color encodes the distance from the ground. a) vertical artifacts (red circle) are caused by a human safety pilot following the MAV; b) after filtering for dynamic obstacles.



(a) **(b)** **(c)**

Figure 5.5.: Top-down views of resulting maps. Using our surfel registration method and and global graph optimization (a), our surfel registration method without global graph optimization (b), and GICP registration with global graph optimization (c).

an indication about the consistency of the aligned 3D scans by the parallel walls. In contrast, Figure 5.4 shows the resulting map from different perspectives, which allows for a better interpretation of the scene. Here, cars and pillars in the parking garage can be identified in the globally aligned 3D scans.

To assess the improvement of the map accuracy that can be attributed to global pose graph optimization, Figure 5.5 compares the resulting maps of different registration methods. Figure 5.5b) shows that without pose graph optimization, the trajectory aggregates drift which results in inconsistencies indicated by the misalignment of the walls. Figure 5.6 shows that even lamps hanging from the ceiling are modeled by the 3D point cloud.

We compare our registration method to a state-of-the-art method by aligning the sparse 3D scans by the Generalised-ICP (GICP) (Segal et al., 2009). Note

Figure 5.6.: The structure of the ceiling showing hanging lamps in a photo (left) and the resulting 3D map (right).

that, similar to our pipeline, scans are assembled by visual odometry and pose graph optimization is used to globally align the local dense 3D maps to have a fair comparison. Figure 5.5c shows that the resulting 3D map is less accurate and smeared. In terms of run-times our method is computationally more efficiency with 145 ± 50 ms compared to GICP with 1555 ± 613 ms. Here, mean run-times and standard deviation over the complete dataset are reported.

In a second experiment we acquire data in an outdoor scenario mapping a gas station during flight. The resulting global maps are shown in Figure 5.7. The resulting map shows that the geometry of the environment is reconstructed correctly, but shows a slight degradation in map quality and sharpness, compared to the previous experiment in the parking garage. The main reason for this degradation is the limited sensor range of the employed Hokuyo 2D LRF. While the parking garage provides sufficient structure in the sensor's FoV in every direction to register acquired scans accurately, the environment around the gas station does not. Experiments with a sensor which allows for higher measurement range are shown in the following experiment.

The previous experiments where carried out with the sensor platform described in Section 2.1.1. The platform has a number of limitations, such as the maximum measurable range of the Hokuyo LRF, the update rate and measurement density of the continuously rotating 3D scanner, but also limited flight time of the MAV platform. The latter does not allow for more comprehensive experiments including longer trajectories with loop closure.

Compared to this, the following experiment was performed with the MAV and

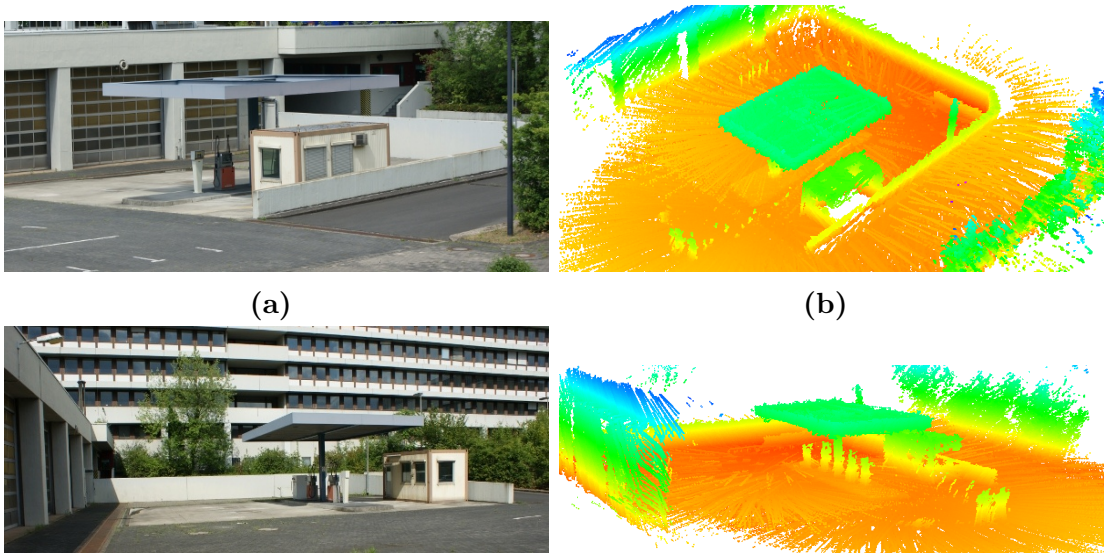


Figure 5.7.: (a+c) photo of the scene from different perspectives. (b+d) the resulting allocentric maps acquired during flight with an MAV.

lidar sensor described in Section 2.1.3. The platform allows for longer flight times, resulting in longer trajectories with a variety of geometric structure. The experiment includes mixed indoor and outdoor environments, changes in elevation, vegetation, and building facades. Throughout the experiments, six levels are used for the map with a cell length of 0.25 m at the finest level, which yields a cell length of 8 m at the coarsest level.

The resulting allocentric map and estimated trajectory after graph optimization is shown in Figure 5.8. The overall length of the trajectory is 820 m including open areas and challenging hallways. The MAV starts in the courtyard of the building complex, traverses the environment around the building, and returns to the courtyard by entering the same hallway. Due to the length of the trajectory loop closure is crucial in this experiment. The resulting map shows that the environment has been reconstructed accurately and that the approach scales to larger environments. However, a current limitation of the implementation is that the local multiresolution maps, including the aggregated 3D measurements are stored in memory. Thus, the memory of the onboard computer poses a limitation of the size of the environment that can be reconstructed. The robot in this experiment was equipped with 32 GB memory and ~26 GB were utilized by the local multiresolution maps. It has to be mentioned, that the size of the local multiresolution is mainly caused by storing the original 3D point measurements—which are only used for visualization—and only a fraction is induced by the surfel elements that

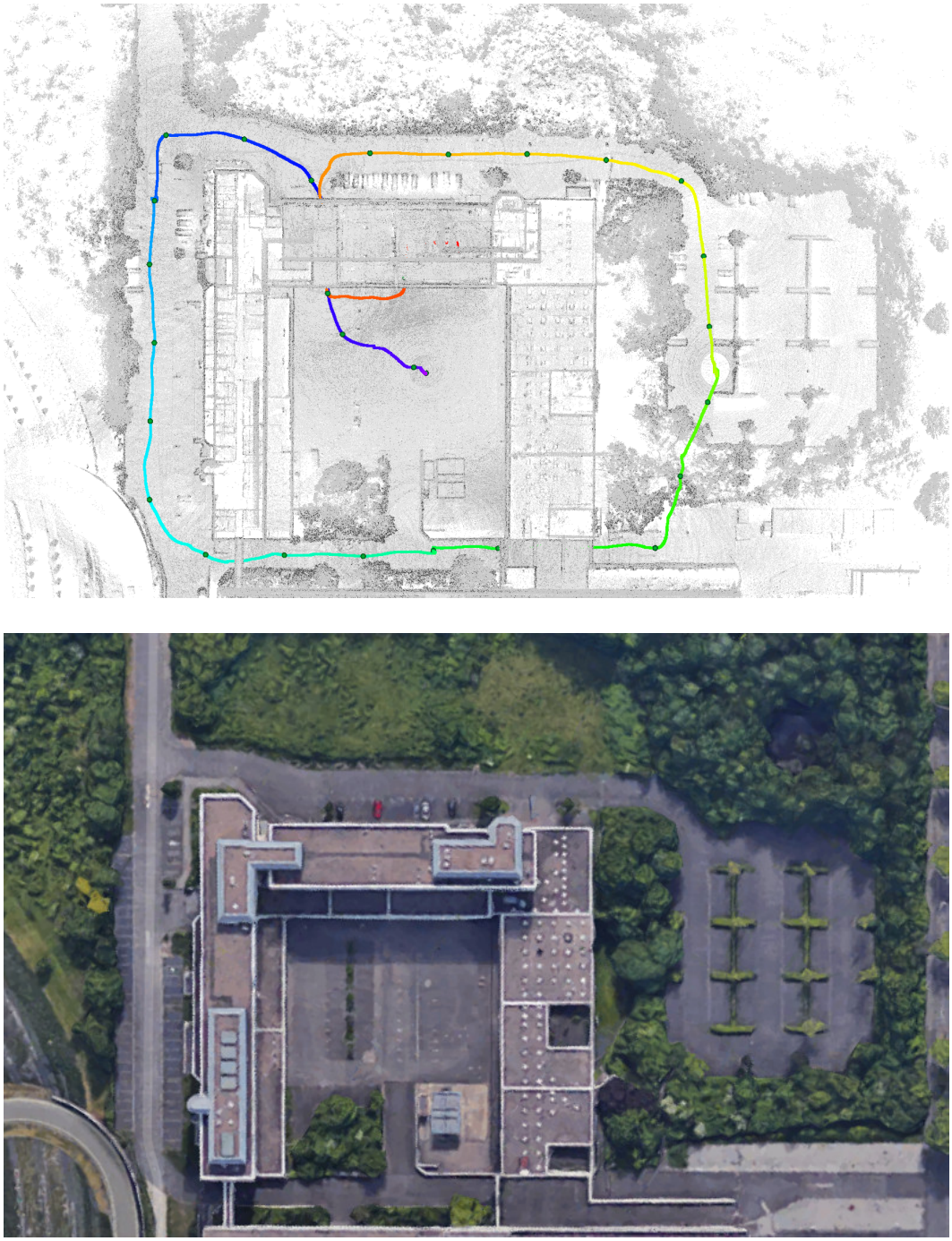


Figure 5.8.: Top-down view of the resulting allocentric map with trajectory (top) and photo of the scene (bottom) of the Courtyard data set. The data set has been acquired with a manual controlled MAV with an overall trajectory length of 820 m. Colored points depict trajectory of the of the robot, from start (blue) to end (red). The green spheres show the view poses of the local multiresolution maps.

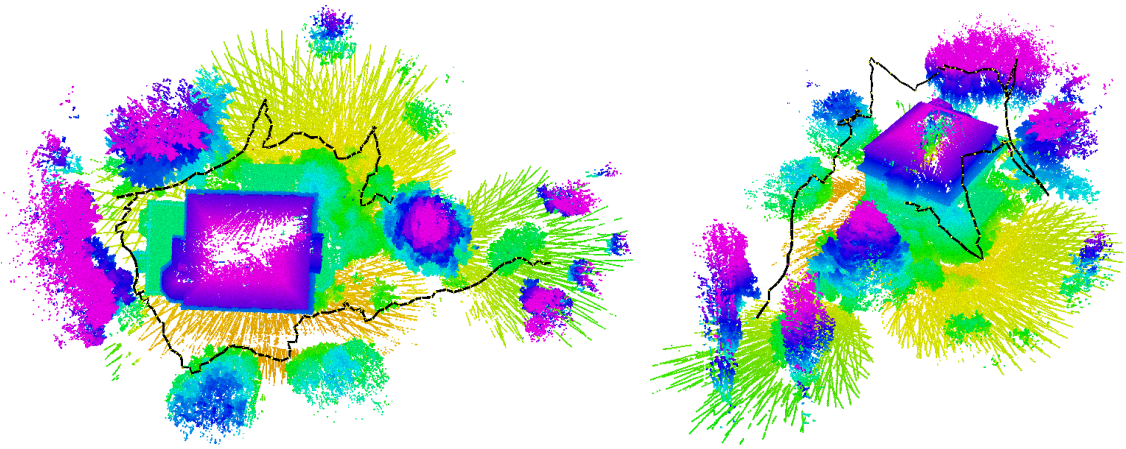


Figure 5.9.: Evaluation of the integrated system by mapping an old manor house. The resulting trajectory (black) traversed by the MAV from top and side view. Color encodes the height from the ground.

are necessary for registration. Consequently, a more efficient data structure for visualization (e.g., a mesh) or serialization of the point measurements to disk, allows to reconstruct even larger environments.

So far, the data used in the experiments was manually acquired by a human pilot controlling the MAV. While this is sufficient to show that the geometry of the environment can be reconstructed, it does not assess the robustness of the approach. Therefore, the following experiments were conducted in an integrated system that add planing and control components, leveraging the presented approach for mapping and localization, and implement a fully autonomous navigation system. Consequently, these experiments are more challenging since an integrated system poses requirements on robustness and run-time. Furthermore, the controller executes planed trajectories not as smooth as the human pilot.

5.3.1. Autonomous Mapping of Outdoor Environments

The following describes the evaluation of the approach integrated into one working MAV mapping system. Since planning and navigation components are not part of this thesis, they are not further described and referred to Droeschel et al. (2015) for details. The experiment describes one exemplary mapping session of the MAV.

The main goal of this experiment was to autonomously map an old manor house as shown in Figure 5.10a. The scenario involves vegetation such as trees and bushes and is difficult to traverse from all sides by humans. Therefore, a manual flight by a human pilot controlling the MAV was not possible in this scenario.

