ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume V-1-2022
XXIV ISPRS Congress (2022 edition), 6–11 June 2022, Nice, France

KINECT BACKPACK FOR RAPID MOBILE INDOOR MAPPING

Michael Bleier*, Joschka van der Lucht, Andreas Nüchter

Informatics VII – Robotics and Telematics, Julius Maximilian University of Würzburg, Germany
{michael.bleier, joschka.lucht, andreas.nuechter}@uni-wuerzburg.de

KEY WORDS: kinect, backpack mapping, mobile mapping, continuous-time slam, rgb-d sensors

ABSTRACT:

In recent years, 3D surveying using mobile mapping has become increasingly popular. Handheld or backpack mounted systems resulted in new uses and applications because they enable fast data acquisition in areas that are difficult to access. Many of these applications do not require high-precision measurement, but they often involve highly complex environments. In this paper, we present a mobile mapping backpack based on four Microsoft Kinect RGB-D sensors and an Intel T265 tracking camera. The data processing takes place in three steps. Starting with a first trajectory and map created using visual-inertial odometry, which is subsequently optimized by means of a continuous-time ICP. Finally, the trajectory is further improved by a continuous-time SLAM approach. In this work, a data set was recorded and analyzed in a bunker facility with the system carried at a walking speed of approximately 0.7 meters per second. For the evaluation, these data are compared with a reference data set, recorded with a Riegl VZ-400 laser scanner.

1. INTRODUCTION

Due to reduced costs and a wide range of sensors, the demand for 3D indoor measurements has increased in recent years. Since interiors are typically rather angled and small, mobile systems are often better suited than static ones. The possibility to capture an object while moving saves a lot of time and simplifies the avoidance of shadowing and incomplete data collection. In addition to 3D mapping of the environment for surveying purposes, indoor 3D maps are often used for virtual tours (Nocerino et al., 2017). Therefore, the acquisition of color data, in addition to the 3D data, is increasingly an important requirement.

A solution specifically designed for this purpose is, for example, the NavVis 3D Mapping Trolley (NavVis, 2021). This trolley has multiple cameras for a 360 degree image and multiple laser scanners. This can be easily pushed through rooms and hallways. Since all sensors are firmly mounted on a trolley, the movement of the sensors is relatively uniform and only in one plane, this is very advantageous for the post-processing of the data. At the same time, this becomes a problem if the area to be detected is no longer on one level and, for example, steps need to be traversed.

Hand-held systems, such as the Zebedee 3D sensor system (Bosse et al., 2012), offer somewhat more flexibility. The user carries a rotating Hokuyo 2D laser scanner and a front-facing camera in his hand. This device allows a high flexibility in movement. Since it is a light and small device, which can be carried well over several levels and stairs. A disadvantage of systems of this type is that it is exhausting to carry the device hand-held for a long duration. In addition, care must be taken to carry the device as smoothly as possible and not to wobble too much back and forth while walking.

Therefore, another approach is to wear the scanner system as a backpack. This still gives the user the flexibility to move freely, similarly to the hand-held system. At the same time, it frees the user from having to hold the heavy system in his hand. In addition, this makes it much easier to achieve a smooth, glide-like movement of the sensor.

Figure 1. Proposed indoor mapping backpack with four Microsoft Kinect 2 RGB-D cameras and an Intel T265 stereo camera for visual-inertial odometry.

Examples include the Google Streetview Backpack System (Frederic Lardinois, TC, 2015, Hess et al., 2016) or the Leica Pegasus Backpack (Leica Geosystems, 2021). The Pegasus system uses two Velodyne VLP-16 scanners for 3D acquisition and five color cameras. The ITC-IMMS backpack (Karam et al., 2019) is another example. This backpack mapper consists of three Hokuyo UTM-30LX scanners mounted on a carrying platform that extends over the user's shoulder and head. This is not always ideal for use in cluttered environments as care must always be taken not to hit any obstacles with the sensor.

Moreover, spinning Lidar systems based on terrestrial laser scanners have been proposed. The backpack of (Lauterbach et al., 2015) consists of a continuously panning

