

Towards Spherical Robots For Mobile Mapping In Human Made Environments*

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

*Informatics VII: Robotics and Telematics, Julius-Maximilians-University Würzburg, Germany
e-mail: dorit.borrmann@uni-wuerzburg.de*

Abstract

Spherical robots are a format that has not been thoroughly explored for the application of mobile mapping. In contrast to other designs, it provides some unique advantages. Among those is a spherical shell that protects internal sensors and actuators from possible harsh environments, as well as an inherent rotation for locomotion that enables measurements in all directions. Mobile mapping always requires a high-precise pose knowledge to obtain consistent and correct environment maps. This is typically done by a combination of external reference sensors such as Global Navigation Satellite System (GNSS) measurements and inertial measurements or by coarsely estimating the pose using inertial measurement units (IMUs) and post processing the data by registering the different measurements to each other. In indoor environments, the GNSS reference is not an option. Hence many mobile mapping applications turn to the second option. An advantage of indoor environments is that human-made environments usually have a certain structure, such as parallel and perpendicular planes. We propose a registration procedure that exploits this structure by minimizing the distance of measured points to a corresponding plane. Further, we evaluate the procedure on a simulated dataset of an ideal corridor and on an experimentally acquired dataset with different motion profiles. We show that we nearly reproduce the ground truth for the simulated dataset and improve the average point-to-point distance to a reference scan in the experimental dataset. The presented algorithms are required to work completely autonomously.

Keywords: Mobile Mapping, SLAM, Spherical Robot

2021 MSC: 00-01, 99-00

1. Introduction

Today's robots for mobile mapping come in all shapes and sizes. State of the art for urban environments are laser scanners mounted to cars. Smaller robotic systems are particularly used when cars no longer have access. Examples for this are human operated systems such as Zebedee [1], a small Hokuyo 2D scanner on a spring, that is carried through the environment, VILMA [2], a rolling FARO scanner operating in profiler mode, RADLER [3], a SICK 2D laser scanner mounted to a unicycle, or a backpack mounted "personal laser scanning system" as in [4] or [5]. Recently more and more autonomous systems gained maturity. A stunning example is Boston Dynamics' quadruped "Spot" that autonomously navigates and maps human environments [6]. Also, the mobile mapping approaches implemented on the ANYmal platform such as [7] were very successful. Of all these formats, one has not been explored thoroughly in the scientific community: The spherical mobile mapping robot. Yet this provides some very promising advantages over the other formats. For one, the locomotion of a spherical robot inherently re-

sults in rotation. That way, a sensor fixed inside the spherical structure will cover the entire environment, given the required locomotion without the need for additional actuators for the sensors. This requires a solution for the spherical simultaneous localization and mapping (SLAM) problem given the six degrees of freedom of the robot. Secondly, a spherical shell that encloses all sensors protects these from possible hazardous environments. For example, the shell stops any dust that deteriorates sensors or actuators when settling at sensitive locations. In contrast to an usual enclosing the shell can separate the sensors entirely from the environment without the need for a number of points-of-connection. A strict requirement then is that the shell is very durable. This is particularly useful for unknown and dangerous environments. E.g., for space applications, in the DAEDALUS study [8], a spherical robot is proposed that is to be lowered into a lunar cave and create a 3D map of the environment. The authors choose this format as the present moon regolith is known to damage instruments and other components.

This paper proposes to use such a spherical robot for mobile mapping man-made environments. In such environments, one advantage are architectural shapes following standard conventions arising from tradition or utility. In particular, there are many flat surfaces such as walls, floors, etc. that are sensed. Exploiting this fact

*To abide the FAIR-Principles of science, all data and code used to produce the results are given in our repository (<https://github.com/fallow24/SphereTDP>).⁴⁵

yields more opportunities for registration as point-to-plane correspondences can be used. The proposed registration method minimizes the distances of each point to its corresponding plane as an objective function. The method is evaluated on a simulated dataset as well as on experimentally acquired datasets.

2. State Of The Art

2.1. Spherical SLAM

While there exist camera-based approaches [9], to the best of our knowledge, SLAM with spherical robots and laser scanners was not done before. However, laser-based SLAM algorithms for motions in six degrees of freedom (DoF) have been thoroughly studied. For outdoor environments [10] provides a first baseline. Adding a heuristic for closed-loop detection and a global relaxation Borrmann et al. yield highly precise maps of the scanned environment [11]. Zhang et al. propose a real-time solution to the SLAM problem in [12]. They achieve the performance at a lower computational load by dividing the SLAM algorithm into two different algorithms: one performing odometry at a high frequency but low fidelity and another running at a lower frequency performing fine matching and registration of the point clouds. More recently Dröschel et al. also propose an online method using a novel combination of a hierarchical graph structure with local multi-resolution maps to overcome problems due to sparse scans [13].

Since these approaches are based on point-to-point correspondences, they require a rather high point density to achieve precise registration. For low-cost LiDARs, this implies slow motion and long integration time. Further, none of the approaches exploit the structure inherent to human-made environments, hence missing out on possible advantages.

2.2. Point Cloud Registration Using Plane Based Correspondences

The de-facto standard for many SLAM algorithms is the Iterative-Closest-Point (ICP) algorithm [14] that employs point-to-point correspondences using closest points, as the name suggests. To overcome the requirements on point-density imposed by the point-to-point correspondences instead other correspondences are used. In human-made environments, planes are abundantly available and hence provide an attractive base for correspondences.

Pathak et al. [15] propose to reduce the complexity of the registration by using correspondences between planar patches instead of points. They demonstrate the effectiveness of their approach even with noisy data in cluttered environments. However, their approach is designed for data acquired in stop-scan-go fashion and not for mobile mapping applications. As they use a region growing procedure with randomized initialization for detecting planar patches the distortions in the data introduced by pose uncertainties are likely to effect the shape of the planar patches and lead to faulty correspondences.

Förster et al. use the planarity of human-made environment successfully in [16]. They register point clouds using plane-to-plane correspondences and include uncertainty measures for the detected planes and the estimated motion. Thereby, they propose a costly exact algorithm and cheaper approximations that yield high-quality maps. Favre et al. [17] use point-to-plane correspondences after preprocessing the point clouds using plane-to-plane correspondences to register two scans with each other successfully.

Both approaches use plane-to-plane correspondences to pre-register the scans. However, for pre-registration the classical point-to-point registration is also very effective. One advantage that point-to-point correspondences have over plane-to-plane correspondences is that they do not require a long stop in order to obtain enough points to detect planes in each pose. For plane-to-plane correspondences, this is necessary to gather enough data to measure planes in each scan robustly. In particular this pause is needed when the field-of-view (FOV) is limited, as the detected planes are thin slices of the true planes which are difficult to find correct correspondences for. The resulting scan procedure is stop-scan-and-go. In particular, for the application of a spherical robot this standstill in each pose cannot be guaranteed or even approximated, making continuous-time approaches using point-to-plane correspondences the method of choice.

