

Unconventional Trajectories for Mobile 3D Scanning and Mapping

Arzberger, Fabian; Zevering, Jasper; Bredenbeck, A.; Borrmann, Dorit ; Nüchter, Andreas

DOI

[10.5772/intechopen.108132](https://doi.org/10.5772/intechopen.108132)

Publication date

2022

Document Version

Final published version

Published in

Unconventional Trajectories for Mobile 3D Scanning and Mapping

Citation (APA)

Arzberger, F., Zevering, J., Bredenbeck, A., Borrmann, D., & Nüchter, A. (2022). Unconventional Trajectories for Mobile 3D Scanning and Mapping. In *Unconventional Trajectories for Mobile 3D Scanning and Mapping* IntechOpen. <https://doi.org/10.5772/intechopen.108132>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,600

Open access books available

177,000

International authors and editors

195M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Unconventional Trajectories for Mobile 3D Scanning and Mapping

Fabian Arzberger, Jasper Zevering, Anton Bredenbeck, Dorit Borrmann and Andreas Nüchter

Abstract

State-of-the-art LiDAR-based 3D scanning and mapping systems focus on scenarios where good sensing coverage is ensured, such as drones, wheeled robots, cars, or backpack-mounted systems. However, in some scenarios more unconventional sensor trajectories come naturally, e.g., rolling, descending, or oscillating back and forth, but the literature on these is relatively sparse. As a result, most implementations developed in the past are not able to solve the SLAM problem in such conditions. In this chapter, we propose a robust offline-batch SLAM system that is able to address more challenging trajectories, which are characterized by weak angles of incidence and limited FOV while scanning. The proposed SLAM system is an upgraded version of our previous work and takes as input the raw points and prior pose estimates, yet the latter are subject to large amounts of drift. Our approach is a two-staged algorithm where in the first stage coarse alignment is fast achieved by matching planar polygons. In the second stage, we utilize a graph-based SLAM algorithm for further refinement. We evaluate the mapping accuracy of the algorithm on our own recorded datasets using high-resolution ground truth maps, which are available from a TLS.

Keywords: 3D LiDAR, mobile mapping, scanning, spherical robot, pendulum, descent, small FOV, Livox, Intel, RealSense

1. Introduction

Mobile systems increasingly gain astonishing capabilities when it comes to 3D sensing, mapping, and environment reconstruction (**Figure 1**). Nowadays, there exist many mobile systems in different shapes and sizes that are able to perform these tasks, e.g., drones, wheeled or tracked robots, or backpack-mounted systems, just to name a few. A key aspect of fulfilling their purpose is the estimation of the systems inertial frame of reference, i.e., the systems local coordinate system, with respect to a global reference frame. This global reference frame has an origin somewhere in the respective environment and is usually initialized with the systems starting position and orientation (pose). By expressing the local measurements in the global frame, the system is able to create a map of the environment whilst localizing itself in it. This process is called simultaneous localization and mapping (SLAM), and only works if the initial pose estimation of the system is sufficiently accurate. The types of robot

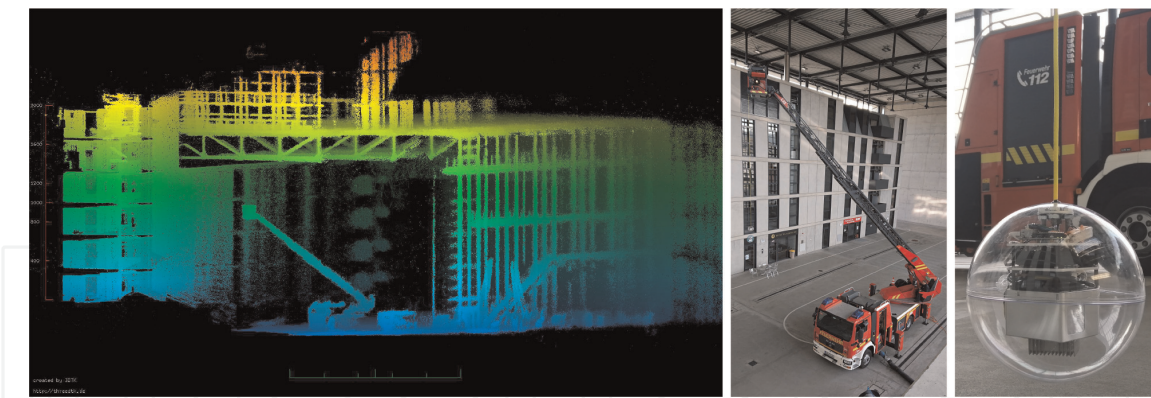
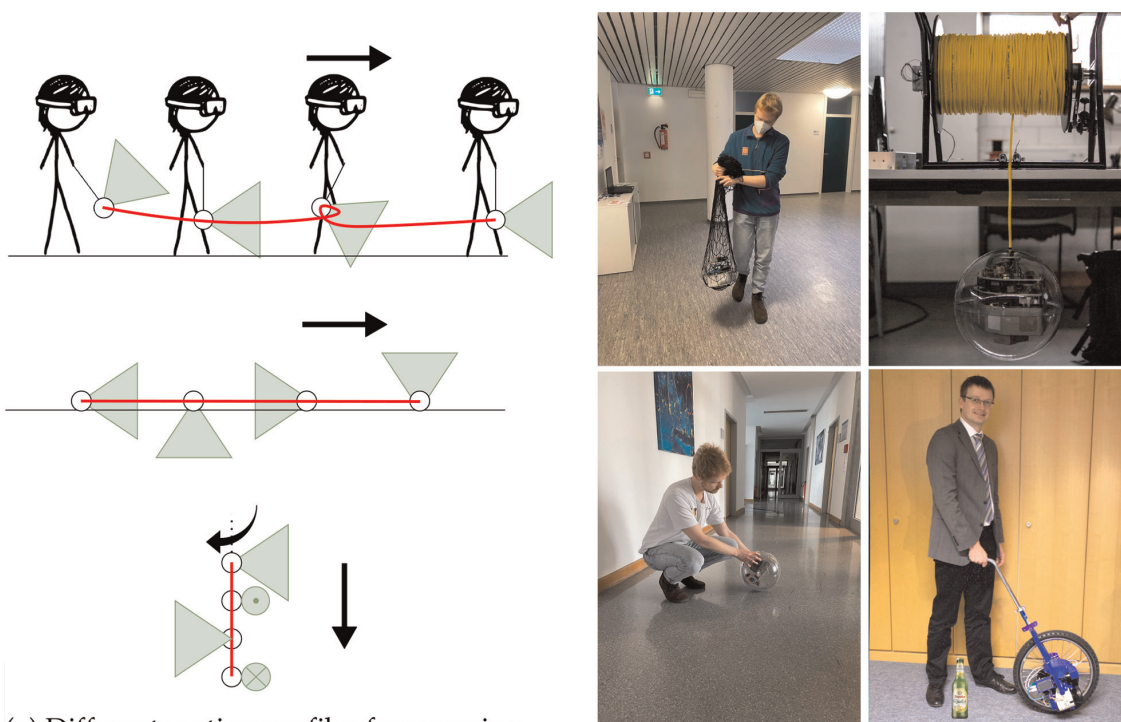


Figure 1.

(Left) post-processed 3D point cloud acquired by a sensor that is rotating freely while being descended with a crane from a fire truck. Prior pose estimates are available from three IMUs and an angular encoder, which encodes the rotation of the cable reel. (Center) the firetruck. (Right) the sensor, which is mounted to the crane of the firetruck.

designs mentioned earlier are often favored since they have access to quite accurate prior pose estimates (GNSS, odometry, visual odometry, etc.) and reliable coverage while sensing the environment with cameras or laser scanners. In such conditions, it is typically easy to perform laser-based SLAM, either online (for autonomous mobile robots) or as a post-processing step. Yet, there are still many situations where conditions are poor, inferring uncertainties to prior pose estimates and thus degrading SLAM performance. Visual feature tracking with a camera, for example, only works in good lighting situations, and GNSS might not be available. Using IMUs as the only pose estimation device usually is also not an option, due to the large accumulation of measurement errors caused by noise and drift which makes position estimation difficult. Even high-end devices, e.g. in aviation systems, must be combined with other references like GNSS in order to be reliable. The ability of LiDAR-based SLAM algorithms to deal with degradation puts constraints on current mobile system designs, as well as their applications. Therefore, extending the capabilities of SLAM algorithms to more unusual scenarios opens up interesting design choices for mobile systems, especially in hazardous environments. The intention of this chapter is to provide a general and robust LiDAR-based solution for the SLAM problem, which is independent with respect to the executed trajectories and sensor setups. We note the utilization of a Livox Mid-40 scanner, which is considered a solid-state LiDAR in [1]. Yet in [2], the family of Livox scanners is only considered to be “semi solid-state” LiDARs, due to their non-repetitive scanning pattern. Solid-state LiDAR got a lot of attention in the past years due to “their superiority in cost, reliability, and [...] performance against the conventional mechanical spinning LiDARs [...]” [3]. While traditional LiDAR is based on electro-mechanic parts which move the sensor head, solid-state LiDAR relies on micro-electro-mechanical systems (MEMS), optical-phase arrays (OPA), or Risley prisms. Despite their potential advantages, solid-state laser-scanners impose new challenges for established SLAM algorithms, e.g., small FOV, an irregular scanning pattern with non-repetitive scanning which makes feature extraction more difficult, and increased motion blur. In this work, we address these challenges by making the assumption that planar structures are available in the environment. Due to the utilization of planes, the envisioned applications are in human-made environments, e.g., old buildings that are in danger of collapsing, narrow underground tunnels, construction sites, or mining shafts. Many mobile mapping systems present in the current literature have access to accurate pose estimates, as well as good sensing coverage. We present four notably interesting experimental setups which do not meet those