* Corresponding author

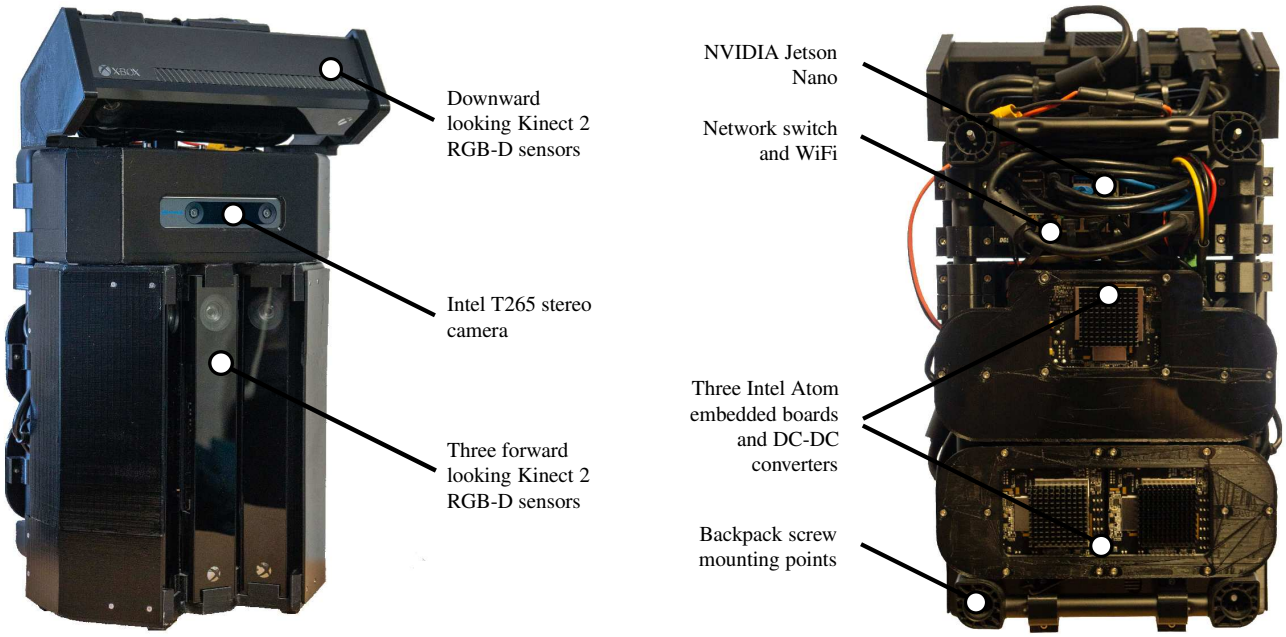

Figure 2. Detail view of the backpack system and its components. Left: Three forward looking Kinects capture the opposite wall and the area directly behind the backpack. An additional downward looking Kinect ensures a reliable detection of the ground. In the center is an Intel Realsense T265 stereo camera for visual-inertial odometry. Right: On the back of the backpack the embedded computing boards for data capture and processing, networking and power supply components are mounted.

Riegl VZ-400 and a SICK LMS-100, which is used for an initial trajectory estimation.

Some systems rely on global navigation satellite system (GNSS), which is often not available in indoor environments, such as the Personal Laser Scanning System proposed by (Liang et al., 2014) using a single FARO scanner. Similarly, the commercially available ROBIN system features a RIEGL VUX scanner and GNSS for positioning.

However, since all these backpacks are relatively large, heavy, and difficult to handle in confined environments, this paper presents a new system based on four Microsoft Kinect cameras. In addition to the smaller size, the backpack presented here is very cost-effective, as it avoids the use of expensive laser scanners or positioning sensors. Instead low-cost RGB-D cameras and off-the-shelf components are employed. Although the sensors used have a considerably shorter range than the laser scanners on the aforementioned backpacks (Khoshelham and Elberink, 2012), this is not as relevant for use in confined indoor spaces. The use of the Kinect cameras also has the advantage that one sensor records depth data and color data simultaneously. Thus, directly a textured 3D data set is obtained. Fig. 1 shows the proposed backpack system in action, mapping a bunker complex with narrow corridors and passages. As is visible, the backpack is very compact and does not protrude beyond the user, neither on the sides, nor upwards.

Due to the easy availability and low cost of a Microsoft Kinect, there are several approaches in the literature to use it for indoor mapping. In the paper by (Hsu et al., 2018) a small portable system consisting of a 2D laser scanner, a Kinect and an IMU is presented. For data fusion and calculation of the maps, they developed a so-called sensor fusion SLAM and were already able to achieve good results with it. Single handheld RGB-D cameras have been widely used in different configurations for indoor mapping (Newcombe et al., 2011, Dai et al., 2017). (Chen et al., 2018) is also working on the use of Kinect cameras for low-cost and efficient 3D indoor mapping. To achieve a larger field of view, three Kinect cameras were used, with a slightly overlapping field of view. In their work, they are mainly concerned with the calibration of the three sensors against each other. Compared to a reference data set from a terrestrial laser scanner, a deviation of more than $0.025\,\mathrm{m}$ was found for only 5% of the recorded points.

In the following, the structure of the system is first described in detail. Then an overview of the methods used for data fusion of the sensors and for calculation and correction of the trajectories is given. To evaluate the accuracy and quality of the obtained 3D data, a ground through data set was recorded using a terrestrial laser scanner. Finally, the results are discussed and improvement approaches for the future are identified.