The user manually defined a set of mission-relevant view poses given the coarse

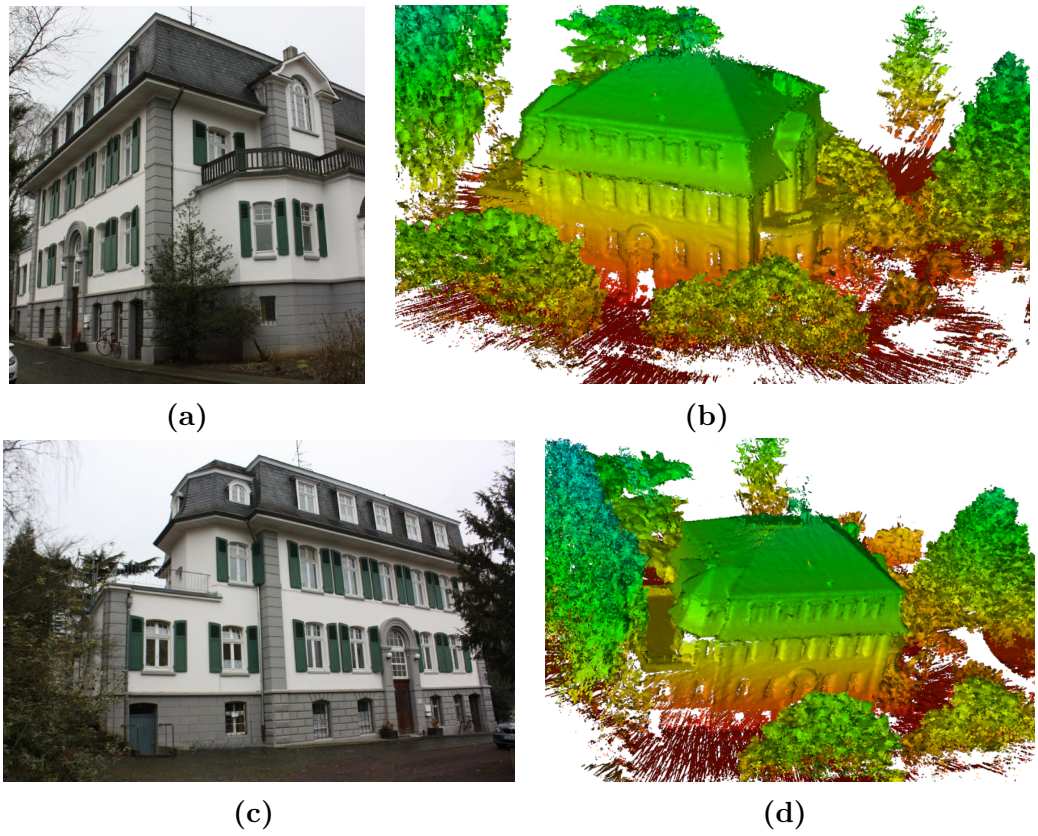


Figure 5.10.: Evaluation of the integrated system by mapping an old manor house. (a + c) Photos of the scene. (b + d) the resulting global maps after the mission. Color encodes the height from the ground.

LoD 2 world model. These view poses, consisting of a GPS position and view direction, were roughly specified to cover the building facade from all sides. Then, they are processed by a mission control layer, incorporating the mission planner. In this experiment, only the local mapping components (described in Chapter 3) were used by the navigation layer, allowing for reactive obstacle avoidance. Therefore, the planner incorporates information of occupied cells in the local multiresolution map to plan a collision free trajectory. The overall flight duration was approximately 8 minutes and the traversed trajectory is shown in Figure 5.9. The MAV successfully traversed the building in all experiments without colliding with an obstacle.

Although the allocentric mapping and localization system was not necessary for the navigation planner to traverse the view poses, it was mapping the environment during mission to gain a 3D reconstruction of the building facade. Resulting 3D maps of the are shown in Figure 5.10. It can be seen that our method is able to successfully map the building. Even details of the structure of the facade, such as

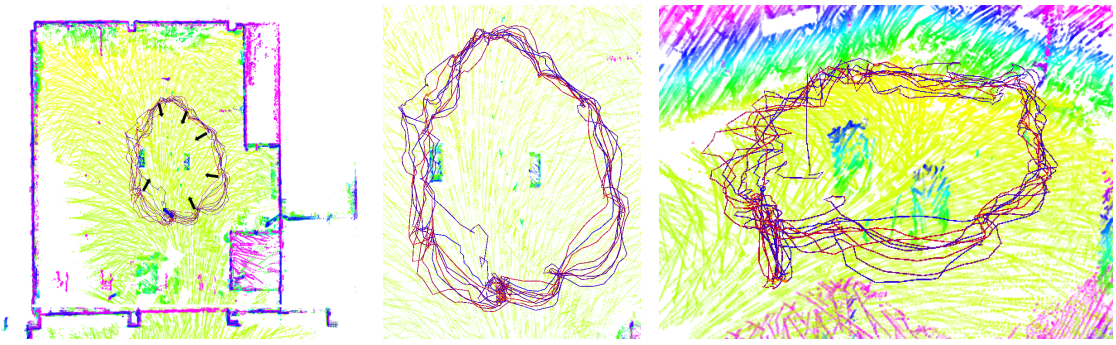


Figure 5.11.: We conducted indoor flight experiments in a 20 x 16 x 6 m hall. The trajectories show the laser-based localization estimates of 10 consecutive autonomous flights. From left to right: map topview, closeup of the trajectories, 3D view. Observation poses are depicted by black arrows, the blue arrow shows the return pose. The map color encodes height.

windows and doors are represented in the map.

5.3.2. Autonomous Navigation in GNSS-denied Environments

In outdoor environments as the one in the previous experiment, GPS can be used as external positioning system and—assuming sufficient accuracy—the navigation planner can navigate based on the measured GPS position. In contrast, GNSS-denied environments, such as warehouses or parking garages necessitate localization without external reference systems. The allocentric mapping and localization system presented in this chapter serves as such and allows indoor localization without an external reference system.

This section describes the evaluation of the approach by integrating it in an MAV mapping and inspection system described in Nieuwenhuisen et al. (2016). The following describes one exemplary indoor inspection session of the MAV. The main goal of this experiment was to autonomously navigate to certain predefined waypoints in a hall employing solely means of localization that are available in both indoor and outdoor environments. Localization was performed employing the presented approach. Since the laser scanner acquires complete 3D scans with a relatively low frame rate, an egomotion estimate from visual odometry and measurements from an IMU are incorporated to track the pose of the MAV between two 3D scans.

A first manual flight was conducted to acquire the allocentric map, that is later used for localization. Based on this map a mission with six observation poses—plus a return pose 2 m above the start pose was defined. A mission planner plans paths between every pair of mission poses and determines the best visiting order. After

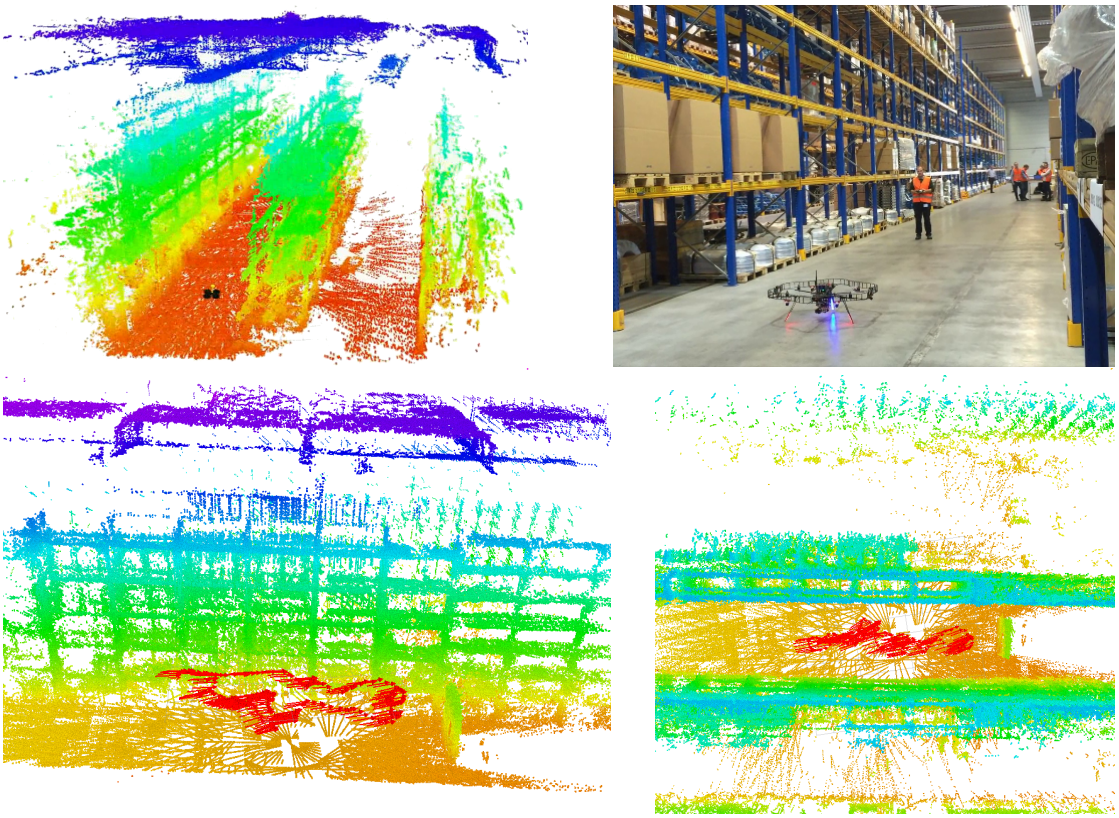


Figure 5.12.: Top: Mapping results. Left: Resulting point cloud after pose graph optimization acquired by a manual flight along a warehouse aisle (color depicts height). Right: Photo of the mapped aisle. Bottom: Localization result. The MAV trajectory (red arrows) is tracked by means of laser scan registration, combined with visual odometry and IMU measurements. This yields 6D pose estimates. Shown is a flight through a warehouse aisle. In the side-view (left), the relation to the accurately mapped storage units can be seen. In the top-down view (right), it can be seen that the pose is tracked despite considerable self similarity of the shelves. Map color encodes height.

takeoff, the global planner begins to continuously plan paths to the next mission relevant pose. The local obstacle avoidance keeps the MAV successfully away from obstacles like hanging cables or people walking around in the hall. In these experiments, the allocentric paths are planned in a grid with a cell size of 0.5 m. An excerpt from the map, the inspection poses, and the traversed trajectories of ten missions are shown in Figure 5.11. The MAV successfully reached all poses in the experiments without colliding with an obstacle and was localized all the time.

The integrated system was also evaluated in an indoor warehouse system. The considered application scenario is an automatic warehouse inventory, performed by an autonomous MAV. The MAV and laser scanner used in this experiment is described in Section 2.1.2. Similar to the previous experiment, the initial map was

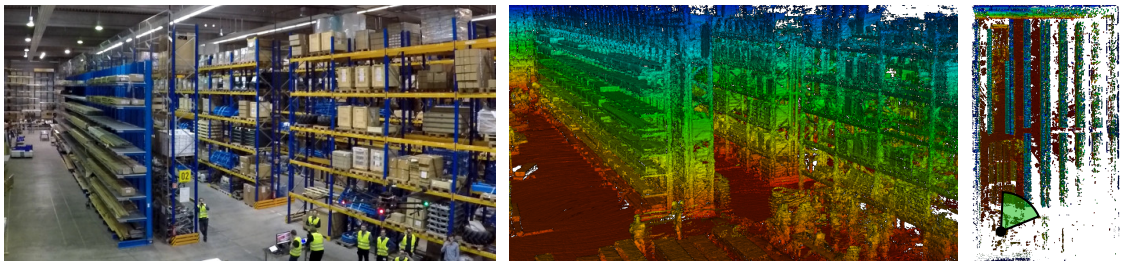


Figure 5.13.: 3D map from the initial manual flight. The top-down view (right) shows the dimensions of the acquired map of the 100 × 60 m warehouse. The camera perspective is highlighted in green. The warehouse contains tight, self-repetitive, and cluttered structures like shelves and stock, and larger, far-away structures like walls. For robust localization, the MAV has to employ a map of the structure of the large building.

built from data acquired in a manual flight, using the presented mapping approach. This map was used to specify view poses that the robot traversed during mission. Figure 5.12 shows the resulting allocentric map and localization poses.

In these indoor experiments, the presented approach allowed for accurate reconstruction of the geometry of the environment and reliable localization throughout the experiments. However, similar to the outdoor experiments in the previous section, the employed 3D scanners pose limitations on update rate and measurement range. This is especially problematic in warehouse environments, where the local structure of the aisles show high ambiguity.

Figure 5.13 show results of the system on the subsequent hardware platform described in Section 2.1.3. Due to the higher measurement range and update rate of the laser sensor, the system allows for faster navigation in larger environments (Beul et al., 2018).

5.3.3. Mapping in Rough Terrain

This section describes the application of the presented mapping and localization on a the mobile manipulation robot Momaro described in section Section 2.2. 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 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 unstruc-

[1] `http://www.theroboticschallenge.org/`

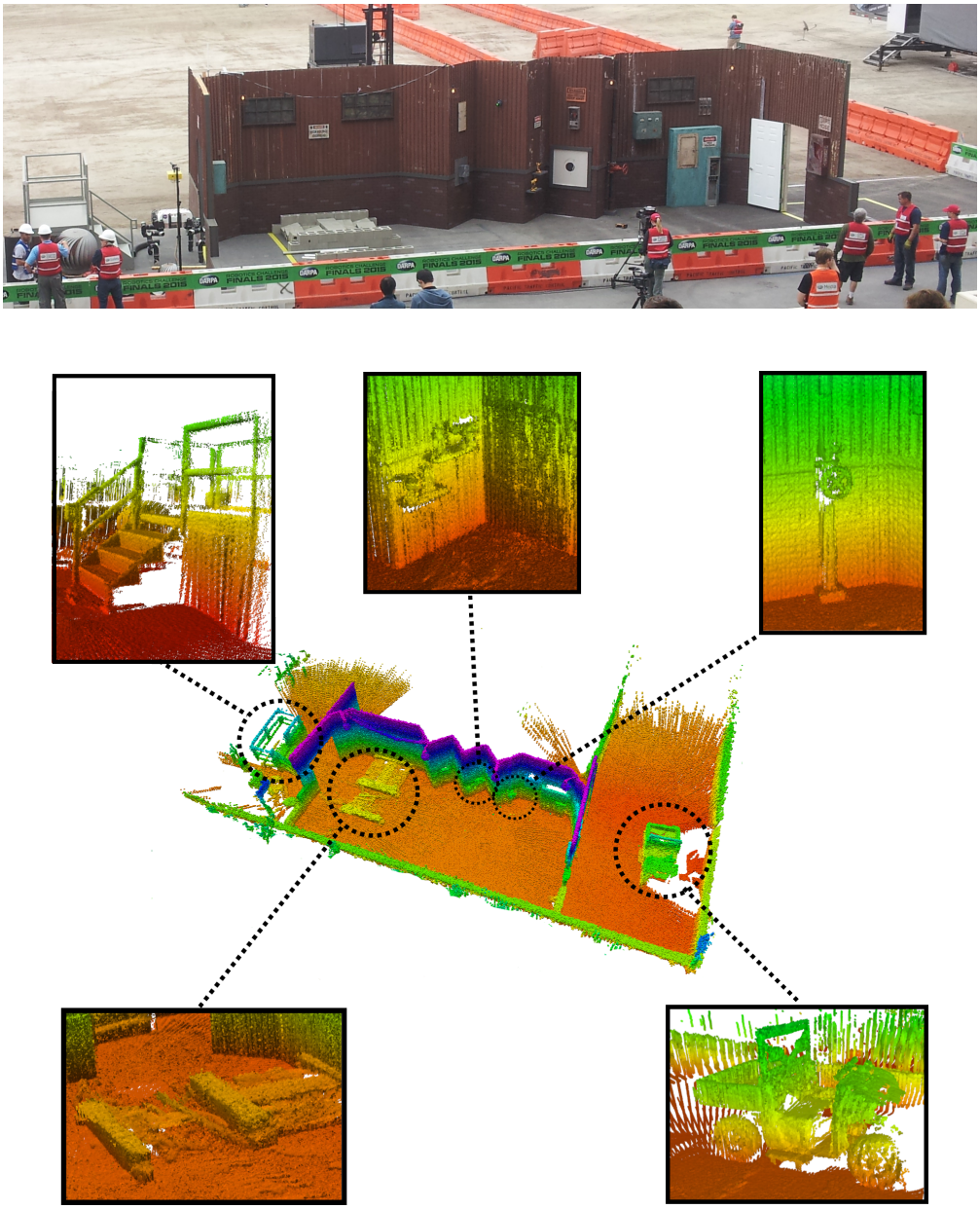


Figure 5.14.: Top: The mock-up disaster scenario of the DRC. Bottom: The resulting allocentric map generated from the data of our first competition run. Color encodes the height from ground.

