2022 IEEE International Symposium on
Safety, Security, and Rescue Robotics (SSRR)
November 8-10, 2022. Seville, Spain

Mobile 3D scanning and mapping for freely rotating and vertically descended LiDAR

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

2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR) | 978-1-6654-5680-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/SSRR56537.2022.10018586

***Abstract*— Situational awareness in search and rescue missions is key to successful operations, e.g., in collapsed buildings, underground mine shafts, construction sites, and underwater caves. LiDAR sensors in robotics play an increasingly important role in this context, as do robust and application-specific algorithms for simultaneous localization and mapping (SLAM). In many of these scenarios mapping requires the utilization of a vertically descended scanning system. This work presents a mobile system designed to solve this task, including a SLAM approach for descended LiDAR sensors with small field of view (FoV), which are in uncontrolled rotation. The SLAM approach is based on planar polygon matching and is not limited to the presented scenario. We test the system by lowering it from a crane inside a tall building at a fire-fighter school, applying our offline SLAM approach, and comparing the resulting point clouds of the environment with ground truth maps acquired by a terrestrial laser scanner (TLS). We also compare the SLAM approach to a state-of-the-art approach with respect to runtime and accuracy of the resulting maps. Our solution achieves comparable mapping accuracy at 0.2% of the runtime.**

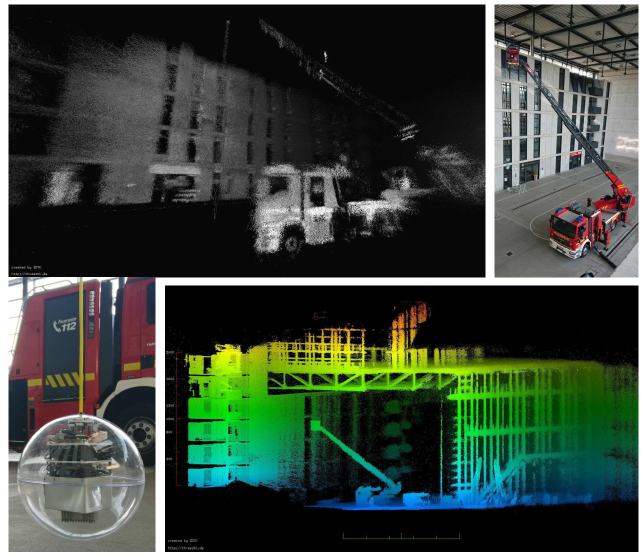


Fig. 1: Post-processed 3D point clouds aquired by a sensor that is rotating freely while beeing descended from the aerial ladder of a fire truck. Prior pose estimates are available from an IMU and an angular encoder, which encodes the rotation of the cable reel. A video of the point cloud is available at `https://youtu.be/N6FGd4y6bJw`.

I. Introduction

Any search and rescue mission relies on situational awareness, which means that the structure and geometry of the environment must be monitored. Often such missions include descending a mobile mapping system down a pit with a crane, e.g., in underground mine shafts, collapsed buildings, construction sites, underwater caves, bridges, dams, and even exoplanetary subsurface exploration [1]. In these scenarios many common pose estimation techniques that are established in different domains, e.g., autonomous cars, drones, are not available. For example, global navigation satellite systems (GNSS) depend on signal reception, and visual-inertial tracking with cameras works only with sufficiently good lightingg conditions. Hence a popular choice in the robotics community are active LiDAR sensors, as they are independent of lighting conditions and inherently yield a precise 3D representation of the environment. However, to aquire a point-cloud scan, the sensor head usually moves, which takes a non-negligible amount of time. If the system itself is moving in the process, this leads to an effect known as motion distortion. Adversely, especially in search and rescue missions, sensor trajectories cannot be ensured to be in a slow and controlled fashion as, e.g., in a lab environment. As a result, the faster and uncontrolled motion leads to less overlap between subsequent scans and more motion distortion, which makes laser-based SLAM especially difficult. When descending an unactuated probe with a cable from a crane, the internal cable twist leads to unrestricted rotation of the probe around the descending axis. The larger the distance of descent, the more oscillation is introduced to the system, and the longer the descending duration is, the more IMU drift accumulates. Furthermore, we deliberately refrain from utilizing cameras, GNSS, or magnetometer measurements, but focus on LiDAR and IMU data. Thus, the problem we address is an LiDAR-Inertial SLAM problem, using only data from a laser scanner, IMU, and a rotational encoder on the cable reel to estimate the traveled distance.

