Preprint of the paper which is accepted for publication in the proceedings of 2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)

HectorGrapher: Continuous-time Lidar SLAM with Multi-resolution Signed Distance Function Registration for Challenging Terrain

Kevin Daun, Marius Schnaubelt, Stefan Kohlbrecher and Oskar von Stryk

***Abstract*— For deployment in previously unknown, unstructured, and GPS-denied environments, autonomous mobile rescue robots need to localize themselves in such environments and create a map of it using a simultaneous localization and mapping (SLAM) approach. Continuous-time SLAM approaches represent the pose as a time-continuous estimate that provides high accuracy and allows correcting for distortions induced by motion during the scan capture. To enable robust and accurate real-time SLAM in challenging terrain, we propose HectorGrapher which enables accurate localization by continuous-time pose estimation and robust scan registration based on multi-resolution signed distance functions. We evaluate the method in multiple publicly available real-world datasets, as well as a data set from the RoboCup 2021 Rescue League, where we applied the proposed method to win the Best-in-Class "Exploration and Mapping" Award.**

I. Introduction

To perform missions within unknown, degraded, and GPS-denied environments, autonomous mobile rescue robots need to localize themselves in such environments and create a map of it using a simultaneous localization and mapping (SLAM) approach. The capability to create accurate maps and precisely locate the robot's pose in the map are key prerequisites for many higher-level autonomous functions such as navigation or exploration.

During search and rescue missions, the motion characteristics can be highly challenging, e.g., traversing obstacles induces aggressive roll-pitch motions and poor odometry estimates as tracks/wheels slip. Environments are typically unstructured and might contain narrow indoor transits in addition to wide, open outdoor spaces with translucent vegetation. These characteristics make both state estimation and mapping highly challenging.

Continuous-time SLAM (CT-SLAM) approaches, such as LIO-SAM [1], the lidar odometry approach by Quenzel et al. [2] or Elastic LiDAR fusion [3] represent the estimate of the trajectory by a continuous function, defined by a discrete set of control points. In contrast to discrete approaches, continuous approaches natively are able to fuse high frequent data as the dimensionality of the state only depends on the number of control points and is independent of the frequency of the sensor data. Furthermore, the continuity of the pose estimation allows to correct for motion distortions during lidar scan acquisition [4] as well as rolling shutter effects for cameras [5]. Thereby, CT-SLAM approaches are well suited for the application in mobile rescue robots.

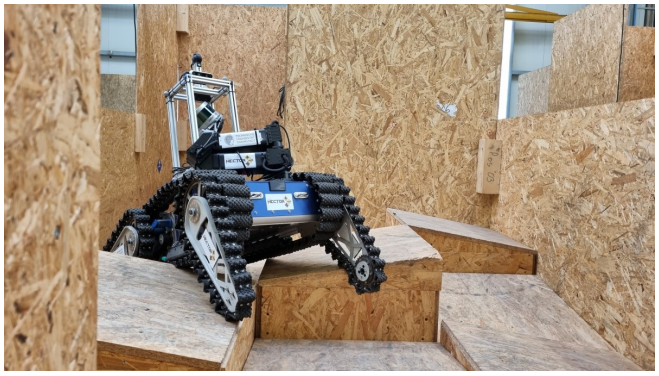


Fig. 1: To enable robust and accurate real-time SLAM in challenging terrain, such as the RoboCup Rescue League Elevated Ramps lane, we propose HectorGrapher which enables accurate localization by continuous-time pose estimation and robust scan registration based on multi-resolution signed distance functions.

We propose *HectorGrapher*, a robust continuous-time framework for mobile rescue robots. The framework is developed for the needs of robust and efficient 3D SLAM for the mobile ground robots used for urban search and rescue research at Team Hector[1] and the German Center for Rescue Robotics[2].

We fuse IMU and wheel odometry[3] as wheel-inertial odometry to reach a low latency prior estimate of the trajectory. Slippage, sensor noise and model errors induce drift which is further reduced by the lidar-inertial odometry which registers lidar data in a continuous time framework and gains robustness by performing scan-to-map registration in a multi-resolution signed distance function (SDF) map. To achieve global consistency, we maintain a global pose graph with efficient branch-and-bound-based loop-closure detection.