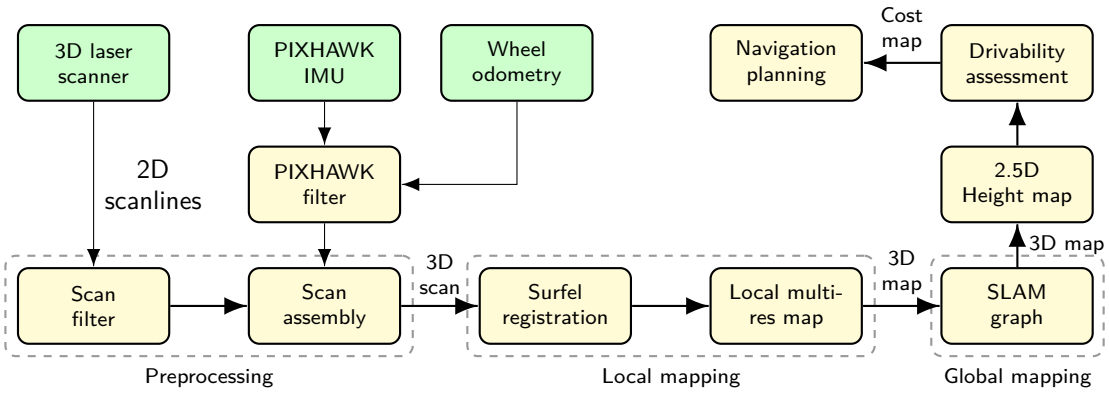


Figure 5.15.: Overview of our mapping, localization and navigation system. The measurements are processed in preprocessing steps described in Section 2.1.2. The resulting 3D point cloud is used to estimate the transformation between the current scan and the map (Chapter 4). Registered scans are stored in a local multiresolution map (Chapter 3). Keyframe views of local maps are registered against each other in a SLAM graph (Section 5.2). A 2.5D height map is used to assess drivability.

tured environments—also due to the lack of applicable perception methods—is still beyond the state of the art.

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 were used 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 5.14 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 (Rodehutskors et al., 2015), 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

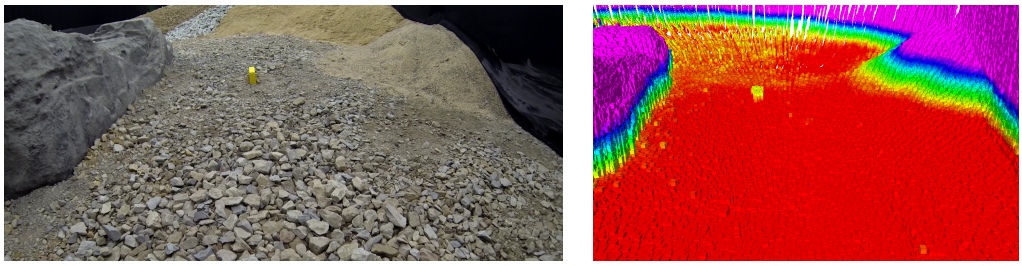


Figure 5.16.: Photo and the corresponding local map of the battery pack—one of the objects to manipulate during the SpaceBot Camp. Color encodes distance from ground.

website[2], including a video or our first day competition run[3].

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.

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—the allocentric mapping described in this chapter was used to allow for fully autonomous navigation. An overview of the system architecture is shown in Figure 5.15.

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-frequent 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 5.17. 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 5.18 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 5.19 and to assess traversability for navigation as shown in Figure 5.20. An overview

[2]Website of our DRC entry `http://www.nimbro.net/Rescue`
[3]Video of first day DRC competition run `http://youtu.be/NJHSFelPsGc`

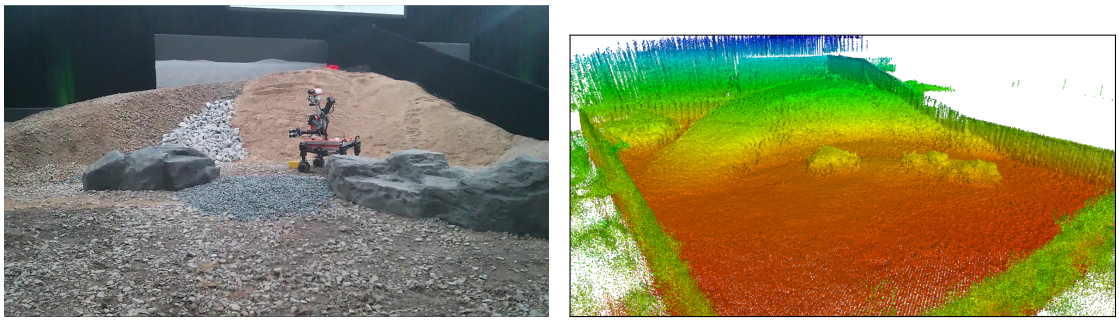


Figure 5.17.: Left: Photo of the planetary-like environment at the DLR SpaceBot Camp consisting of different types of stones and soil. Right: The resulting 3D map built by our mapping component from data that has been collected during our run. Color encodes distance from ground.

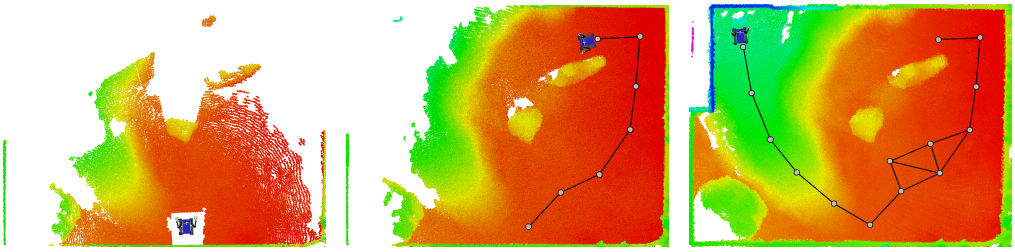


Figure 5.18.: The allocentric map from a top view at different time steps, consisting of 1 (left), 7 (middle) and 14 (right) key frames. Color encodes height. The nodes in the pose graph (grey circles) are connected by spatial constraints (black lines). The robot model shows the current position of the robot.

of the integrated system architecture is shown in Figure 5.15.

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[4], including a video or our demonstration run[5].

Figure 5.21 shows the results of the Hokuyo lidar-based mapping for the UBO hall dataset. In addition to the resulting graph structure and a top-view of the map, different detail views demonstrate the reconstruction quality. Note that for visualization the ceiling has been cut off.

[4]Website of our DLR SpaceBot Camp entry `http://www.nimbro.net/Explorer`
[5]Video of SpaceBot Camp demonstration run `http://youtu.be/q_p5ZO-BKWM`

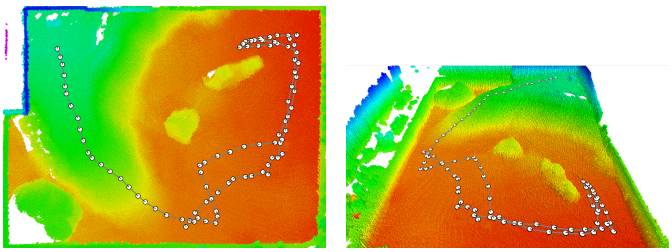


Figure 5.19.: The resulting allocentric map from two different perspectives with the localization poses (black circle) from our run. Color encodes height from ground.

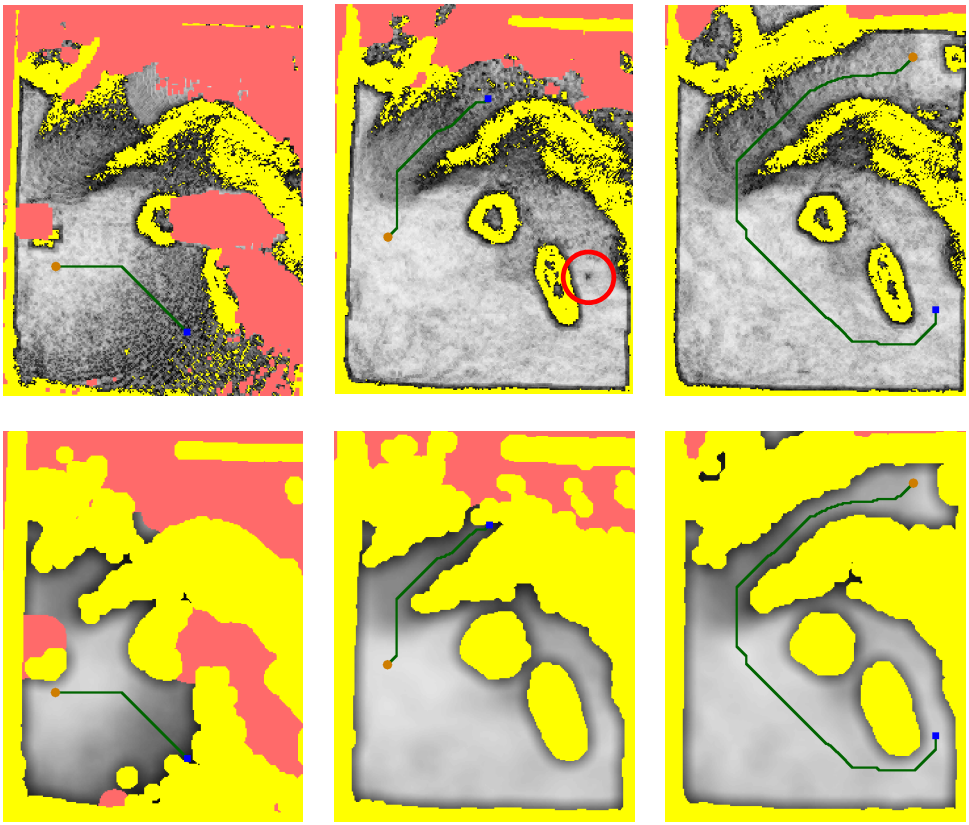


Figure 5.20.: Navigation planning during (left, middle) and after exploration (right) of the SpaceBot Camp arena. The top row shows the calculated traversability costs for each cell. The bottom row shows inflated costs used for A* path planning. The orange dot represents the current robot position, the 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 5.16).

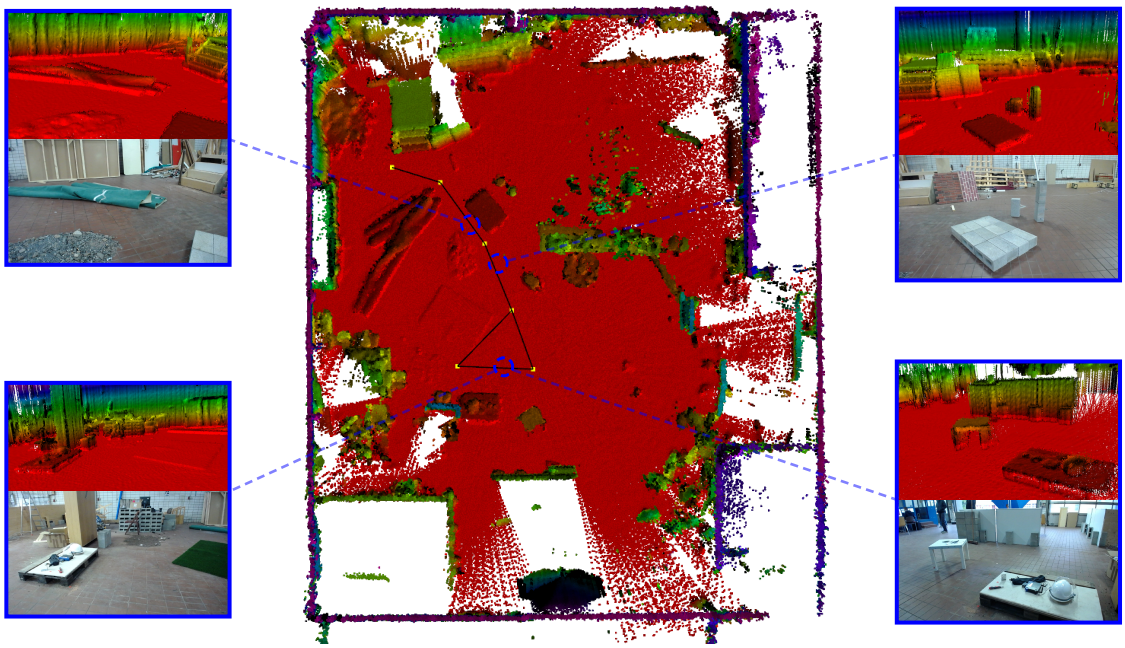


Figure 5.21.: Top-down view of the lidar-based allocentric 3D map from an indoor environment. Color encodes height from the ground. Four detail views of the resulting map with corresponding camera images (blue boxes) from different view poses (blue dashed circles) are shown. The yellow boxes connected by black lines depict the graph-based structure of the allocentric map.

