Remote Sens. **2015**, *7*, 13753-13781; doi:10.3390/rs71013753

OPEN ACCESS

remote sensing

ISSN 2072-4292

www.mdpi.com/journal/remotesensing

Article

Evaluation of a Backpack-Mounted 3D Mobile Scanning System

Helge A. Lauterbach [1], Dorit Borrmann [1], Robin Heß [1], Daniel Eck [2], Klaus Schilling[1,2] and Andreas Nüchter [1,*]

[1] Intitute of Informatics VII–Robotics and Telematics, University of Würzburg, Am Hubland, 97074 Würzburg, Germany; E-Mails: helge.a.lauterbach@stud-mail.uni-wuerzburg.de (H.L.); borrmann@informatik.uni-wuerzburg.de (D.B.); hess@informatik.uni-wuerzburg.de (R.H.); schi@informatik.uni-wuerzburg.de (K.S.)

[2] Zentrum für Telematik, Allesgrundweg 12, 97218 Gerbrunn, Germany; E-Mail: daniel.eck@telematik-zentrum.de

* Author to whom correspondence should be addressed; E-Mail: andreas@nuechti.de; Tel.: +49-931-318-8790.

Academic Editors: Randolph H. Wynne and Prasad S. Thenkabail

Received: 31 July 2015 / Accepted: 23 September 2015 / Published: 21 October 2015

Abstract: Recently, several backpack-mounted systems, also known as personal laser scanning systems, have been developed. They consist of laser scanners or cameras that are carried by a human operator to acquire measurements of the environment while walking. These systems were first designed to overcome the challenges of mapping indoor environments with doors and stairs. While the human operator inherently has the ability to open doors and to climb stairs, the flexible movements introduce irregularities of the trajectory to the system. To compete with other mapping systems, the accuracy of these systems has to be evaluated. In this paper, we present an extensive evaluation of our backpack mobile mapping system in indoor environments. It is shown that the system can deal with the normal human walking motion, but has problems with irregular jittering. Moreover, we demonstrate the applicability of the backpack in a suitable urban scenario.

Keywords: backpack mobile mapping; SLAM; mobile laser scanning; personal laser scanning; man-portable mapping

1. Introduction

Mobile mapping systems consisting of sensors mounted on cars are the state of the art for mapping urban environments. However, they are limited to areas that are accessible by car. Robotic solutions [1] or solutions with scanners mounted on carts, like the VIAmetris iMMS [2,3], the Google Street View Trolley [4] or the NavVis 3D Mapping Trolley [5], are applicable in smaller alleys. For stairs, as well as dirt or gravel roads, these systems still meet their limits. Airborne laser scanning is not restricted to specific terrain and, thus, has advantages, but it is not available in roofed environments or tunnels. Recently, several backpack-mounted systems, also known as personal laser scanning, have been presented as ideal solutions to overcome these issues for indoor mapping. "The Cartographer" by Google relies on Hokuyo laser scanners, inexpensive devices with a low data rate, accuracy and range and thus not well suited for outdoor applications [6]. The Zebedee 3D sensor system [7] currently also consists of Hokuyo 2D laser scanners that are mounted on one ore more springs. Recently, Leica announced its own solution, the Pegasus:Backpack [8]. Equipped with a Dual Velodyne VLP-16 laser scanner, it acquires range measurements up to 50 m, making it suitable for outdoor applications. In previous work, we have presented our backpack mobile mapping system [9], featuring a high-end laser scanner, namely the Riegl VZ-400, for mapping. In contrast to the Akhka-Backpack by Kuko *et al.* [10], which also provides a high-end laser scanner, we do not incorporate a global navigation satellite system (GNSS) for localization. In urban canyons, a GNSS-free approach is advantageous, since the GNSS signal may be disturbed, and thus, localization would fail. Corso and Zakhor and Lehtola *et al.* also presented GNSS-free solutions based on laser scanners. However, they are currently restricted to environments where the 2.5D assumption holds true [11] or to 1D trajectories on flat floors, respectively [12].

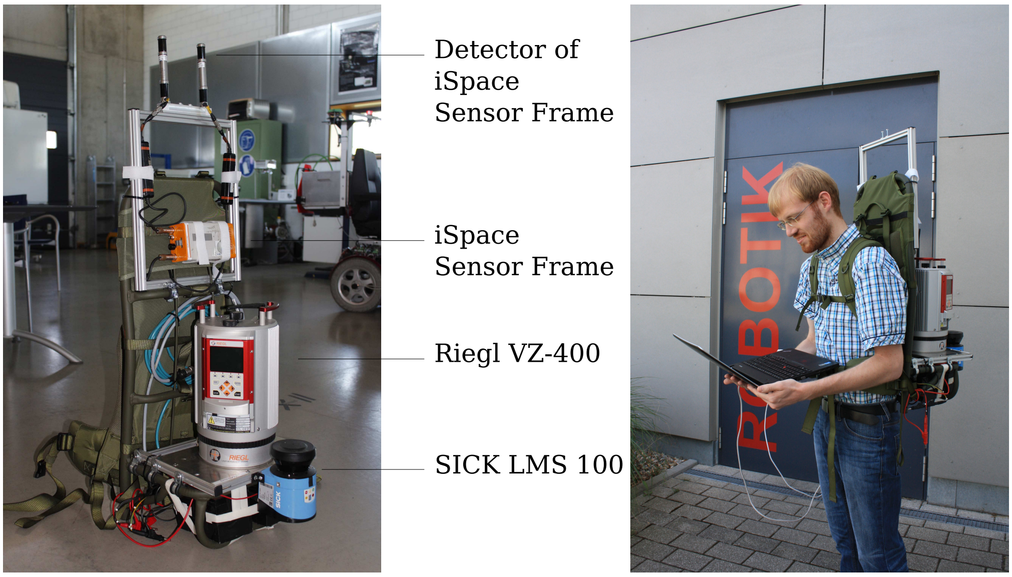


Figure 1. The backpack system. (**Left**): the hardware components. The iSpace sensor frame is attached to the top of the backpack. (**Right**): the operator carries the backpack during a typical experiment.

Our mapping solution, *cf.* Figure 1, relies on a horizontally-mounted 2D profiler, the SICK LMS 100 laser scanner. A simultaneous localization and mapping (SLAM) software, called HectorSLAM [13], generates an initial trajectory of the backpack. The trajectory is then used to "unwind" the data of the Riegl VZ-400. The Riegl scanner itself is rotating around its vertical axis, such that the environment is gaugedmultiple times. This is exploited in our calibration and semi-rigid SLAM solution. While the calibration computes the six degrees of freedom (DoF) pose of every sensor, the semi-rigid SLAM deforms the trajectory of the backpack in six DoF, such that the 3D point cloud aligns well. The system is ready to use and was shown to perform well in indoor scenarios [9], e.g., at MoLaS Technology Workshop 2014 at Fraunhofer IPM [14]. Here, we evaluate the accuracy of the system using ground truth pose information acquired with the iSpace positioning system [15]. Furthermore, we show the applicability of the system to outdoor environments. Most challenging for the system are areas with open space exceeding the range of the 2D laser scanner. We show that the long range of the Riegl VZ-400 compensates this in the semi-rigid registration step to overcome inaccuracies introduced by the 2D mapping subsystem, such that the resulting 3D point cloud aligns well.

2. The Backpack-Mounted Mobile Mapping System

The backpacking system is inspired by the robotic system Irma3D. [1]. Instead of the Volksbot®RT3 chassis, a Tatonka load carrier serves as the basis. Energy is currently provided by two Panasonic 12-V lead-acid batteries with 12 Ah, but to save weight, these will be replaced by lithium polymer batteries. Similar to Irma3D [1], the backpack features a horizontally-scanning SICK LMS 100, which is used to observe the motion of the carrier using a grid mapping variant. To fully exploit the 270° FoV of the SICK LMS 100, the sensors head is positioned slightly above the load carrier. To improve the motion estimation of the 2D LiDAR, an inexpensive, low-end inertial measurement unit (IMU), namely the Phidgets 1044 [16], is mounted on the backpack. The central sensor of the backpack system is the 3D laser scanner Riegl VZ-400. The VZ-400 is able to freely rotate around its vertical axis to acquire 3D scans. Due to the setup, however, there is an occlusion of about 90° from the backside of the backpack and the human carrier. This occlusion causes the trajectory at which data were collected to have a gap when running the VZ-400 in constantly spinning mode. To avoid this gap, we programmed the VZ-400 to spin back and forth.

To evaluate the mapping algorithm, we use the iSpace positioning system. Therefore, an iSpace sensor frame is attached to the backpack, as shown in Figure 1. A laptop, which is carried by the human, collects the data of both scanners and the IMU. All devices are connected via a network switch.

System Architecture

Figure 2 presents the overall architecture of the backpack. For sensor data acquisition and trajectory generation, we use the robot operating system (ROS) [17]. Each step is run as an independent process, a so-called ROS node. The data of 2D LiDAR and IMU are fed into the 2D SLAM subsystem HectorSLAM. The output of the HectorSLAM ROS node serves as the input of the six DoF semi-rigid SLAM, which registers the 3D data from the Riegl VZ-400. This optimization step is implemented using the framework 3DTK (3D Toolkit) [18].

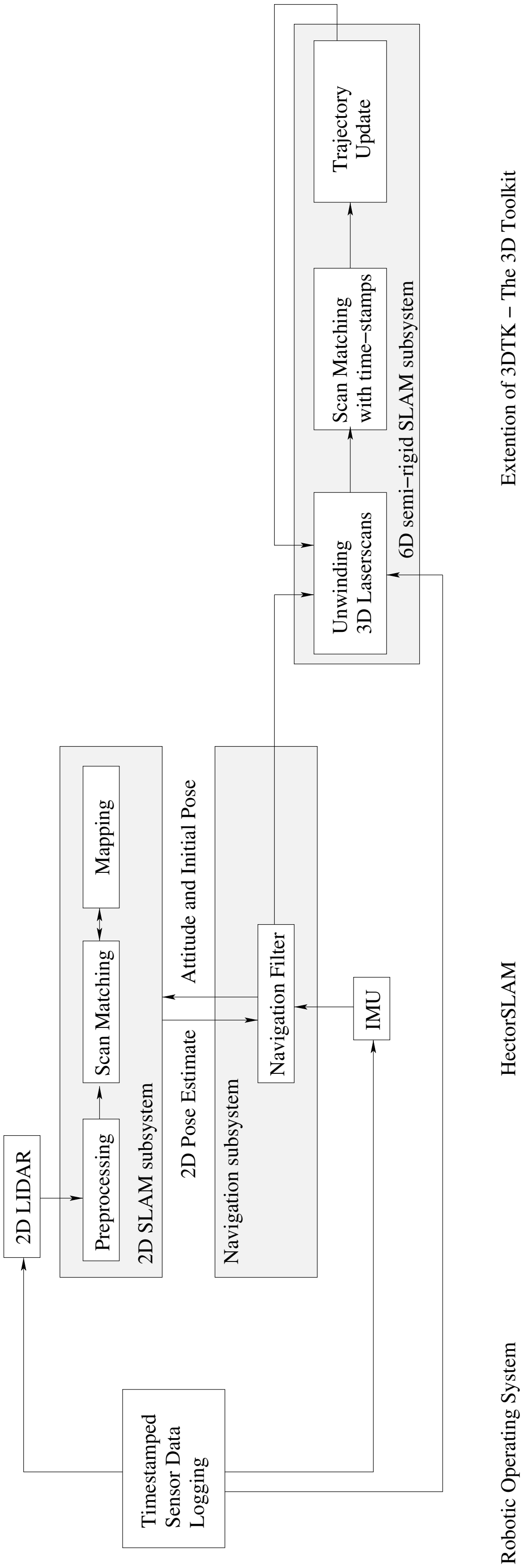


Figure 2. Overview of the system architecture.

3. Initial Trajectory Estimation

Hector SLAM is a state-of-the-art 2D SLAM solution [13]. It represents the environment in a 2D occupancy grid, which is a very well-known representation in robotics. The 2D LiDAR performs six DoF motion while the backpack is carried. First, the scan is transformed into a local stabilized coordinate frame using the IMU-estimated attitude of the LiDAR system. In a scan matching process, the acquired stabilized scan is matched with the existing map. The optimization of the alignment is done using a Gauss–Newton approach, similar to the work in [19], and therefore, neither data association, *i.e.*, point matching, nor an exhaustive search for the optimal pose transformation is needed. The information of the 2D SLAM solution is exchanged using the ROS communication framework with the navigation filter, which is an EKF (extended Kalman filter) in a bi-directional fashion, and, thus, fused with the values of the IMU to produce six DoF pose estimates.

4. 3D Mapping

3D mobile mapping with constantly spinning scanners has been studied in the past by the authors; thus, we summarize our work from [20,21]. The software is suited to turn laser range data acquired with a rotating scanner while the acquisition system is in motion into precise, globally-consistent 3D point clouds.

4.1. Calibration

Calibration is the process of estimating the parameters of a system. We need to estimate the extrinsic parameters, *i.e.*, the three DoF attitude and three DoF position of the two laser scanners with respect to some base frame. Up to now, we worked in the SOCS (scanner own coordinate system) of the SICK scanner. We use the ROS package `tf` (the transform library), which lets us keep track of multiple coordinate frames over time. `tf` maintains the relationship between coordinate frames in a tree structure buffered in time and allows transforming points, vectors, *etc.*, between any two coordinate frames at any desired point in time.

In [21], we presented a general method for the calibration problem, where the 3D point cloud represents samples from a probability density function. We treated the “unwind” process as a function where the calibration parameters are the unknown variables and used the Reny entropy, computed on the closest points regarding a timing threshold, as the point cloud quality criterion. Since computing derivatives of such an optimization is not possible, we employ Powell’s method, which minimizes the function by a bi-directional search along each search vector, in turn, and therefore, it resembles a gradient descent. This optimization usually takes about 20 minutes on a standard platform, but needs to be done only once for a new setup.