We base our work on Cartographer[6], an open-source SLAM system that implements occupancy grid-based scan-to-map matching and loop closure detection. The project contained an early version for batch optimization of range

All authors are with the Simulation, Systems Optimization and Robotics Group, Technical University of Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany {daun, schnaubelt, stryk}@sim.tu-darmstadt.de

Research presented in this paper has been supported in parts by the German Federal Ministry of Education and Research (BMBF) within the subproject "Autonomous Assistance Functions for Ground Robots" of the collaborative A-DRZ project (grant no. 13N14861) and by the LOEWE initiative (Hesse, Germany) within the emergenCITY center.

[1] https://www.teamhector.de/

[2] https://rettungsrobotik.de/en/

[3] For readability, we only refer to wheel odometry although it also covers odometry of tracked robots.

data, odometry and IMU that was discontinued[4], which we use as a basis of our implementation. This paper builds on the findings of our previous work on large-scale 2D SDF SLAM [7]. The main contributions of this paper are:

- Optimization-based odometry and inertial fusion
- Continuous-time, multi-resolution SDF-based lidar-inertial odometry
- Quantitative and qualitative evaluation with ground-truth annotated benchmark data for ground robots in challenging terrain.

Our implementation and benchmark data is publicly available[5].

II. Related Work

The SLAM problem covers two aspects – estimating the robot trajectory and mapping the environment. This requires suitable techniques to model the trajectory and the environment, which influence the choice of methods to optimize the trajectory and the map.

Continuous-time SLAM (CT-SLAM) approaches represent the estimate of the trajectory by a continuous function, defined by a discrete set of control points. In contrast to discrete approaches, continuous approaches easily enable fusing high-frequency data as the dimensionality of the state only depends on the number of control points and is independent of the frequency of the sensor data. Early work towards CT-SLAM approaches was introduced by Bosse and Zlot [4], who propose a linear interpolation-based registration scheme to estimate the continuous trajectory of spinning 2D lidars, by matching the geometric structure of local point clusters. Thereby, they are able to compensate distortions in the scan cloud induced during the scan acquisition. Following up, LOAM [8] proposes a two-fold approach, one part continuously performs a low accuracy registration to achieve a high-frequency velocity update while another part performs a less frequent higher accuracy registration to correct for drift and update the map. Point clouds are matched by extracting edge and plane features to perform efficient scan registration. LeGO-LOAM [9] extends the LOAM approach by separating the ground for scan matching and gain further efficiency by splitting the optimization in solving different components of the six degree-of-freedom transformation separately. LIO-SAM [1] applies the LOAM registration scheme in a smoothing and mapping context, which fuses preintegrated IMU measurements [10] and lidar registration jointly in a pose graph framework.

Instead of linear interpolation various works leverage more complex trajectory representations such as B-Splines [2, 5, 11]. Nüchter et al. [12] apply a global continuous time formulation to improve the registration results of Cartographer [6].

Aiming to provide robust localization in uncertain settings, LOCUS [13] proposes to leverage a multi-stage scan matching scheme fusing multi-modal odometry sources to achieve robust SLAM in cave exploration scenarios.

[4]https://github.com/cartographer-project/cartographer/pull/368

[5]https://github.com/tu-darmstadt-ros-pkg/hectorgrapher

Representing the environment using TSDFs, a volumetric environment representation storing the truncated signed distance to the next surface in each cell was introduced in the seminal work of Curless and Levoy [14]. TSDF gained further attention by the introduction of KinectFusion [15] a method for live 3D tracking and mapping of room-scale environments. Performing a point-to-plane iterative closest point algorithm (ICP) optimized for efficient usage of GPU parallelization they were able to generate high-resolution 3D maps in real-time. Bylow et al. [16] propose an alternative optimization scheme by directly minimize the depth error of the RGB-D image on the TSDF. Thereby, improving the accuracy of the tracked pose. Complementing the previous works, Slavcheva et al. [17] propose directly representing the RGB-D Image as TSDF and performing direct TSDF to TSDF registration. This yields further improvements in the size of the convergence basin, rotational motion estimation and reconstruction quality.

LiDAR systems differ from RGB-D cameras, as range and field of view are significantly are larger, leading to more measurements with steep incident angles which induce projection errors in TSDFs. To correct these errors Fossel et al. [18] compute regression lines in the scan. Daun et al. [7] estimate scan normals to approximate the Euclidean distance. Conventional TSDF approaches suffer from overwriting artifacts when objects are thinner than the truncation distance. Splietker et al. [19] overcome this issue by storing the SDF value for multiple surface orientations separately.