conditions. The main sensor in each scenario is a LIVOX MID-100 laser scanner, which is inconvenient due to its limited FOV and unique scanning pattern. **Figure 2** illustrates the different executed motions. (1) We descent a freely rotating 3D sensor from a crane. Prior pose estimates are available from three IMUs and a rotary encoder on the cable reel. (2) Further, we use the sensor as a pendulum, oscillating back and forth while walking. The robot obtains prior pose estimates from a T265 tracking camera, which uses its own internal IMU. (3) Moreover, we roll the sensor around on flat ground. An IMU-only-based approach with three units estimates the pose of the system [4]. The approach combines two popular filters (Madgwick- and Complementary-filter) and estimates the position by relating the rotational velocity and radius of the sphere to its traveled distance. Further constraints are present in the filter to account for slippage and sliding effects of the sphere. (4) Finally, we repeat the previous experiment but substitute the 3D sensor with a 2D sensor, which measures parts of the environment in planar slices. Rotating the planar slice thus results in



(a) Different motion profiles for scanning. The red line shows the position of the robot over time. The green cones represent the scanning direction and angle of the sensor. (Upper) Walking while swinging the pendulum setup. When the pendulum swings back while moving forward it creates the “looping” shape of the trajectory. (Center) Rolling on flat ground. (Lower) Descending while rotating. The green circles symbolize that either the scanner is facing towards us or away from us.

(b) Setup for the experiments. Upper left: Pendulum setup in the hallways of an office environment. We put the sensors inside a trailer net and swing it back and forth while walking. Upper right: Descending setup with a tear-resistant tether cable, which is rolled around a coil. Bottom left: Rolling on flat surfaces setup in the hallways of an office environment. The sensors are placed inside a protective spherical plastic shell. Bottom right: Rolling on flat surfaces setup with the RADLER [4] device.

Figure 2. Illustration of the executed sensor trajectories (left) and images of system setups during operation (right).

a 3D reconstruction of the scene. For pose estimates, we use a rotary encoder and a single IMU. Section 4 describes all experimental setups in more detail. The prior pose estimates are subject to a significant amount of noise and drift. Some scenarios are more difficult than others since the robot locomotion mechanism causes the sensor to move in five, or even all six degrees of freedom (DOF). In this work, we refine our offline-batch SLAM system from [5] to make it more robust, such that it is able to address the unfavorable conditions imposed by more challenging trajectories. Section 3 introduces the two-staged SLAM algorithm, where the first stage uses a polygon-based approach for a fast coarse alignment, and a graph-based method in the second stage for slow further refinement. We evaluate the accuracy of the resulting maps by means of ground truth point clouds, which are available from high precise terrestrial laser scanning (TLS). Note that in this initial study, no trajectory ground truth has been recorded. This is a task for future work. In particular, the contributions of this work are as follows:

- Fixing the open problem mentioned in previous work [5], where after the application of the SLAM algorithm the resulting trajectories were jagged instead of smooth.
- Reducing the number of hyperparameters for our SLAM algorithm [5] by seven, without the sacrifice of flexibility or performance. We achieve this by substituting the iterative optimization using AdaDelta with a closed-form solution based on singular value decomposition (SVD).
- Introducing major technical improvements regarding our SLAM procedure from previous work [5], which increase the robustness of the algorithm. These include the substitution of the local planar clustering (LPC) with a different plane detection framework according to [6], as well as the implementation of a simple global plane model that builds up sequentially as new range measurements arrive.
- Introducing a globally consistent graph-based method, “semi-rigid SLAM,” according to [7] as an additional post-processing step, to further decrease the amount of accumulated registration errors.
- Publishing the challenging datasets themselves, as we encourage readers to try their own SLAM algorithms on them. A more detailed description of the datasets is found in Section 4.

2. Related work

Many state of the art 3D scanning and mapping approaches for mobile systems are based on wheeled robots, drones, or backpack-mounted solutions. On the other hand, the literature regarding mobile mapping systems with more unconventional trajectories is relatively sparse. In the most recent past, our lab has addressed the subject more frequently with the “RADial LasER scanning device” (RADLER) [8], the “Laser-Mapping Unidirectional Navigation Actuator” (L.U.N.A.) [9], and another publication regarding rolling 3D sensors in human-made environments [5]. RADLER is a modified unicycle where a SICK LMS141 2D scanner is fixed to the wheel. As far as we know it was one of the first systems where the same rotation that is used for

locomotion is also utilized to actuate the sensor. L.U.N.A. extends this idea by employing a self-actuated locomotion approach using internal flywheels. Another noticeable trend, despite being only a concept so far, is the European Space Agency's (ESA) interest in spherical robots capable of SLAM, called DAEDALUS [10]. In their Concurrent Design Facility (CDF) study, we developed a mission to autonomously explore underground caves and lava tubes on the moon with DAEDALUS [11], which emphasizes the potential of this type of system design for hazardous environments. More examples of scanning mobile systems with unconventional trajectories include Zebedee [12], a handheld 2D range scanner mounted on a spring that estimates its pose using an IMU, or VILMA [13], an IMU-less rolling system that uses only range measurements for localization. It later advanced into a commercial solution called "ZEB-Revo" [14]. Leica also provides a handheld sensor for mobile mapping purposes with their BLK2GO [15]. Alismail and Browning [16] provide a marker-less calibration procedure for spinning actuated laser scanners, where the extrinsic parameters with respect to the spinning axis are estimated. In this initial study, we go without fine calibration of extrinsic as the constant calibration errors are less significant than the errors introduced by the factors stated in the previous section. In terms of laser-based SLAM, many algorithms for 6 DOF are available, often based on the well-known Iterative-Closest-Point (ICP) algorithm [17]. Lu and Milos [18] derive a graph-based 2D variant that considers all scans simultaneously in a global fashion. Later, their approach got adopted for 3D point clouds in 6 DOF [19] and extended further to a semi-rigid continuous-time method [7]. This is the method we use in this work as an additional post-registration step, to reduce the amount of accumulated errors during the first scan-matching stage. Another recent continuous-time graph-based framework is "IN2LAAMA" [20], which is able to perform localization, mapping, and extrinsic calibration between a laser-scanner and IMU at the same time. It is an offline-batch method optimized for 360° FOV multichannel LiDAR devices and has been extensively tested with a Velodyne VLP-16, yet is not suited for the application to recently arising solid state laser-scanners. There also exist continuous-time graph-based *online* methods, such as [21] which organize and optimize the system poses using a multilevel hierarchical graph. This method achieves comparable accuracy as similar offline-batch methods by means of multiresolution surfel-based registration. However, the approach is also optimized for traditional multichannel laser scanners and has been tested on carefully controlled micro aerial vehicles (MAVs), which ensures good coverage. More approaches to laser-based SLAM exist that do not rely solely on point-to-point optimization as ICP does. Popular model-based optimization methods often deal with finding planes in the environment, as considering planes is more robust to outliers and noise than considering only points. In Ref. [22], Förster et al. successfully use plane-to-plane correspondences for optimization. Their approach assumes that planar patches got pre-extracted from the point cloud with a method of choice, and incorporates possible uncertainties in the plane model. Further recent examples of laser-based SLAM approaches making use of the existence of planes include [23–26]. Similar registration procedures to ours are [27], which uses plane-to-plane correspondences for pre-registration and point-to-plane correspondences afterward, and [28], which uses point-to-point, as well as plane-to-plane correspondences based on their availability. Two more recent and very interesting SLAM approaches which specialize more on the massively produced LiVOX devices, are "Loam-livox" [3] and "Livox-mapping" [29]. The former is based on the well-known LOAM [30] algorithm, while the latter is provided directly by Livox. Both have been especially optimized for small FOV devices and offer a feature extraction

approach which is suitable for the never repeating, flower-shaped scanning pattern. However, they have been designed with the intention of using them for autonomous driving cars, which follow simpler trajectories compared to this work.