4.2. 6D SLAM

For our backpack system, we need a semi-rigid SLAM solution, which is explained in the next section. To understand the basic idea, we summarize its basis, 6D SLAM.

6D SLAM works similarly to the the well-known iterative closest points (ICP) algorithm, which minimizes the following error function:

$$E(\mathbf{R},\mathbf{t}) = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{m}_i - (\mathbf{R}\mathbf{d}_i + \mathbf{t})||^2 \tag{1}$$

to solve iteratively for an optimal rotation $\mathbf{T} = (\mathbf{R},\mathbf{t})$, where the tuples $(\mathbf{m}_i,\mathbf{d}_i)$ of corresponding model $\mathbf{M}$ and data points $\mathbf{D}$ are given by minimal distance, *i.e.*, $\mathbf{m}_i$ is the closest point to $\mathbf{d}_i$ within a close limit [22]. Instead of the two-scan-Equation (1), we look at the n-scan case:

$$E = \sum_{j\to k}\sum_{i} |\mathbf{R}_j\mathbf{m}_i + \mathbf{t}_j - (\mathbf{R}_k\mathbf{d}_i + \mathbf{t}_k)|^2 \tag{2}$$

where j and k refer to scans of the SLAM graph, *i.e.*, to the graph modeling the pose constraints in SLAM or bundle adjustment. If they overlap, *i.e.*, closest points are available, then the point pairs for the link are included in the minimization. We solve for all poses at the same time and iterate as in the original ICP. The derivation of a GraphSLAM method using a Mahalanobis distance that describes the global error of all of the poses:

$$W = \sum_{j\to k}(\bar{\mathbf{E}}_{j,k} - \mathbf{E}'_{j,k})^T\mathbf{C}_{j,k}^{-1}(\bar{\mathbf{E}}'_{j,k} - \mathbf{E}'_{j,k}) \tag{3}$$

where $\mathbf{E}'_{j,k}$ is the linearized error metric and the Gaussian distribution is $(\bar{\mathbf{E}}_{j,k},\mathbf{C}_{j,k})$ with computed covariances from scan matching, as given in [20], does not lead to different results [23]. Please note, while there are four closed-form solutions for the original ICP Equation (1), linearization of the rotation in Equation (2) or (3) is always required.

4.3. Semi-Rigid SLAM

In addition to the calibration algorithm, we also developed an algorithm that improves the entire trajectory of the backpack simultaneously. The algorithm is adopted from [21], where it was used in a different mobile mapping context, *i.e.*, on wheeled platforms. Unlike other state-of-the-art algorithms, like [24,25], it is not restricted to purely local improvements. We make no rigidity assumptions, except for the computation of the point correspondences. We require no explicit motion model of a vehicle, for instance. The semi-rigid SLAM for trajectory optimization works in six DoF, which implies that the planar trajectory generated by HectorSLAM is turned into poses with six DoF. The algorithm requires no high-level feature computation, *i.e.*, we require only the points themselves.

In the case of mobile mapping, we do not have separate terrestrial 3D scans. In the current state of the art developed by [25] for improving the overall map quality of mobile mappers in the robotics community, the time is coarsely discretized. This results in a partition of the trajectory into sub-scans that are treated rigidly. Then, rigid registration algorithms, like the ICP and other solutions to the SLAM problem, are employed. Obviously, trajectory errors within a sub-scan cannot be improved in this fashion. Applying rigid pose estimation to this non-rigid problem directly is also problematic since rigid transformations can only approximate the underlying ground truth. When a finer discretization is used, single 2D scan slices or single points result that do not constrain a six DoF pose sufficiently for rigid algorithms.

The mathematical details of our algorithm are given in [21]. Essentially, we first split the trajectory into sections and match these sections using the automatic high-precise registration of terrestrial 3D scans, *i.e.*, globally-consistent scan matching [20]. Here, the graph is estimated using a heuristics that measures the overlap of sections using the number of closest point pairs. After applying globally-consistent scan matching on the sections, the actual semi-rigid matching as described in [21] is applied, using the results of the rigid optimization as starting values to compute the numerical minimum of the underlying least squares problem. To speed up the calculations, we make use of the sparse Cholesky decomposition by [26].

A key issue in semi-rigid SLAM is the search for closest point pairs. We use an octree and a multi-core implementation using OpenMP to solve this task efficiently. A time threshold for the point pairs is used, *i.e.*, we match only to points, if they were recorded at least t_d time steps away. This time corresponds to the rotation time of the Riegl scanner, *i.e.*, it is set to 6 s. In addition, we use a maximal allowed point-to-point-distance, which has been set to 0.25 cm.

Semi-rigid SLAM transforms all points; points in a scan line via interpolation over the time-stamps. Finally, all scan slices are joined in a single point cloud to enable efficient viewing of the scene. The first frame, *i.e.*, the first 3D scan slice from the Riegl scanner defines the arbitrary reference coordinate system. By using known landmarks, the acquired point cloud can be transferred into the building coordinate system.

4.4. iSpace Positioning System

iSpace is a high-precision position and tracking system from Nikon Metrology [15]. The optical laser-based system consists of several transmitters. These are mounted on a wall or on tripods to cover the experimental area, both indoors and outdoors. The rotating head of each transmitter emits two perpendicular fan-shaped laser beams at a unique distinguishable frequency near 40 Hz. The vertical aperture angle of the laser beams is limited to 40 degrees, and the detectable range lies between two and 55 meters. Several sensor frames are trackable within the system. A sensor frame consists of at least one detector, a photo diode with a field of view of 360 degrees horizontally and 90 degrees vertically. A small radio frequency module transmits the sensor data wirelessly to the base station of the iSpace system. Each sensor frame receives the laser data from the transmitters and sends the information on to the base station. The iSpace software calculates the elevation and azimuth angles between all detectors for a sensor frame and each visible transmitter based on the received data defining a straight line between transmitter and detector. Given the relative transformation between the transmitters, the length of the lines is calculated using triangulation. In typical environments, the iSpace system is able to perform measurements at a sampling rate of 40 Hz with a maximum error of ± 0.25 mm. In practice, environmental factors, such as size, reflection of the surface and occlusions of the transmitters, have to be taken into consideration.

4.4.1. Calibration between Riegl VZ-400 and iSpace

To localize the laser scanner in the iSpace coordinate system, an iSpace sensor frame is attached to the backpack, as shown in Figure 1. For calibration between the VZ-400 laser scanner and the iSpace sensor frame, several reflective markers were attached to objects in the environment. The centers of the markers are measured with a special sensor frame, the iSpace hand vector bar, thus determining their position in the iSpace coordinate system (Figure 3). Their position in the local scanner coordinate system is determined by using the RiScan Pro software. The markers are labeled manually in a full 3D scan. Afterwards, the software controls the scanner automatically to scan the area around the selected markers with a very high resolution to calculate the precise position of the markers. Then, the coordinates of the markers in the coordinate system defined by iSpace are imported as control points, and the scans are registered to these control points based on the marker position. This yields the position and orientation of the laser scanner in the iSpace coordinate system at the time the scan was taken. Additionally, the pose of the sensor frame is recorded. In the following, poses will be treated as transformation matrices $\mathbf{T}$, consisting of the rotation $\mathbf{R}$ and the translation $\mathbf{t}$. Repeating this procedure for n scans results in n pairs of poses for the Riegl laser scanner $\mathbf{T}_{r,i}$ and the sensor frame $\mathbf{T}_{m,i}$. From these poses, the transformation $\mathbf{T}_{m\to r}$ between the coordinate systems is calculated as:

$$\mathbf{T}_{m\to r,i} = \mathbf{T}_{r,i}\mathbf{T}_{m,i}^{-1} \tag{4}$$

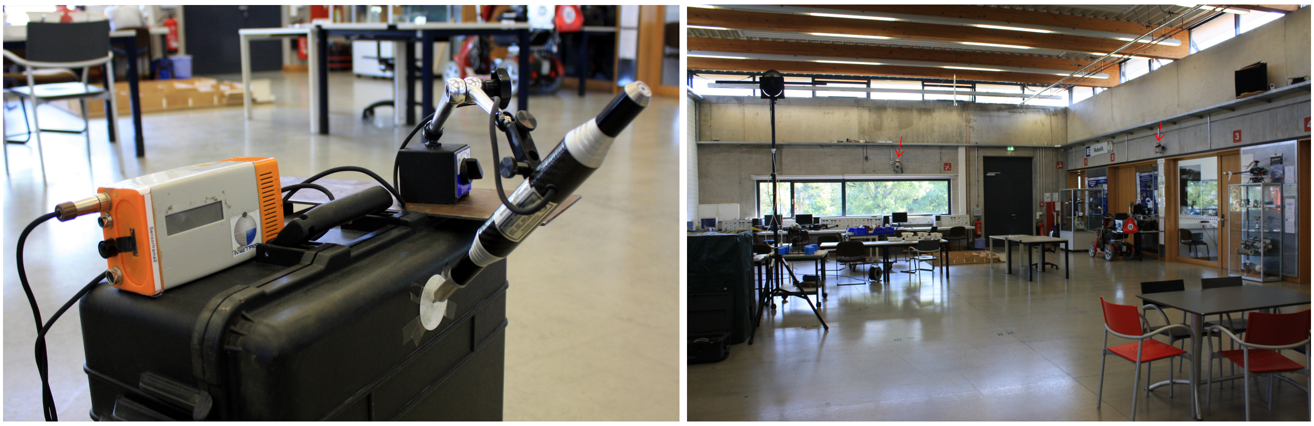


Figure 3. (**Left**): The positions of the reflective markers in the iSpace coordinate system are measured with the hand vector bar. (**Right**): the iSpace transmitters are mounted on the walls of the robotics hall at the University of Würzburg, where the experiments were carried out.

To reduce noise, the average over all transformation matrices $\mathbf{T}_{m\to r,i}$ is calculated as:

$$\bar{\mathbf{T}}_{m\to r} = \frac{1}{n}\sum_{i=0}^{n-1}\mathbf{T}_{m\to r,i} \tag{5}$$

This procedure works for the translation, but is not guaranteed to yield valid solutions for the rotation, *i.e.*, an orthogonal matrix with determinant one. Instead, we compute the nearest orthonormal matrix by projecting $\bar{\mathbf{R}}_{m\to r}$ onto $SO(3)$ [27]. Let $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ be the eigenvectors and λ_1, λ_2, λ_3 the eigenvalues of the square matrix:

$$\mathbf{HHs} = \bar{\mathbf{R}}_{m\to r}^{T} \cdot \bar{\mathbf{R}}_{m\to r} \tag{6}$$

then the final rotation matrix is calculated as:

$$\hat{\mathbf{R}}_{m\to r} = \bar{\mathbf{T}}_{m\to r} \cdot \left(\frac{\mathbf{e}_1 \cdot \mathbf{e}_1^T}{\sqrt{\lambda_1}} + \frac{\mathbf{e}_2 \cdot \mathbf{e}_2^T}{\sqrt{\lambda_2}} + \frac{\mathbf{e}_3 \cdot \mathbf{e}_3^T}{\sqrt{\lambda_3}} \right) \tag{7}$$

Afterwards, for each new scan position, the position of the laser scanner in the iSpace coordinate system is calculated:

$$\hat{\mathbf{T}}_{r,i} = \hat{\mathbf{T}}_{m\to r}\mathbf{T}_{m,i} \tag{8}$$

4.4.2. Metrics for Trajectory Evaluation

To evaluate the trajectory generated from the backpack mobile mapping system quantitatively, it is compared to the ground truth information from the iSpace positioning system. The absolute trajectory error is computed as error metric, similar to [28]. The generated trajectory $\mathbf{T}_b$ is computed in the local coordinate system of the backpack. To align it to the iSpace trajectory $\mathbf{T}_r$, two methods are employed. First, $\mathbf{T}_b$ is transformed, such that the transformed first pose $\mathbf{T}_{bg,0}$ of $\mathbf{T}_b$ and the first pose $\mathbf{T}_{r,0}$ of $\mathbf{T}_r$ are identical. The new poses are given by:

$$\mathbf{T}_{bg,i} = \mathbf{T}_{r,0}\mathbf{T}_{b,0}^{-1}\mathbf{T}_{b,i} \tag{9}$$

This method is highly influenced by the precision of the first pose and emphasizes calibration errors. It is only recommended if the first pose of both trajectories can be measured with high precision. For the current setup, where the poses of the iSpace trajectory are independent measurements while the generated trajectory of the backpack consists of incremental pose estimates, a method with more informative value, concerning the consistency of the result, takes into account the known correspondences between the individual pose measurements. The generated trajectory $\mathbf{T}_b$ is now aligned to the iSpace trajectory $\mathbf{T}_r$ by finding a rigid body transformation S that minimizes the distances between the pose pairs applying the method of [29], *i.e.*, using a closed form solution with singular value decomposition. The transformed trajectory is then $\mathbf{T}_{bg,i} = \mathbf{S}\mathbf{T}_{b,i}$. Let:

$$\mathbf{T}_{r\to bg,i} = \mathbf{T}_{bg,i}\mathbf{T}_{r,i}^{-1} \tag{10}$$

be the difference between the two translation matrices $\mathbf{T}_{bg,i}$ and $\mathbf{T}_{r,i}^{-1}$, *i.e.*, the relative transformation between the two poses. A transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ is composed of the translation vector $\mathbf{t}$ and the rotation matrix $\mathbf{R}$. The position error for each pair of corresponding poses at time step i is computed as:

$$E_i^{\text{pos}} = |\text{trans}\left(\mathbf{T}_{r\to bg,i}\right)| \tag{11}$$

Determining the orientation error of the trajectory requires a valid measure that incorporates the three DoF of the rotation. To reduce the three rotations to a single measure, different representation are commonly used. In urban environments, it is common to reduce the rotation error to the difference of rotation around the yaw axis. This gives an idea of the registration quality for a mobile mapping system, as the changes in roll and pitch angles are often neglectable when moving on smooth surfaces. On a personal mapping system, such as the backpack used here, rotation errors in all directions are to be expected. Therefore, the axis angle representation of a rotation matrix is used to evolve a measure for the rotation error. For the axis angle representation, the rotation matrix $\mathbf{R}$ is transformed into a rotation axis $\mathbf{a} = (a_x, a_y, a_z)$ and the angle θ. Let $\mathbf{R}_{r\to bg,i}$ be the rotational part of $\mathbf{T}_{r\to bg,i}$ and $\mathbf{a}_{r\to bg,i}$ and $\theta_{r\to bg,i}$ the axis angle representation thereof, then the rotational error between the two poses is given by:

$$E_i^{\mathrm{rot}} = |\theta_{r\to bg,i}| \tag{12}$$

Finally, the root mean square error for the entire trajectory is computed according to [28]:

$$\mathrm{RMSE} = \left(\frac{1}{n}\sum_{i=1}^{n} \|E_i\|^2\right)^{\frac{1}{2}} \tag{13}$$

5. Experiments and Results

5.1. Indoor Dataset

To evaluate the accuracy of our semi-rigid optimization algorithm, we acquired datasets in the robotics hall at the Julius Maximilian University of Würzburg, Germany where the iSpace system is fixed to the wall. The hall is an open space of 20×11 m^2 with desks and laboratory equipment.