The main contribution of this work is the development and evaluation of such a mapping system, which we test by descending it from an aerial ladder of a fire truck (cf. Figure 1). The second contribution of this paper is a revision of the SLAM system from our previous work [2]. We substitute the local planar clustering (LPC) with a different plane detection framework according to [3] Furthermore, we implement a dynamic global plane model that builds up sequentially as new range measurements arrive, instead of being initialized

All authors are with Informatics VII, University of Würzburg, Germany. Contact: `fabian.arzberger@uni-wuerzburg.de`

We acknowledge funding from the ESA Contract No. 4000130925 /20/NL/GLC for the “DAEDALUS” [1] Open Space Innovation Platform (OSIP) lunar caves-system study and the Elite Network Bavaria (ENB) for providing funds for the academic program “Satellite Technology”.

only once. The key to situational awareness is creating precise maps as fast as possible, thus we compare the accuracy and runtime of our SLAM approach to an existing state-of-the-art method. In the evaluation, we apply both methods offline to the same input and match the resulting maps against a high-precise ground-truth point cloud, available from a terrestrial laser scanner (TLS). We compare the accuracy of the resulting maps, as well as the algorithms' runtime. Note that in this evaluation the focus lies only on the resulting map and not on the pose estimates. Recording ground truth trajectories and evaluating the pose estimates is a task for future work.

II. Related Work

The system presented here has been part of the "DAEDALUS" project [1]. The project originated from the European Space Agency's (ESA) Open Space Innovation Platform (OSIP). In their Concurrent Design Facility (CDF) study the consortium developed a mission to autonomously explore underground caves and lava tubes on the moon [4]. The mission uses a spherical mapping robot that descends the entrance pit with the help of a crane. Another example where a tethered vehicle is used for the mapping of extremely steep environments is "TReX" [5]. It is a four-wheeled rover-like robot equipped with a rotating 2D laser scanner that has been tested in an open-pit mine. Prior pose estimates are available either from a constant-velocity or visual-odometry (VO) aided approach. Unlike our approach, their work accounts for motion distortion in the LiDAR scans, yet the operations and post-processing itself took several days for a total distance of 1 km. The main drawback is the impact on the walls, increasing the risk of debris loosening and falling off. More examples of suspending tethered mobile mapping systems exist, e.g., the underwater explorer "UX-1" [6]. In [7], terrestrial laser scanners (TLS) were utilized for mapping vertical mine shafts. A recent commercial solution for vertical shaft mapping is [8] provided by Geoslam. They employ data format compatibility to existing mining software such as Deswik and Micromine.

In terms of laser-based SLAM, many algorithms for six degrees of freedom (6 DOF) are available, often based on the well-known Iterative-Closest-Point (ICP) algorithm [9]. Lu and Milos [10] derive a graph-based 2D variant that considers all scans simultaneously in a global fashion. Later, their approach was adopted for 3D point clouds in 6 DOF [11] and extended further to a semi-rigid continuous time method [12]. We use this method to compare it with the proposed SLAM approach. Another recent continuous-time graph-based framework is "IN2LAAMA" [13], which can 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 mechanically actuated multi-channel LiDAR devices and has been extensively tested with a Velodyne VLP-16. We note the utilization of a solid-state LiDAR in this work, which got received a lot of attention in the past years due to "their superiority in cost, reliability, and [...] performance against the conventional mechanical spinning LiDARs [...]" [14]. 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, in particular, small FOV, an irregular scanning pattern with non-repetitive scanning, making feature extraction more difficult, and increased motion blur. There also exist continuous-time graph-based *online* methods such as [15], which organize and optimize the system poses using a multi-level hierarchical graph. This method achieves comparable accuracy as similar offline-batch methods through multiresolution surfel-based registration. However, the approach is also optimized for traditional multi-channel laser scanners and has been tested on carefully controlled micro aerial vehicles (MAVs). Alismail and Browning [16] provide a marker-less calibration procedure for spinning actuated laser scanners, where the extrinsic parameters concerning the spinning axis are estimated. In this initial study, we present the results without fine calibration of extrinsic as the constant calibration errors are orders of magnitude lower than the errors introduced by motion distortion.

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 planar structures are abundantly available in man-made environments. In [17], 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 [18]–[21]. Two more recent SLAM approaches which specialize more on solid state LiDAR, i.e., the massivley produced LiVOX devices, are "Loam-livox" [14] and "Livox-mapping" [22]. The former is based on the well-known LOAM [23] algorithm, while the latter is provided directly by Livox. Both have been specially optimized for small FOV devices and offer a feature extraction approach that is suitable for the never repeating, flower-shaped scanning pattern. In [24], the authors employ a continuous-time model of Livoxs' solid-state LiDARs scanning patterns - the Mid-40, Horizon, and Avia. Using this model in their SLAM, they optimize every single point and thus, account for motion distortion in the scans. The approach runs in real-time on an Nvidia Jetson AGX.

III. System overview

The sensor setup is shown in Figure 2. We use the Livox Mid-100 solid-state LiDAR sensor. It produces 300.000 points per second using three beams that scan in a non-repetitive, flower-shaped fashion, thus point density increases over time. Each beam generates 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°.