2. TECHNICAL APPROACH

The backpack presented here is a low-cost development. Due to its small size and light weight, it is suitable for indoor spaces, narrow passages, cave systems and bunkers. It combines the flexibility of hand-held systems with maximum stability during movement. The following describes the setup, calibration and data processing in more detail.

2.1 Hardware Setup

The backpack consists of four Microsoft Kinect 2 RGB-D sensors. As seen on the left hand in Fig. 2 one of them is looking downward to capture the ground. The remaining three cameras are upright mounted to capture the walls and the ceiling. Here, the middle one is aligned straight back and captures everything behind the backpack. The other two cameras are aligned at a 43.5 degree angle and capture the opposite walls. Due to the upright orientation of the three cameras, the field of view is such that parts of the floor and ceiling is also captured. The

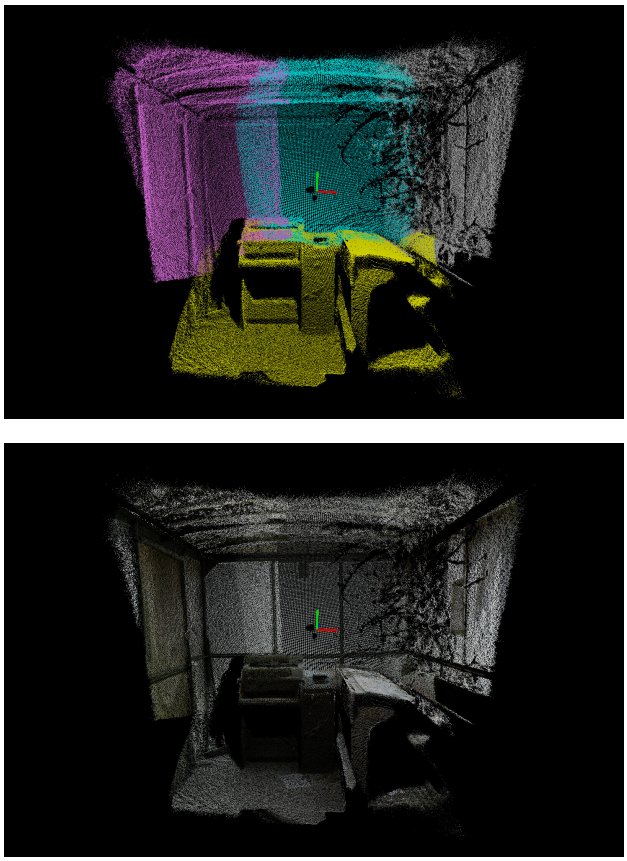

Figure 3. Combined point cloud created by the four RGB-D sensors. Top: Point cloud colored by scan id. Bottom: Point cloud with color data.

individual sensors are aligned such that there is a small overlap of the individual sensor's field of view. There is also an Intel Realsense T265 stereo camera mounted in the upper part of the backpack. This is used to additionally capture an initial trajectory using visual-inertial odometry (VIO), which is later used in the post-processing of the data.

On the right side of Fig. 2 you can see the back side of the backpack. To record the data from the four RGB-D cameras, four embedded computers are used: three Intel Atom based embedded PCs and a NVIDIA Jetson Nano. This is necessary due to the high USB-3 bandwith and CPU load necessary for capturing Microsoft Kinect 2 data. Each embedded computer is dedicated to processing the data of a single Kinect 2. The Jetson Nano board additionally captures the data of the Intel T265 tracking camera, which requires low computing resources since the processing is done directly on the vision processor of the T265 camera.

The embedded computers use the Robot Operating System (ROS) for sensor drivers and data recording. We employ NTP for time synchronization and accurate time stamping of the individual scans. Via network the point cloud data is available using ROS transports. This also enables a first live visualization during the mapping process. To do this, the user can connect to the system via a wifi access point and see a subsampled initial map on their device. This allows an initial assessment and overview of the data obtained while still recording.

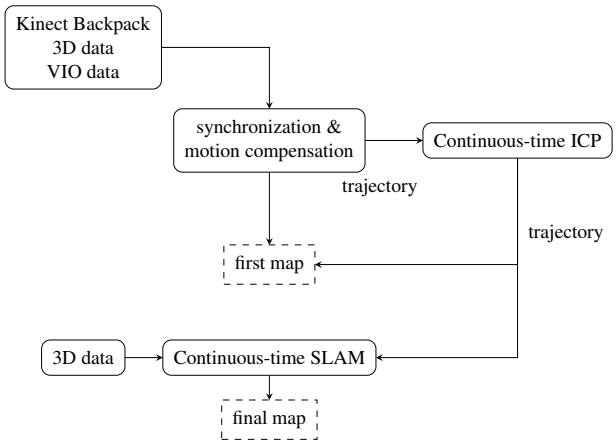

Figure 4. Data processing chain: An initial result created using visual-inertial odometry is further optimized with continuous-time ICP and continuous-time SLAM.