3. SLAM approach

To address the unconventional trajectories, we use a flexible two-staged SLAM approach, which is described in this section. We initially proposed a version of the first stage in [5]. The approach is based on finding planar polygons in the scans and matching them against a global model. In this section, we build upon our previous work and introduce several changes. The second stage of our algorithm is “Semi-Rigid SLAM” [7], which further decreases accumulated registration error from the first stage. It is a continuous-time method where each frame is optimized simultaneously using a partially connected pose graph. **Figure 3** shows a block diagram representing the information flow of the SLAM system, including the additions made in this work. Some basic derivations stay the same (see [5] for further details). Let a point in 3D space be defined as $\mathbf{p}_i = (x_i, y_i, z_i)^\tau$. Further, a homogeneous transformation of that point along the translation $\mathbf{t} = (t_x, t_y, t_z)^\tau$ and rotation defined using the roll-pitch-yaw ($\varphi - \vartheta - \psi$) Tait-Brian angles is given:

$$T(\mathbf{p}_i) = \begin{bmatrix} x_i C_\vartheta C_\psi - y_i C_\vartheta S_\psi + z_i S_\vartheta + t_x \\ x_i (C_\varphi S_\psi + C_\psi S_\varphi S_\vartheta) + y_i (C_\varphi C_\psi - S_\varphi S_\vartheta S_\psi) - z_i C_\vartheta S_\varphi + t_y \\ x_i (S_\varphi S_\psi - C_\psi C_\varphi S_\vartheta) + y_i (C_\psi S_\varphi + C_\varphi S_\vartheta S_\psi) + z_i C_\varphi C_\vartheta + t_z \end{bmatrix}, \quad (1)$$

where C_a and S_a denote cosine and sine with argument a . Additionally, let a plane in 3D space be defined as $\rho_k = \{\mathbf{n}_{\rho_k}, \mathbf{a}_{\rho_k}\}$, where \mathbf{n}_{ρ_k} is the normal vector of the plane and \mathbf{a}_{ρ_k} is its supporting point. The problem we must solve is an optimization problem, where the following error function E has to be minimized:

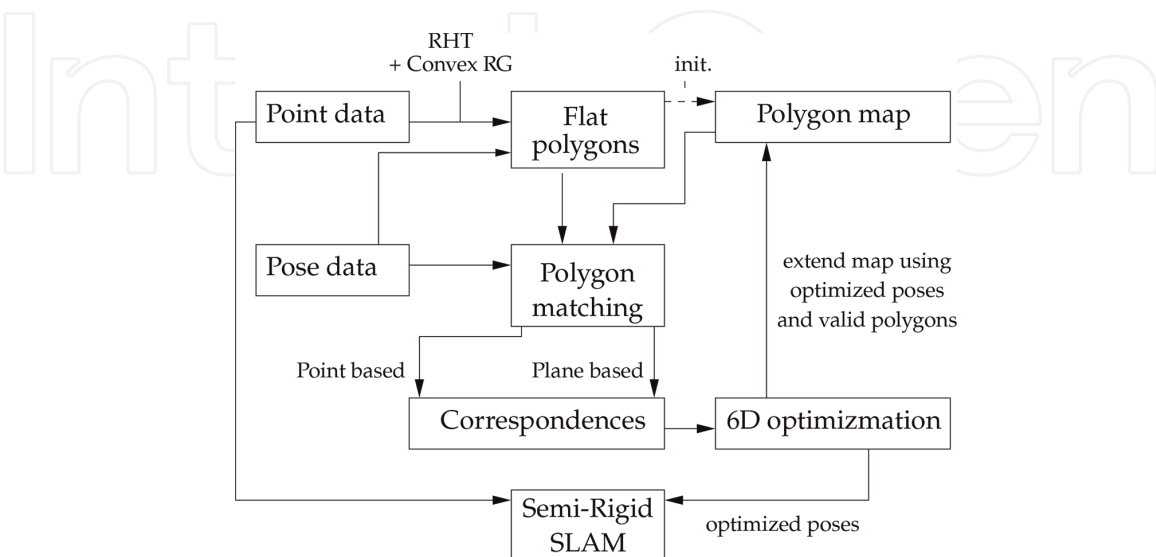


Figure 3. Overview of the proposed SLAM system. The polygon map represents a set of flat, convex-shaped bounding boxes of dominant planes. This model is used to find similarities between polygons from subsequent sensor data.

$$E(T) = \sum_{\rho_k} \sum_{\mathbf{p}_i \in \rho_k} \|\mathbf{n}_{\rho_k} \cdot [T(\mathbf{p}_i) - \mathbf{a}_{\rho_k}]\|^2, \quad (2)$$

Note that point-to-plane correspondences ($\mathbf{p}_i \in \rho_k$) have to be available, which we establish by matching polygons similar to [5]. However, the global polygon model in this work is not extracted only once as an initialization step, as in our previous work. It is instead a new global model, which is initialized and then updated sequentially. Our plane detection approach does also not rely on local planar clustering (LPC) anymore. The old method is a region-growing-based approach to cluster points with similar normals, whereas the new approach uses a Hough-transformation (HT) framework as in Ref. [6]. We describe the abovementioned additions in the following subsections in more detail.

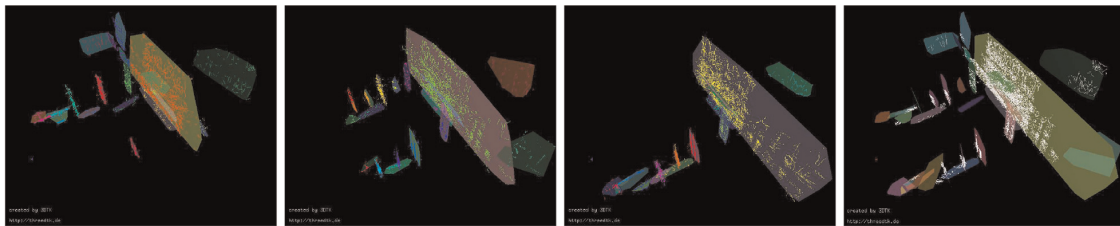
3.1 Local and global plane model

As mentioned earlier, in our previous work [5], we rely on LPC to identify planes in each scan, as well as the points that belong to those planes. The approach calculates normal vectors for each point and clusters them into planar patches based on their distance and angle. Then, after each point in a scan was potentially identified to belong to one plane, correspondences have to be established with respect to the global model. In Ref. [5], we obtain the global model by extracting planes from only a few initial measurements according to Ref. [6].

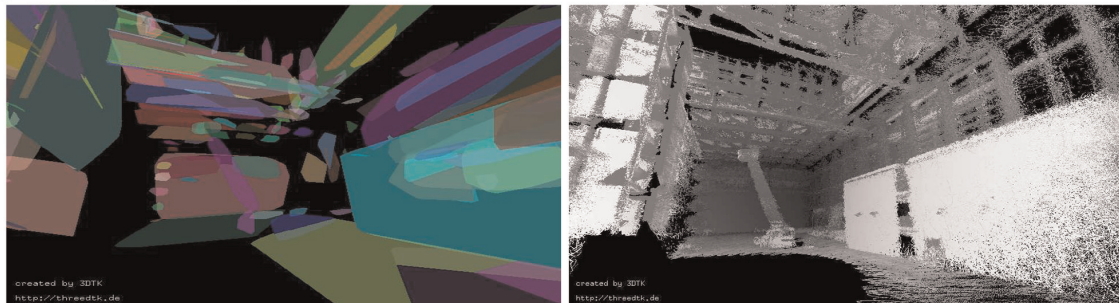
In this work, we replace LPC with that same approach [6] to identify planes in each scan. The new approach is based on a randomized version of the well-known Hough transformation (see Algorithm 5 in Ref. [6]). After a plane has been identified in the Hough space, all the points belonging to that plane are considered, and their convex hull is calculated. However, instead of deleting all the points close to the newly identified plane, we save them in a point cluster and link it to the corresponding plane. That way, we are still able to establish point-to-plane correspondences as in our previous work. **Figure 4** illustrates how the extracted planes from each frame are used to sequentially update the global model. The upper sequence of **Figure 4a** shows the abovementioned point clusters with their corresponding planes for different frames. Note that in the last figure of the sequence, identical planes from the different frames are merged after registration. The bottom **Figure 4b** shows the resulting global plane model, as well as the point cloud after registration of all frames. Merging two planes works by considering all the points on both convex hulls, and recalculate the convex hull and normal vector from those points. Note that this is not fully a dynamic model, as polygons are added sequentially but never refined or even falsified after being added, leading to registration errors such as duplicate and misaligned walls. Now that we have a global plane model, as well as planar point clusters in each subsequent frame, we use the matching function from our previous work [5] to establish correspondences between the two. In the next subsection, we use these correspondences to minimize Eq. (2).

3.2 Closed-form solution with singular value decomposition

In our previous work [5], the minimization of Eq. (2) is achieved by the AdaDelta [31] method, which is based on analytical jacobians and stochastic gradient descent (SGD). However, SGD-based methods require multiple iterations until they converge



(a) From left to right: (1 - 3): Subsequent point cloud and pose data from a laser scanner. The points are grouped in planar clusters, represented by the point color, and a plane is fit through each cluster. (4) The global model that results from sequentially merging the plane observations from 1 - 3, using the first stage. Identical planes from different frames get merged after registration.