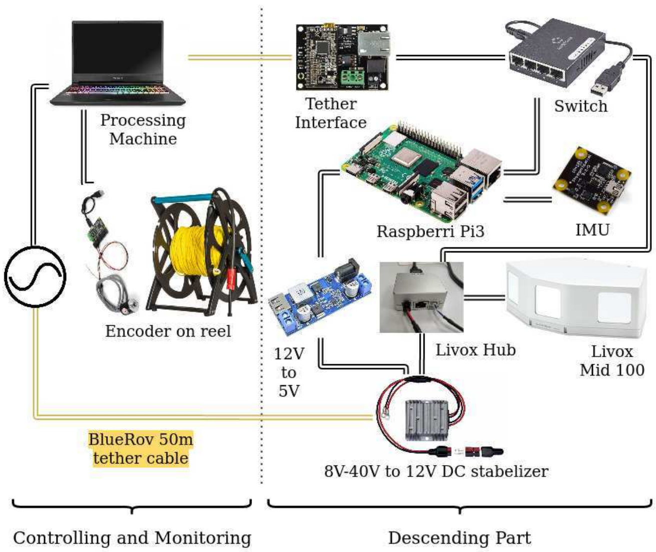


Fig. 2: Overview over the system hardware and connections. The yellow connection represents the same physical cable. It is a BlueRov 50m tether cable that provides power- and data-connection via multiple wires.

The precision at 20 meter scanning distance is 2 cm and the angular accuracy is 0.1 °. A ROS installation on a Raspberry Pi 4 is used for onboard controlling and recording data. Inertial measurements are performed by a PhidgetSpatial Precision 3/3/3 IMU. We connect the system to an outsourced processing machine via a 50 m tear-resistant tether cable (Fathom ROV Tether by BlueRobotics) which was rolled around a cable reel to perform the descending and ascending movement (cf. left image of Figure 4). A PhidgetEncoder HighSpeed spin encoder measures the rotation of the coil which directly corresponds to the height of the system. As the hardware used in this work is widely available for consumers, we consider the setup to be in the low-cost segment.

IV. SLAM APPROACH

We initially proposed a version of our SLAM approach in [2]. 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 derivations of the homogeneous local to global transformation, as well as the error function stay the same (see [2] for further details).

A. Optimization

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:

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

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 $\mathbf{n}_{\rho_k}$ is the normal vector of the plane and $\mathbf{a}_{\rho_k}$ is its supporting point. The problem we must solve is an optimization problem, where the sum of weighted distances of all valid points to their respective plane must be minimized. Thus, the error function $E(T)$ is:

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

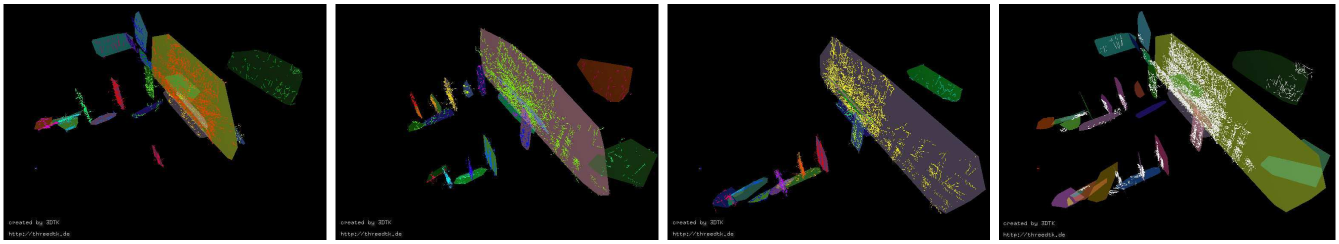
where the weights ω_k^i are defined by a cost function, and D_k^i denotes the distance from a point $\mathbf{p}_i$ to the plane ρ_k. Minimizing $E(T)$ yields the transformation required that optimally matches all points $\mathbf{p}_i$ to their respective plane ρ_k. Note that in our previous work [2], we did not use a cost function. Babin et al. [25] provide a descriptive table of robust cost functions, yet we decide that the L1-norm is the best compromise between simplicity and robustness. Therefore, we set the weight $\omega_k^i = |D_k^i|^{-1}$. Note that point-to-plane correspondences ($\mathbf{p}_i \in \rho_k$) have to be available, which we establish by matching polygons similar to [2]. As in our previous work [2], we minimize Equation (2) using AdaDelta [26]. The method is based on gradient descent and accelerates convergence in the dimensions with large residuals. Let $\mathbf{\Pi}_j = (t_x, t_y, t_z, \phi, \theta, \psi)^\tau$ be the pose corresponding to the j-th scan. Then, AdaDelta computes the optimal pose estimation by iterating

$$\mathbf{\Pi}_{j+1} = \mathbf{\Pi}_j - \alpha_\mathbf{0} \frac{\sqrt{\mathbf{X}_{j-1} + \varepsilon}}{\sqrt{\mathbf{G}_j + \varepsilon}} \cdot \nabla \mathbf{E}. \tag{3}$$