5.4. Summary

This chapter presents an approach to SLAM approach using the map representation and registration method described in the previous chapters. The proposed method aligns local maps from different view poses with each other to construct an allocentric pose graph of the environment. In experiments, qualitative results from data sets acquired on different MAVs and from a mobile ground robot are shown. However, quantitative evaluation e.g., based on data with ground-truth information is difficult due to missing reference systems, such as a MoCap system or highly accurate GPS.

The presented approach in this chapter allows to optimize the view poses of the local maps by registering local maps from different view poses with each other. When the robot revisits parts of the environment, adding loop closure constraints allows to minimize accumulated odometry and registration errors. However, registration errors that have been accumulated during aggregation of the local map, are not addressed. Here, Chapter 6 presents an approach that allows to address this degeneration of the local maps.

6. Map Refinement

In this chapter, the so far presented approach is extended to allow for refining alignments during online mapping. The method is based on efficient local mapping and a hierarchical optimization back-end. The local maps are used in a multi-level graph for allocentric mapping and localization. In order to incorporate corrections when refining the alignment, the individual 3D scans in the local map are modeled as a sub-graph and graph optimization is performed to account for drift and misalignments in the local maps. Furthermore, in each sub-graph, a continuous-time representation of the sensor trajectory allows to correct measurements between scan poses. The approach is evaluated in multiple experiments by showing qualitative results and map quality is assessed based on the map entropy.

Modern 3D laser-range scanners have a high data rate, making online SLAM computationally challenging. Recursive state estimation techniques are efficient but commit to a state estimate immediately after a new scan is made, which may lead to misalignment of measurements.

Laser-based mapping and localization has been widely studied in the robotics community and applied to many robotic platforms (Cole and Newman, 2006; Kohlbrecher et al., 2011; Magnusson et al., 2007; Nüchter et al., 2005). The variety of approaches that exists either focus on efficiency, for example when used for autonomous navigation, or on accuracy when building high-fidelity maps offline. Often, limited resources—such as computing power on a micro aerial vehicle—necessitate a trade-off between the two. A popular approach to tackle this trade-off is to lever-

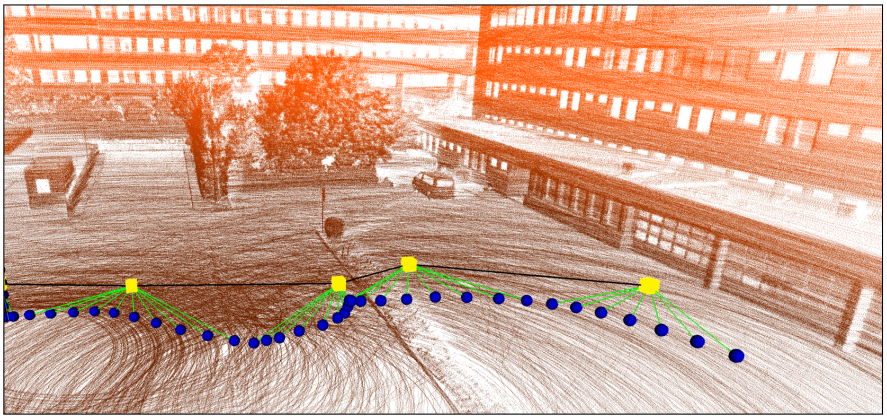


Figure 6.1.: We propose a hierarchical continuous-time SLAM method, allowing for online map refinement. It generates highly accurate maps of the environment from laser measurements. Yellow squares: coarse nodes; Blue circles: fine nodes; Red dots: continuous-time trajectory.

age other sensor modalities to simplify the problem. For example, visual odometry from cameras and inertial measurement units (IMU) are used, to estimate the motion of the laser-range sensor over short time periods. The motion estimate is used as a prior when aligning consecutive laser scans, allowing for fast and relatively accurate mapping.

Often inaccuracies remain, for example caused by wrong data associations in visual odometry. These inaccuracies lead to misalignment and degeneration in the map and require costly reprocessing of the sensor data. To this end, graph-based optimization is popular to minimize accumulated errors (Frese et al., 2005; Kuemmerle et al., 2011; Olson et al., 2006). However, depending on the granularity of the modeled graph, optimization is computationally demanding for large maps.

Another difficulty in laser-based SLAM is the sparseness and distribution of measurements in laser scans. As a result, pairwise registration of laser scans quickly accumulates errors. Registering laser scans to a map, built by aggregating previous measurements, often minimizes accumulated error.

However, errors remain, e.g., due to missing information. For example, incrementally mapping the environment necessitates bootstrapping from sparse sensor data at the beginning—resulting in relatively poor registration accuracy, compared to aligning with a dense and accurate map.

The previous chapters showed that local multiresolution in combination with a surfel-based registration method allows for efficient and robust mapping of sparse laser scans. The key data structure in our previous work is a robot-centric multiresolution grid map to recursively aggregate laser measurements from consecutive

3D scans, yielding local mapping with constant time and memory requirements. While being efficient, the approach presented in this chapter does allow for re-assessing previously aggregated measurements in case of registration errors and poor or missing motion estimates from visual odometry. By modeling individual 3D scans of a local map as a sub-graph, a hierarchical graph structure is built, enabling refinement of the map in case misaligned measurements when more information is available. Furthermore, the approach preserves efficient local and allocentric mapping.

6.1. Related Work

Mapping with 3D laser scanners has been investigated by many groups (Cole and Newman, 2006; Kohlbrecher et al., 2011; Magnusson et al., 2007; Nüchter et al., 2005). 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 (Anderson and Barfoot, 2013; Bosse and Zlot, 2009; Elseberg et al., 2012; Maddern et al., 2012; Stoyanov and Lilienthal, 2009). The mentioned approaches allow creating accurate maps of the environment under certain conditions, but do not allow an efficient assessment and refinement of the map.

Measurements from laser scanners are usually subject to rolling shutter artifacts when the sensor is moving during acquisition. These artifacts are expressed by a deformation of the scan and, when treating laser scans as rigid bodies, these artifacts degrade the map quality and introduce errors when estimating the sensor pose. A common approach to address this problem is to model a deformation in the objective function of the registration approach. Non-rigid registration of 3D laser scans has been addressed by several groups (Anderson et al., 2015; Brown and Rusinkiewicz, 2007; Elseberg et al., 2010; Ruhnke et al., 2012; Zlot and Bosse, 2014).

Ruhnke et al. (Ruhnke et al., 2012) jointly optimize sensor poses and measurements. They extract surface elements from range scans, and seek for close-by surfels from different scans. This data association contributes to the error term of the optimization problem but also results in a relatively high state space. Thus, their approach can build highly accurate 3D maps but does not allow for online processing.

Furthermore, rolling shutter effects can be addressed by modeling the sensor trajectory as a continuous function over time, instead of a discrete set of poses. Continuous-time representations show great advantages when multiple

sensors with different temporal behavior are calibrated (Furgale et al., 2013) or fused (Mueggler et al., 2017), but also to compensate for rolling shutter effects, e.g., for data from a RGB-D camera (Kerl et al., 2015). Continuous-time trajectory representations have been used for laser-based mapping in different works (Alismail et al., 2014; Kaul et al., 2016; Patron-Perez et al., 2015). While most of the continuous-time approaches use a spline to represent the trajectory, Anderson et al. (Anderson et al., 2015) employ Sparse Gaussian Process Regression.

Kaul et al. (Kaul et al., 2016; Zlot and Bosse, 2014) present a continuous-time mapping approach using non-rigid registration and global optimization to estimate the sensor trajectory from a spinning laser scanner and an industrial-grade IMU. The trajectory is modeled as a continuous function and a spline is used to interpolate between the sensor poses.

Recently, Hess et al. presented *Google's Cartographer* (Hess et al., 2016). By aggregating laser scans in local 2D grid maps and an efficient branch-and-bound approach for loop closure optimization. Results of Google's Cartographer have been improved by Nüchter et al. (Nüchter et al., 2017). Their method refines the resulting trajectory by a continuous-time mapping approach, based on their previous work (Elseberg et al., 2013).

Grisetti et al. present a hierarchical graph-based SLAM approach (Grisetti et al., 2010). Similar to the presented approach, a hierarchical pose graph represents the environment on different levels, allowing for simplifying the problem and optimizing parts of it independently.

Following Grisetti et al. (2010), the problem is modeled as hierarchical graph, allowing to optimize simplified parts of the problem independently. Compared to their approach we aggregate scans in local sub-maps to overcome sparsity in the laser scans. Furthermore, local sub-graphs are augmented with a continuous-time representation of the trajectory, allowing to address the mentioned rolling shutter effects.

6.2. Hierarchical Refinement

The system aggregates measurements from a laser scanner in a robot-centric local multiresolution grid maps. If available, information from other sensors, such as an IMU or wheel odometry to account for motion of the sensor during acquisition, are incorporated. Furthermore, these motion estimates are used as prior for the registration.

Local maps that are added to the pose graph are subject to the refinement, reassessing the alignment of 3D scans when more information is available. After

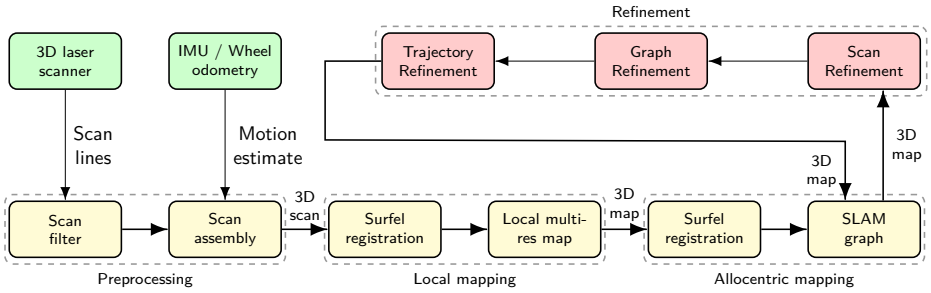


Figure 6.2.: Schematic illustration of our mapping system. Laser measurements are preprocessed and assembled to 3D point clouds. The resulting 3D point cloud is used to estimate the transformation between the current scan and map. Registered scans are stored in a local multiresolution map. Local multiresolution maps from different view poses are registered against each other in a SLAM graph. During mapping, parts of the graph are refined and misaligned 3D scans are corrected.

realigning selected 3D scans from a local map, the sensor trajectory is optimized: first for refined local maps, then for the complete pose graph. Figure 6.2 shows an overview of our mapping system. Since local mapping, allocentric mapping and refinement are independent from each other, our system allows for online mapping while refining previously acquired sensor data when more information is available.

The mapping approach is modeled in a hierarchical graph-based structure as shown in Figure 6.3. The coarsest level is a pose graph, representing the allocentric 6D pose of local maps $\mathcal{M} = \{m_1, \ldots, m_M\}$ with nodes. Each local map aggregates multiple consecutive 3D scans and represents the robot's vicinity at a given view pose.

They are connected by edges $\mathcal{E}_\mathcal{P}$ imposing a spatial constraint from registering two local maps with each other by surfel-based registration. We denote edges $E = ((M_E, M'_E), T_E, I_E) \in \mathcal{E}_\mathcal{P}$ as spatial constraint between the local maps M_E and M'_E with the relative pose T_E and the information matrix I_E, which is the inverse of the covariance matrix from registration.

The scan poses of a local map $\mathcal{M}_j$ are modeled by vertices $\mathcal{S} = \{s_1, \ldots, s_S\}$ in a sub-graph $\mathcal{G}_j$. They are connected by edges $\mathcal{E}_\mathcal{D}$. Registering a scan S_E to a local map M_E poses a spatial constraints $E = ((S_E, M_E), T_E, I_E) \in \mathcal{E}_\mathcal{D}$ with the relative pose T_E and the information matrix I_E.

The 3D scans of the local maps consist of a number of so-called scan lines. A scan line is the smallest element in our optimization scheme. Depending on the sensor setup, a scan line consists of measurements acquired in a few milliseconds. For the Velodyne VLP-16 used in the experiments, a scan line is a single firing sequence (1,33 ms). We assume the measurements of a scan line to be too sparse for robust registration. Thus, we interpolate the poses of scan line acquisitions

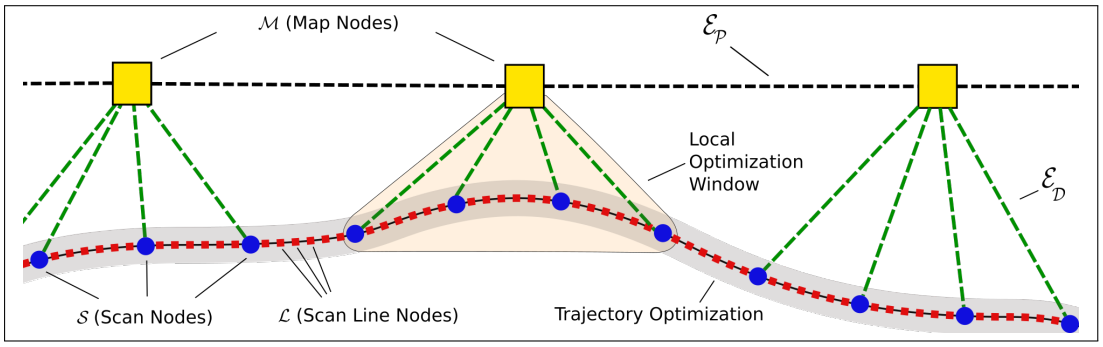


Figure 6.3.: Hierarchical graph representation of the optimization problem at hand: The vertex sets $\mathcal{M}, \mathcal{S}$, and $\mathcal{L}$ correspond to the estimation variables, i.e., the poses of the local multiresolution maps ($\mathcal{M}$), the 3D scans ($\mathcal{S}$), and scan lines ($\mathcal{L}$) of the Velodyne VLP-16. The edge sets $\mathcal{E}_\mathcal{P}$ and $\mathcal{E}_\mathcal{D}$ represent constraints from registration: $\mathcal{E}_\mathcal{P}$ from aligning two local maps to each other and $\mathcal{E}_\mathcal{D}$ from aligning a 3D scan to a surfel map. From $\mathcal{S}$ a continuous-time representation of the trajectory is estimated by a cubic B-spline, allowing to interpolate the pose for each measurement of the 3D scan ($\mathcal{L}$).

with a continuous-time trajectory representation for each sub-graph, as described later.