(b) (Left) Resulting global plane model, which gets created during the first stage of our SLAM approach. (Right) Corresponding point cloud from the same view for comparison. The grayscale values of the points represent reflectivity.

Figure 4.

Illustration of how the global plane model is obtained and sequentially extended from individual measurements.

to a solution. Furthermore, a hyperparameter is added for each of the six degrees of freedom, as well as an additional parameter specifying the maximum number of SGD iterations before updating correspondences. These parameters are not required anymore, as we now introduce a closed-form solution based on singular value decomposition (SVD). The first appearance of SVD in the context of point set registration is in Ref. [32]. The solution assumes that point-to-point correspondences exist, instead of point-to-plane correspondences. To this end, we must first calculate the projection point from our source point to the target plane. Therefore, the source point gets shifted onto the target plane in the direction of the plane's normal vector. Let D be the signed distance of the point to the plane in the normal direction:

$$D_k^i = \mathbf{n}_{\rho_k} \cdot [T(\mathbf{p}_i) - \mathbf{a}_{\rho_k}] . \quad (3)$$

Then, the projection point onto the plane is given as:

$$\mathbf{P}_k^i = T(\mathbf{p}_i) - D_k^i \cdot \mathbf{n}_{\rho_k} . \quad (4)$$

We use this point for point-to-point correspondence. However, we note that \mathbf{P}_k^i is also on the corresponding plane, thus solving the point-to-point problem with SVD also minimizes our initial Eq. (2). First, we need the correspondence centroids, i.e., the centroid of all the plane projection points, and the centroid of all the data points. Let N be the number of correspondences, then the centroid of plane projections is as follows:

$$\mathbf{c}_m = \frac{1}{N} \sum_{\mathbf{p}_i \in \rho_k} \mathbf{P}_k^i, \quad (5)$$

and the centroid of data points is as follows:

$$\mathbf{c}_d = \frac{1}{N} \sum_{\mathbf{p}_i \in \rho_k} T(\mathbf{p}_i). \quad (6)$$

We set up the real 3x3 correlation matrix \mathbf{M} as follows:

$$\mathbf{M} = \sum_{\mathbf{p}_i \in \rho_k} (T(\mathbf{p}_i) - \mathbf{c}_d) (\mathbf{P}_k^i - \mathbf{c}_m)^\tau, \quad (7)$$

which is decomposed using SVD as follows:

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\tau. \quad (8)$$

The interested reader might consider [33] for a detailed description on how the decomposition works. We set up another real 3x3 rotation matrix \mathbf{R} , which solves the rotation needed to minimize Eq. (2):

$$\mathbf{R} = \mathbf{V}\mathbf{U}^\tau. \quad (9)$$

From this, the translation that minimizes Eq. (2) is calculated as follows:

$$\mathbf{t} = \mathbf{c}_m - \mathbf{R}\mathbf{c}_d. \quad (10)$$

Note that this is a closed-form solution that does not need any hyperparameters. In contrast, AdaDelta [31] needs a hyperparameter for each of the 6 degrees of freedom plus an additional parameter for the maximum number of inner iterations.

3.3 Condense and atomize

Our previous work [5] mentions that after the application of the algorithm, i.e., the first stage in this work, the resulting paths look distorted. This is due to two reasons: (1) The individual frames are “condensed” into metascans, which are referenced with only one pose. We call a collection of multiple subsequent scans, which are represented using a shared coordinate system, a metascan. (2) Some scans do not contain any points due to minimum scanning range, e.g., when rolling over the floor, thus they are not optimized. We fix these problems by introducing the inverse operation to “condense,” which is able to distribute the relative transformation that got applied to the metascans, back onto the individual scans. This back-distributing operation is what we call “atomize.” **Figure 5** illustrates the process of condensing, registration in the condensed domain, and atomizing. During the condense operation, one has to transform all points from different frames in the same reference coordinate system. Note that we start counting the frames with one. Let J be the number of all frames, which should be condensed to a total number of M frames, where $M \leq J$. Let S be an arbitrary integer chosen from the interval $[1, \lfloor \frac{J}{M} \rfloor]$. The number S defines which frame we pick as a reference coordinate system for the points from the other scans in the interval. Next, consider all J frames, given has homogeneous 4x4 matrices,

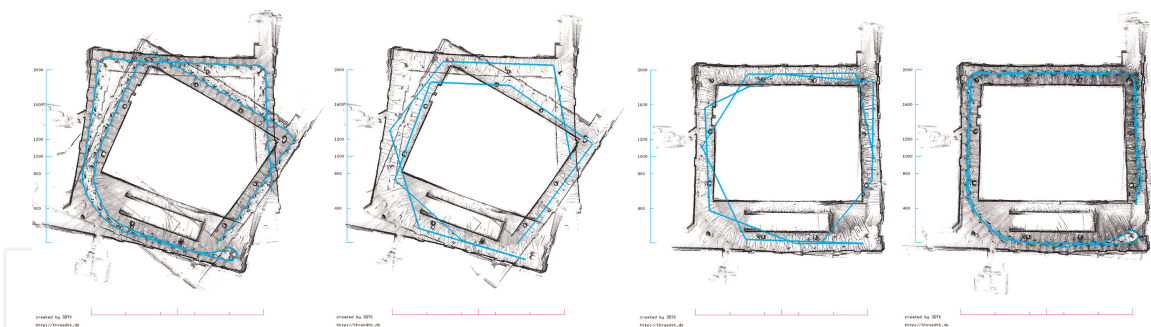


Figure 5.

Trajectories are drawn in blue and result from connecting every subsequent pose with a straight line. From left to right: (1) full point cloud with all initial poses. (2) “condensed” point cloud with only 12 poses. A total of 1000 linescans get combined into one metascan, which has its origin at the pose of the middlest scan. Further, we subsample and filter the metascans themselves. (3) the point cloud from 2 after registration. Only 12 poses got optimized in “condensed” domain. (4) full point cloud and all optimized poses after the “atomize” operation, which is the inverse to “condense.” from the optimized poses in 3, we calculate the relative transformation that has to be applied to all initial poses.

$\{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_S, \dots, \mathbf{H}_{2S} \dots \mathbf{H}_J\}$. The frames with indices $m \cdot S$ (where $m \leq M$) are the indices of the metascan frames, where all points from the other frames, corresponding to the same interval, must be transformed in. Thus, the relative transformation between any single frame with index $j \leq J$ and metascan frame with index m is as follows:

$$\mathbf{H}_{m,j} = \mathbf{H}_{m \cdot S}^{-1} \mathbf{H}_j . \quad (11)$$

We apply this transformation to every point in the j -th frame, for all J frames. Now we have a total of only M metascan frames $\mathbf{H}_{m \cdot S}$, i.e., $\{\mathbf{H}_S, \mathbf{H}_{2S}, \dots, \mathbf{H}_{M \cdot S}\}$, which are input to our first SLAM stage. After the application of the first stage, there are M optimized frames $\hat{\mathbf{H}}_{m \cdot S}$, which we denote with a hat. The atomize operation has to apply the relative pose change between $\mathbf{H}_{m \cdot S}$ and $\hat{\mathbf{H}}_{m \cdot S}$, back onto the individual scans. Thus, the new optimized frames for all J original frames are as follows:

$$\hat{\mathbf{H}}_j = (\hat{\mathbf{H}}_{m \cdot S} \mathbf{H}_{m \cdot S}^{-1}) \mathbf{H}_j . \quad (12)$$

After distributing the relative pose-change onto the individual scans in that way, the second stage of our approach begins. In the second stage, every individual frame is considered simultaneously in a pose graph.

3.4 Post-registration with semi-rigid SLAM

In our first SLAM stage, each scan is considered sequentially. That way, the algorithm creates a global plane model of the environment, without the knowledge of future measurements. This leads to registration inaccuracies during the first stage, which is why we use a second stage afterward: “Semi-Rigid Registration” (SRR) [7]. The method considers all scans simultaneously in a continuous-time fashion using a pose graph. In the graph, each pose is represented by a node and is connected via edges to other poses if the overlap between the corresponding scans is large enough. After one iteration of the algorithm, SRR re-calculates the edges. **Figure 6** illustrates the behavior of SRR on a dataset recorded by a spherical system rolling on a flat ground. Section 4.2 describes the experimental setup and evaluation of the dataset. In

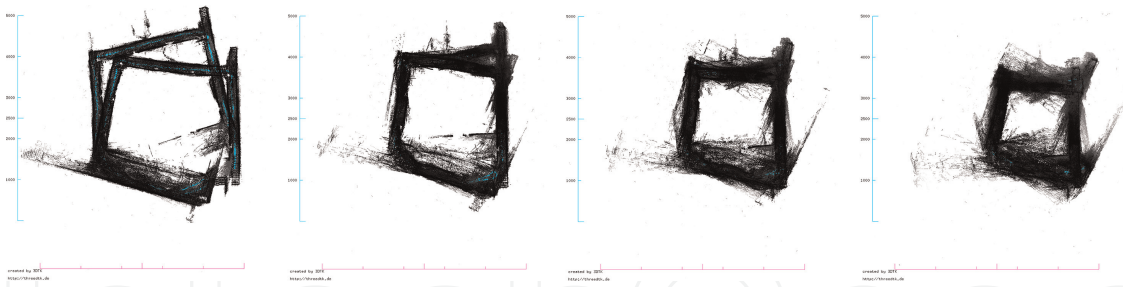


Figure 6. Illustration of multiple iterations of “semi-rigid registration” (SRR) [7]. From left to right: (1) resulting point cloud with initial pose estimations, zero iterations. (2) resulting point cloud after 50 iterations of SRR. (3) after 100 iterations. (4) after 150 iterations.