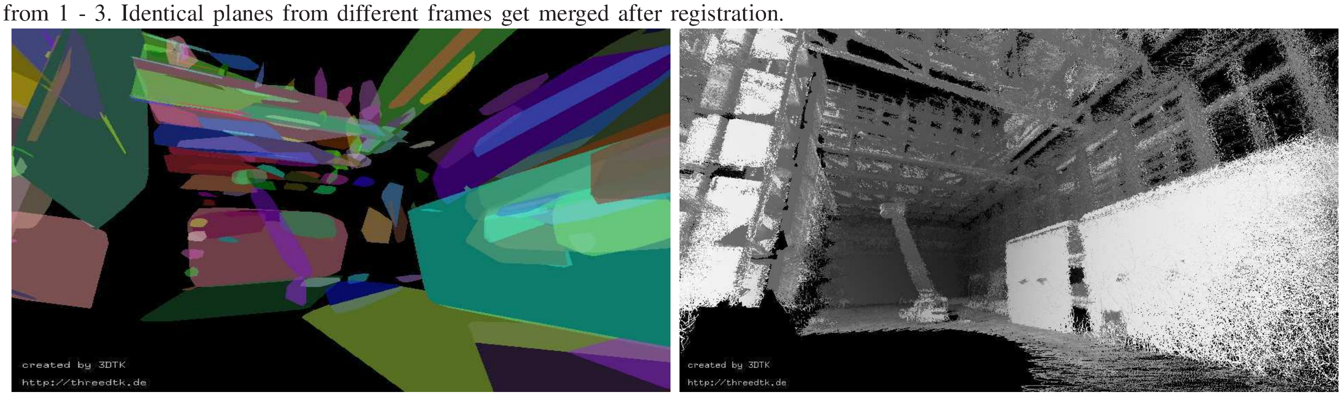
until convergence, where ε is an arbitrary number close to zero, $\alpha_\mathbf{0}$ is the convergence initializion vector, $\mathbf{X}_j$ is an exponentionally decaying average of pose changes, and $\mathbf{G}_j$ is an exponentionally decaying average of the weighted gradient vector $\nabla \mathbf{E}$. See [2] for further details and the analytical jacobians.

B. Dynamic Plane Model

In our previous work [2] we rely on local planar clustering (LPC) to identify planes in each scan, as well as the points that belong to those planes. LPC 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 this work we replace LPC with [3] to identify planes in each scan. The new approach is based on a randomized version of the well-known Hough transformation. The 3D Hough transform maps point data to parameter space, i.e., the Hough space. In this space, the parameters correspond to plane representations, therefore finding local maxima in the parameter space yields a plane model representation of the input points. After a plane has been identified in the Hough space, all points associated with that plane are considered, and their convex hull is computed. However, instead of deleting every point close to the newly identified plane (see



(a) From left to right: (1 - 3): Subsequent point cloud 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. Identical planes from different frames get merged after registration.



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

Fig. 3: Illustration of how the global plane model is obtained and sequentially extended from individual measurements.

Algorithm 5 in [3]), we save them in a point cluster and link them to their corresponding plane. That way, we are still able to establish point-to-plane correspondences. To do that, we compare the individual planar point clusters to the global plane model and match them based on the distance and angle between them as in our previous work [2]. Further, we introduce a dynamic global plane model to our SLAM system, instead of a model that only initializes once.

Suppose that in the dynamic global model there are G planes. We denote this set of planes as $\mathcal{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_g, \cdots, \mathcal{P}_G\}$. Let $\text{M}_k = \{\rho_k, \mathcal{P}_g\}$ be a match between any plane candidate ρ_k and global plane $\mathcal{P}_g$. Further, for a number of K plane candidates, let $\bar{\text{M}} \subseteq \{\rho_1, \cdots, \rho_k, \cdots \rho_K\}$ be the subset of all plane candidates that have no correspondence in the global model. Algorithm 1 describes how the matches and mismatches are used to update the global model. It first calculates the eigenvalues of the local planes ρ_k, by applying Principal Component Analysis (PCA) [27] to the corresponding planar point clusters. PCA is commonly used in computer vision as a plane fitting tool. The resulting eigenvalues $e_1 \leq e_2 \leq e_3$ correspond to the extent of the plane in every dimension, where the smallest eigenvalue corresponds to the normal direction of the plane. Thus, we check the quality of the planes by considering "flatness" of the plane via the eigenvalue ratio $\frac{e_1}{e_1+e_2+e_3}$. If the ratio is above a certain threshold, the plane is not considered in the global model. For the system presented in this work particularly, the threshold is 0.05. Figure 3 illustrates how the extracted planes from each