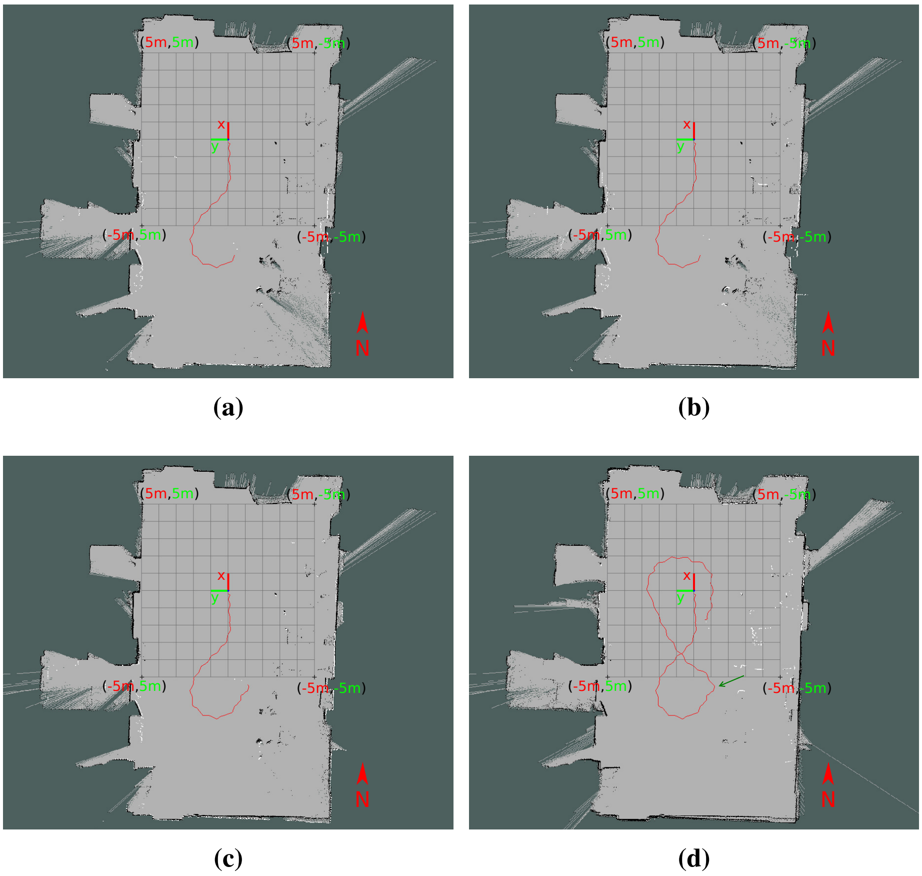
After calibrating the Riegl VZ-400 to the iSpace coordinate system, several datasets were acquired. The backpack was carried through the robotics hall by the first author along varying trajectories. To avoid the blind spot, the Riegl VZ-400 was rotating back and forth with a resolution of 0.5 degrees both horizontally and vertically with a rotational velocity of $0.1\bar{6}$ Hz. The field of view of the scanner is restricted to $180°$ (horizontally) by $100°$ (vertically). Table 1 shows some details of the datasets evaluated in this paper.

Table 1. Properties of the indoor datasets evaluated in this paper.

Experiment	Shape of Trajectory	Line Scans	Duration (s)	Total Points
1	Eight	9525	93	5,078,759
2	Rectangle	10,633	103	7,654,005
3	Circle	6201	60	4,469,067

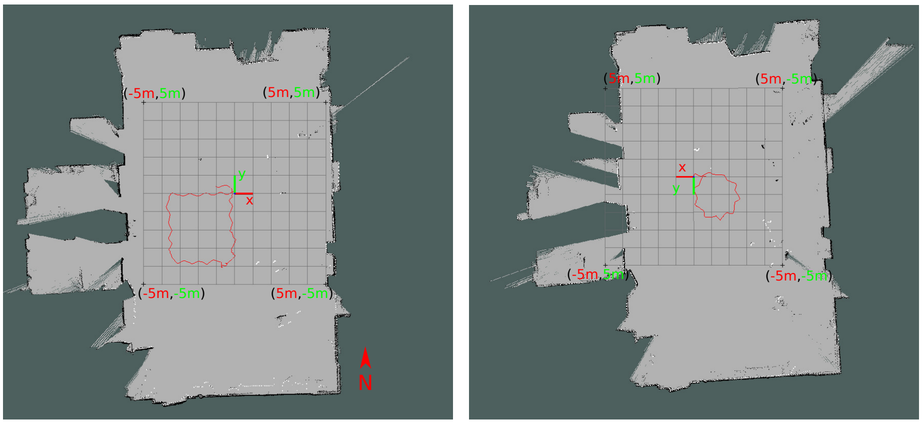
The result of the initial trajectory estimation by HectorSLAM is given in Figure 4 and 5. While the 2D maps for Experiments 2 and 3 are consistent maps, the map for Experiment 1 shows inaccuracies on the upper end (Figure 4). They originate from jittering in the beginning of the experiment before starting to walk. Nevertheless, the motion of normal human walking is compensated well. Difficulties arise with irregular jittering. Figure 4 shows also inaccuracies at the lower right corner of the hall, which arise between Second 40 and Second 43 of the walk. They originate from the sharp curve in the trajectory on the right. The registration process was further affect by people walking around in this area.

Despite the small inaccuracies, the map suffices to serve as input for unwinding the Riegl data, yielding an initial 3D point cloud. Figure 6 compares the point cloud of Experiment 1 from the bird's eye view and two side views before and after the optimization. The quality of the point cloud improves, which is particularly evident at the walls and the floor. The gaps in the walls marked by the red arrows are closed during the optimization step.



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

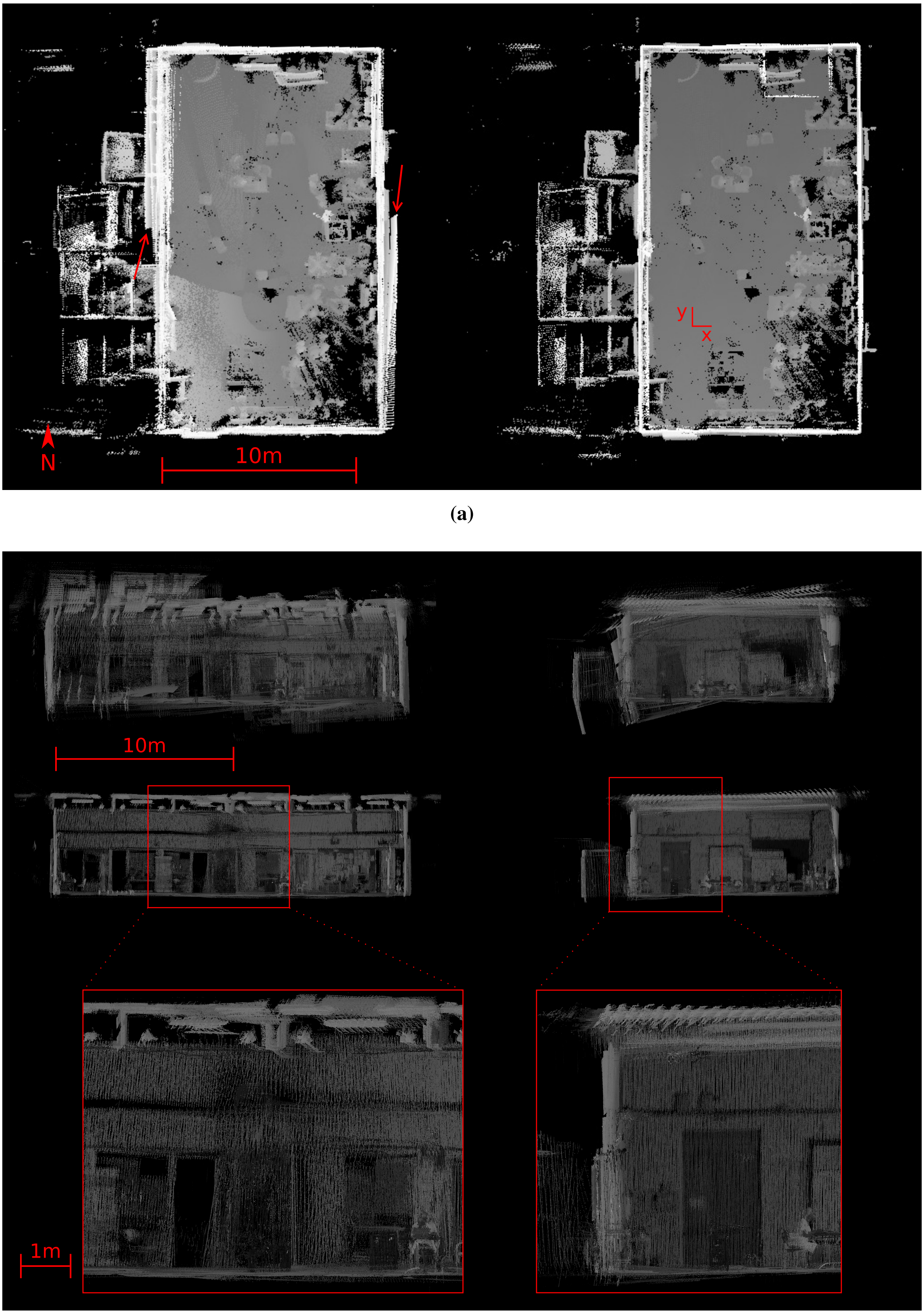
Figure 4. Four frames showing the 2D maps created by HectorSLAM for Experiment 1 in the local coordinate system after 41 seconds (**a**), 43 seconds (**b**), 45 seconds (**c**) and the final result after 93 seconds (**d**). The red lines denote the trajectory. The superimposed grid has a size of 10 × 10 m^2. The green arrow denotes the part of the trajectory where the 2D registration lead to inaccuracies in the lower right corner between (b,c).



(a) Experiment 2: Final map after 103 seconds

(b) Experiment 3: Final Map after 71 seconds

Figure 5. Final 2D maps in the local coordinate system created by HectorSLAM for Experiments 2 (**a**) and 3 (**b**). The maps are consistent and thus serve as input for unwinding the Riegl data.



(a)

(b)

Figure 6. 3D point cloud of Experiment 1. (**a**) Bird's eye view before (left) and after optimization (right). The gaps between the walls (red arrows) are closed by optimization. Depicted is the origin of the iSpace coordinate system; (**b**) Side view with reflectance values before (top) and after optimization (bottom). The quality improved, but the marked areas remain slightly rotated.

The remaining inaccuracies correspond to the aforementioned inaccuracies of the HectorSLAM output. Irregular jittering during the data acquisition leads to trajectory errors between neighboring line scans, which are treated as one sub-scan in the semi-rigid optimization algorithm. Thus, the error cannot be corrected completely. The marked areas in Figure 6 result from such trajectory errors. For instance, the office door is rotated clockwise compared to the office doors outside the marked area.

Figure 7 compares the optimization of this marked area from another perspective. While the rotated floor (Arrow 2) and the boxes (Arrow 1) converge and the windows at the ceiling (Arrow 3) become visible, the marked office doors (Arrows 4 and 5) remain rotated. In general, the structure of the environment improves, but the contours of objects are still blurry after optimization, as Figure 8 depicts.