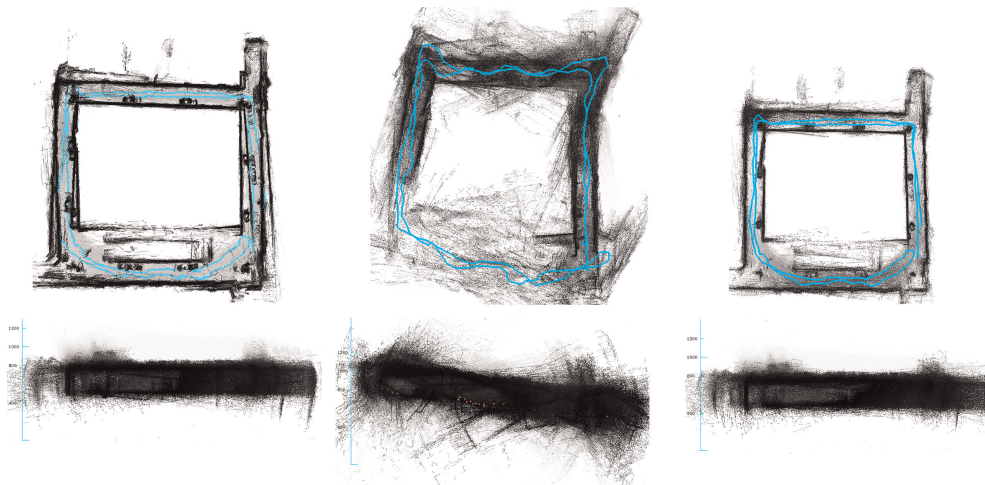


Figure 7. Comparison of the first stage, second stage, and first stage followed by the second stage. Upper row: Point clouds in birds-eye view, with the sliced ceiling. Lower row: Point clouds in side view, aligned to the initial orientation. The initial input to the algorithms is shown in the most left image in **Figure 6**. From left to right: (1) first stage only. (2) second stage (SRR) only. (3) first stage is followed by the second stage.

the given example, the initial pose estimates are subject to a large amount of drift. Although SRR has been designed to also reduce such large-scale errors, the method alone is not able to perform well on more complex trajectories shown in this work. However, if the input to SRR is already coarsely aligned, the quality of point clouds and trajectories improves, as shown in **Figure 7**. The images in the left column show the resulting point cloud after the application of the first stage. The walls are not perfectly aligned, yet the side view reveals that the trajectory is planar. In the centered column, SRR is not able to register the measurements in a meaningful way, using the initial pose estimates. Moreover, the point cloud and trajectory are less planar. In the right column, both stages get applied, resulting in better overall map quality. Using the input from the first stage, the graph-based second stage is able to reduce the accumulated registration error from the first stage.

3.5 Comparison with ICP

The previous section has demonstrated that using SRR alone is not an option with the given trajectories. **Figure 6** shows how SRR is not able to converge to a meaningful solution when being applied to a dataset from a rolling spherical system. For this reason, the input to SRR is usually preregistered using the well-known ICP algorithm.

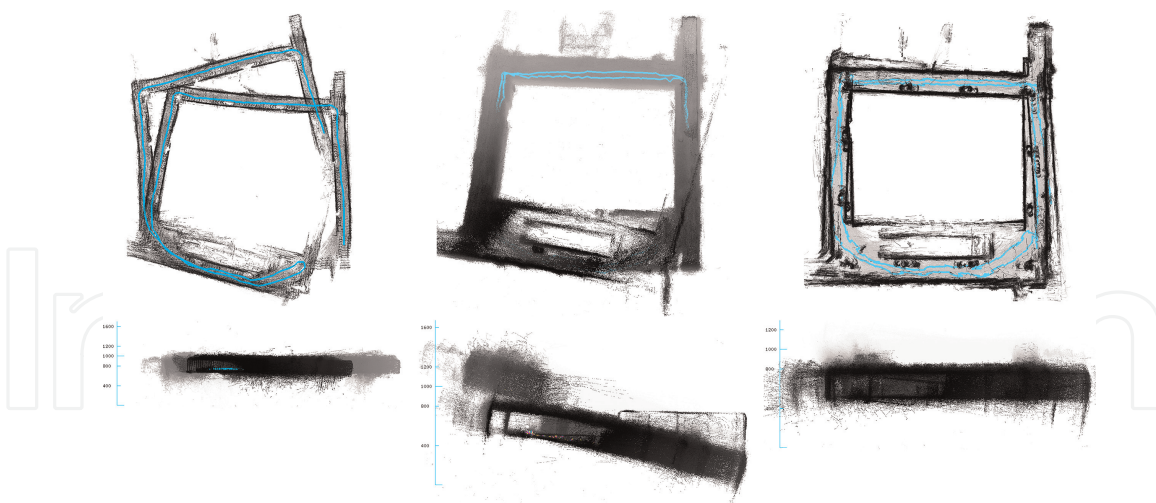


Figure 8.

Comparison of point clouds and trajectories after application of different SLAM algorithms. Upper row: Birds-eye sliced view of the point clouds. Lower row: Side-view of the point clouds, aligned with the initial orientation. From left to right: (1) initial point cloud and trajectory. (2) after ICP, metascan variant, available in 3DTK—The 3D toolkit [34]. (3) after the first stage of the proposed method.

The SRR preregistration uses a highly precise metascan implementation, available from 3DTK—The 3D Toolkit [34]. However, the polygon-based approach outperforms ICP, which is illustrated in **Figure 8**. In the images of the left column, one sees a birds-eye view of the input to both algorithms. The center images show the resulting point cloud and trajectory after ICP. Due to the given trajectory, the algorithm is not able to establish meaningful point-to-point matches on the dataset, thus the output is no longer planar. The first stage we present in this work is shown in the images in the right column. Although some walls are not aligned, the result is more planar and resembles the environment better than the output of ICP. Using this method before applying SRR leads to a faster convergence and more accurate solution in the second stage. In the next section, we analyze the accuracy of the resulting maps qualitatively, as well as quantitatively using high-precise ground truth point clouds.

4. Experiments

In this section, we describe the system setup and procedure of four experiments, which demonstrate unconventional trajectories. Note that all datasets are available at <http://kos.informatik.uni-osnabrueck.de/3Dscans/>. We compare three systems to each other that have been tested in the same environment. One other system had to be tested in a different building, which allowed for a long descent from a crane, which we therefore analyze separately. For our experiments, we use three kinds of motion: rolling on the ground, moving forward while swinging, and descending while rotating. All motion profiles were shown previously in **Figure 2a**. A birds-eye view of the environments in which the experiments were carried out is illustrated in **Figure 9**. In the left image, a square hallway can be seen which we used for the pendulum and rolling experiments, whereas the right image shows the firefighter school which we used for the descending experiment. We use the same LiDAR sensor in three experiments: the Livox Mid-100. It produces 300.000 points per second using three beams that scan in a non-repetitive, flower-shaped fashion, thus point density increases over