Algorithm 1: Updating the global plane model

```
Data: Global Model P, Matches M_k = {ρ_k, P_g},
      Missmatches M̄ ⊆ {ρ_1, ··· , ρ_K}
Calc. PCA eigenvalues for {ρ_1, ··· , ρ_K};
for each Match {ρ_k, P_g} do
    if eigenvalues of ρ_k are OK then
        Merge ρ_k into P_g;
        if eigenvalues of P_g are BAD then
            Undo merge;
for each Missmatch M̄_k = ρ_k do
    if eigenvalues of ρ_k are OK then
        Insert ρ_k into P;
```

frame are used to sequentially update the global model by merging corresponding planes. Merging two planes works by considering all points on both convex hulls, and recalculate the convex hull and normal vector from those points. The bottom Figures 3b show the resulting global plane model, as well as the point cloud after registration of all frames.

V. Experiment and Evaluation

We executed the experiment in an environment that allowed for a long descent, i.e., in the building of the state firefighters school in Würzburg, as shown in Figure 4. In this experiment the sensor freely rotates around the descending axis, corresponding to the cable direction. Note that the

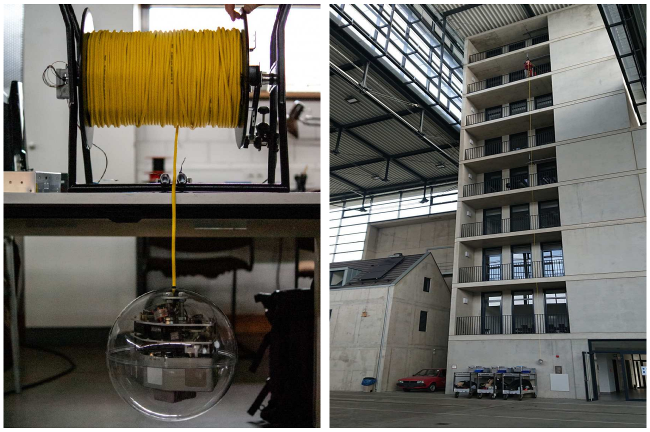


Fig. 4: Setup for the experiment. We descent the system with a tear-resistent tether cable, which is rolled around a coil. The internal sensors are protected using a spherical plastic shell.

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 [28]. The descent of the system covered a distance of $22\,\mathrm{m}$ and was performed within a duration of $402\,\mathrm{s}$. The post-processing is performed after the experiments on a separate server. Figure 6 shows a birds-eye view of the 3D point clouds before and after we apply our algorithm. Note that the initial pose estimates are especially erroneous in one rotational dimension, i.e., the yaw angle, 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. Despite the large rotational errors, the resulting map resembles the environment well. Some error remains due to the significant effect of motion distortion, considering the fast rotations. However, we consider that qualitatively, the resulting point cloud is sufficient for basic situational awareness, e.g., the walls intersect at right angles, the map is true to scale, and objects are recognizable despite being noisy. The resulting point cloud is analyzed in the next subsection in terms of the achieved mapping accuracy, which we perform by matching it against a high-precise ground-truth model of the environment, given by a RIEGL VZ400 terrestrial laser-scanner (TLS), which has an angular resolution of $0.08°$ and accuracy of $5\,\mathrm{mm}$. To establish the error distribution we match the resulting point cloud against the ground truth map, using ICP from 3DTK [29]. Then, we create a three-dimensional difference image by measuring all point-to-point errors. We note that since the resulting point cloud is still subject to motion distortion and noise, the alignment to ground truth due to ICP might be imperfect, i.e., the remaining root mean square of point-to-point errors (RMSE) after matching is non-zero. For matching, we discard any point-to-point correspondences that have distances larger than $20\,\mathrm{cm}$. After matching, 3DTK's ICP reports approx. $11.4\,\mathrm{cm}$ remaining RMSE.

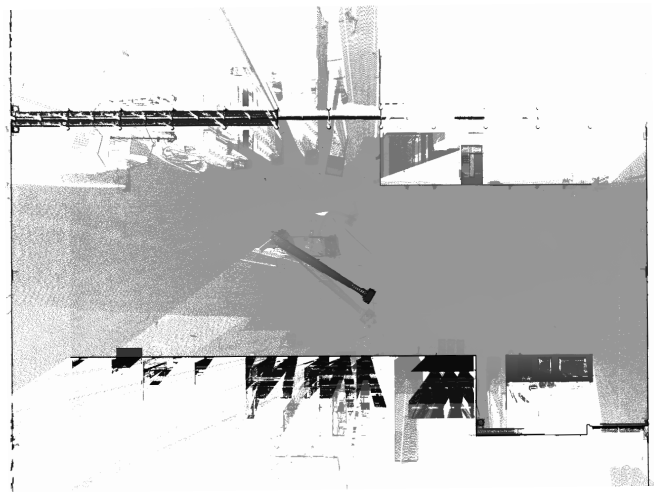


Fig. 5: Birds-eye view of the ground truth point cloud, aquired with a terrestrial laser scanner (TLS). The ceiling has been cropped for a better view. Ground truth point cloud of the state fire fighters school in Würzburg.