An established technique to improve scalability and robustness of grid-based methods are multi-resolution approaches. Hector SLAM [20] performs robust 2D scan-to-map matching against a pyramid of occupancy grids, starting at the coarsest resolution, forwarding each result as initialization for the next finer resolution. Quenzel et al. [2] apply an adaptive resolution selection scheme to perform efficient surfel-based scan registration. Chen et al. [21] leverage a hierarchical data structure reconstruction of large-scale scenes with fine geometric details from depth cameras on GPUs. Vespa et al. [22] adaptively choose the octree-resolution based on depth image resolution and the distance to the object for SLAM in room-scale environments and demonstrate up to six-fold execution speed-ups to single resolution grids.

III. Method

Our pipeline for solving the SLAM problem consists of three main components: wheel-inertial odometry, lidar-inertial odometry and the pose-graph back-end with loop closure detection. In the wheel-inertial odometry, we fuse wheel odometry and IMU observations to gain a low latency and high frequency pose estimate. The result is forwarded to lidar-inertial odometry, where lidar pointclouds are registered in multi-resolution TSDF submaps to achieve an accurate and robust lidar-inertial odometry. Lidar-inertial odometry still induces small errors, leading to drift in the pose estimate over time. To maintain global consistency of the map, the poses from the lidar-inertial odometry are stored in pose-

graph, where we perform loop-closure detection and global optimization. Optimizing the global pose-graph yields a globally consistent map and pose estimate.

A. Wheel-Inertial Odometry

Track and wheel encoders enable a high-frequency, low-latency estimate of the current robot motion state. Due to slippage and model errors, this estimate is only a rough approximation. Such errors are particularly strong for rotational motions in slippage-based drive kinematics such as skid-steer-kinematics. Inertial sensors provide high-frequency estimates of the linear acceleration and the rotational velocity. Estimating the position requires a double integration of the acceleration, which strongly amplifies even small errors and induces unbound drift after short periods of time. In contrast, integration of the angular components only induces small drift.

To reach an accurate motion estimate at a high frequency we fuse inertial and wheel odometry measurements in an optimization problem which can be stated as a pose-graph, as shown in Fig. 3. Given the last known state $x_i = \{p_i, v_i, R_i, b_i\}$, with the position vector p_i, linear velocity vector v_i, orientation matrix R_i and the imu biases b_i, and the unknown state x_j we apply the IMU preintegration method introduced in [10] to estimate the changes in state Δv_{ij}^{pre}, Δp_{ij}^{pre} and ΔR_{ij}^{pre} to define the residuals:

$$r_{ij}^{v} = R_i^T(v_j - v_i - g\Delta t_{ij}) - \Delta v_{ij}^{pre} \tag{1}$$

$$r_{ij}^{p} = R_i^T(p_j - p_i - v_i\Delta t_{ij} - \frac{1}{2}g\Delta t_{ij}^2) - \Delta p_{ij}^{pre} \tag{2}$$

$$r_{ij}^{R} = \Delta R_{ji}^{pre} R_i^T R_j \tag{3}$$

$$r_{ij}^{b} = b_j - b_i \tag{4}$$

Additionally, we directly add the estimated linear velocity from the track/wheel odometry v_j^{enc} as unary constraints

$$r_j^{v} = v_j^{enc} - v_j \tag{5}$$

The residuals of Eq. (1) - Eq. (5) are added to a non-linear least squares problem which we solve using the Levenberg-Marquardt implementation in GTSAM[6]. Furthermore, we assume to be in an steady state when both, IMU and odometry indicate no motion, which improves the estimate of the biases.