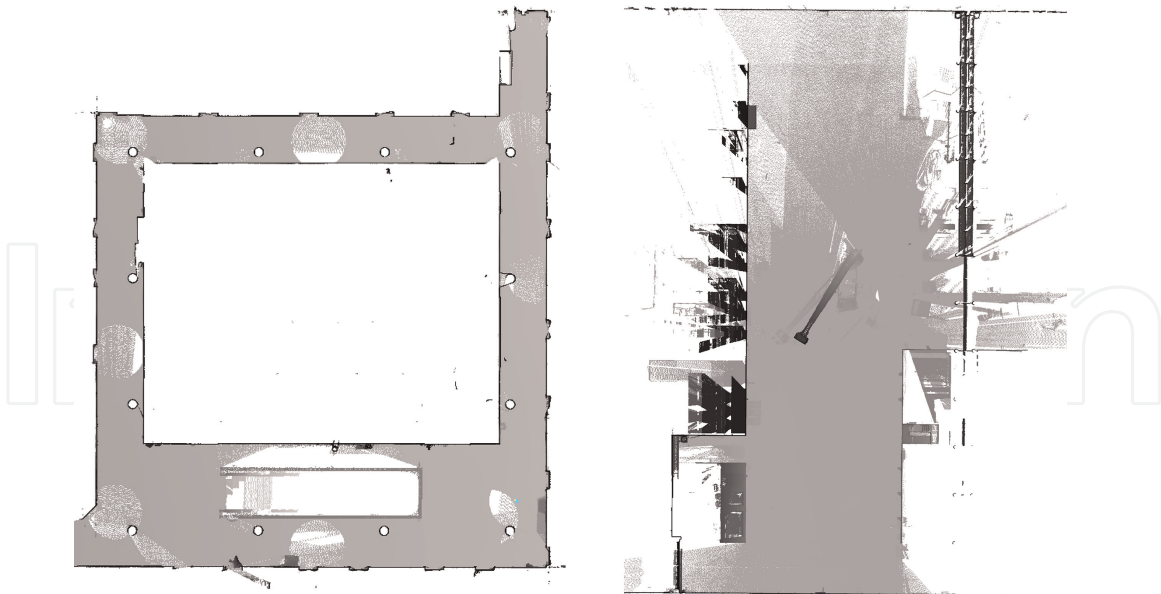


Figure 9. Birds-eye view of the ground truth point clouds, acquired with a terrestrial laser scanner (TLS). The ceiling has been cropped for a better view. Left: Ground truth point cloud of an office hallway used for the rolling and pendulum experiments. Right: Ground truth point cloud of the state firefighters school in Würzburg, used for the descending experiment.

time. Each beam has a circular field of view (FOV) of 38.4° . Thus, three beams aligned in a row create a vertical FOV of 38.4° and horizontal FOV of 98.4° . The precision at 20-meter scanning distance is 2 cm and the angular accuracy is 0.1° . Further, the minimum scanning distance of the laser scanner is 1 m and the output frequency is set to 10 Hz. The maximum output frequency of the sensor is 50 Hz, yet point density then decreases. In future work, though, we want to test the systems also with 50 Hz LiDAR frequency. For all setups, a ROS installation on a Raspberry Pi 4 is used for onboard controlling and recording data. Additionally, we apply the rolling motion to a SICK LMS141 2D laser scanner with a maximum range of 40 m that operates at a scanning frequency of 50 Hz and resolution of 0.5° . Here a Raspberry Pi 3 is used for data collection. Inertial measurements are performed by PhidgetSpatial Precision 3/3/3 IMUs. The post-processing is performed after the experiments on a separate server.

4.1 Pendulum

As to the pendulum setup the system is equipped with an Intel T265 tracking camera, which uses a combination of feature tracking and internal IMUs to estimate its pose. The T265 unfortunately has been discontinued by Intel and is only available on the secondary market, which increased its price. As a budget alternative, we consider the Intel T261, which is still available, or performing visual-inertial-odometry manually, e.g. Intels RealSense-SLAM or VINS-Fusion [35] with Intels D435i. To test this setup in the hallways of an office-like environment (cf. upper left image of **Figure 2b**), we put the sensors inside a trailer net and swing them back and forth while walking. The movement itself consists of de- and accelerations to the front and back and slight movements up and down. Depending on the walking speed of the person, even an overall negative velocity of the sensor is possible if the pendulum swings faster backward than the person moves forward. Note that the view of the tracking camera is partially obscured by the trailer net, reflections off the shell, and

the thread of the shell. As the camera and laser scanner are mounted near to each other with the same orientation, we assume that their coordinate systems coincide and thus, use no external calibration between the two. The duration of this motion was 281 s and covered a total distance of approx. 162 m (only walking, oscillation not included).

4.2 Rolling on flat surfaces

In this experiment, we put the sensors inside a spherical plastic shell and roll it on a flat surface manually (cf. bottom left image of **Figure 2b**). This time there is no tracking camera included. Rather, rotation, as well as position, estimates come from a combination of three Phidget 3/3/31044-0b IMUs. We use an IMU filter that is specialized for pose estimation of spherical robots [4]. The rolling duration was 691 s and covered the same distance as before of 162 m. Further, a similar experiment has been executed in the same environment and with the same trajectory, with the RADLER system as described in [8]. RADLER is a modified unicycle where the 2D laser scanner is mounted with its scanning plane parallel to the wheel axle thus creating a radial scanning pattern while rotating.

4.3 Crane descending

Unlike the previous experiments, this one was executed in a different environment that allowed for a long descent, i.e., in the building of the state firefighters school in Würzburg. We connected the robot to an outsourced processing machine via a 50 m tear-resistant tether cable (Fathom ROV Tether by BlueRobotics) which was rolled around a coil to perform the descending and ascending movement (cf. right image of **Figure 2b**). In this experiment, the sensor unrestrictedly rotates around the descending axis, corresponding to the cable direction. Note that the rotation itself is induced by the internal twist of the cable, not by any actuators. A spin encoder estimates the position, as it measures the rotation of the coil which directly corresponds to the height of the robot according to the helix arc length formula. The descent of the sphere covered a distance of 22 m and was performed within a duration of 402 s.

5. Evaluation

The resulting point clouds after application of the presented SLAM approach are now analyzed in terms of their accuracy, which we do by matching them against high-precise ground-truth models of the environment, given by a RIEGL VZ400 terrestrial laser scanner (TLS). It has an angular resolution of 0.08° and an accuracy of 5 mm. For the evaluation, we consider histograms that show a distribution of point-to-point errors. To create such histograms, we match the resulting point clouds against the corresponding ground truth maps, using ICP from 3DTK [34]. Then, we create a three-dimensional difference image by measuring all point-to-point errors. When calculating the difference images for any dataset, we use an octree-based filter where the voxels are smaller than 10 cm with a maximum of 5 points. Thus, the resolution of the different images is the same, i.e., histograms do not depend on point density anymore and are comparable to each other as long as they have been created for the same environment. **Figure 10** contains three histograms that were created in the office hallway environment. Broadly speaking, a distribution is better if its tail is short and its peak is located toward the left, i.e., zero point-to-point error. In particular, the