2.2 Calibration and Synchronization

We use the ROS driver of (Wiedemeyer, 2015) for capturing and color mapping of the Kinect 2 data. Calibration of the intrinsics and extrinsics of the individual RGB-D sensors is performed using stereo calibration based on a chessboard pattern. We calibrate the IR- and RGB-sensors of the individual Kinect 2 as well as perform a calibration between the RGB-D sensors and the T265 stereo camera to find the relative poses. This way we reference all sensors to a common coordinate system. For this work we do not re-calibrate the depth data of the Kinect 2 sensor and rely on the factory data. A compensation of systematic depth errors further optimizes the 3D point measurement quality as shown in the literature (Lachat et al., 2015, Lindner et al., 2010).

Since the four Kinect 2 sensors have little overlap we create a combined point cloud of all sensors based on time synchronization using the time stamps of the sensor data. We ensure that a combined point cloud has the data of exactly the four Kinect 2 and the time difference between the sensor images is small. However, the Kinect 2 cannot be triggered using a global trigger to ensure RGB and depth data of all sensors is captured at the exact same time. Therefore, we apply motion compensation using the odometry data of the T265 tracking camera in order to create consistent point clouds during fast movements of the backpack. The data is captured with 10 Hz. Each joint point cloud has depending on the environment roughly 500.000 to 1 million points.

2.3 Data Processing

The data processing pipeline, as seen in Fig. 4 is mainly split in two steps. The first step initializes the trajectory using the visual-inertial odometry data and applies continuous-time ICP to compensate drift. This creates a preliminary map. The second step the 3D data, together with the initial trajectory from the first processing step, are processed to the final map. We use 3DTK - The 3D Toolkit (Nüchter et al., 2022) for point cloud processing.

Let's have a closer look into these steps of processing. For the first processing step, the poses of the individual scans are initialized by interpolating the VIO data captured using the Intel T265 tracking camera and Intel's proprietary visual-inertial

odometry solution. Alternatively, an initial trajectory is created using the iterative closest point algorithm (ICP). For this a metascan created from a sliding window of registered preceding scans is used to provide more structure during ICP registration. The result is a first preliminary point cloud map and trajectory. Over time, larger parts of the map are merged into sub-maps and the trajectory is improved using a continuous-time ICP. Loops that have already been found are also closed.

Thus, the preliminary map displayed at runtime is also constantly improved. In second processing step, the recorded 3D data is optimized with a continuous-time SLAM in several iterations. For this purpose, the trajectory calculated at the first step by the continuous-time ICP is used as the initial trajectory.

2.3.1 Continuous-time ICP Since the remaining residual errors accumulate and the existing drift of visual-inertial odometry or metascan ICP registration is not completely eliminated, a registration with a continuous-time ICP method is performed in the next step. The basic idea is that the error of the trajectory in temporal proximity of a considered pose is negligible. The trajectory is then split into subsections and several successive 3D scans around a chosen reference scan are combined to form a submap. The partial maps are again registered against their predecessors. The change in pose of a reference scan is then distributed to the poses between two reference scans to maintain the continuity of the trajectory.

Since an ICP is prone to angular errors and these add up to significant drifts over longer distances, this step also attempts to find loops and close this. The corrections of a trajectory calculated in this way are then distributed among the poses of the individual scans of a submap. For small changes a linear distribution (translation) or SLERP (rotation) is sufficient. This step provides a good initial trajectory, which is used in the next processing step as a starting point for the continuous-time SLAM.

2.3.2 Continuous-time SLAM Given a sufficiently estimated trajectory, the entire point cloud can be improved by optimizing the entire trajectory. We use the approach from (Elseberg et al., 2013) that is based on the ICP concept known for rigid registration algorithms. The initial point cloud is a set of individual scans, each of which is assigned a time stamped pose during the trajectory estimation. We first split the trajectory into overlapping sections and match these using the automatic high-precision registration of terrestrial 3D scans, i.e., the graph-based SLAM approach presented in (Borrmann et al., 2008). The graph is estimated using a heuristic that measures the overlap of sections based on the number of closest point pairs. After applying globally consistent scan matching on the sections the actual continuous-time or semi-rigid matching, as described in (Elseberg et al., 2013), is applied, using the results of the rigid optimization as starting values to compute the numerical minimum of the underlying least square problem. The choice of the subdivision is important for the results. Local trajectory errors within a sub-scan cannot be directly improved. Here, we build junks of roughly 10 scans, which corresponds to $1\,\mathrm{s}$ of trajectory.