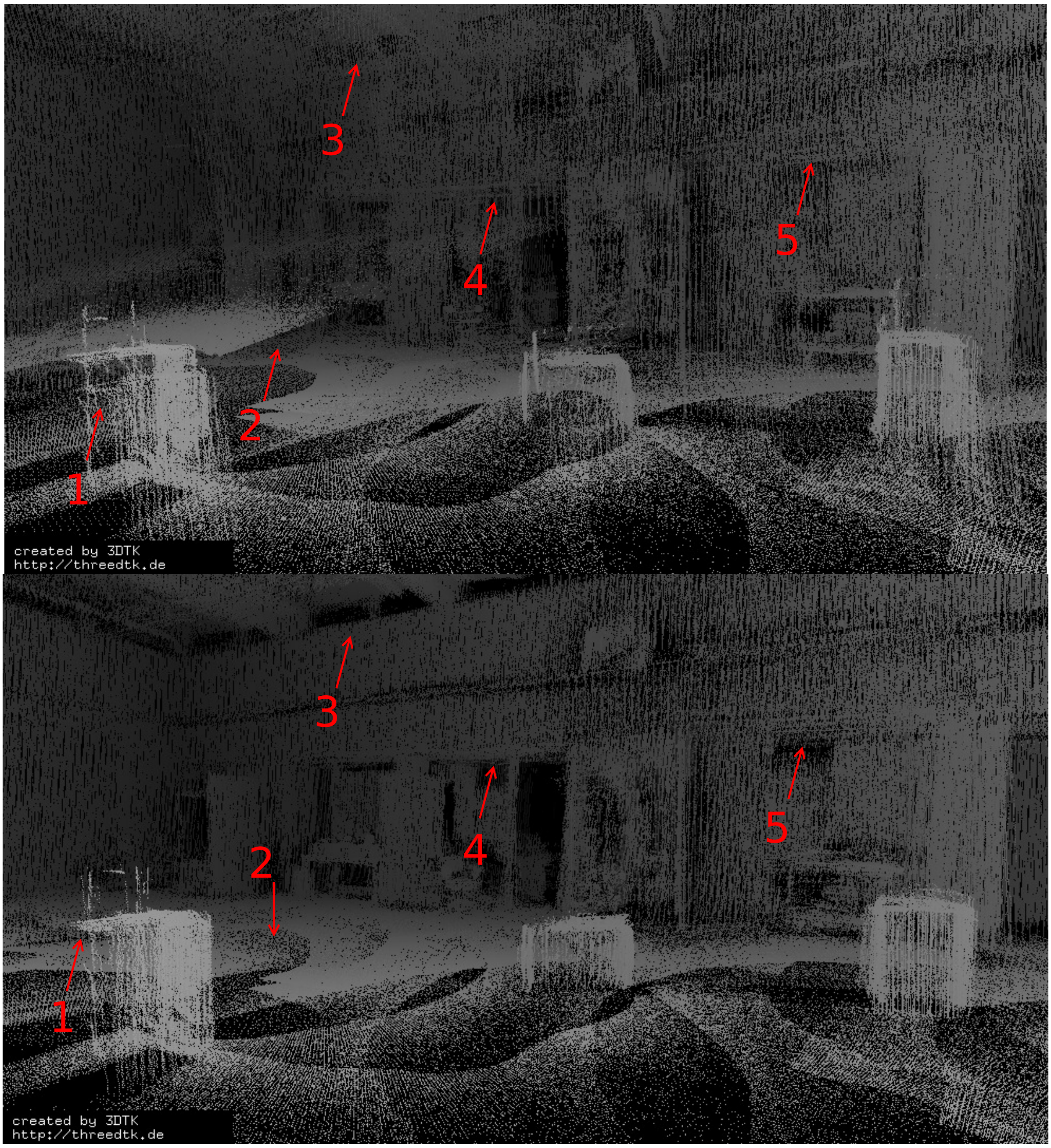


Figure 7. 3D point cloud of Experiment 1 without reflectance values. The color denotes the height. The boxes (1) and the rotated floor (2) converge, the windows at the ceiling (3) become visible. However, the office doors (4 and 5) remain rotated.

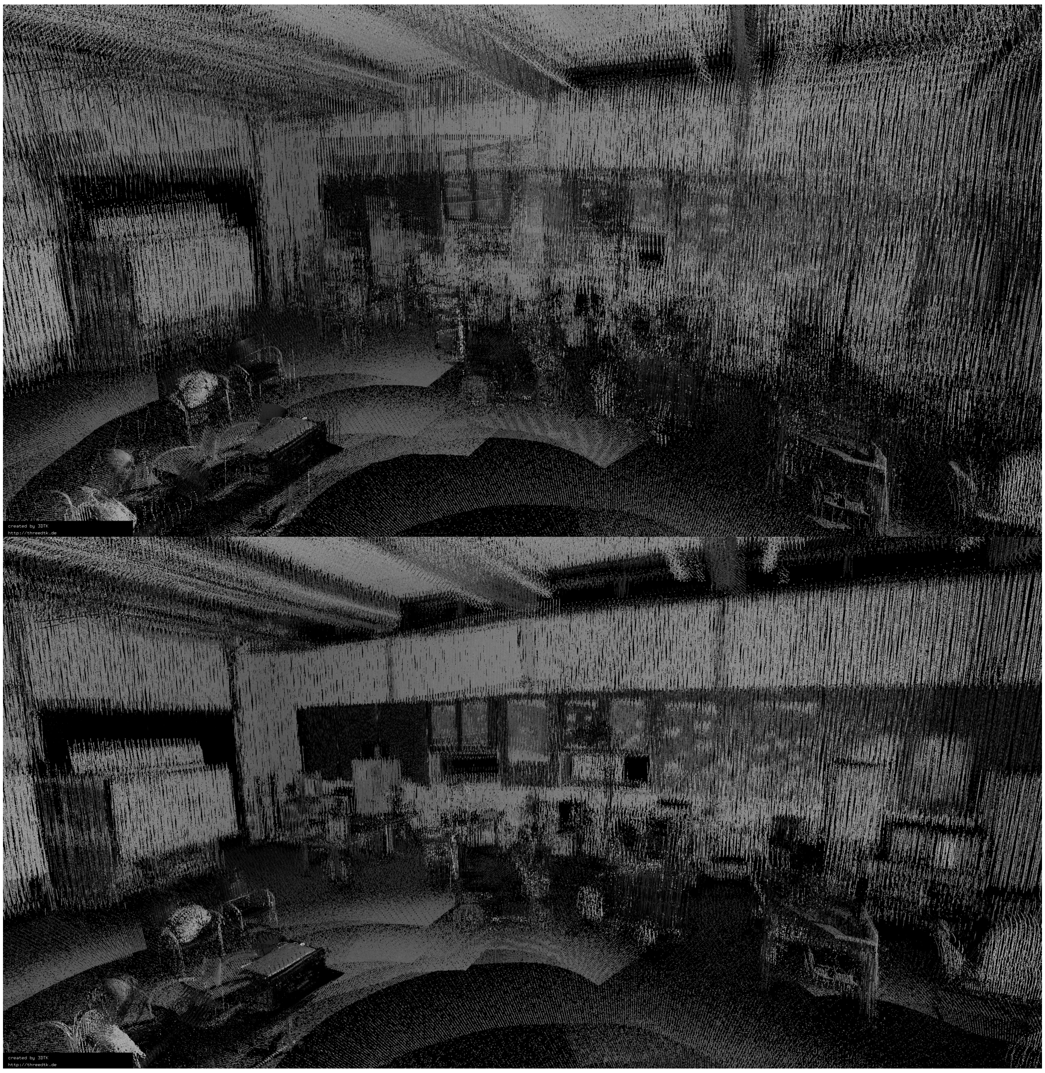
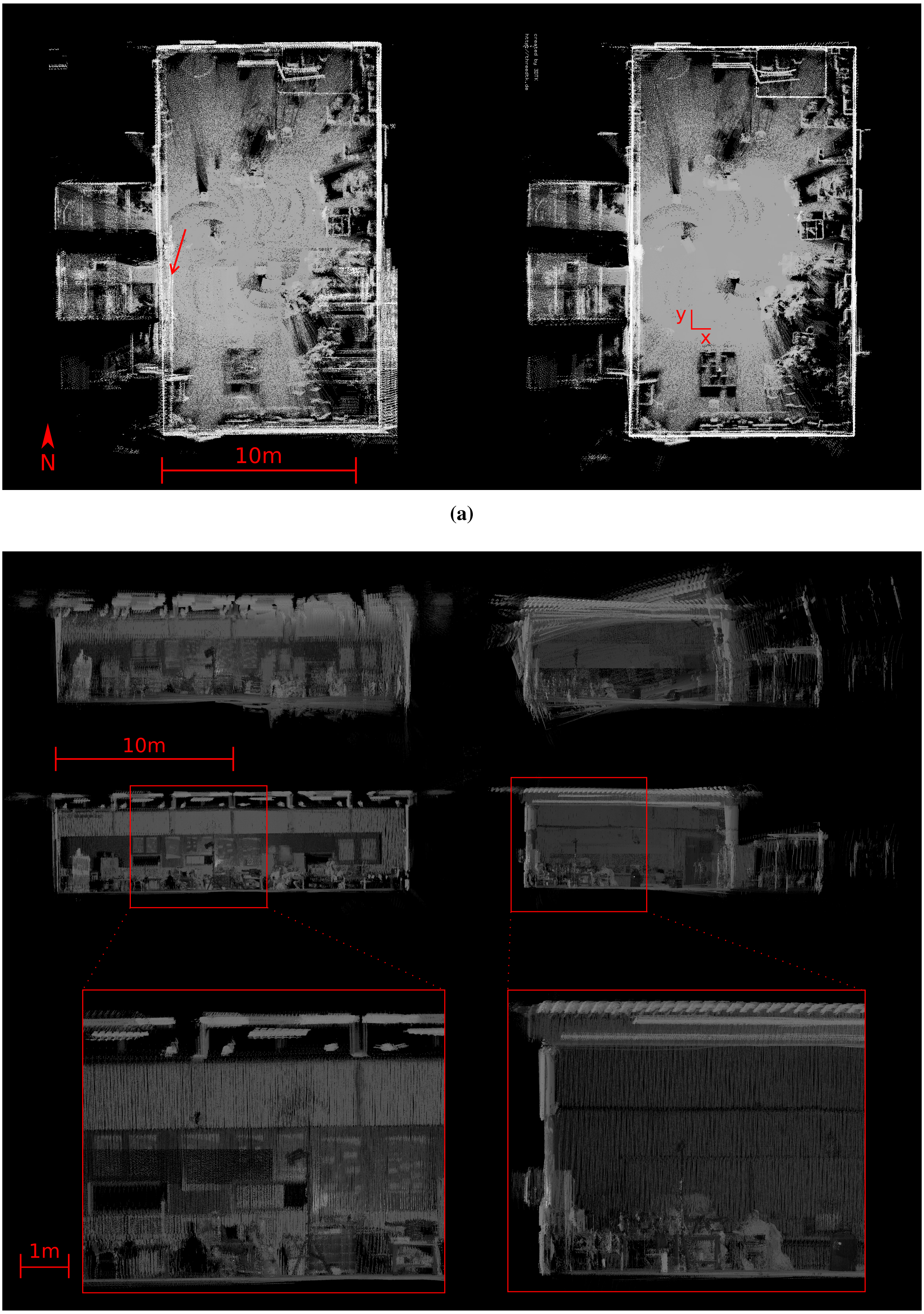


Figure 8. 3D point cloud of Experiment 1 with reflectance values. The quality improves, but the contours are still not sharp after optimization.

The higher quality of the initial 2D map is also reflected in the results of Experiments 2 and 3. In both experiments, the quality of the floor and the walls improve as Figure 9 and 10 depict. Only a few line scans remain slightly rotated. In Experiment 2 a large rotational error is eliminated, as visible in Figure 9. For Experiment 3, the main improvements appear at the floor and the ceiling.

The red arrows in Figure 11 mark some examples where improvements from the semi-rigid optimization are clearly visible in the 3D point cloud. Sharp edges emerge, and details become distinguishable, for instance the structure of the ceiling (Arrow 1) or the objects on the table on the right (Arrow 5) (Figure 11). Objects that appeared twice are converged successfully, e.g., the stand of the lamp (Arrow 4 in Figure 11). Both resulting point clouds represent the environment well. The changes in Figure 12 are not as noticeable, because the errors in the original point cloud directly after unwinding were already small.

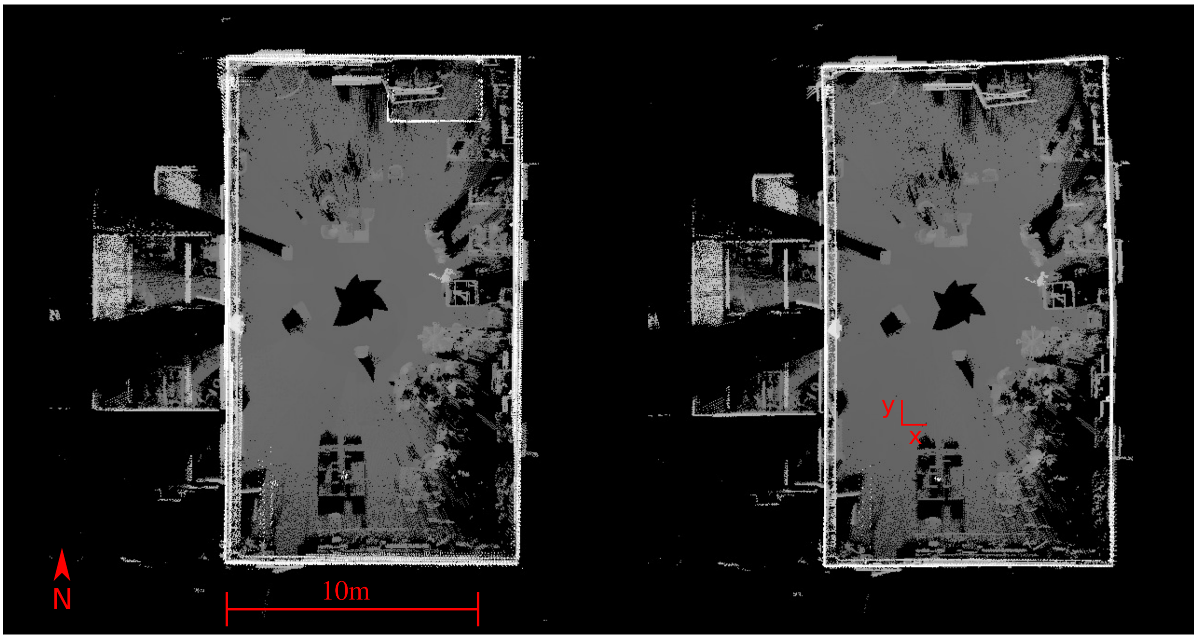
In addition to the visual inspection, the accuracy of the backpack mobile mapping system is evaluated quantitatively using the ground truth information from the iSpace positioning system. The generated trajectories in the local coordinate system of the backpack are compared to the ground truth trajectories from the iSpace system as described in Section 4.4.2.