Optimization of the sub-graphs and the pose graph is efficiently solved using the g^{2}o framework by (Kuemmerle et al., 2011), yielding maximum likelihood estimates of the view poses $\mathcal{S}$ and $\mathcal{M}$. On their local time scale, sub-graphs are independent from each other, allowing to minimize errors independent from other parts of the graph. Optimization results from sub-graphs are incorporated in the higher level pose graph, correcting the view poses of the local maps. Therefore, we define the last acquired scan node in a local sub-graph as reference node and update the pose of the map node according to it.

During operation, we iteratively refine sub-graphs in parallel, depending on the available resources. Global optimization of the full graph is only performed when the local optimization has changed a sub-graph significantly or a loop closure constraint was added. Similarly, if global optimization was triggered by loop closure, sub-graphs are refined when the corresponding map node changed. To determine if optimization of a sub-graph necessitates global optimization or vice-versa, we compare the refined pose of the representative scan node s_r in a sub-graph to the view pose of the corresponding map node. For our experiments, we choose a threshold of 0.01 m in translation and 1° in rotation.

6.2.1. Local Sub-Graph Refinement

After a local map has been added to the pose graph, the corresponding sub-graph $\mathcal{G}_{\mathcal{M}}$ is refined by realigning selected 3D scans with its local map. Realigning only selected 3D scans, instead of all scans in a sub-graph, allows for fast convergence while resulting in similar map quality, as shown later in the experiments.

For a sub-graph, we determine a 3D scan s_k for refinement by the spatial constraints and their associated information matrix, which is the inverse of the covariance matrix Σ of the registration result. Following (Kerl et al., 2013), we determine a scalar value for the uncertainty in the scan poses based on the entropy $H(T, \Sigma) \propto ln\left(\left|\Sigma\right|\right)$. It allows to select the 3D scan with the largest expected alignment error.

Furthermore, the same measure is used to compare the spatial constraints that have been added to the local map after s_k, to determine if realigning s_k can decrease the alignment error. The selected 3D scan is then refined, by realigning it to its local map, resulting in a refined spatial constraint in the sub-graph. From the sub-graph of spatial constraints, we infer the probability of the trajectory estimate given all relative pose observations

$$p(\mathcal{G}_{\mathcal{M}} \mid \mathcal{E}_{\mathcal{D}}) \propto \prod_{e_{d_{ij}} \in \mathcal{E}_{\mathcal{D}}} p(s_i^j \mid s_i, s_j). \tag{6.1}$$

The sensor trajectory is optimized for each local sub-graph independently. Results from sub-graph optimization are later incorporated when optimizing the allocentric pose graph.

6.2.2. Local Window Alignment

Registration errors are often originated from missing information in the map, e.g., due to occlusions or unknown parts of the environment. Thus, registration quality can only increase if the map has been extended with measurements that provide previously unknown information. In other words, realigning a 3D scans can only increase the map quality if more scans—in best case from different view poses—have been added to the map. Therefore, we increase the local optimization window by adding 3D scans to a local map from neighboring map nodes in the higher level. For example, when the robot revisits a known part of the environment, loop closure is performed and scan nodes from neighboring map nodes are added to a local map.

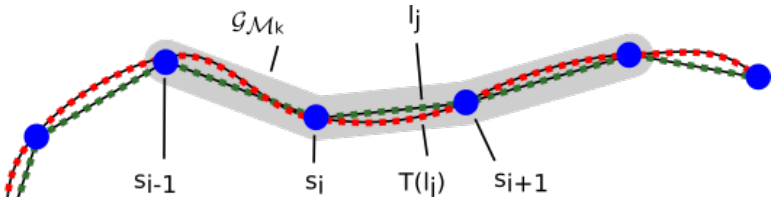


Figure 6.4.: Scan line poses (green squares) originated from odometry measures are refined (red squares) by interpolating with a continuous-time trajectory representation built from scan poses (blue dots) in a local sub-graph (gray).

6.2.3. Continuous-Time Trajectory Representation

Acquiring 3D laser scans often involves mechanical actuation, such as rotating a mirror or a diode/receiver array, during acquisition of the scan. Especially for 3D laser scans—where the acquisition of measurements for a full scan can take multiple hundred milliseconds or seconds—a discretization of the sensor pose to the time where the scan was acquired, leads to artifacts degrading the map quality. However, since a finer discretization of the scan poses makes the state size intractable, temporal basis functions have been used to represent the sensor trajectory (Furgale et al., 2012).

We represent the trajectory of the sensor as cubic B-spline in $SE(3)$ due to their smoothness and the local support property. The local support property allows to interpolate the trajectory from the discrete scan nodes in our local sub-graph. Following (Lovegrove et al., 2013), we parameterize a trajectory by cumulative basis functions using the Lie algebra.

To estimate the trajectory spline, we use the scan nodes $s_0, \ldots, s_m$ with the acquisition times $t_{s_0}, \ldots, t_{s_m}$ as control points for the trajectory spline and denote the pose of a scan node s_i as T_{s_i}. In our system, scan poses follow a uniform temporal distribution. In other words, the difference between the acquisition times of consecutive scans can be assumed to be constant.

As illustrated in Figure 6.4, we use 4 control points to interpolate the sensor trajectory between two scan nodes s_i and s_{i+1}. For time $t \in \left[t_{s_i}, t_{s_{i+1}}\right)$ the pose along the spline is defined as

$$T(u(t)) = T_{s_{i-1}} \prod_{j=1}^{3} exp\big(\widetilde{B}_j(u(t))\Omega_{i+j}\big). \tag{6.2}$$

Here, $\widetilde{B}$ is the cumulative basis, Ω is the logarithmic map, and $u(t) \in [0, 1)$ transforms time t in a uniform time (Lovegrove et al., 2013). Finally, the spline trajec-

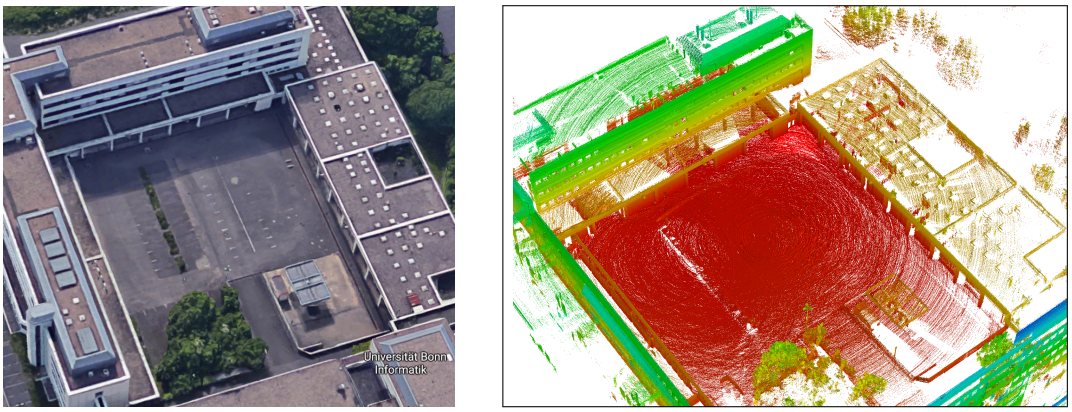


Figure 6.5.: The resulting 3D map from an out/in-door environment. Color encodes height from the ground.

tory is used to update the scan line poses between two scan nodes.

6.2.4. Loop-Closure and Global Optimization

After a new local map has been added to pose graph, we check for one new constraint between the current reference m_{ref} and other map nodes m_{cmp}. We determine a probability

$$p_{\text{chk}}(v_{\text{cmp}}) = \mathcal{N}\left(d(m_{\text{ref}}, m_{\text{cmp}}); 0, \sigma_d^2\right) \tag{6.3}$$

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

When a new spatial constraint has been added, the pose graph is optimized globally on the highest level. When the optimization modifies the estimate of a map node, the changes are propagated to the sub-graph. Similarly, when a sub-graph changed significantly, global optimization of the highest level is carried out.

6.3. Experiments

We assess the accuracy of our refinement method on two different data sets with different sensor setups. The first data has been recorded with a MAV equipped with a Velodyne VLP-16 lidar sensor. The second data set has been recorded in the Deutsches Museum in Munich and is provided by the Google Cartographer team (Hess et al., 2016). Throughout the experiments, we use a distance threshold

of 5 m for adding new map nodes to the graph.

To measure map quality, we calculate the *mean map entropy* (MME) (Droeschel et al., 2014a) from the points $\mathcal{Q} = \{q_1, \ldots, q_Q\}$ of the resulting map. The entropy h for a map point q_k is calculated by

$$h(q_k) = \frac{1}{2} \ln \ |2\pi e \, \Sigma(q_k)|, \tag{6.4}$$

where $\Sigma(q_k)$ is the sample covariance of mapped points in a local radius r around q_k. We select $r = 0.5\,\text{m}$ in our evaluation. The mean map entropy $H(\mathcal{Q})$ is averaged over all points of the resulting map

$$H(\mathcal{Q}) = \frac{1}{Q} \sum_{k=1}^{Q} h(q_k). \tag{6.5}$$

It represents the *crispness* or *sharpness* of a map. Lower entropy measures correspond to higher map quality.

To examine the improvement of the map quality and the convergence behavior of our method, we first run the experiments without online refinement and perform the proposed refinement as a post-processing step. In each iteration, we refine one scan in every sub-graph and run local graph optimization. Local sub-graphs are refined in parallel and after refining all sub-graphs, global optimization is performed. To assess the number of iterations necessary for refinement, entropy measurements are plotted against the number of iterations. Afterwards, we run the proposed method with online refinement and compare the resulting map quality.

Evaluation was carried out on an Intel® Core™ i7-6700HQ quadcore CPU running at 2.6 GHz and 32 GB of RAM. For the reported runtime, we average over 10 runs for each data set.

6.3.1. Courtyard

The first data set has been recorded by a MAV during flight in a building courtyard. The MAV in this experiment is a DJI Matrice 600, equipped with a Velodyne VLP-16 lidar sensor and an IMU, measuring the attitude of the robot. The Velodyne lidar measures $\approx$ 300,000 range measurements per second in 16 horizontal scan rings, has a vertical field of view of 30° and a maximum range of 100 m.

It measures the environment with 16 emitter/detector pairs mounted on an array at different elevation angles from the horizontal plane of the sensor. The array is continuously rotated with up to 1200 rpm. In our experiments, a scan line corresponds to one data packet received by the sensor, i.e., 24 so-called firing

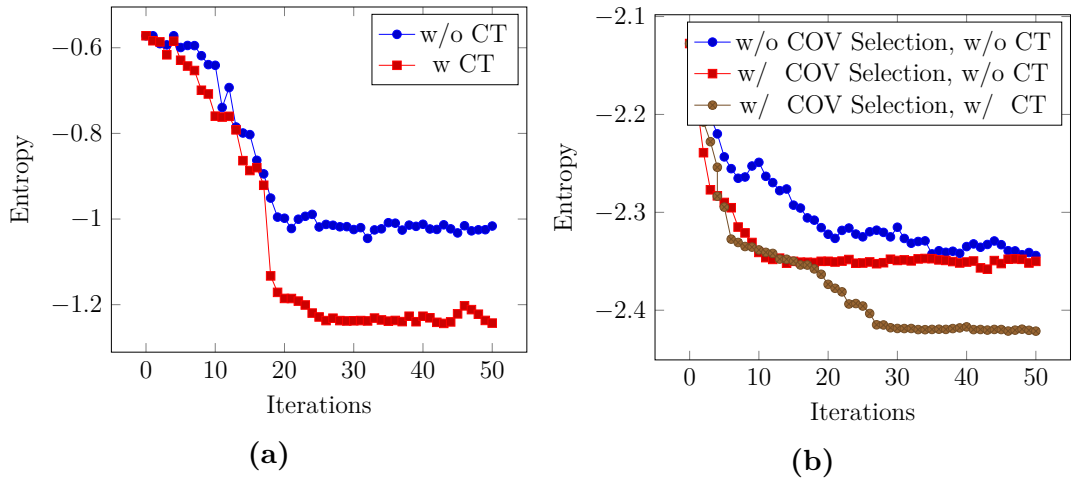


Figure 6.6.: a) The resulting map entropy for the courtyard data set with and without continuous-time trajectory interpolation (CT) Section 6.2.3. b) The resulting map entropy for the Deutsches Museum data set with and without our covariance based scan selection (COV Selection) described in Section 6.2.1 and continuous-time trajectory interpolation (CT). Scan selection leads to faster convergence of our method.

sequences. During one firing sequence (1,33 ms) all 16 emitter/detector pairs are processed.

In total, 2000 scans were recorded during 200 s flight time. Controlled by a human operator, the MAV traversed a building front in different heights. The resulting graph consists of 16 map nodes with several loop closures, resulting in 27 edges between map nodes.

Figure 6.5 shows the environment and a resulting map. In a first experiment, we compare the method from our previous work with the proposed method. The resulting point clouds are shown in Figure 6.7. The figure shows, that the proposed method corrects misaligned 3D scans and increases the map quality. Figure 6.6a shows the resulting entropy plotted against the number of iterations when running the refinement as post-processing step. During one iteration, a single 3D scan in each map node is refined. We measure an average runtime of 54 ms per iteration for refining a single map node and 380 ms per iteration for refining all 16 map nodes in parallel.