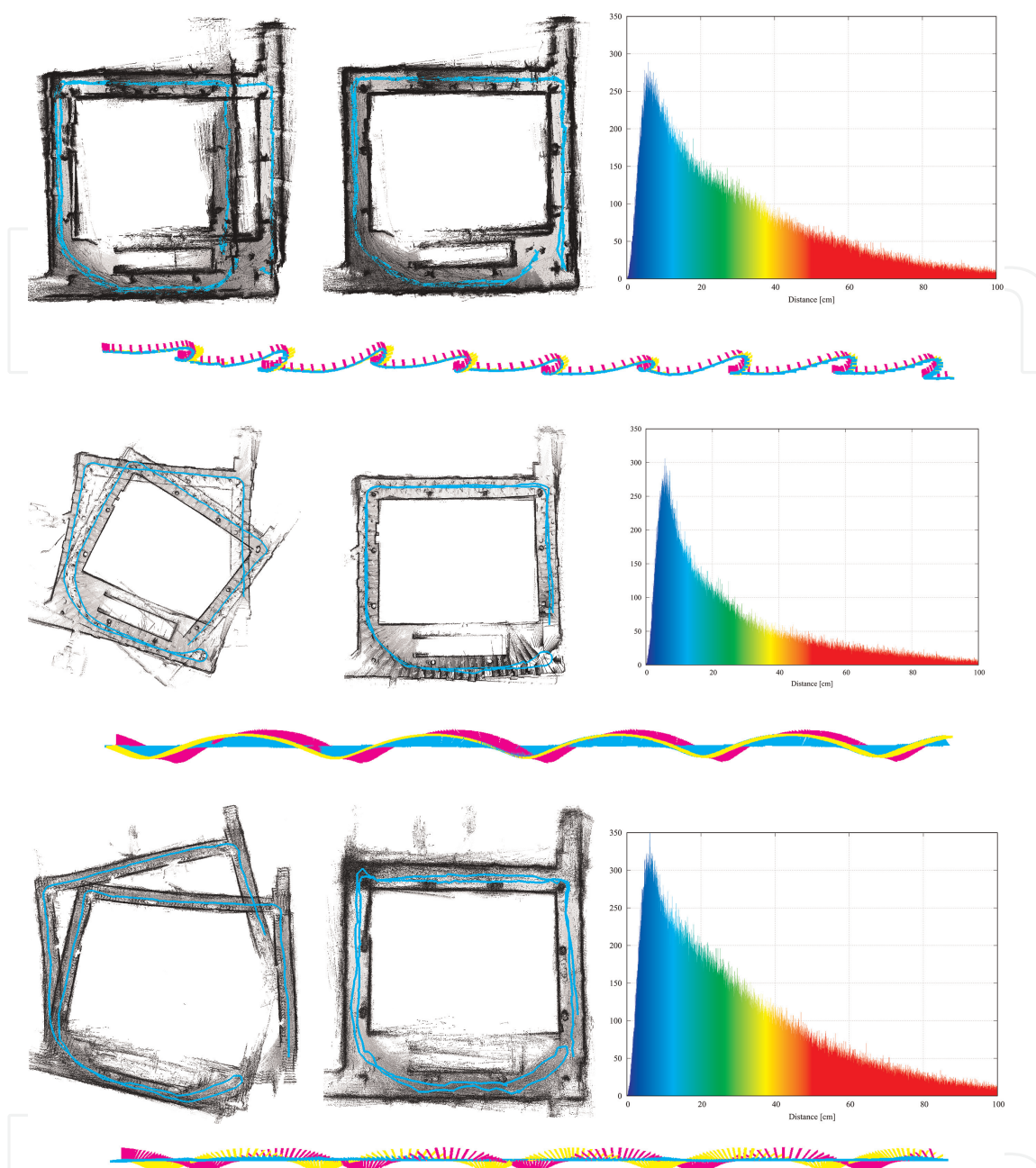


Figure 10. Birds-eye view of point clouds acquired with the different setups. The ceiling has been cropped for a better view. A profile view showing sensor poses after registration, movement from left to right is also shown. Left column: Unprocessed point cloud with initial pose estimates, from IMUs and tracking camera (pendulum), IMUs and wheel encoder (RADLER) and IMUs only (spherical). Center column: Post-processed point cloud after application of proposed SLAM algorithm. Right column: Histograms showing the occurrences of certain point-to-point errors from the compared post-processed 3D point clouds to the reference ground truth point cloud. The colors denote distance, where blue corresponds to zero distance and red corresponds to 50 cm distance or more. (a): Pendulum setup. Mean point-to-point error is 28.31 cm with peak at 10.94 cm. (b): Spherical system. Mean point-to-point error is 28.70 cm with peak at 11.64 cm. (c): RADLER. Mean point-to-point error is 24.48 cm with peak at 12.10 cm.

quantity of most interest is the distribution mean, as it tells you the average point-to-point error. The individual histograms correspond to the pendulum, spherical system, and RADLER results, respectively. From the naked eye, one might suspect that RADLER performed best, while the spherical system performed worst. The following sections confirm this statement quantitatively and give a more detailed interpretation of the results.

5.1 Pendulum

Figure 10a shows that the initial pose estimations using the Intel T265 tracking camera is the most accurate, when compared to the methods without feature tracking (cf. left image of **Figure 10b** and **c**). In the other datasets, the IMUs struggle with yaw angle estimations especially at the corners, whereas here the feature tracking compensates for that. After registration, the map represents the environment better as before since there are no duplicate corridors left in the point cloud, and the optimized poses are consistent with the map. Yet the result is not perfect, e.g., the walls appear thick due to the large amount of motion distortion, the pillars are not perfectly aligned, and a sizeable duplicate wall remains uncorrected. Note that this is presumably because there is no external calibration between the tracking camera and laser scanner. Employing such is a task for future work and potentially increases the mapping accuracy. According to the mean point-to-point error (\bar{E}) from the histograms in **Figure 10**, this result ($\bar{E} = 28.31$ cm) resembles ground truth better than the rolling system ($\bar{E} = 28.70$ cm), but worse than RADLER ($\bar{E} = 24.48$ cm).

5.2 Rolling on flat surfaces

The following sections sum up the results for the mobile systems with rolling sensor trajectories, i.e., RADLER and the spherical system. As mentioned above, RADLER has the best similarity to ground truth according to **Figure 10**, whereas the spherical system has the worst.

5.2.1 2D LiDAR

Figure 10b presents the results of the experiment with RADLER, which were carried out in the same environment as before. The left image shows that the initial pose estimates have significantly more drift compared to the pendulum system, especially regarding the yaw angle. However, the walls appear thinner, and there is overall less noise due to the missing shell. After our SLAM, there are a few spots where the walls are not perfectly aligned, and a lot of noise remains between the walls. In comparison to the maps created with the pendulum and spherical system, though, RADLERs result resembles ground truth best. We suppose that this is because RADLER's 2D scanner operates at a higher frequency (50 Hz) and uses the rotational encoder in addition to the IMU for determining the systems pitch. The Livox Mid-100 scanner used for the other experiments operates on only 10 Hz, which is why fast trajectories lead to a more obscure scan and, thus thicker walls. Further, the 2D scanner from SICK is optimized for short-range measurements, whereas the 3D scanner from Livox is for long medium- to long-range measurements.

5.2.2 3D LiDAR

Figure 10c shows the results of the experiment with the spherical system. The initially estimated trajectory distance is the largest when compared against the other datasets (211.77 m compared to RADLER: 141.01 m, and pendulum: 148.30 m), indicating an overestimated radius parameter in the pose estimation model [4]. The resulting map resembles the actual scale of the environment better, the pillars are well aligned, and the corrected poses are consistent with the map. However, there are also

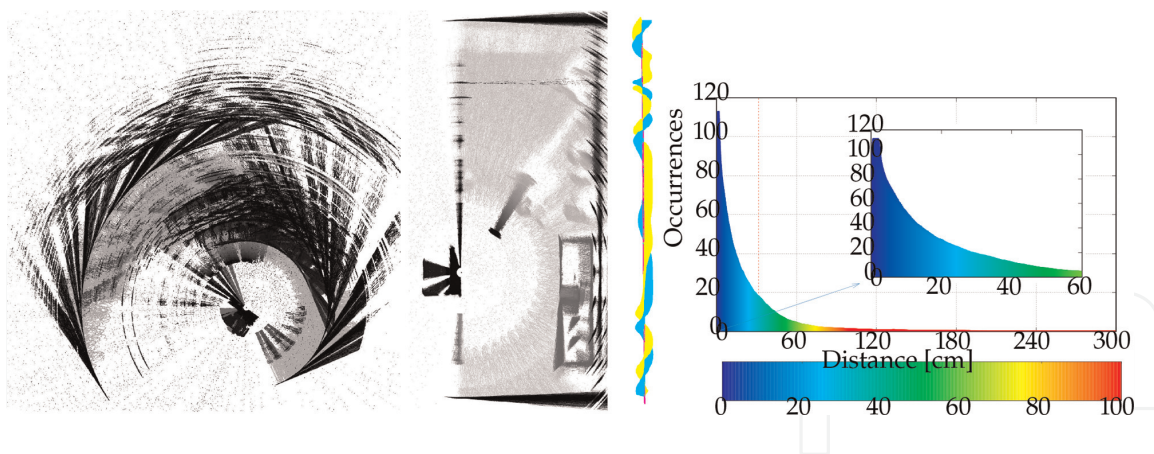


Figure 11. In the images of 3D point clouds, the ceiling has been cropped for a better view. From left to right: (1) birds-eye view of the resulting 3D point cloud, acquired with the descending system using a spin-encoder and IMUs for pose estimation. (2) birds-eye view of the post-processed 3D point cloud. (3) profile view of the mobile systems pose, movement from top to bottom. (4) histogram showing the occurrences of certain point-to-point errors to ground truth. The colors denote distance, where blue corresponds to zero distance and red corresponds to 1 m distance or more. Mean point-to-point error is 31.6 cm with peak at 3.60 cm.

duplicate walls, remaining outliers, and noise due to remaining registration errors. According to the histograms in **Figure 10**, the resulting map has the least similarity to ground truth when compared to the pendulum and RADLER, which is consistent with previous observations. Note that in this dataset, the shell is attached to the sphere, which makes range measurements more noisy and adds outliers due to reflections off the shell.

5.3 Crane descent

This section presents the results for the crane descent experiment, which was conducted in a different environment than the previously mentioned results. Thus, the histogram is not really comparable to the ones in **Figure 10**, although it uses the same voxel filter to create the distance image. We still analyze the shape and mean point-to-point error of the distribution and to interpret them. The upper half of **Figure 11** shows a birds-eye view of the 3D point clouds in the same fashion as before. Note that the initial pose estimates are especially erroneous in one rotational dimension. This is the yaw rotation, which is especially difficult to detect for IMUs without the use of a magnetometer. As this experiment originated in the context of a space mission, using the magnetometer for inertial measurements was not an option. In the first stage of an out SLAM algorithm, we locked each other dimension but yaw from being optimized. The resulting map resembles the environment well, yet error remains due to low scanning frequency and motion distortion. The mean point-to-point error when comparing against ground truth (cf. right image of **Figure 9**) is 31.6 cm. However, the peak of the histogram is located at 3.60 cm, indicating that there is room for further improvement. We seek to improve on these results by accounting motion distortion and further reducing IMU drift in future work.