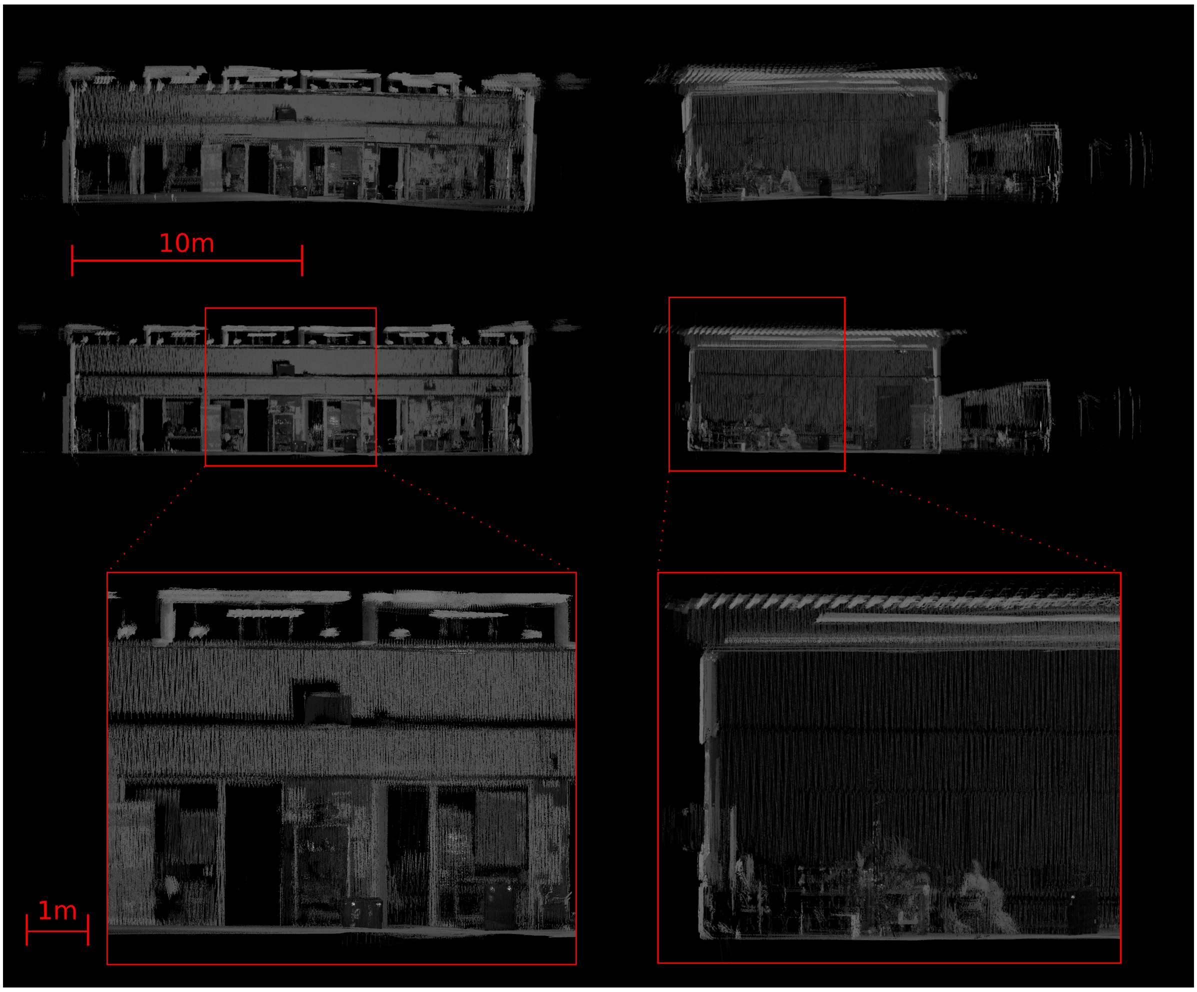
(a)

(b)

Figure 9. 3D point cloud of Experiment 2. (**a**) Bird's eye view. The thickness of the walls decreased by optimization (red arrow). Depicted is the origin of the iSpace coordinate system. (**b**) Side views before (top) and after optimization (bottom). The scans are rotated by semi-rigid registration, such that the floor matches.



(a)



(b)

Figure 10. 3D point cloud of Experiment 3. (**a**) Bird's eye view. The thickness of the walls decreased by optimization. Depicted is the origin of the iSpace coordinate system; (**b**) Side view before (top) and after (bottom) optimization.

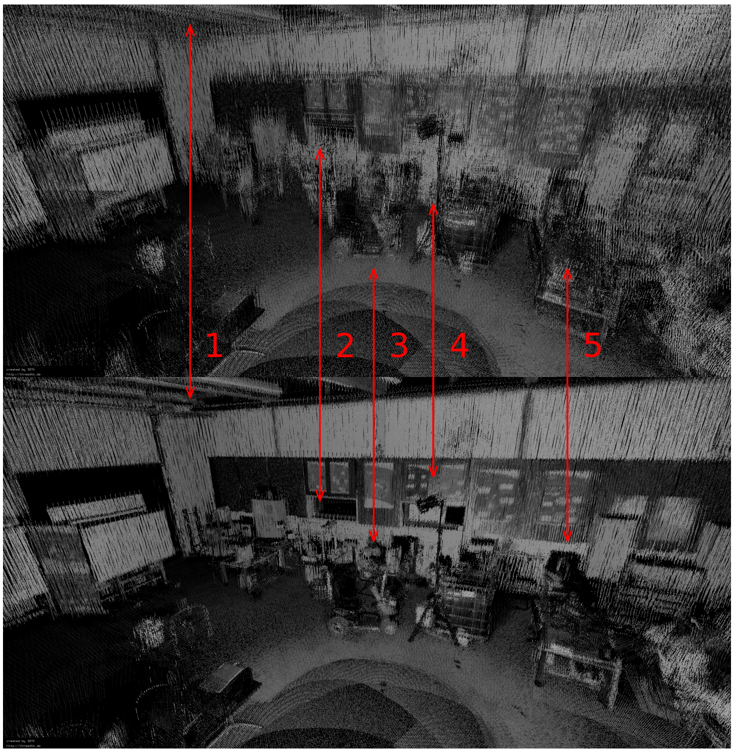


Figure 11. Details from the 3D point cloud of Experiment 2. The arrows mark some examples for the improvements by semi-rigid registration. 1: the structure of the ceiling gets clearer. 2: the edges of the windows do match. 3: the scooter appears separated from the structure behind. 4: the tripod of the lamp appearing twice converged. 5: the objects on the table are distinguishable.

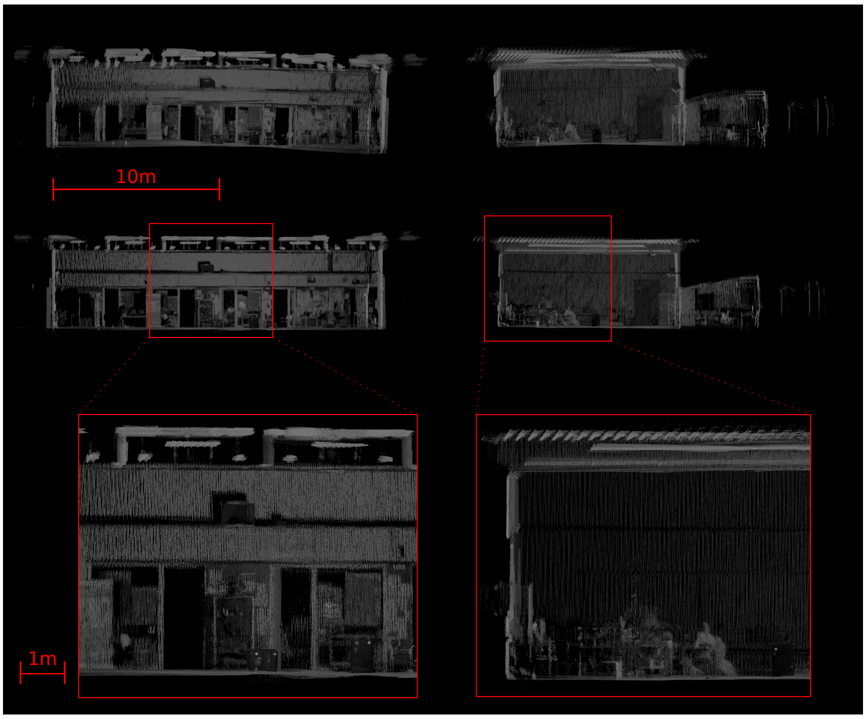


Figure 12. 3D view of Experiment 3 with trajectory. The structure of the scene improves, e.g., at the ceiling.

The resulting position and orientation errors for each experiment are given in Table 2. For all three experiments, the error after applying the semi-rigid optimization is noted. Additionally, for comparison, the error before optimization is noted for Experiment 2. The visualization of the position and orientation errors in axis angle representation for the entire trajectories is shown in Figures 13–15.

Table 2. Absolute trajectory error of the experiments. The * denotes the error before optimization.

	Position Error (cm)			**Orientation Error (°)**					
Exp.	**RMSE**	**Min**	**Max**	**RMSE**	**Min**	**Max**	**RMSE**	**Min**	**Max**
1	11.493	1.807	23.469	2.960	0.152	11.039	2.114	0.00034	6.687
2 (*)	6.135	0.232	29.369	6.685	0.380	19.927	2.270	0.00011	5.239
2	6.008	0.450	25.555	2.110	0.106	7.440	1.437	0.00040	4.495
3	3.405	0.401	8.520	1.714	0.033	4.158	1.335	0.00017	3.372
					Axis angle			Yaw rotation	

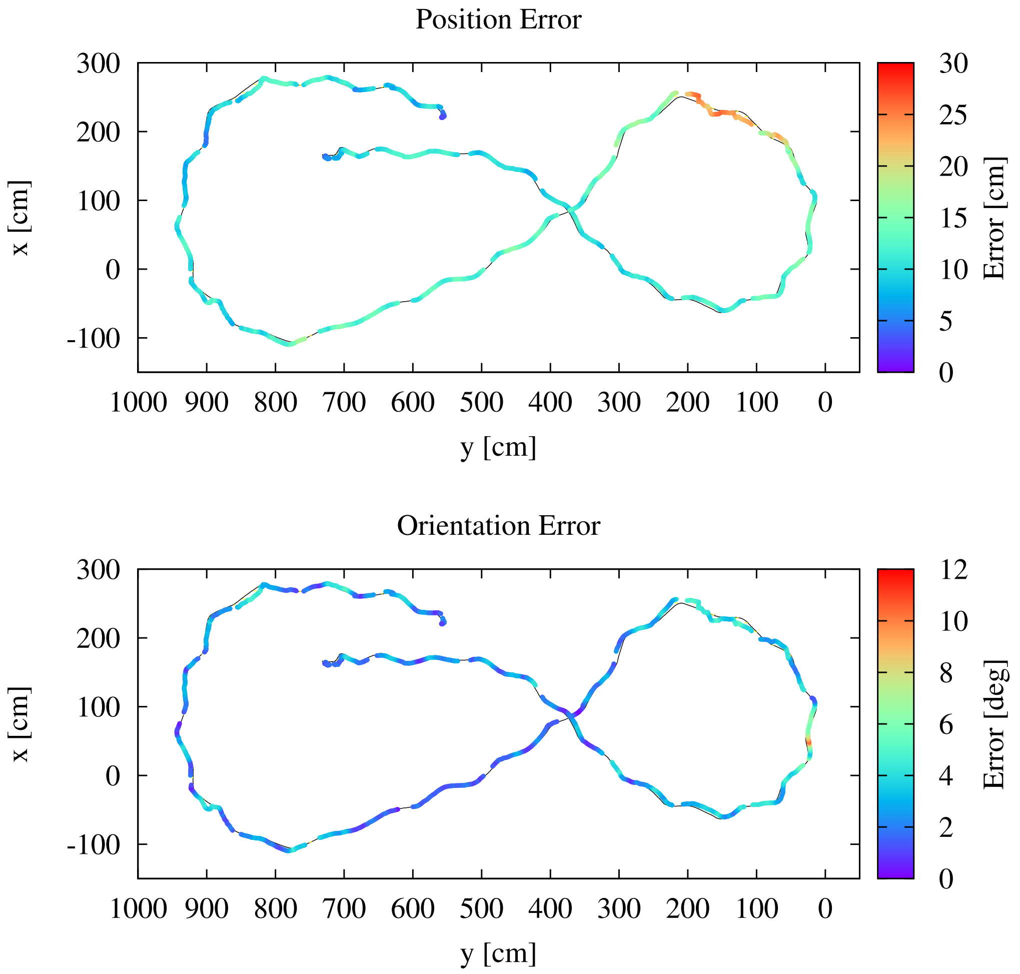
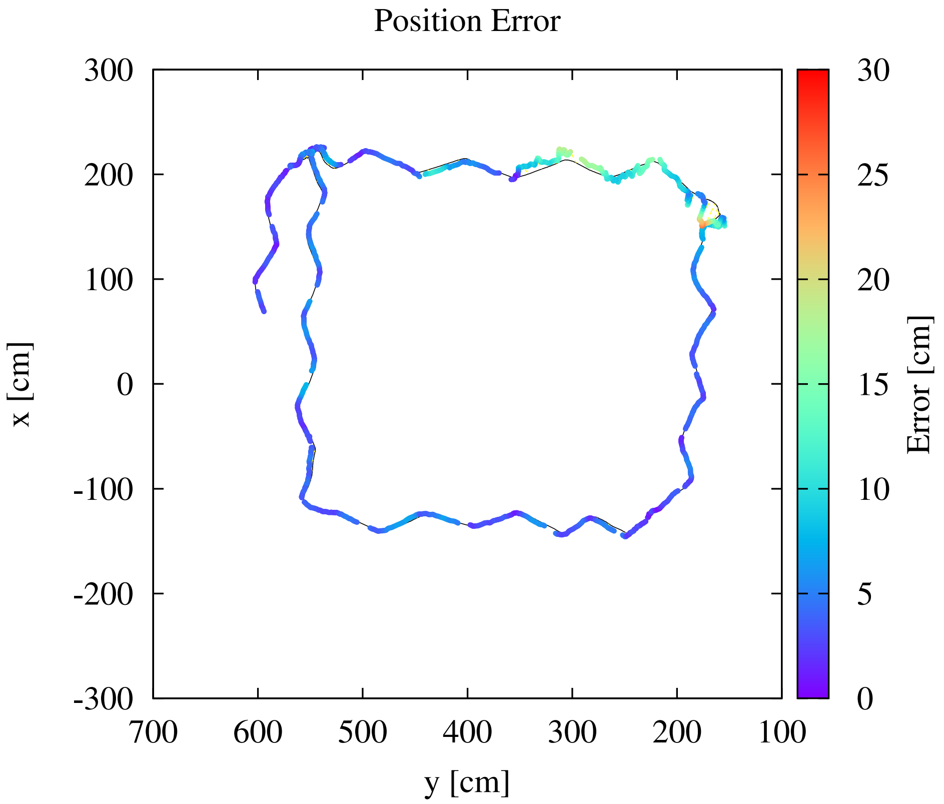


Figure 13. Experiment 1. (**Top**): position error after applying semi-rigid registration. (**Bottom**): orientation error. The black line denotes the ground truth trajectory and the colored points the initial estimated trajectory.