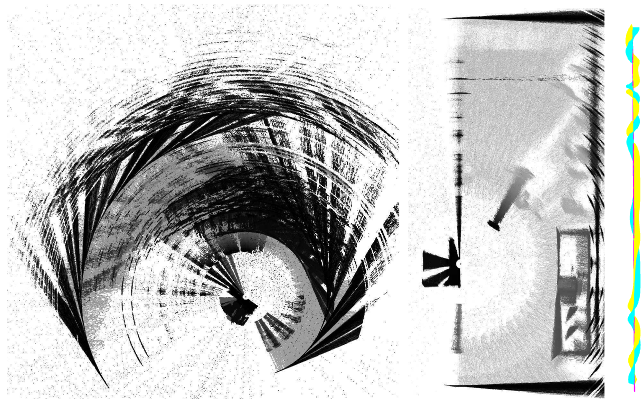
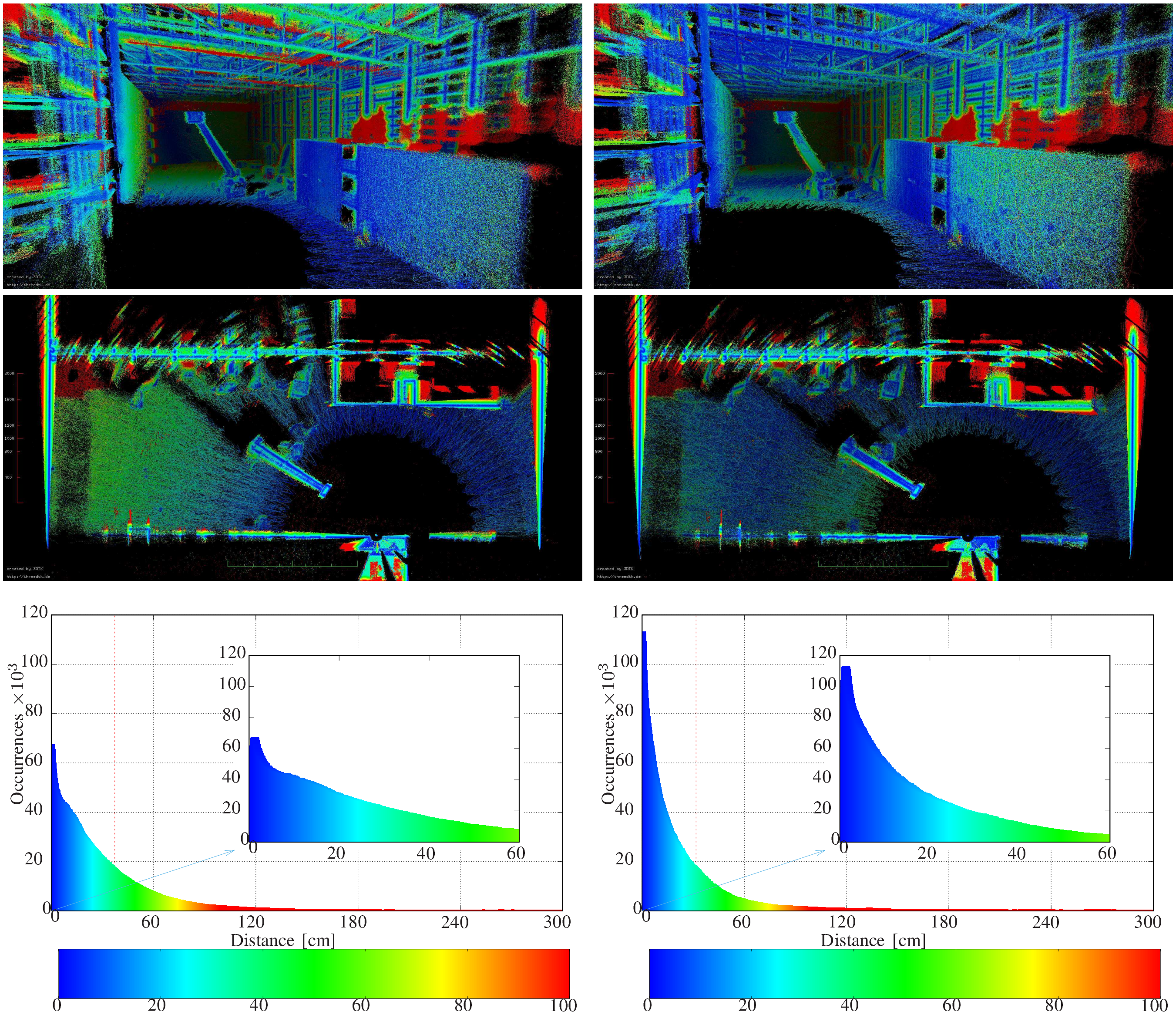