In the following, we propose this combination of point-to-point based pre-registration followed by a point-to-plane based optimization.

3. Registration Algorithm

Consider a line scan as the smallest chunk of range data we obtain from the scanner device driver. In the case of a SICK LMS1xx it is a line, and in case of a Livox scanner it has a flower shape. More details are provided in the following sections. We start with transforming each line scan into the project coordinate system, which is defined by the pose of the first acquired line scan. For map improvement, the individual scans need to be registered to another. We propose an algorithm that consists of multiple steps outlined in algorithm 1. Based on ideas described in [18] we first pre-register the scans and then further improve the overall map by exploiting the fact that human-made environments often consist of planes. We then find the planes in the pre-registered point cloud and then optimize the poses associated with the scans to minimize the distance of all points to their respective planes.

3.1. Pre-Registration

For the plane extraction the linescans need to be transformed into the project coordinate system. Various pre-registration methods are suitable for this. The most simple method is to use the data from an inertial measurement unit for a coarse alignment. Alternative solutions combine

Algorithm 1: Registration algorithm for man-made environments

Result: A corrected map

1. Pre-register the scans;
 2. Extract planes from the registered map;
 3. Further improve map by solving the optimization that minimizes the distance of all points to their respective planes;
-

multiple linescans into a metascan and perform registration methods known from terrestrial laser scanning or use a small number of iterations of continuous-time SLAM approaches such as the one presented in [19]. This yields maps that can be used for plane registration and hence a point-to-plane correspondences based registration.

3.2. Plane Extraction

After having done the pre-registration, the scans are aligned well enough to make statements about the potential planes in the environment. To find the planes in the environment, a Randomized Hough transform with an accumulator ball as described in [20] is used as this method prefers dominant planes such as the building structure over smaller planar surfaces. In the next subsection, the Hesse-normal form of the plane is required. For an ideal plane the orthogonal distance from the origin $\rho_{\mathcal{P}_k}$ is computed via $\mathbf{n}_{\mathcal{P}_k} \cdot \mathbf{p}_i$, where $\mathbf{p}_i = [x_i \ y_i \ z_i]^T$ is an arbitrary point on the plane and $\mathbf{n}_{\mathcal{P}_k} = [n_{\mathcal{P}_k}^x \ n_{\mathcal{P}_k}^y \ n_{\mathcal{P}_k}^z]^T$ is the normal vector of that plane. To find this point, we first define a convex hull that encloses the plane. It is defined by the points that lie on the plane with the furthest distance to one another. We then choose the center point of the convex hull of the plane as \mathbf{p}_i .

3.3. Point-to-Plane Correspondences

There are many ways of solving the segmentation problem of assigning each point to a plane often based on the RANSAC method [21], [22] but also deep learning approaches [23]. The quality of the preregistration directly influences the quality of the plane detection and therefore the quality of the final registration result. We assume small enough errors in pre-registration to allow for plane detection. Which is why a simple algorithm is sufficient: For each plane, consider adding a correspondence for all points closer to the plane than a threshold ε . Depending on the threshold value, some points now have multiple correspondences if their distance to multiple planes is less than ε . We then omit all points with correspondences to multiple planes from the optimization to avoid introducing errors due to the plane ambiguity.

3.4. Optimization

Assuming we know now the Hesse-normal form of all planes and all assigned points to these planes, we register

the points to have a minimal distance to their respective planes. The transformation $T(\mathbf{p}_i)$ of each point \mathbf{p}_i with respect to a 6 DoF motion is described in homogeneous coordinates using the roll-pitch-yaw ($\varphi - \vartheta - \psi$) Tait-Brian angles as in [24]. Transforming the result back from homogeneous coordinates and using C_a and S_a to denote the cosine and sine of the angle in the subscript, and t_a denoting the translation along the axis in the subscript yields:

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

From this we define the function $D(\varphi, \vartheta, \psi, t_x, t_y, t_z, \mathbf{p}_i)$ that computes the distance of a point \mathbf{p}_i to its corresponding plane \mathcal{P}_k . Omitting the arguments of the function for simplicity:

$$\begin{aligned} D &= T(\mathbf{p}_i) \cdot \mathbf{n}_{\mathcal{P}_k} \\ &= n_{\mathcal{P}_k}^x (x_i C_\vartheta C_\psi - y_i C_\vartheta S_\psi + z_i S_\vartheta + t_x) \\ &\quad + n_{\mathcal{P}_k}^y (x_i (C_\varphi S_\psi + C_\psi S_\varphi S_\vartheta) \\ &\quad + y_i (C_\varphi C_\psi - S_\varphi S_\vartheta S_\psi) - z_i C_\vartheta S_\varphi + t_y) \\ &\quad + n_{\mathcal{P}_k}^z (x_i (S_\varphi S_\psi - C_\varphi C_\psi S_\vartheta) \\ &\quad + y_i (C_\psi S_\varphi + C_\varphi S_\vartheta S_\psi) + z_i C_\varphi C_\vartheta + t_z) \\ &\quad - \rho_{\mathcal{P}_k} \end{aligned} \quad (2)$$

This distance function is what we want to minimize for all points and their respective planes. Hence the error function E is chosen as the square of the L2-norm of the distance:

$$E = \sum_{\forall \mathcal{P}_k} \sum_{\mathbf{p}_i \in \mathcal{P}_k} \|D(\varphi, \vartheta, \psi, t_x, t_y, t_z, \mathbf{p}_i)\|_2^2 \quad (3)$$

Its gradient follows then immediately:

$$\nabla E = \sum_{\forall \mathcal{P}_k} \sum_{\mathbf{p}_i \in \mathcal{P}_k} \left[\frac{\partial}{\partial \varphi} E \quad \frac{\partial}{\partial \vartheta} E \quad \frac{\partial}{\partial \psi} E \quad \frac{\partial}{\partial t_x} E \quad \frac{\partial}{\partial t_y} E \quad \frac{\partial}{\partial t_z} E \right]^T \quad (4)$$

$$\Rightarrow \nabla E = \sum_{\forall \mathcal{P}_k} \sum_{\mathbf{p}_i \in \mathcal{P}_k} 2D(\Pi, \mathbf{p}_i) [\nabla E_\varphi \quad \nabla E_\vartheta \quad \nabla E_\psi \quad n_{\mathcal{P}_k}^x \quad n_{\mathcal{P}_k}^y \quad n_{\mathcal{P}_k}^z]^T \quad (5)$$