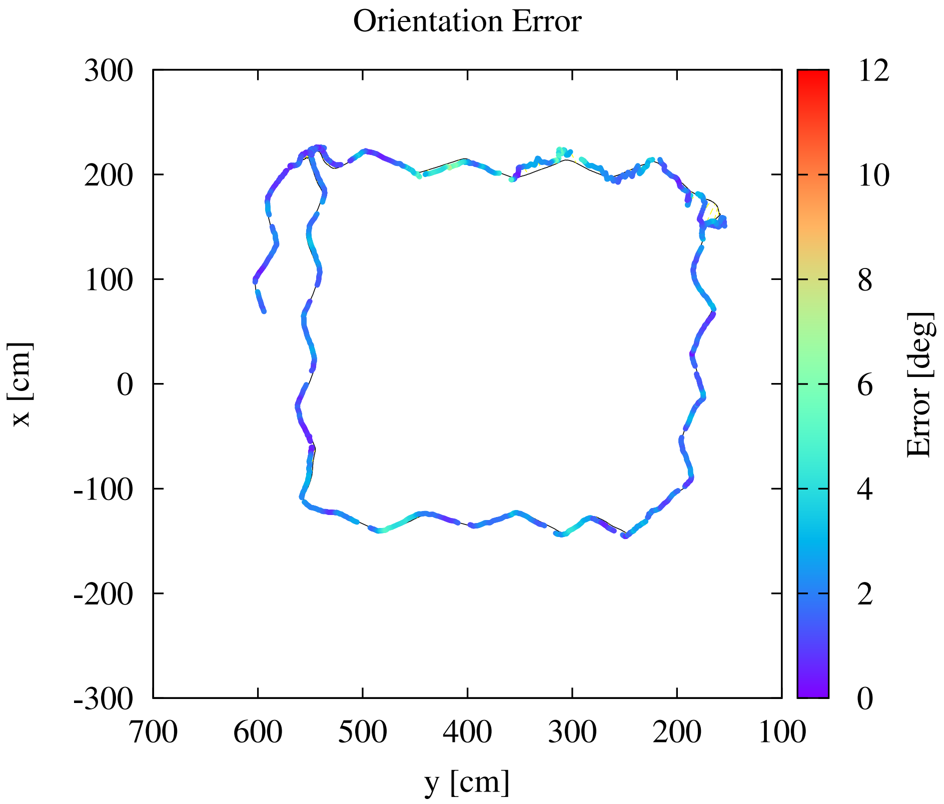


Figure 14. Experiment 2. (**Top**): position error. (**Bottom**): orientation error.

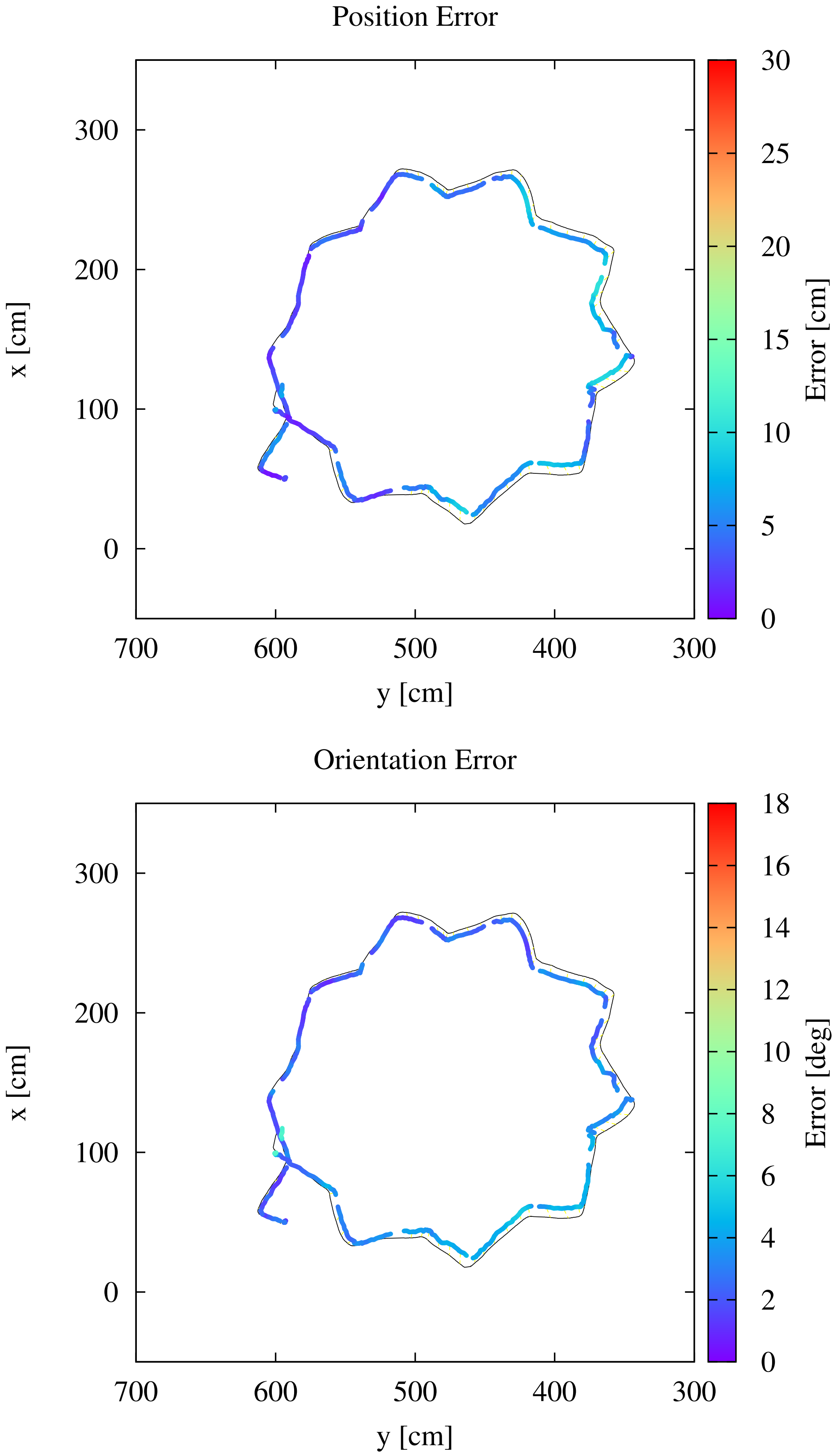


Figure 15. Experiment 3. (**Top**): position error. (**Bottom**): orientation error.

Figure 13 shows the error for Experiment 1. The main part of the optimized trajectory has an accuracy of 10 cm to 15 cm. Only the part in the upper left, which corresponds to the aforementioned inaccuracies in the HectorSLAM output, shows an error above 20 cm. The orientation is corrected by the semi-rigid optimization. Even in those areas where the position error remains above 20 cm, the resulting orientation error is below 6°. Only the small part on the right has an error up to 11°, which causes the rotated area marked in Figure 6. In contrast to that, Figure 16 gives an example for the error aligning the start poses of the trajectories for Experiment 1. Since the estimated starting pose is erroneous, the position error increases with growing distance from the starting pose up to a maximum of 41.177 cm. Therefore, this method is not evaluated further.

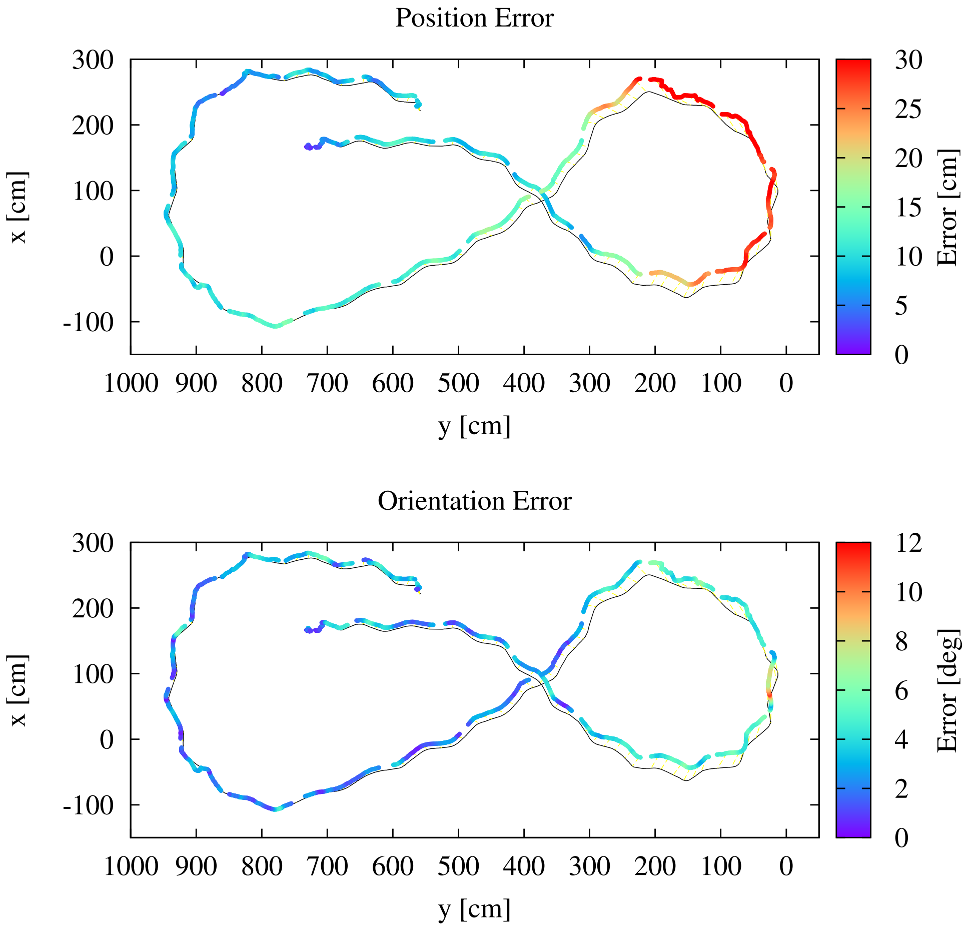


Figure 16. Experiment 1. Aligning trajectories without ICP. (**Top**): position error after applying semi-rigid registration. (**Bottom**): orientation error.

Comparing the trajectories before and after the semi-rigid optimization points out the improvements. This is shown exemplarily for Experiment 2. For Experiment 2, the inaccuracy in the position of the initially generated trajectory (Figure 17) remains below 10 cm, except for the small part in the upper right corner where the error grows up to 29 cm. Deforming the trajectory by the optimization step even seems to increase the position error in some parts of the trajectory in order to decrease the maximum error (Figure 14). Table 2 points out that the optimization changes the position errors only by a few centimeters. Considering the bird's eye view in Figure 9, which shows that already, the initial point cloud represents the floor plan of the robotics hall well, this result can be expected. The major optimization work is done by reducing the orientation errors. They concern mainly the orientation in roll and pitch direction, as the side views in Figure 9 visualize. The initial trajectory has an orientation error near 20° over a distance of two meters. This error is reduced such that the maximum error of the entire trajectory does not exceed 7°. Experiment 2 also demonstrates the deficits of considering only the error in yaw rotation. Table 2 shows that the maximal orientation error in yaw rotation is 5° in contrast to the maximal error of 20° using the axis angle representation. Moreover, the errors in yaw rotation before and after optimization do not differ much, while the root mean square error is reduced to one third by considering the axis angle representation.