For long trajectories in unstructured environments this global approach is problematic. If the trajectory error is larger than the features in the scene, wrong point correspondences are likely to occur and to move the point cloud into local minima. Additionally, memory requirements and runtime increase. Thus, a sequential method is developed to minimize local errors before the global optimization.

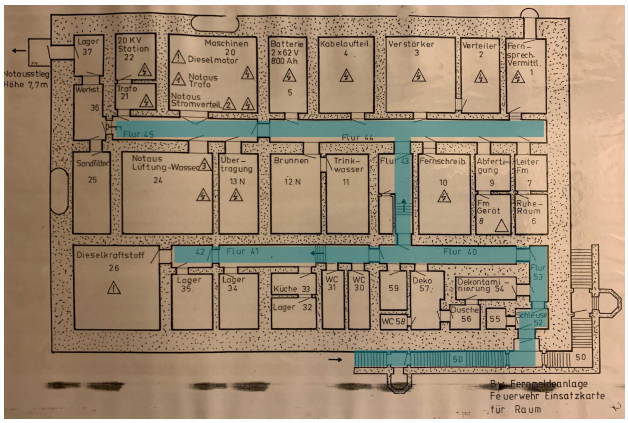

Figure 5. The floor plan of the bunker. The areas marked in blue are included in the dataset examined here.

3. EXPERIMENTAL RESULTS

3.1 Dataset

An old bunker facility was chosen as the test environment. This nuclear bomb-proof bunker was build on the Dillberg in Lengfurt in the 1960s through the German Federal Armed Forces. It is $50\,\mathrm{m}$ long and $30\,\mathrm{m}$ wide, the bunker is six meters deep in the ground and covered by $80\,\mathrm{cm}$ of soil. The building complex conceals 58 rooms, a power generator, tanks for 26000 liters of diesel, an air filtration system and a 112-meter-deep well with a diameter of three meters. The bunker could house up to 65 people for about four months. In the event of a nuclear attack, it was intended to provide protection. In the event of war, it was intended to serve as a bug-proof communications center.

As a reference, the entire plant was scanned with a Riegl VZ-400 laser scanner. The data set consists of 32 individual scans that were recorded statically with the help of a tripod. These were first roughly registered by hand and then refined with global ICP scan matching (Borrmann et al., 2008). The Riegl VZ-400 achieves an accuracy of $5\,\mathrm{mm}$ and a precision of $3\,\mathrm{mm}$. This provides a point cloud with very high accuracy and allows a direct comparison of the mobile recorded data with the statically generated data in order to validate the accuracy of the mobile backpack system.

The data set of the mobile system considered here covers a length of approx. $180\,\mathrm{m}$ and was recorded over a period of $260\,\mathrm{s}$. This results in an average movement speed of 0.7 meters per second. The staircase and the two main corridors, which are connected by an intermediate corridor, were recorded. This can be seen in blue marked in the Fig. 5. In order to keep the complexity in the first attempts within a manageable range, the adjacent rooms were not recorded in this data set.

3.2 Results

In the first step, we consider the improvement of the data according to the different trajectory optimization steps. After this we compare the optimized data with the reference scans of the bunker created using a Riegl VZ-400.

Fig. 6 shows the dataset after the different trajectory optimization steps. The trajectory is shown in red. The top row shows the data right after the visual-inertial odometry without any further correction. In this step, very large errors are visible. The

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-1-2022-121-2022 | © Author(s) 2022. CC BY 4.0 License.