Where

$$\begin{aligned} \nabla E_\varphi &= n_{\mathcal{P}_k}^y (x_i [-S_\varphi S_\psi + C_\varphi C_\psi S_\vartheta] \\ &\quad + y_i [-S_\varphi C_\psi - C_\varphi S_\vartheta S_\psi] \\ &\quad - z_i C_\varphi C_\vartheta) \\ &\quad + n_{\mathcal{P}_k}^z (x_i [C_\varphi S_\psi + C_\varphi C_\psi S_\vartheta] \\ &\quad + y_i [C_\varphi C_\psi - S_\varphi S_\vartheta S_\psi] \\ &\quad - z_i S_\varphi C_\vartheta) \end{aligned} \quad (6)$$

$$\begin{aligned} \nabla E_\vartheta &= n_{\mathcal{P}_k}^x (-x_i S_\vartheta C_\psi + y_i S_\vartheta S_\psi + z_i C_\vartheta) \\ &\quad + n_{\mathcal{P}_k}^y (x_i C_\psi S_\varphi C_\vartheta - y_i S_\varphi C_\vartheta S_\psi + z_i S_\vartheta S_\varphi) \\ &\quad + n_{\mathcal{P}_k}^z (-x_i C_\varphi C_\psi C_\vartheta + y_i C_\varphi C_\vartheta S_\psi - z_i C_\varphi S_\vartheta) \end{aligned} \quad (7)$$

$$\begin{aligned} \nabla E_\psi &= n_{\mathcal{P}_k}^x (-x_i C_\vartheta S_\psi - y_i C_\vartheta C_\psi) \\ &\quad + n_{\mathcal{P}_k}^y (x_i [C_\varphi C_\psi - S_\varphi S_\vartheta S_\psi] \\ &\quad + y_i [-C_\varphi S_\psi - S_\varphi S_\vartheta C_\psi]) \\ &\quad + n_{\mathcal{P}_k}^z (x_i [S_\varphi C_\psi + C_\varphi S_\psi S_\vartheta] \\ &\quad + y_i [-S_\psi S_\varphi + C_\varphi S_\vartheta C_\psi]) \end{aligned} \quad (8)$$

195 and $\Pi = [\varphi \ \vartheta \ \psi \ t_x \ t_y \ t_z]^T$.

As the gradient is well-defined we minimize the error function with any gradient based method. The commonly used, well-known stochastic gradient descent (SGD) algorithm computes

$$\Pi_{k+1} = \Pi_k - \alpha \nabla E \quad (9)$$

where α is the learning rate. To accelerate convergence and to improve the found solution further modifications are made.

Since we have vastly different effects on the error function by each dimension, the first consideration for improving the SGD is the following: Typically, changes in orientation, i.e., the first three elements of the gradient vector $\frac{\partial}{\partial \varphi} E$, $\frac{\partial}{\partial \vartheta} E$, and $\frac{\partial}{\partial \psi} E$, have much more impact on the error function than a change in position. This is intuitively explained since translating the scan makes the error grow linearly for all points. However, when rotating the scan, points with a larger distance to the robot are moved drastically, leading to a higher sensibility on the error function. For this reason, the α applied on orientation has to be much smaller than the α applied on the position. It becomes obvious that α needs to be extended into vector form, $\boldsymbol{\alpha}$, therefore weighting each dimension differently. 200 205 210 215 220 225 230 235

Another consideration to speed up SGD is to adaptively recalculate $\boldsymbol{\alpha}$ for each iteration. We employ and modify ADADELTA as a technique to do so, which is described in detail in [25]. The main idea is the following: It extends the SGD algorithm by two terms. First, an exponentially decaying average of past gradients \mathbf{G}_k , which is recursively defined as

$$\mathbf{G}_{k+1} = \zeta \mathbf{G}_k + (1 - \zeta) \nabla E^2 \quad (10)$$

and second, an exponentially decaying average of past changes \mathbf{X}_k , which is defined as 235

$$\mathbf{X}_{k+1} = \zeta \mathbf{X}_k + (1 - \zeta) \boldsymbol{\alpha} \nabla E^2 \quad (11)$$

where $\zeta \leq 1$ is a decay constant, typically close to 1. The root mean squared (RMS) of these quantities are 240 245

$$RMS[\mathbf{G}]_k = \sqrt{\mathbf{G}_k + \varepsilon_{num}} \quad (12)$$

and

$$RMS[\mathbf{X}]_k = \sqrt{\mathbf{X}_k + \varepsilon_{num}} \quad (13)$$

where $\varepsilon_{num} > 0$ is a very small constant, typically close to 0 (Note: this is a different threshold as the one used in section 3.3). It will prevent dividing by zero in the recalculation of $\boldsymbol{\alpha}$, which is as follows:

$$\boldsymbol{\alpha}_k = \frac{RMS[X]_{k-1}}{RMS[G]_k} \quad (14)$$

For our particular application, ADADELTA behaves a little too aggressively. Despite giving a good measure on how 250 255

to adapt $\boldsymbol{\alpha}$, the algorithm sometimes overshoots, and does not converge. Therefore, we employ another scaling factor, typically not found in ADADELTA, extending eq. (14) to:

$$\boldsymbol{\alpha}_k = \boldsymbol{\alpha}_0 \cdot \frac{RMS[X]_{k-1}}{RMS[G]_k} \quad (15)$$

where $\boldsymbol{\alpha}_0$ holds the scaling factors for each dimension.

Finally, the SGD model is improved using eq. (15) and extends to

$$\Pi_{k+1} = \Pi_k - \boldsymbol{\alpha}_0 \cdot \frac{RMS[X]_{k-1}}{RMS[G]_k} \cdot \nabla E \quad (16)$$

Using this algorithm once after finding correspondences from points to planes leads to convergence to a local minimum, which is often not an optimal solution. Even if we increase the number of iterations dramatically, no better solution than the local minimum is found. That is unless you consider updating the correspondence model after i iterations of gradient descent. Re-assigning point-to-plane correspondences this way k times, with a large enough k , leads to an optimal solution after maximum $n = k \cdot i$ iterations of gradient descent.

3.5. Further Optimizations of the Algorithm

The algorithm was extended by two further optimizations, which are helpful in different scenarios.

Firstly, we introduce the possibility to choose a lock on some dimensions from being optimized by setting the corresponding α_i to zero. Although 6D optimization generally works, reducing the optimization space is particularly useful if the source of error in the system is known and a model exists. That way, e.g., as we expect a spherical robot to move on a plane the position along the axis perpendicular to the plane is constant and should not be used for optimization.

Secondly, we employ a continuous iteration, where the scans are processed sequentially. This is especially useful for mobile systems, where pose errors accumulate due to increasing tracking uncertainties. The assumption is that the error in one scan is also present in the following scan, plus some unknown new error. We eliminate the error from scan m which is also present in scan $m+1$ by applying the pose change from scan m , $\Pi_{n,m} - \Pi_{0,m}$ after n gradient descent iterations, to scan $m+1$, before restarting gradient descent. To quantify the quality of the proposed registration method, it is tested on simulated and real-world data.