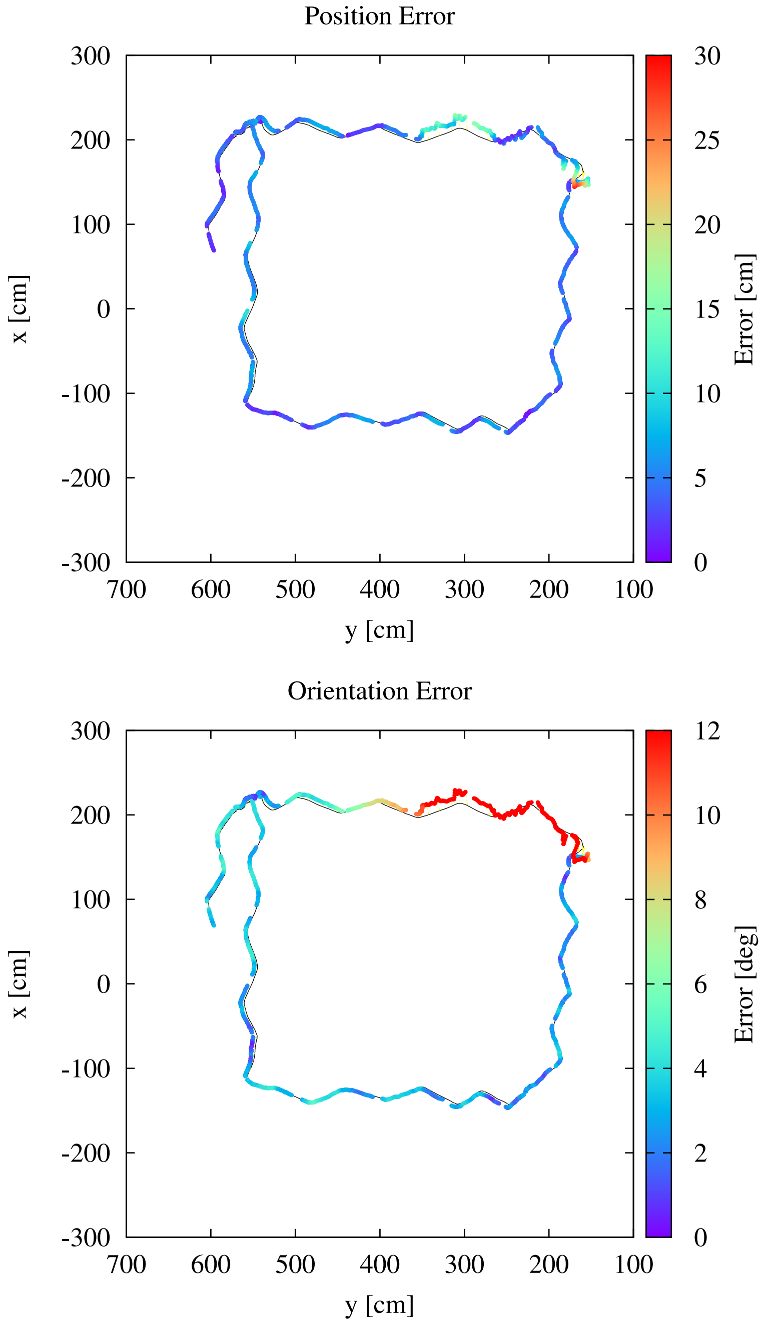


Figure 17. Error for Experiment 2 before optimization. (**Top**): position error. (**Bottom**): orientation error.

As expected from visual inspection, the position error, as well as the orientation error for Experiment 3 are small (Figure 15). They do not exceed 9 cm in position and 4° in orientation, respectively. This experiment shows that the backpack mobile mapping system is able to produce a good point cloud representation of the environment.

5.2. Outdoor Dataset

Up to now, the backpack was only tested in indoor environments. Characteristic for those environments are optimal circumstances, like a planar ground. Moreover, the size of the scene is most often restricted; hence, all of measurements are within the range of the 2D LiDAR. In contrast to this, outdoor environments are more complex. Besides the rough terrain, large distances outside the range of the SICK scanner predominate.

For evaluation in an urban environment, we walked a loop around the robotics hall, as depicted in Figure 18. The outdoor robotic test area consists of challenging analog moon and mars environments, a crater, trees and plants and other structures. Figure 19 shows part of the urban scene. The data was acquired during a walk of 276 seconds. The Riegl VZ-400 captured a total of 28,811 line scans, *i.e.*, 7,936,966 point measurements. The results were inspected visually.

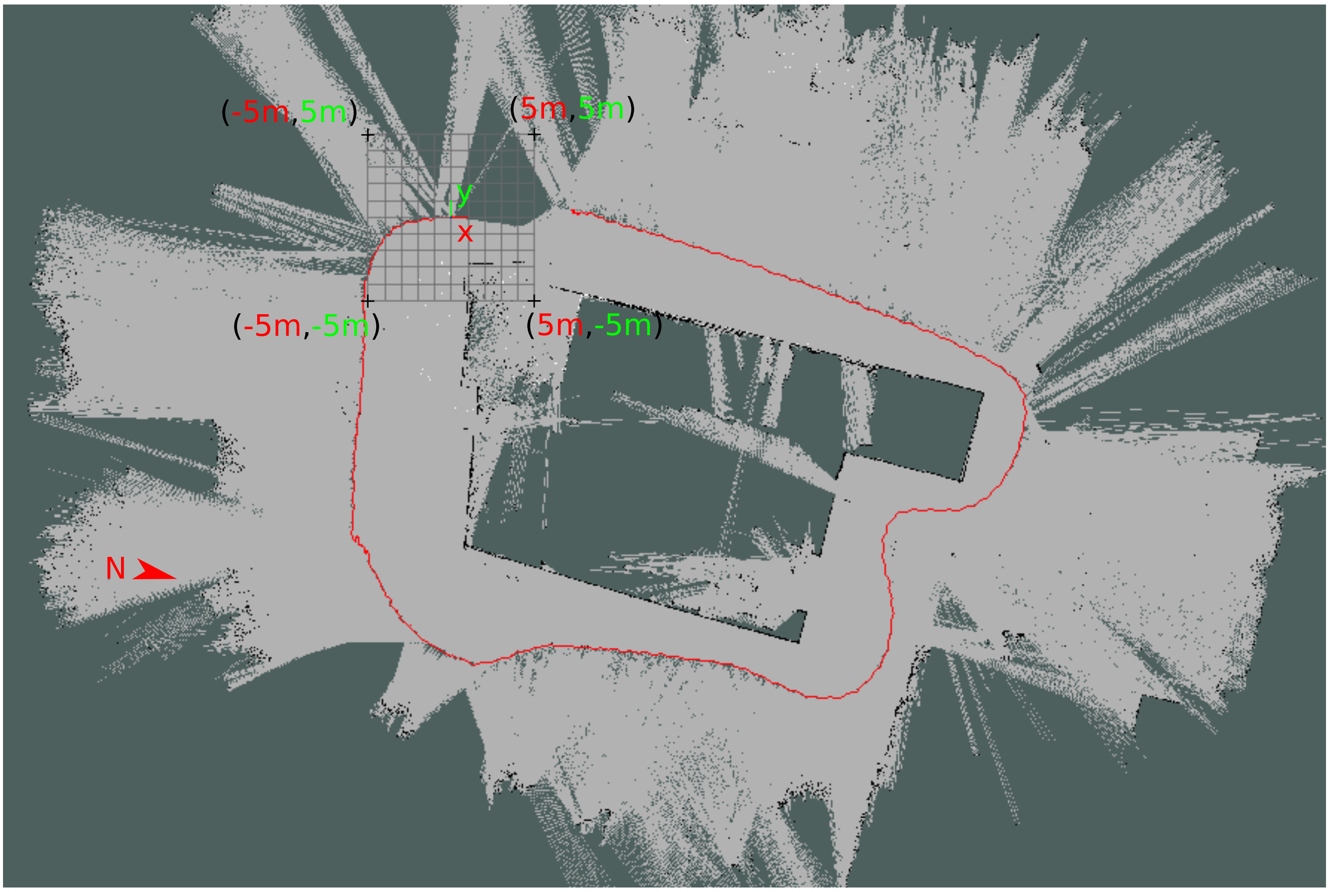


Figure 18. 2D map of the outdoor dataset created by HectorSLAM in the local coordinate system. Inaccuracies arose when the SICK scanner covered open terrain. The superimposed grid has a size of 10 × 10 m^2, and the red line denotes the trajectory.

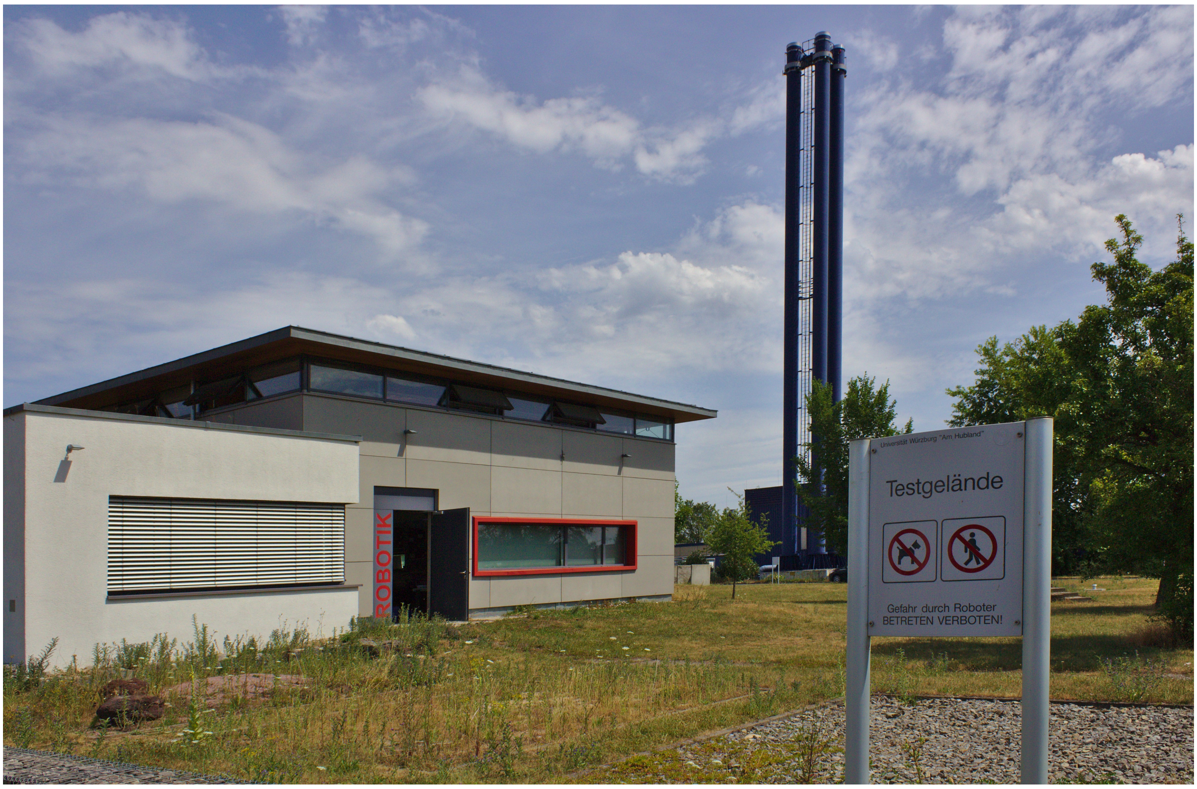


Figure 19. The robotics hall and the robotic test field were the experiment took place. In the foreground, the areas for simulating different planetary surfaces. In the background, the chimney of the Technischer Betrieb of the university.

The output of HectorSLAM is shown in Figure 18. Predominant large distances out of the range of the SICK scanner cause inaccuracies in the 2D map. The bottom left corner was not registered correctly by HectorSLAM, because the SICK field of view covered only open terrain in this area. Furthermore, the left wall of the building appears too short. Although enough features were in range, like parked cars, the homogeneous structure of the wall led to the wrong matches.

However, aligning the 3D scans is still possible by exploiting the longer range of the Riegl VZ-400. To reduce the errors from the 2D pose estimation, we apply an additional preregistration step using a modified version of the well-known ICP algorithm. One complete rotation of the Riegl VZ-400 consisting of 540 line scans is treated as one scan. These scans are registered by ICP with a maximum point-to-point distance of 150 cm. The computed transformation is distributed over all 540 line scans before applying the semi-rigid registration step, thus reducing the initial error.

The bird's eye view in Figure 20 shows that the rotation error in the bottom right corner at No. 1 decreases when applying semi-rigid SLAM. Despite some inaccuracies, the walls on the left match. The parked cars at No. 3 do not appear twice anymore. Figure 21 visualizes the optimization of the walls from another perspective. However, some inaccuracies remain after optimization. In Figure 20, the walls at No. 2 were not matched perfectly, although the length of the wall on the left was corrected. One drawback of the preregistration step is visible at the corner of the robotics hall at No. 4. Due to the distribution of the initial transformation error over all scans, some inaccuracies increased. In general, the resulting point cloud represents the environment well. Figure 22 shows some details from the scene. In this area, even the trees appear consistent. For comparison photos of the scene are shown in Figure 23.