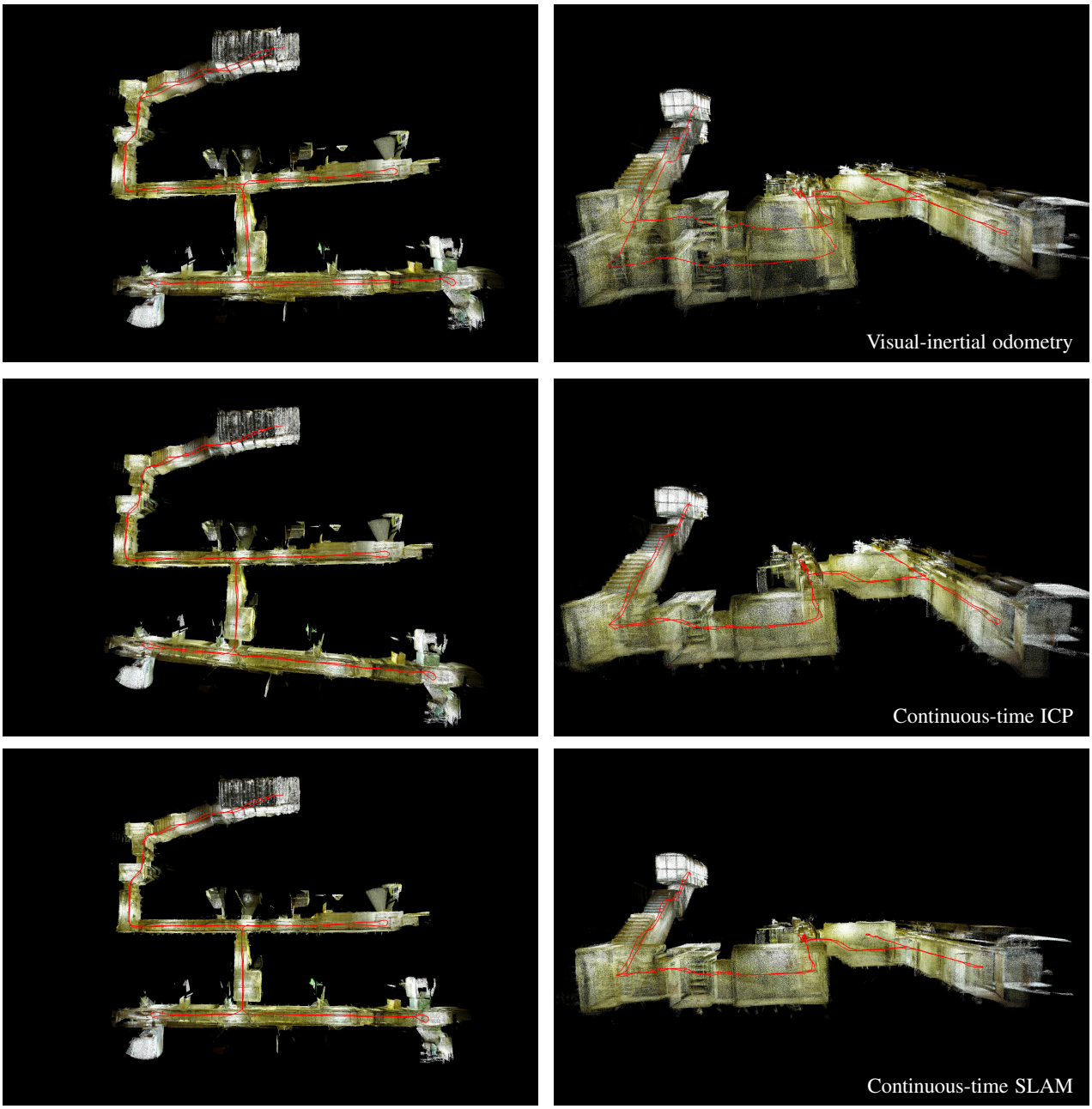

Figure 6. Point cloud result after the different trajectory optimization steps. The trajectory of the backpack is overlayed in red color. Top row: Visual-inertial odometry result. Middle row: Result after continuous-time ICP registration. Bottom row: Final result created using continuous-time SLAM.

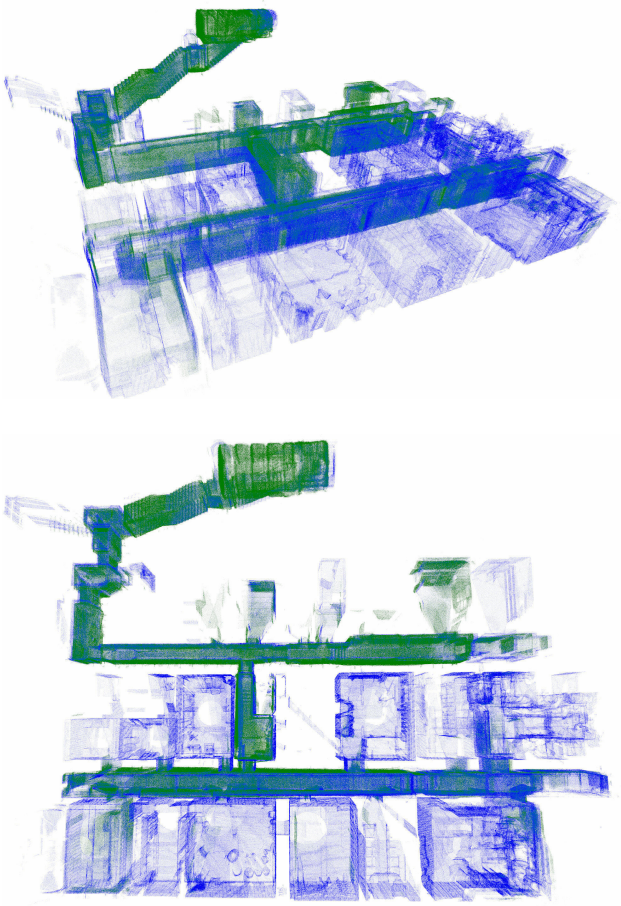

Figure 7. Comparison of the backpack point cloud with reference scans of the bunker created using a Riegl VZ-400 laser scanner. Co-registration of the mobile mapping point cloud (green) and the reference scans (blue).

walls of the corridors in the left figure do not lie on top of each other. In the right figure, it can also be seen that even the floor is drifting apart over time. The drift of the visual-inertial odometry is mainly in the height axis.