4. Simulation

For the simulated data, we implemented a noisy world-robot-sensor model. The simulated sensor is modeled after a Livox-Mid100 [26] laser scanner with customizable noise level on the range measurements inside a robot with different motion capabilities, subject to noise in its pose estimation. This yields scan results similar to the ones obtained in the real world.

@Fab
ify in
ments
locks
active

4.1. Simulated Sensor

The Livox-Mid100 laser scanner consists of three Mid40 scanners arranged to provide an overall field-of-view of 98.4° (horizontal) by 38.4° (vertical). It provides 300.000 points with a range precision of 2 cm up to 20 m range [26]. A particular advantage of the Livox laser scanners is their unique scanning pattern. The scanners use two motorized, so-called Risley prisms to steer the beam deflectance. For the specific ratio of rotation speeds used in the Livox scanners, the beam traces out a non-repetitive, flower-shaped scanning pattern [27]. Since the pattern is non-repetitive, the point density increases with integration time, enabling denser measurements.

4.2. Range Noise Model

Most real laser scanners are subjected to range measurement noise that is proportional to the measured range. Therefore the simulated sensors are as well. To achieve this, we sample a noise percentage n_P from some normal distribution $\mathcal{N}(\mu, \sigma^2)$. The resulting range r given the true range r_t is then

$$r = r_t(1 + n_P) \quad (17)$$

For each measurement ray the noise percentage is sampled independently.

4.3. Pose Estimation Noise Model

Regarding the pose measurement, one cannot simply add white noise to the current pose estimate as this does not capture the integration error, which is common among inertial measurement units (IMU). Therefore we assume a disturbance torque about the two axes that lie in the ground plane. This is equivalent to a case where a slightly shifted ground plane or an unbalanced locomotion system of the robot is present.

Accordingly, a spherical robot that is assumed to be translating exactly along one axis by rotating about one other axis at an angular velocity ω experiences random disturbance torques at each time instant that accelerate the rotation of the sphere about the respective axes. This additional motion is determined by integrating the disturbance torques, which are sampled from some normal distribution $n_\phi, n_\psi \in \mathcal{N}(\mu, \sigma^2)$ each. The pose update after a discrete time step Δt in the simulation is therefore given by:

$$\begin{bmatrix} \varphi \\ \vartheta \\ \psi \\ x \\ y \\ z \end{bmatrix}_{k+1} = \begin{bmatrix} \varphi \\ \vartheta \\ \psi \\ x \\ y \\ z \end{bmatrix}_k + \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ R \\ 0 \\ 0 \end{bmatrix}}_{\text{Difference in angular velocity}} \omega \Delta t + \Delta t \sum_{i=0}^k \underbrace{\begin{bmatrix} n_\varphi[i] \Delta t \\ 0 \\ n_\psi[i] \Delta t \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{(18)}$$

where Δt is generally chosen as some small value (e.g. 1 ms)

Assuming intended locomotion along the z -axis and unintended locomotion along the x -axis where R is the sphere radius. No upwards movement is possible since the robot moves on a flat ground, hence the entry is zero.

4.4. Simulated Datasets

Figure 1 shows a noise-free trajectory through a simulated environment and a trajectory with noisy sensors while figure 2 shows a rendering of the simulated robot as well as the simulation sequence. In the noise-free case, the trajectory slightly bends in the direction of disturbance torque sideways while also varying in velocity in the intended direction of travel. In the noisy case, the ideal straight trajectory is assumed. Hence the planes enclosing the room are sensed multiple times. In particular, we see that the trajectory of the robot leads through one of the sensed planes. Also, the further the sensed points are, the noisier they appear, which is consistent with the larger influence of the pose error at higher distances.

5. Experimental Setup

We also tested our algorithm on a real-world dataset that adheres to different motion profiles and laser scanners type with different scanning patterns than the simulated data.

The dataset is collected by a line scanner, in particular a SICK LMS141 industrial scanner, inside a sphere of a diameter of 20 cm that lies on a floating desk, which is air-pressurized, such that the sphere is hovering. On this floating desk, the sphere can rotate freely about all axis while being fixed in position. Hence, in this experiment the optimization space can be reduced to a rotation. Further, without the motion, the 2D laser cannot obtain a 3D model from the environment, thus requiring registration. The sphere is equipped with a low-cost IMU, i.e., a PhidgetSpatial Precision 3/3/3, to estimate the orientation, a battery pack and a raspberry pie for processing. This IMU provides a precision of 76.3 µg in linear acceleration, 0.02°s^{-1} in angular velocity about the x and y axis, and $0.013^\circ \text{s}^{-1}$ in angular velocity about the z axis [28].

Figure 3 shows the experimental setup. The left column of figure 5 shows the pre-registered resulting point cloud. We see that the point cloud is rather noisy, and in particular, the walls are sensed multiple times, hence appearing very blurry.

6. Results

6.1. Simulated Results

The plane-based registration is applied to the simulated dataset with noisy pose and range measurements (cf. Figure 1) without further processing. Assuming this represents a coarsely pre-registered 3D point cloud, the distances to the ground truth were evaluated before and after the plane-based registration. Figure 4