6.3.2. Deutsches Museum

For further evaluation, we compare our method on a data set that has been recorded at the Deutsches Museum in Munich. The data set is provided by Hess et al. (Hess et al., 2016). Two Velodyne VLP-16 mounted on a backpack are carried

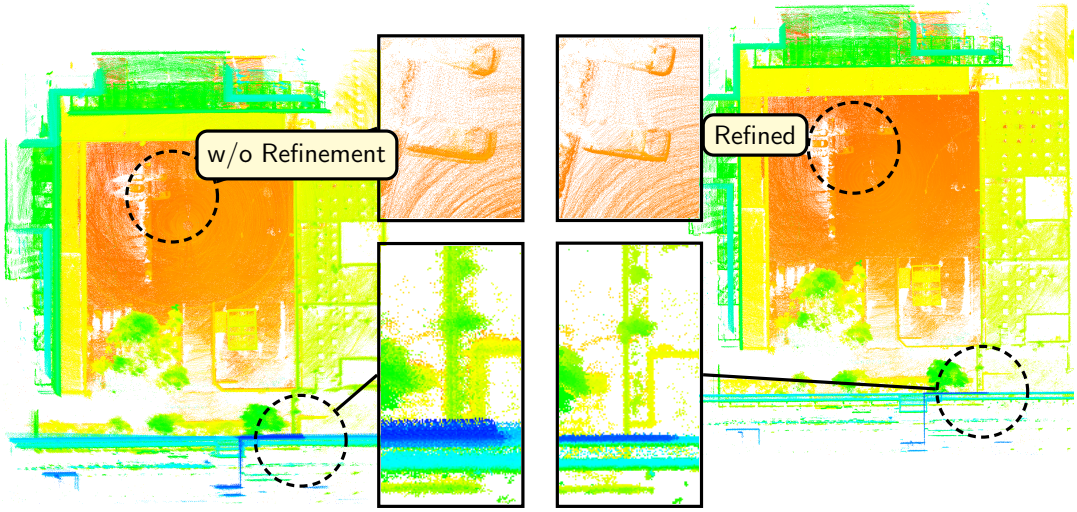


Figure 6.7.: Resulting point clouds from the Courtyard data set. Left: results from the presented method without refinement. Right: results from the presented method. Color encodes the height.

through the museum. Parts of the data set contain dynamic objects, such as moving persons. The provided data set includes a calibration between the two laser scanners—one mounted horizontal, one vertical. We use the provided calibration between the two sensors as initial calibration guess and refine it by adding additional constraints to our pose graph and the local sub-graphs. Similar to the alignment of two local maps, we refine the calibration parameters by our surfel-based method, registering the scans of the graph from the horizontal scanner to the scans of the graph from the vertical scanner.

Following (Nüchter et al., 2017), we select a part of the data set and run our method on it. Besides visual inspection of the resulting point cloud we compute the entropy as described before. Figure 6.6b shows the convergence behavior of our method with and without the covariance-based scan selection. It indicates that our covariance-based scan selection leads to faster convergence. Furthermore, we compare the presented method with the method from our previous work. We also compare our method to Google's Cartographer (Hess et al., 2016) and the continuous-time slam method from (Elseberg et al., 2013) that has been evaluated in (Nüchter et al., 2017). We summarize our results for each method in Table 6.1. The resulting point clouds from the three different floors are shown in Figure 6.9.

Method	MME
Cartographer (Hess et al., 2016)	-2.04
Droeschel et al. (Droeschel et al., 2017)	-2.12
Nüchter et al. (Nüchter et al., 2017)	-2.34
Ours	-2.42

Table 6.1.: Resulting best mean map entropies (MME) for the data set recorded at Deutsches Museum.

6.4. Summary

This chapter presents an hierarchical, continuous-time approach for laser-based 3D SLAM. The method is based on efficient local mapping and a hierarchical optimization back-end. Measurements from a 3D laser scanner are aggregated in local multiresolution maps, by means of surfel based registration. The local maps are used in a graph-based structure for allocentric mapping. The individual 3D scans in the local map model a sub-graph to incorporate corrections when refining these sub-graphs. Graph optimization is performed to account for drift and misalignments in the local maps. Furthermore, a continuous-time trajectory representation allows to interpolate measurements between discrete scan poses. Evaluation shows that our approach increases map quality and leads to sharper maps. In summary, the contribution of the chapter is a novel combination of a hierarchical graph structure—allowing for scalability and efficiency—with local multiresolution maps to overcome alignment problems due to sparsity in laser measurements, and a continuous-time trajectory representation.

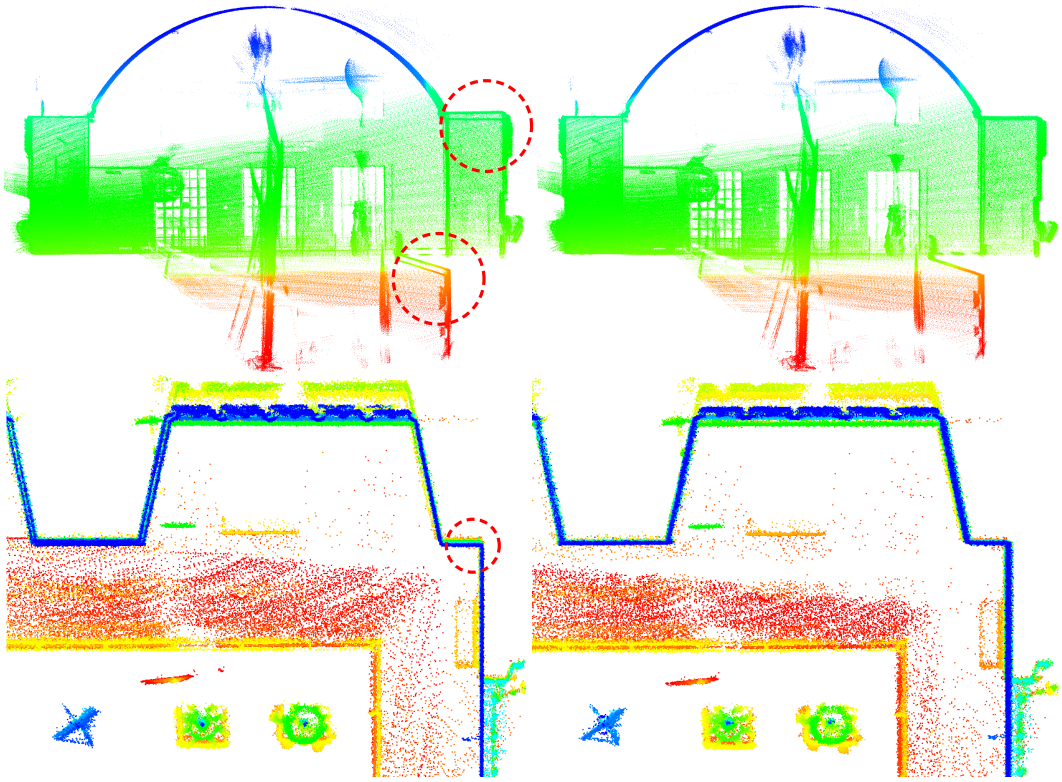


Figure 6.8.: Resulting point clouds from a part of the trajectory of the Deutsches Museum data set. Left: results from (Nüchter et al., 2017). Right: results from our method. Red (dashed) circles highlight distorted parts of the map. Color encodes the height.

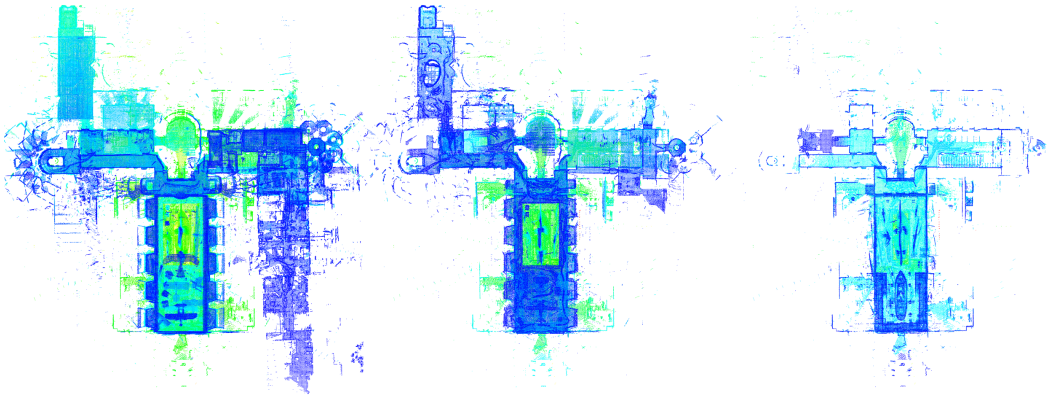


Figure 6.9.: Resulting point clouds from the complete trajectory of the Deutsches Museum data set. The three point clouds show the floors of the resulting allocentric map. Color encodes the height.

7. Conclusion and Outlook

Autonomous navigation requires perceptual capabilities that are self-evident for humans. These capabilities include the ability to build an internal representation of the environment from perceptual inputs. This internal representation, called the map, is used to infer about location from new perceptual inputs and to plan paths. When entering an unknown environment, we simultaneously build this representation and localize within. In robotics this problem is known as simultaneous localization and mapping (SLAM). SLAM has been studied in the robotics community for decades, aiming to allow robots to navigate autonomously in scenarios where a map is not available beforehand. Consequently, a number of approaches addressing the SLAM problem exist (Cadena et al., 2016). However, SLAM is an extensive problem and it depends on a number of aspects, such as the type of the robot, the configuration of the environment, and the desired performance requirements. Thus, depending on the specific aspects, research in this field deserves more or attention or is more mature. For instance, for a wheeled robot platform, equipped with a 2D laser range finder and wheel encoders, moving on a planar indoor environment the SLAM problem is sometimes considered as solved and reliable approaches exist. In contrast, SLAM on a highly dynamic platform— such as a MAV— with limit computing resources in a cluttered 3D environment is still actively discussed in the research community. Here, especially considering the vast amount of data today's sensors produce, efficiency, scalability and robustness are major limiting factors. For example, a modern lidar sensor produces up to 1 million 3D measurements per second that need to be processed with a certain performance, to allow for consistent mapping and robust localization.

Often the huge amount of data and the aforementioned limitations are addressed by relaxing the problem. For example, by modeling the environment in lower dimensionality, such as projecting 3D measurements in a 2D representation. Furthermore, various simplification to the geometry of data and environment, by considering only significant parts of the data or modeling the environment by compressing geometry. On the contrary, these relaxations require assumptions about the environment, for example supposing that geometry in the environment is either planar or has a certain structure, which can not be assumed in general. Similarly, modeling the environment in lower dimensionality makes certain assumptions about

the environment which are not feasible in many situations, such as representing the environment in a 2D map for overhanging structures, or robots that move on non-planar trajectories.

This thesis presents an approach to SLAM that differs from the previously mentioned approaches. Based on the idea, that a proper representation of the geometry in the environment is key, a map representation is proposed to facilitate online mapping and localization.

Central to our approach are so-called local multiresolution grid maps, which is a concise map representation, allowing for efficient aggregation of sensor measurements in a dense map, modeling the geometry of the environment. Local multiresolution grid maps partition the environment in 3D voxels with varying resolution. Local multiresolution hereby refers to the property of having a fine resolution close to the robot and a coarser resolution farther away. This inherent discretization correlates with the characteristics in relative distance accuracy and measurement density of modern 3D lidars and reduces memory and computational requirements without loosing information. Furthermore, local multiresolution grid maps are robot-centric maps that *move* with the robot resulting in a constant number of grid cells necessary to model the robots surrounding, independent of the traveled distance. The latter allows to accomplish constant memory consumption and computation time. Chapter 3 details local multiresolution grid maps as the central data structure in this thesis.

However, to aggregate sensor measurements in local multiresolution grid maps—and thereby building a dense representation of the geometry in the environment—registration of newly acquired sensor data is key. Registration means to estimate the sensor motion between the current sensor measurement and the map by aligning them. For this alignment, a probabilistic registration method is presented in Chapter 4, leveraging the local multiresolution property and allowing for efficient and accurate registration. It allows to align new sensor data to a local multiresolution map, but also for the alignment of two maps with each other.

The latter allows to generate spatial constraints between two maps from different view poses. These spatial constraints and the local maps from different view poses are used to model the environment in a graph-based structure. When the robot revisits a part of the environment, a spatial constraint between the current local map and the previously acquired map allows to compensate for accumulated inaccuracies in the motion estimate (called drift). The graph is optimized in order to minimize the accumulated error. The resulting pose graph maps larger environments and provides a allocentric frame of reference for localization. Registration of scans to the local map—and thereby estimating the sensor's motion with respect to a local frame—is called the *front-end*. In contrast, construction of the

pose graph, its optimization and the alignment of the current local map to it, is called the *back-end*. The proposed back-end using local multiresolution maps is described in Chapter 5.

Inaccuracies, caused by wrong data associations or missing information, may remain. For example, incrementally mapping the environment necessitates bootstrapping from sparse sensor data at the beginning—resulting in relatively poor registration accuracy, compared to aligning with a dense and accurate map. Consequently, map quality degrades due to misaligned sensor data. To overcome degradation of the map quality, Chapter 6 proposes an approach for reassessing the registration of previously added 3D scans. By modeling individual 3D scans of a local map as a sub-graph, we build a hierarchical graph structure, enabling refinement of the map in case misaligned measurements when more information is available. Furthermore, the sensor trajectory is modeled by a continuous-time representation, allowing to incorporate refinement results on on the finest-possible granularity.

Extensive evaluation on different datasets, with different robotic platforms and sensor setups, has been carried out to assess the versatility, robustness, efficiency and accuracy of the approach. The experiments demonstrate that the approach allows for efficient—in both memory and computation—and accurate reconstruction of the environment. Furthermore, it allows to localize the robot in the environment

7.1. Future Directions

The presented approach is based purely on geometry. While this is sufficient in most environments, it might pose a challenge in scenes without sufficient 3D structure. Here additional information, such as intensity from the lidar measurements or color from additional camera sensors might be beneficial. Incorporating texture into the approach mainly involves adding the information to surfels and incorporating this information in the registration method. Incorporating texture from color cameras involves calibration, synchronization, and occlusion handling. In contrast, utilizing the intensity information of the LRF, which corresponds to the reflectance of the measured surface is straightforward since it is already associated with the depth measurement. For example, the intensity information could leveraged for registration by introducing an additional term in the objective function, which corresponds to the similarity of the intensity.

The local multiresolution maps as described, aggregate the individual 3D measurements for each grid cell along with the surfel that summarizes them. The main reason for storing 3D points in each cell is the ability to regenerate surfels

from past measurements when the map has been translated and new grid cells are initialized from the coarser levels. Furthermore, the 3D measurements are used for visualization. Increasing memory efficiency can be accomplished by reducing the number of points stored for each cell, for instance by downsampling. Furthermore, new surfels on the finer levels might be generated from surfels of the coarser levels directly instead of the points they summarize. One way to accomplishing this could be to project the surfel information in the finer grid cell or to sample points in the surfel volume.

Loop-closure is one of the most important parts of a SLAM approach, since it allows to reduce accumulated drift when revisiting known parts of the environment. Currently, edges between the actual key frame and key frames from previously visited parts of the environment are added only based on distance. While this is sufficient for manageable amounts of drift (see experiments) larger environments might necessitate methods for place recognition. This would allow to add spatial constraints between key poses not only based on distance, but also based on their appearance or texture. Here, surfels can be used for efficient computation of feature descriptors.