6. Conclusion

We have shown in this work that unconventional trajectories still pose problems for current SLAM algorithms, especially when using low FoV LiDARs. We built a

flexible SLAM approach that shows the capabilities to register unconventional trajectories with large-scale pose estimation errors reliably. Further, we tested our SLAM system with three different unconventional movements: rolling, pendulum, and rotating crane descend. According to the previous accuracy evaluation, rolling on the floor is the most difficult scenario. The spherical shell of the system adds noise and outliers to the range measurement. Additionally, low overlap and sometimes no overlap make scan matching hard, even using polygons. Therefore, the success of SLAM using this trajectory type, compared with the other scanning methods, relies the most on the initial pose estimations. Moreover, this scenario has the most difficult initial pose estimation, due to the large accumulation of errors both in translation and rotation, which makes SLAM especially difficult. RADLER seems to have the best results regarding accuracy. This is because of the higher scanning frequency compared to the other experiments, but also because the rotational encoder on the wheel helps a lot with position estimation when compared to the spherical setup, which relies on constrained IMU integration. Therefore, it is sufficient to compensate mostly the accumulated rotational error via SLAM. Descending from the crane shows similar behavior: the rotational encoder on the cable reel makes position estimation fairly easy. Further, there are only negligible rotations in two principal axes. However, the faster and uncontrolled rotation around the cable leads to a much larger error in the corresponding axis of revolution, as well as to larger motion distortion. Since pose errors mostly accumulate in one rotational degree of freedom, our SLAM is still able to correct these via constrained optimization in the first stage. The pendulum setup, on the other hand, shows almost no rotational error in the initial pose estimations, because the visual-inertial odometry (VIO) of the Intel T265 camera compensates for IMU drift. VIO works reliably although the view of the camera is partially obscured by the trailer net, the thread of the shell, and reflections from the shell. Yet the relocalization module of the camera, which uses an internal feature map, fails once, as the visual features of the hallways are ambiguous. This leads to large positional errors in the initial pose estimates at the end of the trajectory. Our SLAM is able to correct these errors using the polygon-based optimization in the first stage. However, a lot of work remains to be done. In particular, we need to address the large drift of the IMUs and employ a more accurate external calibration, to improve the initial pose estimates even more. Furthermore, we aim to improve the quality of the maps by revisiting the global polygon model of the algorithms first stage and making it more dynamic. Moreover, we want to mitigate the effects of motion distortion in future work. Finally, the pose estimations need to be evaluated in terms of the achieved positional and rotational errors, e.g. with an external optical tracking system. Nevertheless, we provide significant insight and datasets with ground truth maps and are excited to contribute more to this rather unexplored field of research in future work.

IntechOpen

Author details


Fabian Arzberger^{1*}, Jasper Zevering¹, Anton Bredenbeck², Dorit Borrmann¹
and Andreas Nüchter¹

¹ Robotics and Telematics, Julius Maximilian University of Würzburg, Würzburg, Germany

² Delft University of Technology, Delft, The Netherlands

*Address all correspondence to: fabian.arzberger@uni-wuerzburg.de

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Li K, Li M, Hanebeck UD. Towards high-performance solid-state-LiDAR-inertial odometry and mapping. *IEEE Robotics and Automation Letters*. 2021; **6**(3):5167-5174
- [2] Tan W, Zhang D, Ma L, Wang L, Qin N, Chen Y, et al. Semantic segmentation of UAV lidar point clouds of a stack interchange with deep neural networks. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS. Brussels, Belgium: IGARSS; 2021. pp. 582-858
- [3] Lin J, Zhang F. Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV. In: 2020 IEEE International Conference on Robotics and Automation. (ICRA). 2020. pp. 3126-3131. DOI: 10.1109/ICRA40945.2020.9197440
- [4] Zevering J, Bredenbeck A, Arzberger F, Borrmann D, Nuechter A. IMU-based pose-estimation for spherical robots with limited resources. In: 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. Karlsruhe, Germany: MFI; 2021. pp. 1-8
- [5] Arzberger F, Bredenbeck A, Zevering J, Borrmann D, Nüchter A. Towards spherical robots for mobile mapping in human made environments. *ISPRS Open Jour Photogrammetry and Remote Sensing*. 2021;**1**:100004
- [6] Borrmann D, Elseberg J, Lingemann K, Nüchter A. The 3D Hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*. 2011;**2**:1-13
- [7] Elseberg J, Borrmann D, Nuchter A. 6DOF Semi-rigid SLAM for Mobile Scanning. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. Vilamoura-Algarve, Portugal: IROS; 2012. pp. 1865-1870
- [8] Borrmann D, Jörissen S, Nuchter A. Radler – A RADial LasER scanning device. In: Proc. of the Int. Symp. on Experimental Research. Buenos Aires, Argentina; 2020. pp. 655-664
- [9] Zevering J, Bredenbeck A, Arzberger F, Borrmann D, Nüchter A. LUNA-A laser-mapping unidirectional navigation actuator. In: International Symposium on Experimental Robotics. Valletta, Malta: ISER; 2020. pp. 85-94
- [10] Rossi AP, Maurelli F, Dreger H, Mathewos K, Pradhan N, Pozzobon R, et al. DAEDALUS - Descent and Exploration in Deep Autonomy of Lava Underground Structures. *JMU Würzburg: Inst.für Informatik*; 2021. p. 21
- [11] ESA. ESA plans mission to explore lunar caves; 2021. Available from: https://www.esa.int/Enabling_Support/Preparing_for_the_Future/Discovery_and_Preparation/ESA_plans_mission_to_explore_lunar_caves
- [12] Bosse M, Zlot R, Flick P. Zebedee: Design of a Spring-Mounted 3-D range sensor with application to Mobile mapping. *IEEE Transactions on Robotics*. 2012;**28**(5):1104-1119
- [13] Lehtola VV, Virtanen JP, Vaaja MT, Hyypä H, Nüchter A. Localization of a Mobile laser scanner via dimensional reduction. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*. 2016;**121**:48-59
- [14] Geoslam. ZEB Revo RT; Accessed: February 24, 2022. <https://www.geoslam.com/solutions/zeb-revo-rt/>

- [15] Leica. Leica BLK2GO;. Accessed: February 24, 2022. <https://shop.leica-geosystems.com/global/blk2go-overview>
- [16] Alismail H, Browning B. Automatic calibration of spinning actuated Lidar internal parameters. *Jour Field Robotics*. 2015
- [17] Besl PJ, McKay ND. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992;14(2): 239-256
- [18] Lu F, Milios EE. Globally consistent range scan alignment for environment mapping. *Auton Robots*. 1997;4:333-349
- [19] Borrmann D, Elseberg J, Lingemann K, Nüchter A, Hertzberg J. Globally consistent 3D mapping with scan matching. *Robotics and Auton Systems*. 2008;56(2):130-142
- [20] Gentil CL, Vidal-Calleja T, Huang S. IN2LAMA: INertial Lidar Localisation and MApping. In: 2019 International Conference on Robotics and Automation. Montreal, QC, Canada: ICRA; 2019. pp. 6388-6394
- [21] Droschel D, Behnke S. Efficient Continuous-Time SLAM for 3D Lidar-Based Online Mapping. In: IEEE Int Conf on Robotics and Automation. Brisbane, Australia: ICRA; 2018
- [22] Förstner W, Khoshelham K. Efficient and accurate registration of point clouds with plane to plane correspondences. In: 2017 IEEE Int. Conf. on Computer Vision Workshops. Venice, Italy: ICCVW; 2017. pp. 2165-2173
- [23] Grant WS, Voorhies RC, Itti L. Efficient Velodyne SLAM with point and plane features. *Auton Robots*. 2019;43: 1207-1224
- [24] Geneva P, Eckenhoff K, Yang Y, Huang G. LIPS: LiDAR-Inertial 3D Plane SLAM. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS '18). 2018. pp. 123-130
- [25] Zhou L, Wang S, Kaess M. LSAM: LiDAR smoothing and mapping with Planes. In: Proc. of IEEE Int. Conf. On Robotics and Automation (ICRA '21). Xi'an, China; 2021
- [26] Wei X, Lv J, Sun J, Pu S. Ground-SLAM: Ground Constrained LiDAR SLAM for Structured Multi-Floor Environments; 2021. DOI: 10.48550/ARXIV.2103.03713
- [27] Favre K, Pressigout M, Marchand E, Morin L. A plane-based approach for indoor point clouds registration. In: ICPR 2020 - 25th Int. Conf. on Pattern Recognition. Milan, Italy; 2021
- [28] Taguchi Y, Jian YD, Ramalingam S, Feng C. Point-plane SLAM for hand-held 3D sensors. In: 2013 IEEE International Conference on Robotics and Automation. Karlsruhe, Germany: IEEE; 2013. pp. 5182-5289
- [29] LIVOX. Livox mapping;. Accessed June 30, 2022. https://github.com/Livox-SDK/livox_mapping
- [30] Zhang J, Singh S. LOAM: Lidar odometry and mapping in real-time. In: Robotics: Science and Systems Conf. Berkeley, California. 2014
- [31] Zeiler MD. ADADELTA: An adaptive learning rate method. *CoRR*. 2012
- [32] Arun KS, Huang TS, Blostein SD. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1987; 9(5):698-700

[33] Golub G, Kahan W. Calculating the singular values and Pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*. 1965;2(2): 205-224

[34] Nüchter A, Lingemann K. 3DTK—The 3D Toolkit. 2011. <https://slam6d.sourceforge.io/index.html>

[35] Qin T, Cao S, Pan J, Shen S. A General Optimization-based Framework for Global Pose Estimation with Multiple Sensors; 2019