Fig. 6: In the images of 3D point clouds, the ceiling has been cropped for a better view. (Left) Birds-eye view of the resulting 3D point cloud, aquired with the descending system using a spin-encoder and IMUs for pose estimation. (Center) Birds-eye view of the post-processed 3D point cloud. (Right) Profile view of the mobile systems pose, movement from top to bottom.

A. *Comparison with SRR*

In this section, we quantitatively compare the presented approach with another state-of-the-art, high-precise offline-batch method: "Semi Rigid Registration" [12] (SRR). SRR uses a metascan-ICP-based implementation as a preregistration and afterward 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 recalculates the edges. We first discuss the mapping accuracy that both algorithms achieved.

Figure 7 shows the evaluation of the point distances to



(a) SRR (b) Presented approach

Fig. 7: Evaluation of point distances after SRR (left) and after the presented method (right). Lateral images always have the same orientation. The color-space maps all values with a distance greater than 100 cm to the same color. Points that have distances larger than 300 cm are excluded from the analysis. The means of the distributions are represented by the red dotted line and are 37.2 cm for SRR (left), and 31.6 cm for the presented approach (right).

ground truth after SRR and after the presented method has been applied to the same input. The color in the images denotes point-to-point error and corresponds with the color in the histogram. The presented method combines 25 successive frames into metascans which ensures that planes are robustly found and registers them. SRR optimizes every single frame individually. Further, both methods process only a subsampled version of the input, where the smallest voxels of size 10 cm are allowed to have only one point. This is especially useful considering the less dense flower-shaped scanning pattern of the Livox Mid-100, but also decreases processing time. When evaluating the accuracy we ensure similar point density at any distance from the sensor, by considering only subsampled versions of the resulting point clouds, where the smallest voxels of 10 cm must only contain a maximum of 50 points. The difference images of SRR and the presented method look similar, i.e., both methods achieve a good approximation to ground truth. In particular, the mean point-to-point error according to the histograms are 37.2 cm after the application of SRR, and 31.6 cm after the application of the presented method. Table I also shows that for lower percentiles (90% of the points), the presented method achieves smaller point-to-point distances than SRR. However, for larger percentiles (95% or above), the points

TABLE I: Comparison of point-to-point-distance percentiles to ground truth, as well as runtimes for SRR and the presented method.

	P90	P95	P98	Runtime
SRR	82.5 cm	127.8 cm	213.4 cm	6069.04 min
presented method	74.3 cm	133.7 cm	217.2 cm	13.63 min

have distances less than 213.1 cm after the application of SRR, and 217.2 cm after the application of the presented method. Thus, we argue that the accuracy of both methods is comparable in the presented scenario. The runtime evaluation yields a more distinctive result, though, as the runtime for SRR is significantly higher. We run both methods multi-threaded on a mobile Intel i7-10750H 12-core CPU with 5 GHz frequency per core and 64 GB of RAM. Overall, SRR needs 6069.04 min (approx. 4.2 days) to achive the result shown in Figure 7 (left column). The presented method achieves comparable accuracy in only 13.63 min, which corresponds to a 445271 % speed-up. Note that the duration of descent was 6.73 min, which is approximately half of the processing duration.

VI. Conclusion

In this work, we presented a cost-efficient approach to vertical mapping, which is applicable in search and rescue scenarios such as collapsed buildings, underground mines, construction sites, etc., as well as for exploration missions. While many existing approaches utilize high-priced terrestrial laser scanners or mechanical actuated LiDAR, we rely on more consumer-available solid-state LiDAR. Thus, our prototype showcases how future affordable solutions might evolve. We employed the proposed approach to the SLAM problem on a freely rotating, vertically suspended system. The approach is a revision of our previous work [2], which we want to make real-time capable in the future. We compared the presented method with a state-of-the-art high-precise globally consistent graph-based method, SRR [12]. Our method relies on polygon-matching and has comparable accuracy to SRR while being significantly faster (445271 %). Nevertheless, a lot of work remains to be done. As of now, the processing time needed to create the full map is twice the duration of the descending process. We plan to address this issue soon by further revising the correspondence model according to [14], which uses a specialized feature extraction technique for Livox devices. This will allow us to skip the creation of metascans, reducing processing time. Further, there would no longer be a need to subsample the scans but rather use only feature points, which decreases processing time even more. To verify the accuracy of our system, we also want to compare the trajectories against ground truth measurements, e.g., from an opti-track system. Moreover, we seek to reduce the effects of motion distortion by considering a model of the Livoxs' scanning pattern and the timestamp of every point, as in [24]. We also want to reduce IMU drift by considering more accurate sensor calibration.