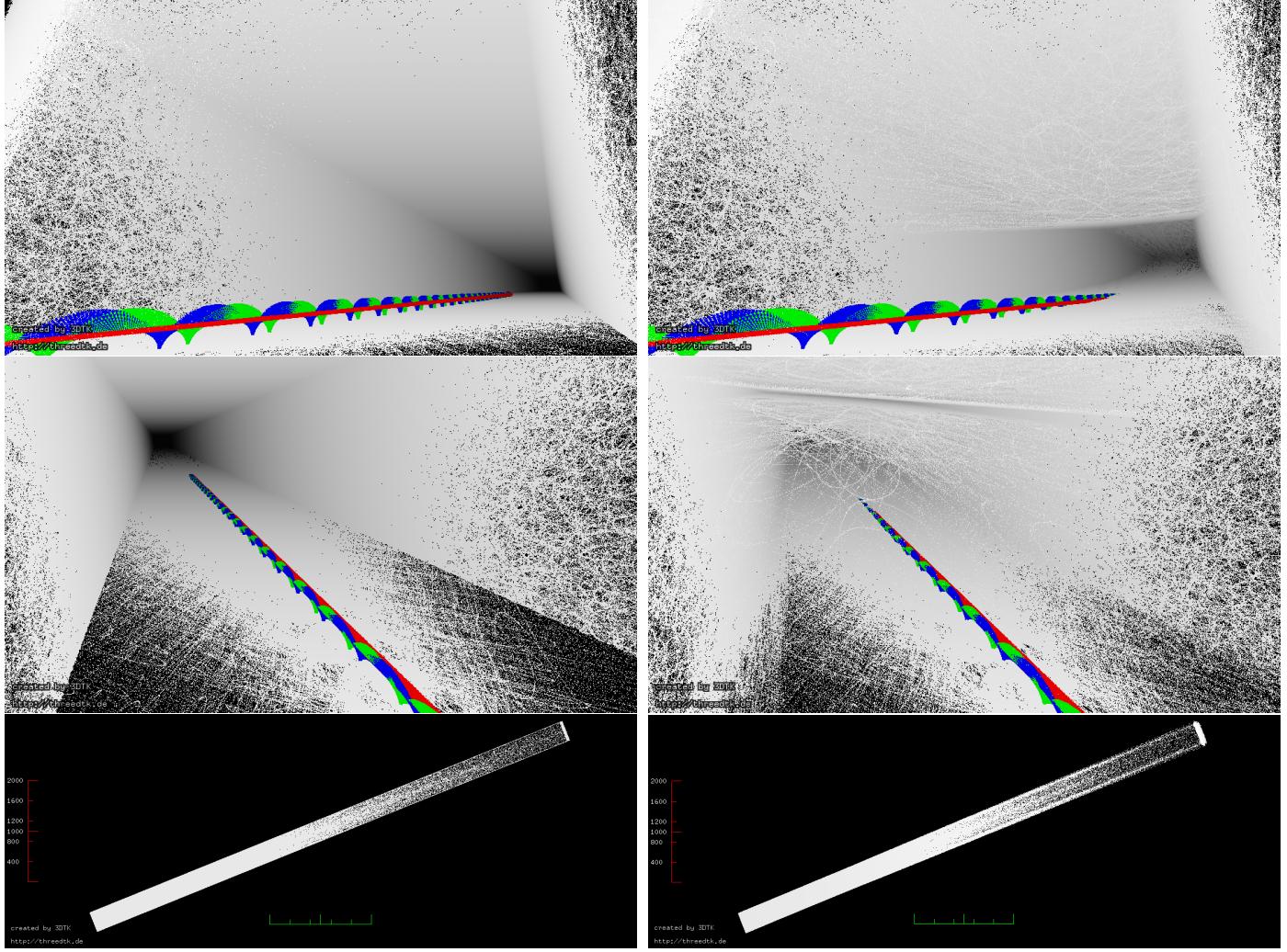


Figure 1: Two simulated datasets of a long hallway of size $4\text{ m} \times 3\text{ m} \times 100\text{ m}$. Ideal noise free (left), noisy pose ($\mathcal{N}_\varphi = \mathcal{N}_\psi(\mu = 0.0001, \sigma = 0.00001)$) and noisy range measurements ($\mathcal{N}_r(\mu = 0, \sigma = 0.001)$) (right). The normal distributions are define in such a way, that in the resulting dataset plane detection is still possible.

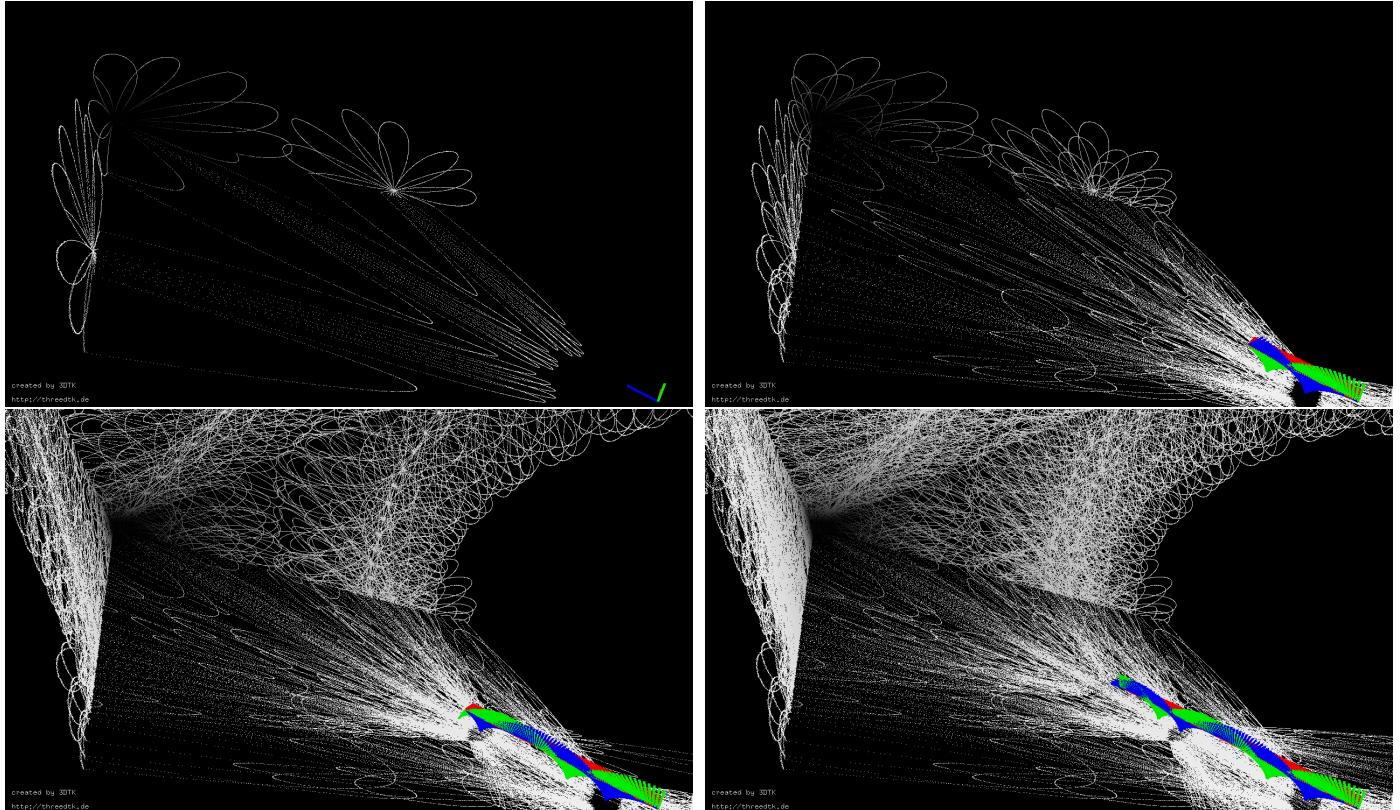


Figure 2: Top: Initial sequence of the simulation of the ideal dataset showing the simulated motion of the robot. Below: Rendering of the simulated robot, a Livox laser scanner inside a spherical shell.



Figure 3: Data acquisition using floating sphere. Top: A sequence of orientations the sphere assumes during data acquisition. Bottom: Hardware setup of the sphere.