The result of the second optimization step is displayed in the middle row of Fig. 6. This is the result after continuous-time ICP registration. As is visible, this is able to pull the walls and floor on each other. The major errors from the first step have thus been corrected. This is also visible in the path of the trajectory in the right image. Since the sensor is a backpack, it is assumed that it was carried at approximately the same height both on the way there and on the way back. While the height of the two lines differed significantly in the first step, they are now almost on top of each other. The floor and ceiling now also create a uniform image and are visibly well placed on top of each other. In the right picture, we see that the walls are now also well aligned. In addition, in comparison to the right picture from the first row, the two corridors now no longer have a large curvature. However, the continuous-time ICP registration yields some residual angular errors. The two long corridors are no longer parallel to each other, but are visibly twisted.

The final optimization step is the continuous-time SLAM and the result is shown in the bottom row of Fig. 6. This method is able to correct the angular errors from the previous step. As you can see in the picture on the left, the two corridors are now parallel to each other. The angles to the connecting corridor are now also right-angled. In addition, as shown in the picture on the right, the angular errors of the floor have also been corrected. The floors of the lower corridors now also lie in one plane, without being twisted against each other as the result in the previous steps.

Between the first and the last step, a clear improvement can be seen. While the errors are immediately obvious in the first step, after the last optimization step no gross errors are visible. In the next section, these results are compared with the reference data.

Fig. 7 and Fig. 8 show the comparison of the mobile backpack point cloud and the reference point cloud of the bunker created using a Riegl VZ-400 laser scanner. Fig. 7 shows the co-registration between the backpack point cloud and the laser scans. The mobile backpack point cloud is shown in green and the reference scans are colored in blue. At first glance, the two data sets lie well on top of each other. The distance of the corridors and the angles have been corrected properly.

In order to view the deviation of the two data sets in more detail, Fig. 8 shows the registration error and error histogram. The error is calculated in centimeter and the point cloud is colored from blue to red according to the registration error. The errors over approx. $30\,\mathrm{cm}$ are in areas that are only contained in one data set. Therefore, all errors above $50\,\mathrm{cm}$ are assumed to be errors from non-overlapping parts and were removed from the calculation. If we now look at the remaining errors, we notice that most of the errors are in the range smaller than $15\,\mathrm{cm}$.

4. CONCLUSIONS

In this paper we demonstrated the feasibility of a low-cost backpack mapping system build from off-the-shelf RGB-D sensor for mobile indoor mapping. The scanning trajectory is bootstrapped using visual-inertial odometry and is further optimized by applying continuous registration to minimize errors.

This enables rapid 3D acquisition of indoor spaces. The errors accumulated over a $180\,\mathrm{m}$ trajectory are in the magnitude of $1\,\mathrm{dm}$ compared to laser scanning. The residual errors are mostly introduced by the lower accuracy of the point cloud created using RGB-D sensors. This could be further improved by applying geometric re-calibration of the time-of-flight measurements using a spline fitting approach as suggested, for example, by (Lindner et al., 2010).

ACKNOWLEDGEMENTS

The backpack was built in the project TASTSINN/VR, funded by the Federal Ministry of Education and Research due to an enactment of the German Bundestag under the grant 16SV8159. The authors thank the consortium partners Fraunhofer ISC and Awesome Technologies for the cooperation in the project.

A big thank you goes to Karl Dengel from Bauzentrum Kuhn, who allowed us to enter the bunker facility and record our data sets there.

REFERENCES

Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A., Hertzberg, J., 2008. Globally consistent 3D mapping with scan matching. *Robotics and Autonomous Systems*, 56(2), 130–142.