References

[1] A. P. Rossi, F. Maurelli, H. Dreger, K. Mathewos, N. Pradhan, R. Pozzobon, S. Ferrari, C. Pernechele, D. Borrmann, A. Nüchter, A. Bredenbeck, J. Zevering, and F. Arzberger, "DAEDALUS - Descent And Exploration in Deep Autonomy of Lava Underground Structures," Inst.für Informatik, Tech. Rep. 21, 2021.

[2] F. Arzberger, A. Bredenbeck, J. Zevering, D. Borrmann, and A. Nüchter, "Towards spherical robots for mobile mapping in human made environments," *ISPRS Open Jour. Photogrammetry and Remote Sensing*, vol. 1, p. 100004, 2021.

[3] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter, "The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design," *3D Research*, vol. 2, pp. 1–13, 2011.

[4] Z. Zhongming, L. Linong, Y. Xiaona, Z. Wangqiang, L. Wei, *et al.*, "ESA plans mission to explore lunar caves," 2021.

[5] P. McGarey, D. Yoon, T. Tang, F. Pomerleau, and T. D. Barfoot, "Field Deployment of the Tethered Robotic eXplorer to Map Extremely Steep Terrain," in *Field and Service Robotics*, M. Hutter and R. Siegwart, Eds., 2018, pp. 303–317.

[6] R. A. S. Fernandez, Z. Milošević, S. Dominguez, and C. Rossi, "Motion Control of Underwater Mine Explorer Robot UX-1: Field Trials," *IEEE Access*, vol. 7, pp. 99 782–99 803, 2019.

[7] T. Lipecki and T. T. H. Kim, "The development of terrestrial laser scanning technology and its applications in mine shafts in poland," *Inzynieria Mineralna*, vol. 1, pp. 301–310, 10 2020.

[8] Geoslam, "Vertical shaft inspection," https://geoslam.com/solutions/vertical-shaft-inspection/, accessed on 22.07.2022.

[9] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[10] F. Lu and E. E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robots*, vol. 4, pp. 333–349, 1997.

[11] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg, "Globally consistent 3D mapping with scan matching," *Robotics and Auton. Systems*, vol. 56, no. 2, pp. 130–142, 2008.

[12] J. Elseberg, D. Borrmann, and A. Nuchter, "6DOF semi-rigid SLAM for mobile scanning," 10 2012, pp. 1865–1870.

[13] C. L. Gentil, T. Vidal-Calleja, and S. Huang, "IN2LAMA: INertial Lidar Localisation And MApping," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6388–6394.

[14] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 3126–3131.

[15] D. Droeschel and S. Behnke, "Efficient Continuous-Time SLAM for 3D Lidar-Based Online Mapping," *2018 IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2018.

[16] H. Alismail and B. Browning, "Automatic calibration of spinning actuated lidar internal parameters," *Jour. Field Robotics*, 2015.

[17] W. Förstner and K. Khoshelham, "Efficient and accurate registration of point clouds with plane to plane correspondences," in *2017 IEEE Int. Conf. on Computer Vision Workshops (ICCVW)*, 2017, pp. 2165–2173.

[18] W. S. Grant, R. C. Voorhies, and L. Itti, "Efficient Velodyne SLAM with point and plane features," *Auton. Robots*, vol. 43, 2019.

[19] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "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.

[20] L. Zhou, S. Wang, and M. Kaess, "π-LSAM: LiDAR smoothing and mapping with planes," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA '21)*, Xi'an, China, May 2021, to appear.

[21] X. Wei, J. Lv, J. Sun, and S. Pu, "Ground-SLAM: Ground Constrained LiDAR SLAM for Structured Multi-Floor Environments," 2021.

[22] LIVOX, "Livox mapping," github.com/Livox-SDK/livox_mapping, accessed on 30.06.2022.

[23] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems Conf. (RSS)*, 07 2014.

[24] X. Zheng and J. Zhu, "Effective solid state lidar odometry using continuous-time filter registration," *arXiv preprint arXiv:2206.08517*, 2022.

[25] P. Babin, P. Giguère, and F. Pomerleau, "Analysis of robust functions for registration algorithms," *CoRR*, vol. abs/1810.01474, 2018.

[26] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012.

[27] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1, pp. 37–52, 1987, proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[28] E. W. Weisstein, "Helix," *https://mathworld. wolfram. com/*, 2003.

[29] A. Nüchter and K. Lingemann, "3DTK—The 3D Toolkit. 2011," https://slam6d.sourceforge.io/index.html, 2011.