	P90	P95	P98
Uncorrected	372.1 cm	553.4 cm	827.9 cm
Corrected	35.9 cm	64.1 cm	122.8 cm

Table 1: Comparison of point-distances in the uncorrected and corrected simulated dataset.

Reupload matching process

shows the different point-to-point distances.

Before the registration, the corridor is only represented acceptably in the front part. The further into the corridor, i.e., the longer the robot accumulates errors, the more imprecise the data becomes. Finally, we see that many points exceed the threshold of 1 m and thus being mapped to the same color value. After registration, we see that, qualitatively, the ideal corridor was nearly restored from the noisy data. In particular a very large portion of points (90%) have distances of less than 35.9 cm. Table 1 shows the comparison of further percentiles of both datasets.

Further, the square and straight shape of the corridor is restored well, and especially the large amount of points with an errors of greater than 1 m is removed. Any such errors tend to occur at the back and the front of the corridor where the measured range is the largest hence has the largest contribution of the range error.

6.2. Floating Sphere Results

Figure 5 shows the results obtained before and after employing the plane based registration on the dataset ac-

quired by the floating sphere. The pre-registration is obtained by determining the orientation of the sphere via Madgwick filtered [29] IMU measurements. To increase the number of possible point-to-plane correspondences we combine ten temporally successive scans into one meta-scan, which is then globally registered. For this we must assume constant pose over this time duration. However, given the high measurement frequency of the Lidar this introduces only a small error. This does not only speed up convergence due to the proportional effect on the error function but also decreases the risk of transforming the scan in a senseless way. Transformations like these happen in particular for a small collection of points as outliers have more influence.

After the registration, the walls of the room are significantly more prominent in the point cloud. In particular, the noise in the top left corner of the top view (cf. figure 5) has been visibly reduced. Further, the deviation of points around the walls is notably smaller as the points are moved on the plane.

However, some imprecisions remain. In particular, at the initially very noisy top left corner of the top view, a few points cannot be matched to the plane. The scans that cause these imprecisions are problematic as they measure a number of plane-like structures in the interior of the room. When relabeling those scans, the interior points are associated with a plane, and hence the corresponding distance is minimized.

7. Conclusion

In this paper, we proposed an approach to mobile mapping using a spherical robot. Using the inherent rotation of the sphere used for locomotion to also rotate a laser scanner also for measurements of the entire environment. We propose to post-process the acquired data using a novel registration method for man made environments that exploits the structure of those environments. Given that in human-made environments straight planes are abundantly available, we employ point-to-plane correspondences to improve a pre-registered 3D point cloud. We have evaluated the procedure on a simulated dataset and on an experimentally acquired dataset with different motion profiles. In this evaluation, we have shown that the procedure improves both datasets and yields maps that better resemble human-made, structures. In particular, the qualitative structure of the environments is reconstructed well. In the resulting maps, the parallel walls are clearly improved.

However, as section 5 shows, some outliers remain in the registered point cloud due to the simplistic point-to-plane correspondence model. This is to be improved in future studies. One idea to do so is to employ a local clustering technique onto the individual scans before registering them globally. Points from the scan with a similar normal get clustered together if their distance is small enough. This way, a cluster represents a local plane while

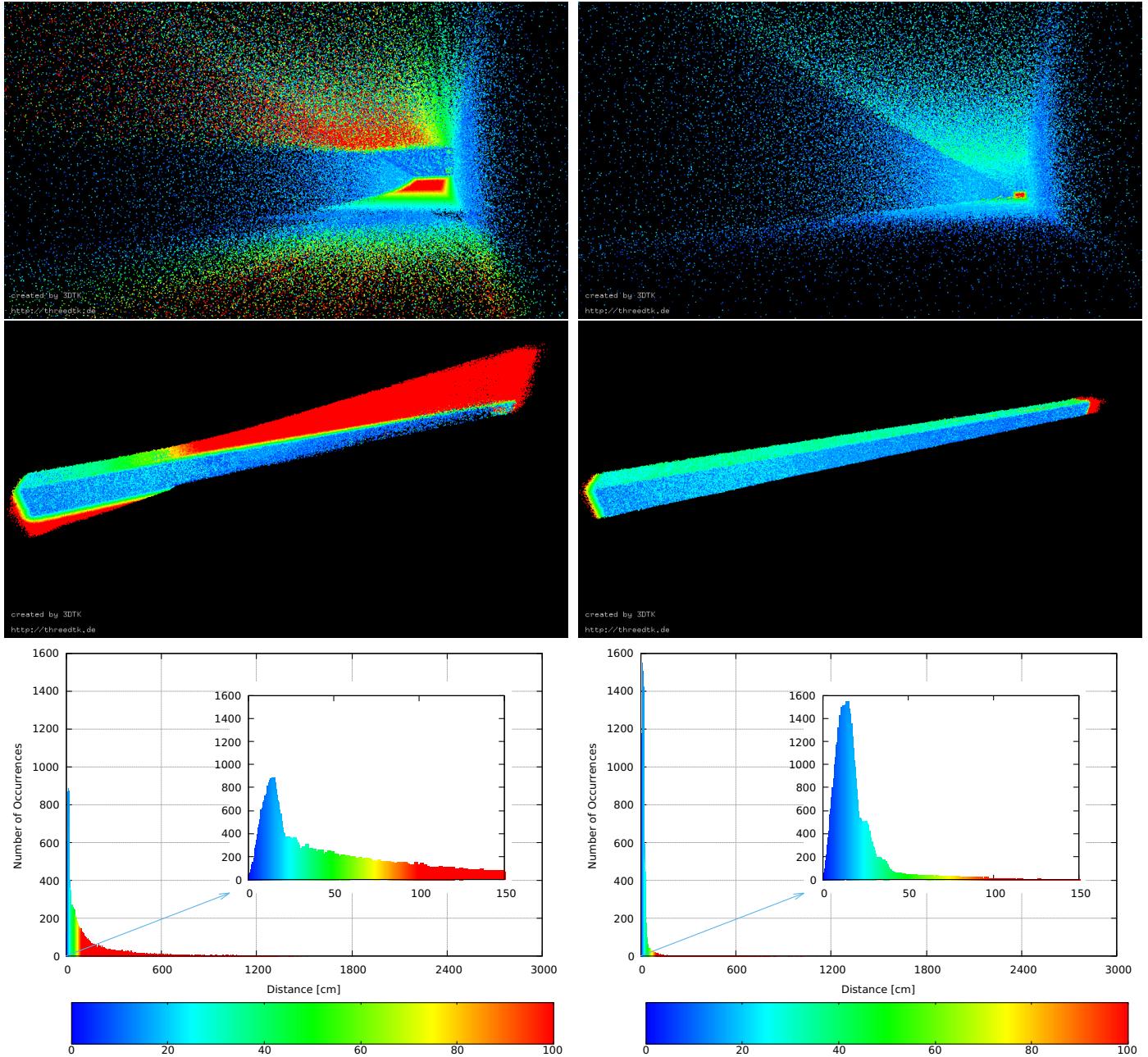


Figure 4: Evaluation of point distances before (left) and after (right) the plane-based registration on a simulated dataset. Lateral images always have the same orientation. A maximal distance of 30 m is set, such that all points that display a higher distance value are excluded from the analysis. Both point-clouds were reduced before evaluating point-to-point distances to the ground-truth. Further, the color-space maps all values with a distance greater than 1m to the same color. The top two columns show a heat map of distances, while the bottom shows the corresponding histogram. The color mapping is equivalent in both. An animation of the matching process is given at <https://youtu.be/0ps5Pg4qo4E>.