Dynamic objects are currently addressed by modeling occupancy in each grid cell and adapting the occupancy when new measurements are incorporated. This adaptation is addressed by ray casting in the grid cells. Another source of information for dynamics in the environment can come from data association during registration. Associations that do not comply with the estimated motion from the majority of associations indicate dynamics. In contrast to that, implicit modeling of dynamic objects might improve accuracy and allows a variety of tracking approaches to be incorporated. Besides modeling dynamics in the map, the information can also be leveraged in the data association step during registration. Neglecting surfels that are likely to be part of a dynamic object, improves robustness.

A. Acronyms

MAV micro aerial vehicle

MVOG multi-volume occupancy grid

SLAM simultaneous localization and mapping

GNSS global navigation satellite systems

GPS global positioning system

FoV field-of-view

MoCap motion capture

IMU inertial measurement unit

ICP iterative closest points (Besl and McKay, 1992)

GICP Generalised-ICP by Segal et al. (2009)

NDT Normal-Distribution Transform by Biber and Strasser (2003)

3D-NDT 3D Normal-Distribution Transform by Magnusson et al. (2007)
by Arun et al. (1987)

EM Expectation Maximization by Dempster et al. (1977)

voxel volume pixel

surfel surface element

ATE absolute trajectory error

MME mean map entropy

LIDAR light detection and ranging

LRF laser range finder

DRC DARPA Robotics Challengehttp://www.theroboticschallenge.org/

ATE absolute trajectory errorSturm et al. (2012)

Bibliography

Alismail, H., L. D. Baker, and B. Browning (2014). "Continuous trajectory estimation for 3D SLAM from actuated lidar". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*. IEEE, pp. 6096–6101 (cit. on p. 102).

Amanatides, J. and A. Woo (1987). "A Fast Voxel Traversal Algorithm for Ray Tracing". In: *In Eurographics '87*, pp. 3–10 (cit. on p. 44).

Anderson, S. and T. D. Barfoot (2013). "Towards relative continuous-time SLAM". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)* (cit. on pp. 77, 101).

Anderson, S., T. D. Barfoot, C. H. Tong, and S. Särkkä (2015). "Batch Nonlinear Continuous-Time Trajectory Estimation as Exactly Sparse Gaussian Process Regression". In: *Auton. Robots* 39, pp. 221–238 (cit. on pp. 101, 102).

Arun, K. S., T. S. Huang, and S. D. Blostein (1987). "Least-squares fitting of two 3-D point sets". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 5, pp. 698–700 (cit. on pp. 55, 117).

Bachrach, A., R. He, and N. Roy (2009). "Autonomous flight in unstructured and unknown indoor environments". In: *European Micro Aerial Vehicle Conf (EMAV)*, pp. 1–8 (cit. on pp. 22, 77).

Belter, D., P. Łabcki, and P. Skrzypczyński (2012). "Estimating terrain elevation maps from sparse and uncertain multi-sensor data". In: *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*. IEEE, pp. 715–722 (cit. on pp. 37, 78).

Bergstra, J., D. Yamins, and D. D. Cox (2013). "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures". In: (cit. on p. 65).

Bergstra, J. S., R. Bardenet, Y. Bengio, and B. Kégl (2011). "Algorithms for hyper-parameter optimization". In: *Advances in Neural Information Processing Systems*, pp. 2546–2554 (cit. on p. 65).

Besl, P. J. and N. D. McKay (1992). "A Method for Registration of 3-D Shapes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 14.2, pp. 239–256 (cit. on pp. 54, 117).

Beul, M., D. Droeschel, M. Nieuwenhuisen, J. Quenzel, S. Houben, and S. Behnke (2018). "Fast Autonomous Flight in Warehouses for Inventory Applications". In: *IEEE Robotics and Automation Letters* 3.4, pp. 3121–3128 (cit. on p. 91).

Biber, P. and W. Strasser (2003). "The normal distributions transform: a new approach to laser scan matching". In: *Proceedings 2003 IEEE/RSJ International*

Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453). Vol. 3, 2743–2748 vol.3 (cit. on pp. 37, 42, 55, 117).

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. (cit. on p. 58).

Böhm, J. and S. Becker (2007). "Automatic marker-free registration of terrestrial laser scans using reflectance". In: *Proceedings of the Proceedings of 8th Conference on Optical 3D Measurement Techniques, Zurich, Switzerland*, pp. 9–12 (cit. on p. 55).

Borrmann, D., J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg (2008). "Globally consistent 3D mapping with scan matching". In: *Robotics and Autonomous Systems* 56.2, pp. 130–142 (cit. on p. 59).

Bosse, M. and R. Zlot (2009). "Continuous 3D scan-matching with a spinning 2D laser". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)* (cit. on pp. 77, 101).

Bosse, M., P. Newman, J. Leonard, and S. Teller (2004). "Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework". In: *The International Journal of Robotics Research* 23.12, pp. 1113–1139. eprint: `https://doi.org/10.1177/0278364904049393`. URL: `https://doi.org/10.1177/0278364904049393` (cit. on p. 78).

Brown, B. J. and S. Rusinkiewicz (2007). "Global non-rigid alignment of 3-D scans". In: *ACM Transactions on Graphics (TOG)*. Vol. 26. 3. ACM, p. 21 (cit. on p. 101).

Cadena, C., L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard (2016). "Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age". In: *IEEE Transactions on Robotics* 32.6, pp. 1309–1332 (cit. on pp. 15, 113).

Carlevaris-Bianco, N. and R. M. Eustice (2013). "Generic factor-based node marginalization and edge sparsification for pose-graph SLAM". In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, pp. 5748–5755 (cit. on p. 78).

Censi, A. (Apr. 2007). "An accurate closed-form estimate of ICP's covariance". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, pp. 3167–3172 (cit. on p. 80).

Chan, T. F., G. H. Golub, and R. J. LeVeque (1979). *Updating Formulae and a Pairwise Algorithm for Computing Sample Variances*. Tech. rep. Stanford, CA, USA (cit. on p. 42).

Cole, D. M. and P. M. Newman (2006). "Using laser range data for 3D SLAM in outdoor environments". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* Pp. 1556–1563 (cit. on pp. 36, 99, 101).

Cover, H., S. Choudhury, S. Scherer, and S. Singh (2013). "Sparse Tangential Network (SPARTAN): Motion Planning for Micro Aerial Vehicles". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)* (cit. on p. 77).

Curless, B. and M. Levoy (1996). "A Volumetric Method for Building Complex Models from Range Images". In: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '96. New York, NY, USA: ACM, pp. 303–312. URL: http://doi.acm.org/10.1145/237170.237269 (cit. on p. 37).

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38 (cit. on pp. 56, 58, 117).

Droeschel, D. and S. Behnke (2014). "3D Local Multiresolution Grid for Aggregating Omnidirectional Laser Measurements on a Micro Aerial Vehicle". In: *ISR/Robotik 2014; 41st International Symposium on Robotics*, pp. 1–7 (cit. on p. 20).

– (2018). "Efficient Continuous-time SLAM for 3D Lidar-based Online Mapping". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)* (cit. on p. 19).

Droeschel, D., J. Stückler, and S. Behnke (2014a). "Local Multi-Resolution Representation for 6D Motion Estimation and Mapping with a Continuously Rotating 3D Laser Scanner". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)* (cit. on pp. 20, 108).

Droeschel, D. and S. Behnke (2017). "MRSLaserMap: Local Multiresolution Grids for Efficient 3D Laser Mapping and Localization". In: *RoboCup 2016: Robot World Cup XX*. Ed. by S. Behnke, R. Sheh, S. Sarıel, and D. D. Lee. Cham: Springer International Publishing, pp. 319–326 (cit. on p. 19).

Droeschel, D., J. Stückler, and S. Behnke (2014b). "Local Multi-Resolution Surfel Grids for MAV Motion Estimation and 3D Mapping". In: *Proc. of the Int. Conference on Intelligent Autonomous Systems (IAS)* (cit. on p. 19).

Droeschel, D., D. Holz, and S. Behnke (Oct. 2014c). "Omnidirectional Perception for Lightweight MAVs using a Continuously Rotating 3D Laser". In: *PFG Photogrammetrie, Fernerkundung, Geoinformation* 2014.5, pp. 451–464. URL: http://dx.doi.org/10.1127/1432-8364/2014/0236 (cit. on p. 19).

Droeschel, D., M. Nieuwenhuisen, M. Beul, D. Holz, J. Stückler, and S. Behnke (2015). "Multilayered Mapping and Navigation for Autonomous Micro Aerial Vehicles". In: *Journal of Field Robotics (JFR)* (cit. on pp. 19, 87).

Droeschel, D., M. Schwarz, and S. Behnke (2017). "Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner". In: *Robotics and Autonomous Systems* 88, pp. 104 –115 (cit. on pp. 19, 111).

Dryanovski, I., W. Morris, and J. Xiao (2010). "Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles". In: *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, pp. 1553–1559 (cit. on p. 36).

Einhorn, E., C. Schröter, and H.-M. Gross (2011). "Finding the adequate resolution for grid mapping-cell sizes locally adapting on-the-fly". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1843–1848 (cit. on p. 37).

Elfes, A. (1987). "Sonar-based real-world mapping and navigation". In: *IEEE Journal on Robotics and Automation* 3.3, pp. 249–265 (cit. on pp. 33, 34, 36).

Elseberg, J., D. Borrmann, and A. Nuechter (2012). "6DOF semi-rigid SLAM for mobile scanning". In: *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)* (cit. on pp. 77, 101).

Elseberg, J., D. Borrmann, K. Lingemann, and A. Nüchter (2010). "Non-rigid registration and rectification of 3D laser scans". In: *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1546–1552 (cit. on p. 101).

Elseberg, J., D. Borrmann, and A. Nüchter (2013). "Algorithmic solutions for computing precise maximum likelihood 3D point clouds from mobile laser scanning platforms". In: *Remote Sensing* 5.11, pp. 5871–5906 (cit. on pp. 102, 110).

Evangelidis, G. D., D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis (2014). "A generative model for the joint registration of multiple point sets". In: *European Conference on Computer Vision*. Springer, pp. 109–122 (cit. on p. 56).

Fankhauser, P, M Bloesch, C Gehring, M Hutter, and R Siegwart (2014). "Robot-centric elevation mapping with uncertainty estimates". In: *International Conference on Climbing and Walking Robots (CLAWAR), Poznań, Poland*, pp. 433–440 (cit. on p. 78).

Frese, U., P. Larsson, and T. Duckett (2005). "A multilevel relaxation algorithm for simultaneous localization and mapping". In: *IEEE Transactions on Robotics* 21.2, pp. 196–207 (cit. on p. 100).

Furgale, P., T. D. Barfoot, and G. Sibley (2012). "Continuous-time batch estimation using temporal basis functions". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, pp. 2088–2095 (cit. on p. 106).

Furgale, P., J. Rehder, and R. Siegwart (2013). "Unified temporal and spatial calibration for multi-sensor systems". In: *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1280–1286 (cit. on p. 102).

Grisetti, G., R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg (2010). "Hierarchical optimization on manifolds for online 2D and 3D mapping". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, pp. 273–278 (cit. on p. 102).

Grzonka, S., G. Grisetti, and W. Burgard (2009). "Towards a Navigation System for Autonomous Indoor Flying". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)* (cit. on pp. 22, 77).

– (2012). "A Fully Autonomous Indoor Quadrotor". In: *IEEE Trans. on Robotics* 28.1, pp. 90–100 (cit. on p. 77).

Hebert, M., C. Caillas, E. Krotkov, I. S. Kweon, and T. Kanade (1989). "Terrain Mapping for a Roving Planetary Explorer". In: *Proceedings of the IEEE Inter-*

national Conference on Robotics and Automation (ICRA '89). Vol. 2, pp. 997–1002 (cit. on pp. 36, 78).

Hess, W., D. Kohler, H. Rapp, and D. Andor (2016). "Real-time loop closure in 2D LIDAR SLAM". In: *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, pp. 1271–1278 (cit. on pp. 78, 102, 107, 109–111).

Holz, D. and S. Behnke (2014). "Mapping with Micro Aerial Vehicles by Registration of Sparse 3D Laser Scans". In: *Proc. of the Int. Conference on Intelligent Autonomous Systems (IAS)* (cit. on p. 68).

Holz, D. and S. Behnke (Dec. 2015). "Registration of Non-uniform Density 3D Laser Scans for Mapping with Micro Aerial Vehicles". In: *Robot. Auton. Syst.* 74.PB, pp. 318–330 (cit. on p. 55).

Horaud, R., F. Forbes, M. Yguel, G. Dewaele, and J. Zhang (2011). "Rigid and articulated point registration with expectation conditional maximization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.3, pp. 587–602 (cit. on p. 56).

Hornung, A., K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard (2013). "OctoMap: an efficient probabilistic 3D mapping framework based on octrees". In: *Autonomous Robots* 34, pp. 189–206 (cit. on pp. 37, 46, 48, 50).

Huh, S., D. Shim, and J. Kim (2013). "Integrated navigation system using camera and gimbaled laser scanner for indoor and outdoor autonomous flight of UAVs". In: *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)* (cit. on p. 77).

Jian, B. and B. C. Vemuri (2011). "Robust point set registration using gaussian mixture models". In: *IEEE transactions on pattern analysis and machine intelligence* 33.8, pp. 1633–1645 (cit. on p. 56).

Johnson, A. E. and M. Hebert (1999). "Using spin images for efficient object recognition in cluttered 3D scenes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.5, pp. 433–449 (cit. on p. 55).

Kaul, L., R. Zlot, and M. Bosse (2016). "Continuous-Time Three-Dimensional Mapping for Micro Aerial Vehicles with a Passively Actuated Rotating Laser Scanner". In: *Journal of Field Robotics* 33.1, pp. 103–132 (cit. on p. 102).

Kerl, C., J. Sturm, and D. Cremers (2013). "Dense visual SLAM for RGB-D cameras". In: *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2100–2106 (cit. on p. 105).

Kerl, C., J. Stückler, and D. Cremers (2015). "Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 2264–2272 (cit. on p. 102).

Khan, S., D. Wollherr, and M. Buss (2015). "Adaptive rectangular cuboids for 3D mapping". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, pp. 2132–2139 (cit. on p. 37).