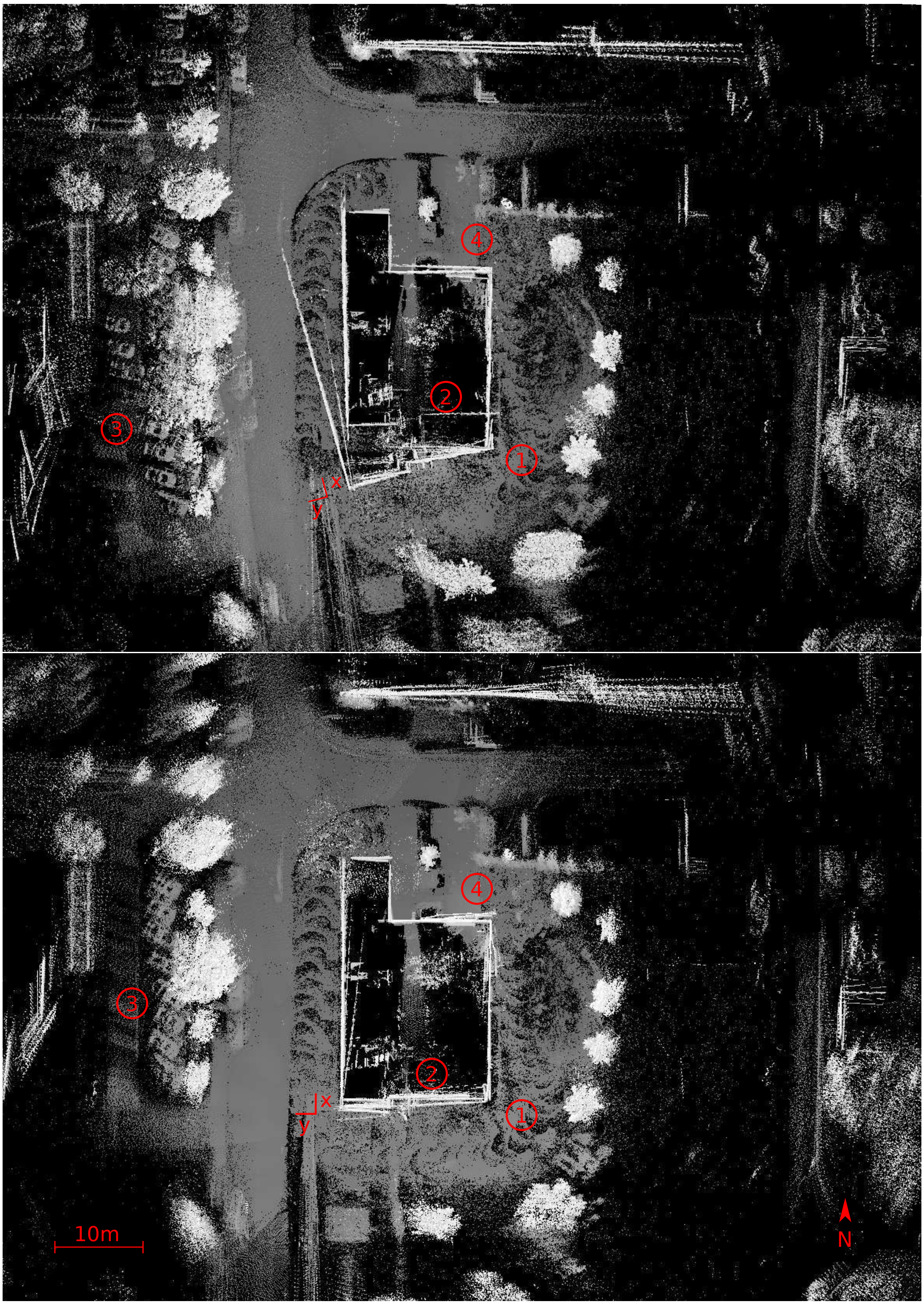


Figure 20. Bird's eye view before (top) and after optimization (bottom). The rotation error at (1) decreased by semi-rigid optimization. The walls at (2) converged, but still do not match perfectly. The parking cars at (3) do not appear twice any more. At (4), inaccuracies increased due to distributing the initial transformation error during preregistration. The origin of the local coordinate system corresponds to the starting point of the trajectory.

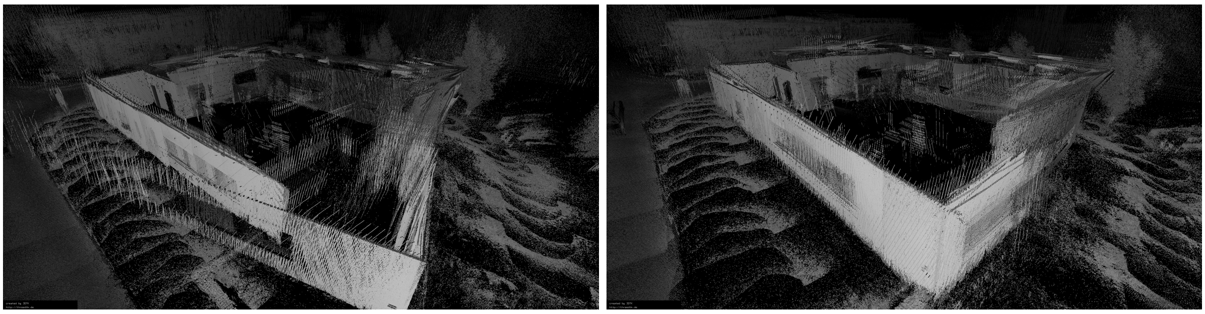


Figure 21. The walls on the left do match after optimization.

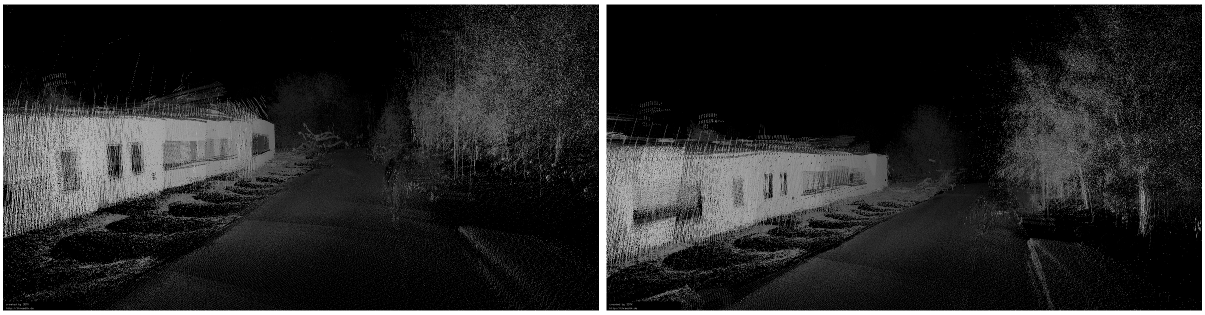


Figure 22. Reducing the rotation error leads to matching walls and a straight road.

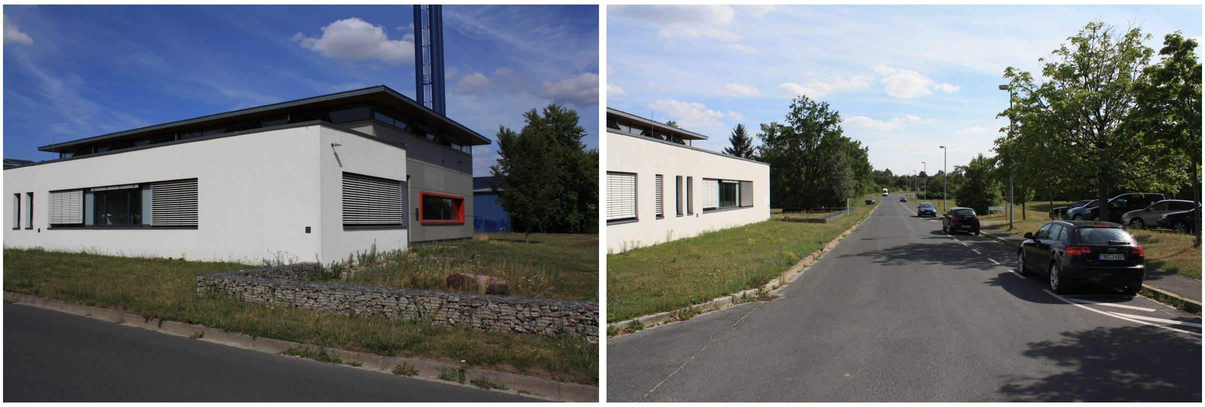


Figure 23. Photos showing the scene corresponding to the point clouds seen in Figures 21 and 22.

6. Conclusions

The paper evaluates the accuracy of the mapping process using our backpack mobile mapping system. The system was originally designed for indoor applications. It uses an inexpensive IMU and a 2D laser scanner to acquire initial pose estimates for the main sensor, the Riegl VZ-400 laser scanner. The evaluation shows that even in cases where the 2D SLAM algorithm yields poor results, the effective semi-rigid SLAM algorithm manages to align the 3D point cloud data well. The normal human walking motion was nicely compensated by the two mapping algorithms, yielding a good point cloud representation of the environment with maximum pose errors of $\approx 25\,\text{cm}$ in translation and $\approx 7^\circ$ in rotation. Inaccuracies are mostly limited to small fractions of the trajectory, especially in areas where the operator failed to walk smoothly. In the remaining part, the errors are significantly smaller. Furthermore, the yaw rotation accounts for only small parts of the overall error. By exchanging the lead batteries with lighter lithium polymer batteries, we expect to make walking easier for the operator and to reduce the jittering. Due to the nature of the iSpace system, the evaluation was limited to an environment containing mostly open space. Future work will focus on comparing the resulting point cloud to results acquired from terrestrial laser scanning and other personal laser scanning systems in more challenging environments.

For the first time, the system was also tested in an urban outdoor environment. Despite the relatively low range of the 2D laser scanner and the resulting inaccuracies in the map, the final 3D point cloud represented the environment well. Challenges, such as uneven and rough terrain and little to no point features within the range of the 2D laser scanner in open spaces, were successfully mastered. In future work, we plan to investigate further into the urban use of the system. By incorporating a GNSS device, we expect to increase the accuracy of the initial map and, thus, to reach better preconditions for the semi-rigid optimization, yielding better results. The 2D laser scanner works best in narrow areas where many items are within range, while GNSS performs best in open space. Thus, the combination of both seems an ideal solution. In addition, we plan to incorporate an urban model for improving the GNSS measurements based on sensed 3D geometry, similar to [30].

Acknowledgments

This publication was funded by the German Research Foundation (DFG) and the University of Würzburg in the funding program Open Access Publishing.

Author Contributions

All authors contributed equally to the work.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Nüchter, A.; Elseberg, J.; Borrmann, D. Irma3D—An intelligent robot for mapping applications. In Proceedings of the 3rd IFAC Symposium on Telematics Applications (TA'13), Seoul, South Korea, 11-13 November 2013; pp. 119–124.
2. VIAmetris. Mobile Mapping Technology. Available online: http://www.viametris.com/ (accessed on 30 July 2015).
3. Thomson, C.; Apostolopoulos, G.; Backes, D.; Boehm, J. Mobile laser scanning for indoor modelling. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *II-5/W2*, 289–293.
4. Google. Street View Trolley. Available online: http://maps.google.com/intl/en/maps/about/behind-the-scenes/streetview (accessed on 6 September 2015).

5. NavVis. M3 Trolley. Available online: https://www.navvis.com/explore/trolley/ (accessed on 30 July 2015).
6. Frederic Lardinois, TC. Google Unveils The Cartographer, Its Indoor Mapping Backpack. Available online: http://techcrunch.com/2014/09/04/ google-unveils-the-cartographer-its-indoor-mapping-backpack/ (accessed on 30 July 2015).
7. Bosse, M.; Zlot, R.; Flick, P. Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping. *IEEE Trans. Robot.* **2012**, *28*, 1104–1119.
8. Leica. Leica Pegasus:Backpack. Available online: www.leica-geosystems.com/de/ Leica-PegasusBackpack_106730.htm (accessed on 30 July 2015).
9. Nüchter, A.; Borrmann, D.; Elseberg, J.; Redondo, D. A Backpack-mounted 3D mobile scanning system. *Allg. Vermess.-Nachr.* **2015**, in press.
10. Kukko, A.; Kaartinen, H.; Hyyppä, J.; Chen, Y. Multiplatform mobile laser scanning: Usability and performance. *Sensors* **2012**, *12*, 11712–11733.
11. Corso, N.; Zakhor, A. Indoor localization algorithms for an ambulatory human operated 3D mobile mapping system. *Remote Sens.* **2013**, *5*, 6611–6646.
12. Lehtola, V.V.; Virtanen, J.P.; Kukko, A.; Kaartinen, H.; Hyyppä, H. Localization of mobile laser scanner using classical mechanics. *ISPRS J. Photogramm. Remote Sens.* **2015**, *99*, 25–29.
13. Kohlbrecher, S.; Von Stryk, O.; Meyer, J.; Klingauf, U. A flexible and scalable slam system with full 3D motion estimation. In Proceedings of the 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Kyoto, Japan, 1-5 November 2011; pp. 155–160.
14. Fraunhofer IPM. MoLaS Technology Workshop 2014. Available online: http://www.ipm. fraunhofer.de/en/tradefairs-events/molas-workshop-2014.html (accessed on 30 July 2015).
15. Nikon Metrology. iSpace—Portable Metrology System User Manual and Startup Guide. Available online: http://www.nikonmetrology.com (accessed on 30 July 2015).
16. Phidgets. Products for USB Sensing and Control. Available online: http://www.phidgets.com/ products.php?product_id=1044 (accessed on 30 July 2015).
17. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 17 May 2009; Volume 3, p. 5.
18. Nüchter, A.; Borrmann, D.; Elseberg, J. 3DTK—The 3D Toolkit. Available online: http://slam6d.sourceforge.net/ (accessed on 30 July 2015).
19. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), Vancouver, BC, Canada, 24–28 August 1981; Volume 2, pp. 674–679.
20. Borrmann, D.; Elseberg, J.; Lingemann, K.; Nüchter, A.; Hertzberg, J. Globally consistent 3D mapping with scan matching. *Robot. Auton. Syst.* **2008**, *56*, 130–142.
21. Elseberg, J.; Borrmann, D.; Nüchter, A. Algorithmic solutions for computing precise maximum likelihood 3D point clouds from mobile laser scanning platforms. *Remote Sens.* **2013**, *5*, 5871–5906.
22. Besl, P.; McKay, N. A method for registration of 3D-shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256.

23. Nüchter, A.; Elseberg, J.; Schneider, P.; Paulus, D. Study of parameterizations for the rigid body transformations of the scan registration problem. *Comput. Vis. Image Underst.* **2010**, *114*, 963–980.
24. Stoyanov, T.; Lilienthal, A.J. Maximum likelihood point cloud acquisition from a mobile platform. In Proceedings of the 2009 International Conference on Advanced Robotics (ICAR2009), Munich, Germany, 22–26 June 2009; pp. 1–6.
25. Bosse, M.; Zlot, R. Continuous 3D scan-matching with a spinning 2D laser. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA'09), Kobe, Japan, 12–17 May 2009; pp. 4312–4319.
26. Davis, T.A. Algorithm 849: A concise sparse Cholesky factorization package. *ACM Trans. Math. Softw.* **2005**, *31*, 587–591.
27. Horn, B.K.P.; Hilden, H.M.; Negahdaripour, S. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A* **1988**, *5*, 1127–1135.
28. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; pp. 573–580.
29. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *PAMI-9*, pp. 698–700.
30. Maier, D.; Kleiner, A. Improved GPS sensor model for mobile robots in urban terrain. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–7 May 2010; pp. 4385–4390.