B. Lidar-Inertial Odometry

1) Notation: We model the robot state as a time continuous trajectory, similar to the formulation in [3]. The trajectory is represented by a linear Lie-group valued spline, which is defined by a set of timestamped control points C. Each control point $c_i = [\boldsymbol{T}_i, \tau_i]$ is given by a rigid transformation $\boldsymbol{T}_i \in SE(3) = [\boldsymbol{R}|\boldsymbol{p}]$ at time τ_i, which transforms from the robot body frame into the map frame. For simplicity, we assume that the IMU frame, robot body frame and lidar frame coincide. Poses at time τ between two consecutive control points c_i and c_j with $\tau_i < \tau < \tau_j$ are evaluated by linear interpolation

$$\alpha = (\tau - \tau_i)/(\tau_j - \tau_i) \tag{6}$$

$$\overline{\boldsymbol{p}} = (1-\alpha)p_i + \alpha p_j \tag{7}$$

$$\overline{\boldsymbol{R}} = slerp(\alpha, R_i, R_j) \tag{8}$$

with the spherical linear interpolation operator $slerp$. We denote the interpolated transform as $\overline{\boldsymbol{T}}(\tau) = [\overline{\boldsymbol{R}}|\overline{\boldsymbol{p}}]$.

We denote the observations of single structured lidar scan $\boldsymbol{H} = \{\boldsymbol{h}_i\}_{i=1,\dots,N}$ consisting of N timestamped range observations $\boldsymbol{h} = (h_x, h_y, h_z, h_t)$ in the scan coordinate frame with the sensor pose as the origin of the scan frame.

2) Environment representation: We model the map as a multi-resolution TSDF. In the following, we first outline the the principle of a single-resolution TSDF and then extend it to the multi-resolution formulation in the next section.
The Signed Distance Function $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}$ maps from for each position in space to the scalar, signed distance to the nearest surface . Φ is positive outside of objects and negative inside of objects. Therefore, object surfaces are encoded as the zero isocontour ($\Phi = 0$). As an exhaustive evaluating Φ quickly becomes computational intractable, TSDFs evaluate Φ only close to the surface. Each evaluation of a position further away from the closest surface than the truncation distance τ is truncated

$$\Phi_\tau(\boldsymbol{x}) = \begin{cases} -\tau, & \text{if } \Phi(\boldsymbol{x}) < -\tau \\ \Phi(\boldsymbol{x}), & \text{if } |\Phi(\boldsymbol{x})| \leq \tau \\ \tau, & \text{if } \Phi(\boldsymbol{x}) > \tau. \end{cases} \tag{9}$$

To represent a 3D scene as a TSDF, space is discretized in a regular grid. Each grid cell c contains the current estimate of the TSDF $M_\Phi(c)$, and a scalar weight $M_w(c)$ indicating the confidence in the TSDF value.

As estimating the Euclidean distance for every cell is computationally intense, most approaches approximate it. Most common are projective approaches [15] from the context of depth image integration. They update the the cells along a ray from the origin towards the observed scan point. The less orthogonal the viewing angle becomes, the more inaccurate are the update distances. Especially for sparse, long-range lidar sensors, this significantly reduces the quality of the resulting map. To compensate for the distance biases induced by using the projective distance, we showed in [7] that the use of scan normals to approximate the Euclidean distance improves the accuracy of the resulting map for long range sensors.

We update cells around a range observation $\boldsymbol{h}_i$ which are on the ray in the direction of the scan normal $\boldsymbol{n}_i$

$$\boldsymbol{v}(u) = \boldsymbol{h}_i - u\boldsymbol{n_i}. \tag{10}$$

with the interpolation parameter u.

All cells c along the ray $\boldsymbol{v}(u)$ within the truncation distance ($u \in [-\tau, \tau]$) are updated by taking a weighted

[6]https://gtsam.org/

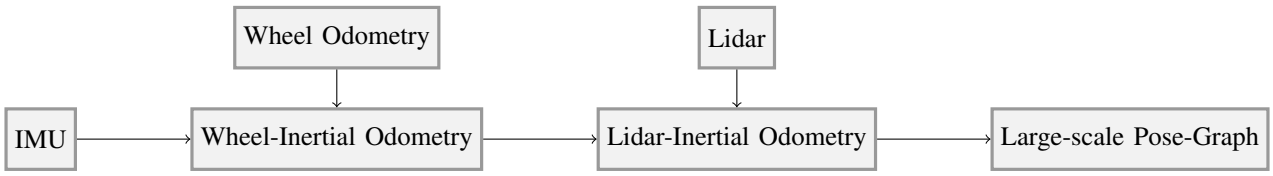


Fig. 2: SLAM System Overview

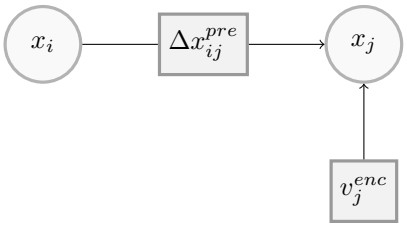


Fig. 3: We fuse preintegrated IMU measurementents Δx_{ij}^{pre} and linear velocity estimates v_j^{enc} from track or wheel encoders in a small factor graph to compute a high-frequency, low-latency estimate of the robot motion.