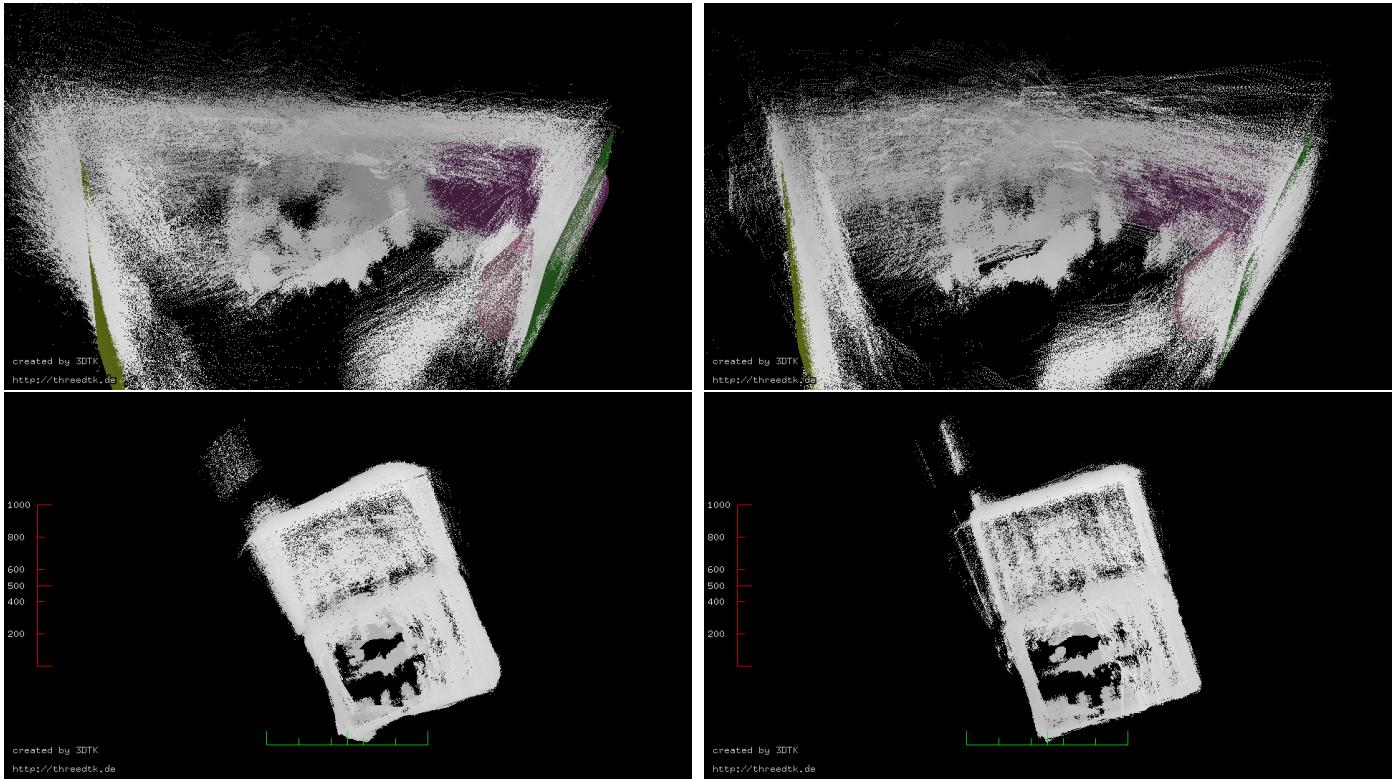


Figure 5: The point cloud acquired by the floating sphere before (left) and after (right) applying the plane based registration. View from the interior that shows the used planes (top) and a birds-eye view (bottom). An animation of the registration process is given at https://youtu.be/8XdIUN_9VpY.

at the same time storing all the points that belong to that plane. Point-to-plane correspondences then arise from the intersection of local clusters with global planes.

Right now, not all steps in the procedure are autonomous, in particular the parameter tuning. In the future, one goal is to increase the autonomy of the system. One approach is to introduce soft-locks for the optimization dimensions. I.e., instead of locking some dimensions entirely from being used for optimization, they are weighted based on the dynamics of the system that encode which noise source is more likely. Currently, we are assembling the sphere of Figure 2 to test it in real environments. Unfortunately, this sphere will be too large to fit our floating setup, thus it will be tested in an office like environment. Furthermore, we will test our rolling and scanning spheres on a rail system to achieve repeatable trajectories.

Acknowledgement

The authors thank the Cylon team (Ignacio Dorado Llamas, Timothy Randolph, Camilo Andres Reyes Mantilla) for designing and building the prototype and Timo Burger for acquiring the floating table data set. Special thanks to Dieter Ziegler and Sergio Montenegro for supporting our work. We acknowledge funding from the ESA Contract No. 4000130925/20/NL/GLC for the study “DAEDALUS – Descent And Exploration in Deep Auton-

omy of Lava Underground Structures” within the Open Space Innovation Platform (OSIP) lunar caves-system and the Elite Network Bavaria (ENB) for providing funds for the academic program “Satellite Technology”.

References

- [1] M. Bosse, R. Zlot, P. Flick, Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping, *IEEE Transactions on Robotics* 28 (5) (2012) 1104–1119. doi:[10.1109/TRO.2012.2200990](https://doi.org/10.1109/TRO.2012.2200990).
- [2] V. V. Lehtola, J.-P. Virtanen, M. T. Vaaja, H. Hyppä, A. Nüchter, Localization of a Mobile Laser Scanner via Dimensional Reduction, *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)* 121 (2016) 48–59. doi:[10.1016/j.isprsjprs.2016.09.004](https://doi.org/10.1016/j.isprsjprs.2016.09.004). URL <https://robotik.informatik.uni-wuerzburg.de/telematics/download/jprs2016.pdf>
- [3] D. Borrmann, S. Jörissen, A. Nüchter, RADLER – A RADial LasER scanning device, in: Proceedings of the International Symposium on Experimental Research, Buenos Aires, Argentina, 2020, pp. 655–664. doi:[10.1007/978-3-030-33950-0_56](https://doi.org/10.1007/978-3-030-33950-0_56).
- [4] H. A. Lauterbach, D. Borrmann, R. Heß, D. Eck, K. Schilling, A. Nüchter, Evaluation of a backpack-mounted 3d mobile scanning system, *Remote Sensing* 7 (10) (2015) 13753–13781. doi:[10.3390/rs71013753](https://doi.org/10.3390/rs71013753). URL <https://www.mdpi.com/2072-4292/7/10/13753>
- [5] Leica, Leica pegasus:backpack, www.leica-geosystems.com/de/Leica-PegasusBackpack_106730.htm (May 2015).
- [6] Boston Dynamics, Spot robot, <https://www.bostondynamics.com/spot> (2021).