Kohlbrecher, S., O. von Stryk, J. Meyer, and U. Klingauf (2011). "A flexible and scalable SLAM system with full 3D motion estimation". In: *Safety, Security,*

and Rescue Robotics (SSRR), IEEE International Symposium on, pp. 155–160 (cit. on pp. 99, 101).

Konolige, K., G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent (2010). "Efficient Sparse Pose Adjustment for 2D mapping". In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 22–29 (cit. on p. 78).

Kretzschmar, H., C. Stachniss, and G. Grisetti (2011). "Efficient information-theoretic graph pruning for graph-based SLAM with laser range finders". In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, pp. 865–871 (cit. on p. 77).

Kschischang, F. R., B. J. Frey, and H. A. Loeliger (Sept. 2006). "Factor Graphs and the Sum-product Algorithm". In: *IEEE Trans. Inf. Theor.* 47.2, pp. 498–519. URL: http://dx.doi.org/10.1109/18.910572 (cit. on p. 77).

Kuemmerle, R., G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard (2011). "G2o: A General Framework for Graph Optimization". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)* (cit. on pp. 79, 81, 100, 104).

Lenac, K., A. Kitanov, R. Cupec, and I. Petrovi (2017). "Fast planar surface 3D {SLAM} using {LIDAR}". In: *Robotics and Autonomous Systems* 92, pp. 197 –220 (cit. on p. 78).

Lovegrove, S., A. Patron-Perez, and G. Sibley (2013). "Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras". In: *British Machine Vision Conference* (cit. on p. 106).

Lu, F. and E. Milios (1997). "Globally consistent range scan alignment for environment mapping". In: *Autonomous robots* 4.4, pp. 333–349 (cit. on pp. 59, 77).

Maddern, W., A. Harrison, and P. Newman (2012). "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)* (cit. on pp. 77, 101).

Magnusson, M., T. Duckett, and A. J. Lilienthal (2007). "Scan Registration for Autonomous Mining Vehicles Using 3D-NDT". In: *Journal of Field Robotics* 24.10, pp. 803–827 (cit. on pp. 37, 42, 55, 56, 65, 77, 99, 101, 117).

Magnusson, M., A. Nüchter, C. Lörken, A. J. Lilienthal, and J. Hertzberg (2009). "Evaluation of 3D Registration Reliability and Speed — A Comparison of ICP and NDT". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, pp. 3907–3912 (cit. on pp. 59, 69, 70).

Mateo, X., X. Orriols, and X. Binefa (2014). "Bayesian perspective for the registration of multiple 3d views". In: *Computer Vision and Image Understanding* 118, pp. 84–96 (cit. on p. 56).

Meagher, D. (1982). "Geometric modeling using octree encoding". In: *Computer graphics and image processing* 19.2, pp. 129–147 (cit. on p. 37).

Moravec, H. P. (1988). "Sensor fusion in certainty grids for mobile robots". In: *AI magazine* 9.2, p. 61 (cit. on pp. 36, 44).

Mueggler, E., G. Gallego, H. Rebecq, and D. Scaramuzza (2017). "Continuous-Time Visual-Inertial Trajectory Estimation with Event Cameras". In: *arXiv preprint arXiv:1702.07389* (cit. on p. 102).

Myronenko, A. and X. Song (2010). "Point Set Registration: Coherent Point Drift". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12, pp. 2262–2275 (cit. on pp. 56, 57).

Newcombe, R. A., S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon (2011). "KinectFusion: Real-time dense surface mapping and tracking". In: *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136 (cit. on p. 38).

Ni, K. and F. Dellaert (2010). "Multi-level submap based slam using nested dissection". In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, pp. 2558–2565 (cit. on p. 78).

Nieuwenhuisen, M., D. Droeschel, M. Beul, and S. Behnke (2016). "Autonomous navigation for micro aerial vehicles in complex GNSS-denied environments". In: *Journal of Intelligent & Robotic Systems* 84.1-4, pp. 199–216 (cit. on p. 89).

Nüchter, A., K. Lingemann, J. Hertzberg, and H. Surmann (2005). "6D SLAM with approximate data association". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, pp. 242–249 (cit. on pp. 55, 77, 99, 101).

Nüchter, A., K. Lingemann, and J. Hertzberg (2007a). "Cached k-d tree search for ICP algorithms". In: *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*, pp. 419–426 (cit. on p. 55).

Nüchter, A, M Bleier, J Schauer, and P Janotta (2017). "Improving Google's Cartographer 3D Mapping by Continuous-Time Slam". In: *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 543–549 (cit. on pp. 102, 110–112).

Nüchter, A., K. Lingemann, J. Hertzberg, and H. Surmann (2007b). "6D SLAM—3D mapping outdoor environments". In: *Journal of Field Robotics* 24.8-9, pp. 699–722 (cit. on pp. 36, 59).

Olson, E., J. Leonard, and S. Teller (2006). "Fast iterative alignment of pose graphs with poor initial estimates". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2262–2269 (cit. on p. 100).

Patron-Perez, A., S. Lovegrove, and G. Sibley (2015). "A spline-based trajectory representation for sensor fusion and rolling shutter cameras". In: *International Journal of Computer Vision* 113.3, pp. 208–219 (cit. on p. 102).

Payeur, P., P. Hébert, D. Laurendeau, and C. M. Gosselin (1997). "Probabilistic octree modeling of a 3d dynamic environment". In: *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*. Vol. 2. IEEE, pp. 1289–1296 (cit. on p. 37).

Pfaff, P., R. Triebel, and W. Burgard (2007). "An efficient extension to elevation maps for outdoor terrain mapping and loop closing". In: *The International Journal of Robotics Research* 26.2, pp. 217–230 (cit. on p. 78).

Pomerleau, F., F. Colas, R. Siegwart, and S. Magnenat (Apr. 2013a). "Comparing ICP Variants on Real-world Data Sets". In: *Auton. Robots* 34.3, pp. 133–148. URL: http://dx.doi.org/10.1007/s10514-013-9327-2 (cit. on p. 55).

Pomerleau, F., F. Colas, and R. Siegwart (2013b). "A Review of Point Cloud Registration Algorithms for Mobile Robotics". In: *Foundations and Trends in Robotics* 4.1, pp. 1–104 (cit. on p. 55).

Rocha, R., J. Dias, and A. Carvalho (2005). "Cooperative multi-robot systems:: A study of vision-based 3-d mapping using information theory". In: *Robotics and Autonomous Systems* 53.3, pp. 282–311 (cit. on p. 36).

Rodehutskors, T., M. Schwarz, and S. Behnke (2015). "Intuitive Bimanual Telemanipulation under Communication Restrictions by Immersive 3D Visualization and Motion Tracking". In: (cit. on pp. 27, 93).

Ruhnke, M., R. Kümmerle, G. Grisetti, and W. Burgard (2012). "Highly accurate 3D surface models by sparse surface adjustment". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*. IEEE, pp. 751–757 (cit. on p. 101).

Rusinkiewicz, S. and M. Levoy (2001). "Efficient variants of the ICP algorithm". In: *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, pp. 145–152 (cit. on p. 55).

Rusu, R. B., N. Blodow, and M. Beetz (2009). "Fast Point Feature Histograms (FPFH) for 3D registration". In: *2009 IEEE International Conference on Robotics and Automation*, pp. 3212–3217 (cit. on p. 55).

Ryde, J. and H. Hu (2010). "3D mapping with multi-resolution occupied voxel lists". In: *Autonomous Robots* 28, pp. 169 –185 (cit. on p. 36).

Schadler, M., J. Stückler, and S. Behnke (2013). "Multi-Resolution Surfel Mapping and Real-Time Pose Tracking using a Continuously Rotating 2D Laser Scanner". In: *Proc. of IEEE Int. Symp. on Safety, Security, and Rescue Robotics (SSRR)* (cit. on pp. 37, 56).

Scherer, S., J. Rehder, S. Achar, H. Cover, A. D. Chambers, S. T. Nuske, and S. Singh (2012). "River mapping from a flying robot: State estimation, river detection, and obstacle mapping". In: *Autonomous Robots* 32.5. Ed. by G. Sukhatme, pp. 1–26 (cit. on p. 77).

Schwertfeger, S., A. Jacoff, C. Scrapper, J. Pellenz, and A. Kleiner (2010). "Evaluation of Maps Using Fixed Shapes: The Fiducial Map Metric". In: *10th Performance Metrics for Intelligent Systems Workshop (PerMIS)*. Baltimore, Maryland, pp. 339–346 (cit. on p. 59).

Segal, A., D. Haehnel, and S. Thrun (2009). "Generalized-ICP". In: *Proc. of Robotics: Science and Systems (RSS)* (cit. on pp. 55, 63, 83, 117).

Shen, S., N. Michael, and V. Kumar (2011). "Autonomous multi-floor indoor navigation with a computationally constrained micro aerial vehicle". In: *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, pp. 2968 –2969 (cit. on pp. 22, 77).

Stachniss, C., S. Thrun, and J. J. Leonard (2016). "Simultaneous Localization and Mapping". In: *Springer Handbook of Robotics*. Ed. by B. Siciliano and O. Khatib. 2nd. Springer. Chap. 46, pp. 1153–1176 (cit. on p. 77).

Stachniss, C. and W. Burgard (2003a). "Exploring unknown environments with mobile robots using coverage maps". In: *IJCAI*, pp. 1127–1134 (cit. on p. 36).

– (2003b). "Mapping and exploration with mobile robots using coverage maps". In: *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Vol. 1. IEEE, pp. 467–472 (cit. on p. 36).

Stachniss, C. and H. Kretzschmar (2017). "Pose Graph Compression for Laser-Based SLAM". In: *Robotics Research : The 15th International Symposium ISRR*. Ed. by H. I. Christensen and O. Khatib. Cham: Springer International Publishing, pp. 271–287. URL: http://dx.doi.org/10.1007/978-3-319-29363-9_16 (cit. on p. 77).

Stoyanov, T. and A. Lilienthal (2009). "Maximum likelihood point cloud acquisition from a mobile platform". In: *Proc. of the Int. Conf. on Advanced Robotics (ICAR)* (cit. on pp. 77, 101).

Stoyanov, T., M. Magnusson, H. Andreasson, and A. J. Lilienthal (2012). "Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations". In: *The International Journal of Robotics Research* 31.12, pp. 1377–1393 (cit. on p. 55).

Stückler, J. and S. Behnke (2014). "Multi-resolution surfel maps for efficient dense 3D modeling and tracking". In: *Journal of Visual Communication and Image Representation* 25.1, pp. 137–147 (cit. on pp. 37, 51, 56, 59).

Stueckler, J. (2014). "Efficient Dense Registration, Segmentation, and Modeling Methods for RGB-D Environment Perception". PhD thesis. Faculty of Mathematics and Natural Sciences, University of Bonn, Germany. URL: http://hss.ulb.uni-bonn.de/2014/3852/3852.htm (cit. on pp. 42, 43).

Sturm, J., N. Engelhard, F. Endres, W. Burgard, and D. Cremers (2012). "A Benchmark for the Evaluation of RGB-D SLAM Systems". In: *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)* (cit. on pp. 59, 60, 118).

Takahashi, M., G. Schulein, and M. Whalley (2008). "Flight control law design and development for an autonomous rotorcraft". In: *Proceedings of the 64th Annual Forum of the American Helicopter Society* (cit. on p. 77).

Tamaki, T., M. Abe, B. Raytchev, and K. Kaneda (2010). "Softassign and em-icp on gpu". In: *Networking and Computing (ICNC), 2010 First International Conference on*. IEEE, pp. 179–183 (cit. on p. 56).

Teniente, E. H., R. Valencia, and J. Andrade-Cetto (2011). "Dense outdoor 3D mapping and navigation with Pose SLAM". In: *Proc. III Workshop de Robótica: Robótica Experimental, ROBOT11*. Citeseer, pp. 567–572 (cit. on p. 77).

Thrun, S. et al. (2002). "Robotic mapping: A survey". In: *Exploring artificial intelligence in the new millennium* 1, pp. 1–35 (cit. on p. 33).

Thrun, S., M. Diel, and D. Hähnel (2003). "Scan Alignment and 3-D Surface Modeling with a Helicopter Platform." In: *FSR*. Vol. 24. Springer Tracts in Advanced Robotics. Springer, pp. 287–297 (cit. on p. 77).

Thrun, S., W. Burgard, and D. Fox (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press (cit. on pp. 43, 77).

Tombari, F., S. Salti, and L. Di Stefano (2010). "Unique Signatures of Histograms for Local Surface Description". In: *Computer Vision – ECCV 2010*. Ed. by K. Daniilidis, P. Maragos, and N. Paragios. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 356–369 (cit. on p. 55).

Tomić, T., K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka (2012). "Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue". In: *Robotics Automation Magazine, IEEE* 19.3, pp. 46–56 (cit. on pp. 22, 77).

Triebel, R., P. Pfaff, and W. Burgard (2006). "Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing". In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 2276–2282 (cit. on pp. 36, 78).

Ulaş, C. and H. Temeltaş (2013). "3D multi-layered normal distribution transform for fast and long range scan matching". In: *Journal of Intelligent & Robotic Systems* 71.1, pp. 85–108 (cit. on p. 55).

Wolcott, R. W. and R. M. Eustice (2015). "Fast LIDAR localization using multiresolution Gaussian mixture maps". In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Accepted, To Appear. Seattle, WA, USA (cit. on p. 36).

Wulf, O., A. Nüchter, J. Hertzberg, and B. Wagner (2007). "Ground truth evaluation of large urban 6D SLAM". In: *Proc. of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, pp. 650–657 (cit. on p. 59).

Yguel, M., O. Aycard, and C. Laugier (2008). "Update Policy of Dense Maps: Efficient Algorithms and Sparse Representation". In: *Field and Service Robotics: Results of the 6th International Conference*. Ed. by C. Laugier and R. Siegwart. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 23–33. URL: `https://doi.org/10.1007/978-3-540-75404-6_3` (cit. on p. 46).

Zhao, L., S. Huang, and G. Dissanayake (2013). "Linear SLAM: A linear solution to the feature-based and pose graph SLAM based on submap joining". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 24–30 (cit. on p. 78).

Zlot, R. and M. Bosse (2014). "Efficient Large-scale Three-dimensional Mobile Mapping for Underground Mines". In: *Journal of Field Robotics* 31.5, pp. 758–779 (cit. on pp. 101, 102).