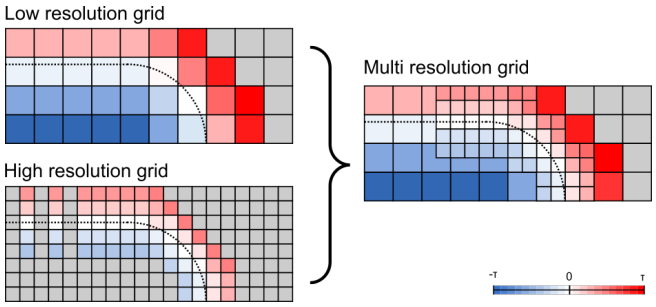


Fig. 4: We maintain two grids a high-resolution grid and a low-resolution grid with the same relative truncation distance. During scan matching, for each point we evaluate the SDF at the highest resolution providing a valid interpolation.

moving average of the distance measurements

$$M_\Phi(c) := \frac{M_W(c)M_\Phi(c) + \omega(u)\Phi_\tau(u)}{W(c) + \omega(u)} \tag{11}$$

$$M_W(c) := M_W(c) + \omega(u) \tag{12}$$

with the update weighting function $\omega(u)$. We apply constant weighting function $\omega(u) = const$.

3) Optimization: We register the lidar scans and wheel-inertial odometry in a joint optimization. For registering scans in TSDF, two paradigms are prevalent: ICP based registration [15] and direct optimization of the pose on the TSDF [16]. As [16] indicates benefits in accuracy and efficiency for the direct optimization, we use the direct formulation as a basis and extend it by the mult-resolution registration and wheel-inertial odometry terms.

For the original direct optimization approach the convergence basin is limited by the truncation distance, as outside the truncation distance the gradient is constant. Furthermore, it requires an interpolation of the discrete TSDF grid to compute gradients, which might not be possible if the scan data is sparse. Finally, expressiveness and thereby the accuracy depends on the grid resolution.

To leverage the precision of the high grid resolution and the robustness of a lower resolution we apply a multi-resolution scan matching approach (see Fig. 4). We maintain two TSDF grids with different resolutions, a high resolution grid M_Φ^{high} and a low resolution grid M_Φ^{low}, but both with the same relative truncation distance with respect to the resolution. For scan matching we evaluate SDF at the highest resolution providing a valid gradient. This leads to coarse and robust gradients in large area around the surface and precise gradient close to the surface.

Following the derivations in [16], we phrase the registration as a Nonlinear Least Squares Problem

$$\min_{\boldsymbol{C}} \sum_{i=1}^{N} \left(\Phi_I^{MR}(\overline{\boldsymbol{T}}(\tau_i)\boldsymbol{h}_i)\right)^2 + \sum_{i=1}^{|C|-1} (r(T_{i+1}^{-1}T_i\Delta T_{i;i+1}^{odom}))^2, \tag{13}$$

$$r(\boldsymbol{T}_i) := [\boldsymbol{p}|RPY(\boldsymbol{R})] \tag{14}$$

Φ_I^{MR} is the tri-linear interpolation of the highest resolution grid available providing a valid interpolation. We consider an interpolation to be valid, when none of the eight neighboring cells is uninitialized. $RPY(\boldsymbol{R})$ is the extraction of the Cardanian angles from the rotation matrix $\boldsymbol{R}$. We solve the optimization with the Levenberg-Marquardt method using the Ceres Solver [23] and compute gradients with Automatic Differentiation.

C. Loop Closure and pose-graph

Scan matching induces small registration errors, resulting in an accumulation of errors over large distances. Thereby, it yields global inconsistencies in the pose estimate and the map.

To correct these errors, Cartographer generates many small, locally consistent submaps connected in a pose-graph. Optimizing the pose-graph for its constraints yields a globally consistent map if sufficient constraints have been found. The algorithm uses an efficient branch-and-bound based approach [6] to compute loop closure constraints for occupancy grids. By adapting the scan matching problem and bounding function for the method can be generalized for SDFs as shown in [7].