- [7] P. Fankhauser, M. Bloesch, M. Hutter, Probabilistic terrain mapping for mobile robots with uncertain localization, *IEEE Robotics and Automation Letters* 3 (4) (2018) 3019–3026. doi:10.1109/LRA.2018.2849506.
- [8] A. P. Rossi, F. Maurelli, V. Unnithan, H. Dreger, K. Matthevos, N. Pradhan, D.-A. Corbeanu, R. Pozzobon, M. Massironi, S. Ferrari, C. Pernechele, L. Paoletti, E. Simioni, P. Maurizio, T. Santagata, D. Borrmann, A. Nüchter, A. Bredenbeck, J. Zevering, F. Arzberger, C. A. R. Mantilla, Daedalus – descent and exploration in deep autonomy of lava underground structures, *Tech. Rep.* 21, Institut für Informatik (2021). doi:10.25972/OPUS-22791.
- [9] J. Guo, X. Chen, S. Guo, J. Xu, Study on map construction of spherical robot based on statistical filtering, in: 2020 IEEE International Conference on Mechatronics and Automation (ICMA), 2020, pp. 1269–1274. doi:10.1109/ICMA49215.2020.9233654.
- [10] A. Nüchter, K. Lingemann, J. Hertzberg, H. Surmann, 6D SLAM – 3D Mapping Outdoor Environments, *Journal of Field Robotics (JFR)*, Special Issue on Quantitative Performance Evaluation of Robotic and Intelligent Systems 24 (8–9) (2007) 699–722. doi:10.1002/rob.20209. URL <https://robotik.informatik.uni-wuerzburg.de/telematics/download/jfr2007.pdf>
- [11] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, J. Hertzberg, Globally consistent 3d mapping with scan matching, *Journal Robotics and Autonomous Systems (JRAS)* 56 (2) (2008) 130–142. doi:10.1016/j.robot.2007.07.002. URL <https://robotik.informatik.uni-wuerzburg.de/telematics/download/ras2007.pdf>
- [12] J. Zhang, S. Singh, Loam: Lidar odometry and mapping in real-time, in: *Robotics: Science and Systems Conference (RSS)*, 2014. doi:10.15607/RSS.2014.X.007.
- [13] D. Droschel, S. Behnke, Efficient continuous-time slam for 3d lidar-based online mapping, 2018 IEEE International Conference on Robotics and Automation (ICRA) doi:10.1109/icra.2018.8461000. URL <http://dx.doi.org/10.1109/ICRA.2018.8461000>
- [14] P. J. Besl, N. D. McKay, A method for registration of 3-d shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (2) (1992) 239–256. doi:10.1109/34.121791.
- [15] K. Pathak, A. Birk, N. Vaskevicius, J. Poppinga, Fast registration based on noisy planeswith unknown correspondences for 3d mapping, *IEEE Transactions on Robotics* 26 (3) (2010) 424–441. doi:10.1109/TRO.2010.2042989.
- [16] W. Förstner, K. Khoshelham, Efficient and accurate registration of point clouds with plane to plane correspondences, in: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), 2017, pp. 2165–2173. doi:10.1109/ICCVW.2017.253.
- [17] K. Favre, M. Pressigout, E. Marchand, L. Morin, A Plane-based Approach for Indoor Point Clouds Registration, in: *ICPR 2020 - 25th International Conference on Pattern Recognition*, Milan (Virtual), Italy, 2021. URL <https://hal.archives-ouvertes.fr/hal-03108891>
- [18] J. Elseberg, D. Borrmann, K. Lingemann, A. Nüchter, Non-rigid registration and rectification of 3d laser scans, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 1546–1552. doi:10.1109/IROS.2010.5652278.
- [19] J. Elseberg, D. Borrmann, A. Nüchter, Algorithmic solutions for computing accurate maximum likelihood 3D point clouds from mobile laser scanning platforms, *Remote Sensing* 5 (11) (2013) 5871–5906. doi:10.3390/rs5115871. URL <https://robotik.informatik.uni-wuerzburg.de/telematics/download/remotesensing2013.pdf>
- [20] D. Borrmann, J. Elseberg, A. Nüchter, K. Lingemann, The 3D Hough Transform for Plane Detection in Point Clouds – A Review and A new Accumulator Design, *Journal of 3D Research* 2 (2) (2011) 1–13. doi:10.1007/3DRes.02(2011)3. URL <https://robotik.informatik.uni-wuerzburg.de/telematics/download/3dresearch2011.pdf>
- [21] R. Honti, J. Erdélyi, A. Kopacik, Plane segmentation from point clouds, *Pollack Periodica* 13 (2018) 159–171. doi:10.1556/606.2018.13.2.16.
- [22] B. Gaspers, J. Stückler, J. Welle, D. Schulz, S. Behnke, Efficient multi-resolution plane segmentation of 3d point clouds, 2011, pp. 145–156. doi:10.1007/978-3-642-25489-5_15.
- [23] F. Engelmann, T. Kontogianni, J. Schult, B. Leibe, Know what your neighbors do: 3d semantic segmentation of point clouds, *CoRR* abs/1810.01151. arXiv:1810.01151. URL <http://arxiv.org/abs/1810.01151>
- [24] J. Diebel, Representing attitude: Euler angles, unit quaternions, and rotation vectors, *Matrix* 58 (15–16) (2006) 1–35.
- [25] M. D. Zeiler, Adadelta: An adaptive learning rate method (2012). arXiv:1212.5701.
- [26] Livox, Livox Mid40 and Mid100, <https://www.livoxtech.com/mid-40-and-mid-100> (2021).
- [27] ThorLabs, Application note risley prism scanner, https://www.thorlabs.com/images/tabcimages/Risley_Prism_Scanner_App_Note.pdf.
- [28] Phidget, PhidgetSpatial Precision 3/3/3, <https://www.phidgets.com/?&prodid=32> (2021).
- [29] S. Madgwick, An efficient orientation filter for inertial and inertial/magnetic sensor arrays, *Report x-io and University of Bristol (UK)* 25 (2010) 113–118.