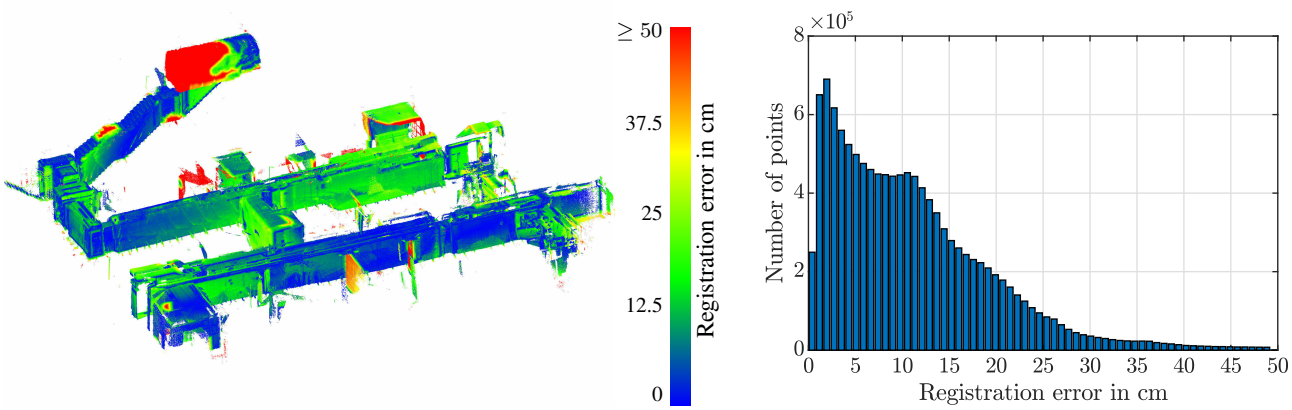

Figure 8. Comparison of the backpack point cloud with reference scans of the bunker created using a Riegl VZ-400 laser scanner. Left: Point cloud colored by the registration error. Right: Error histogram. Errors larger than 50 cm are removed since they are assumed to only occur in regions without any overlap between the two point clouds.

Bosse, M., Zlot, R., Flick, P., 2012. Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping. *IEEE Transactions on Robotics*, 28(5), 1104–1119.

Chen, C., Yang, B., Song, S., Tian, M., Li, J., Dai, W., Fang, L., 2018. Calibrate multiple consumer rgb-d cameras for low-cost and efficient 3d indoor mapping. *Remote Sensing*, 10(2), 328.

Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C., 2017. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4), 1.

Elseberg, J., Borrmann, D., Nüchter, A., 2013. Algorithmic solutions for computing precise maximum likelihood 3D point clouds from mobile laser scanning platforms. *Remote Sensing*, 5(11), 5871–5906.

Frederic Lardinois, TC, 2015. Google Unveils The Cartographer, Its Indoor Mapping Backpack. `https://techcrunch.com/2014/09/04/google-unveils-the-cartographer-its-indoor-mapping-backpack/`.

Hess, W., Kohler, D., Rapp, H., Andor, D., 2016. Real-time loop closure in 2d lidar slam. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 1271–1278.

Hsu, Y.-W., Huang, S.-S., Perng, J.-W., 2018. Application of multisensor fusion to develop a personal location and 3D mapping system. *Optik*, 172, 328–339.

Karam, S., Vosselman, G., Peter, M., Hosseinyalamdary, S., Lehtola, V., 2019. Design, calibration, and evaluation of a backpack indoor mobile mapping system. *Remote Sensing*, 11(8), 905.

Khoshelham, K., Elberink, S. O., 2012. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2), 1437–1454.

Lachat, E., Macher, H., Landes, T., Grussenmeyer, P., 2015. Assessment and calibration of a RGB-D camera (Kinect v2 Sensor) towards a potential use for close-range 3D modeling. *Remote Sensing*, 7(10), 13070–13097.

Lauterbach, H. A., Borrmann, D., Heß, R., Eck, D., Schilling, K., Nüchter, A., 2015. Evaluation of a backpack-mounted 3D mobile scanning system. *Remote Sensing*, 7(10), 13753–13781.

Leica Geosystems, 2021. Leica pegasus backpack. `https://leica-geosystems.com/de-de/products/mobile-sensor-platforms/capture-platforms/leica-pegasus-backpack`.

Liang, X., Kukko, A., Kaartinen, H., andY. Xiaowei, J. H., Jaakkola, A., Wang, Y., 2014. Possibilities of a Personal Laser Scanning System for Forest Mapping and Ecosystem Services. *Sensors*, 14(1), 1228–1248.

Lindner, M., Schiller, I., Kolb, A., Koch, R., 2010. Time-of-Flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding*, 114(12), 1318-1328. Special issue on Time-of-Flight Camera Based Computer Vision.

NavVis, 2021. NavVis M6 trolley. `https://www.navvis.com/m6`.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A., 2011. Kinectfusion: Real-time dense surface mapping and tracking. *2011 10th IEEE international symposium on mixed and augmented reality*, IEEE, 127–136.

Nocerino, E., Lago, F., Morabito, D., Remondino, F., Porzi, L., Poiesi, F., Chippendale, P., Locher, A., Havlena, M., Van Gool, L. et al., 2017. A smartphone-based 3D pipeline for the creative industry-the replicate EU project. *3D Virtual Reconstruction and Visualization of Complex Architectures*, 42, 535–541.

Nüchter, A. et al., 2022. 3DTK — The 3D Toolkit. `http://threedtk.de/`.

Wiedemeyer, T., 2015. IAI Kinect2. `https://github.com/code-iai/iai_kinect2`.