IV. Evaluation

In this section, we give an evaluation of the performance in various settings, from small isolated tests up to a large-scaled mixed indoor-outdoor scenario. We perform the evaluation

(a) The Asterix robot. (b) The EC scout robot.

Fig. 5: The rescue robots used in the evaluation are equipped with lidar, IMU, onboard compute and various use-case specific sensors.

with two different ground robots, each with a different sensor configuration (see Figure 5).
We demonstrate the capability to accurately estimate the robot motion while traversing complex obstacles and NIST test methods for mobility and maneuvering [24] and quantify the accuracy by comparing against the reference of a motion capture system. Furthermore, we demonstrate the capability to map a larger complex environment with various obstacles on the ground with a data-set from RoboCup Rescue League 2021. Finally, we demonstrate the capabilities to map a large-scale mixed indoor-outdoor scenario with multiple loops.

A. System

In the following, we shortly present the robotic systems used to evaluate the SLAM approach. The tracked robot Asterix shown in Figure 5a is equipped with a continuously rotating Velodyne VLP16 lidar, a LORD MicroStrain 3DM-GX3-45 IMU and performs the onboard computing on a Intel NUC with an i7-10510U hexacore processor. The lidar is mounted on a pitched rotating mount to achieve comprehensive geometry data acquisition coverage, increasing the effective vertical FOV from 30° (in case of fixed mounting) to 120°. For high mobility in challenging terrain, the robot is equipped with adjustable flippers, see [25] for more details. EC Scout, shown in Figure 5b, is a 4-wheel drive platform targeted at urban environments. It carries an autonomy module containing an Ouster OS0-128 lidar and a Xsens MTi-G-710 IMU. For onboard-processing of the data, an AMD Ryzen 9 5950X CPU with 16 cores is used.

B. Terrain Sequences

To evaluate the performance of our approach in challenging terrains, we captured four sequences with challenging terrain, each tracked with with a high-performance Qualisys optical motion capture system. The four sequences (see Figure 6) contain 1) double pitch ramps, which induce a fast pitch motion when traversing 2) a loose woodpile which slips when traversed 3) the RoboCup Rescue League "Maneuvering 3 - Traverse" lane which contains a 2.4m long 30° incline and 4) the the RoboCup Rescue League "Mobility 4 - Elevated Ramps" lane containing a diagonal hill terrain consisting of $60\,\mathrm{cm}$ ramps with sloped tops. The sequences are between $59\,\mathrm{s}$ and $149\,\mathrm{s}$ in duration.

We compare the accuracy of HectorGrapher and Cartographer[6] with reference to the measurements from a visual motion capture system. Other state-of-the-art systems such as LIO-SAM[1] are not included as, to our best knowledge, they do not support the spinning lidar configuration. We use the odometry benchmark measure suggested in [26], splitting the motion capture trajectory in $0.5\,\mathrm{s}$ sequences and comparing the errors for the translation and rotation component of each sequence. Both, HectorGrapher and Cartographer are able to generate qualitatively comparable maps of the environment. The error metrics are shown in Figure 7.

For the rotation component both methods achieve comparable results, with outliers in the same value region. HectorGrapher performs slightly worse for small errors. In contrast, for the translation significant differences are notable. The median error of Cartographer is more than two times the median error of Hectorgrapher in the woodpile and elevated ramps scenarios, and even in the continuous ramps and traverse scenario $40\,\%$- $60\,\%$ larger. Notable are also the outliers with high errors with $12\,\mathrm{cm}$-$14\,\mathrm{cm}$ in three scenarios, whereas the errors for HectorGrapher are no larger than $6\,\mathrm{cm}$. The improved localization accuracy can also be observed in the scan registration quality. Figure 8 shows the registered pointclouds for the "Continuous ramps" scenario. While the Cartographer result shows shift artifacts at the pallet stack and the wall, such artifacts are less notable in the HectorGrapher result.

Runtimes for HectorGrapher were 4-5.5 times higher than for Cartographer, which seems reasonable as the optimization problem becomes significantly larger and the update of the TSDF require more computation than the update of the occupancy grid in Cartographer.

C. RoboCup Rescue League 2021

We applied the proposed method for the RoboCup Rescue League 2021, winning the best-in-class "Exploration and Mapping" award. Each team had to setup a scenario following the same rules, following the NIST guidelines for evaluation of rescue robots[7], distributing 10 barrels at two different heights as visual fiducials. The fiducials appear as circles in the 2D projection of the map and are utilized to measure accuracy and completeness. To make the terrain challenging, every $4.8\,\mathrm{m}$ had to contain a small obstacle (see Figure 9a) such as a wooden bar or a ramp. As part of the scenario we traversed the RoboCup German Open - "EXP 1 Map on Continuous Ramps" Arena (see Figure 9b) which is a narrow, $1.2\,\mathrm{m}$ wide corridor in wave form in a $7.2\,\mathrm{m}\times2.4\,\mathrm{m}$ area continuously paved with ramps. The ramps induce fast roll-pitch motions. In combination with the narrow environment, both tasks - navigation and mapping - are challenging.

The data set covers $622\,\mathrm{s}$ of data. As the data set is rather small in scale and does not cover large loops we only perform

[7]https://rrl.robocup.org/forms-guides-labels/

(a) Continuous ramps (b) Woodpile (c) RoboCup Rescue League: Maneuvering 3 - Traverse (d) RoboCup Rescue League: Mobility 4 - Elevated Ramps

Fig. 6: Overview of the evaluation scenarios with ground truth data.

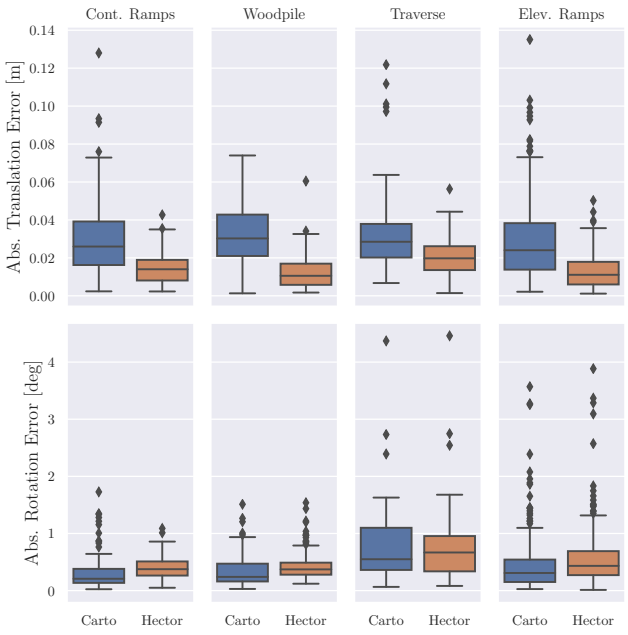


Fig. 7: Quantitative Error Comparison of Cartographer (Carto) and HectorGrapher (Hector) in the four evaluation scenarios.

the lidar odometry part and do not need to check for loop closures.

The resulting map, trajectory and pointcloud are shown in Figure 10. The map overall shows a high consistency, even in the narrow parts such as the EXP1 arena. The fiducials are clearly visible in the 2D projection and demonstrate the accuracy and coverage. On a desktop computer with an AMD Ryzen 9 3900X processor, computations took 193.4 s wall time and 191.9 s CPU time on a single core with a peak memory usage of 321.39 MB yielding a real-time factor of 3.24. The low computational cost makes the approach well suited for localization and mapping on mobile robots with limited hardware.

D. Scout DRZ Loop

To evaluate large-scale mapping capabilities we captured a mixed indoor-outdoor dataset with three loops at the DRZ Living Lab in Dortmund with the EC Scout platform. In contrast to the previous evaluations, this dataset also covers three loops and outdoor terrain, including an unpaved path through a scrapyard and an unpaved trail through a small forest. The unstructuredness and translucency of these environments makes scan registration challenging. The dataset covers 704 s of data and a distance of approx. 920 m.

As the data set is large in scale, we run the full SLAM pipeline with lidar odometry, loop closure detection and pose-graph optimization. As the EC Scout is equipped with an Ouster OS-0 128 scanner capturing data at more than 10x the rate of the Velodyne VLP-16, we downsample the pointcloud and only insert 10 % of the points in the TSDF to keep computations manageable.

The resulting map, trajectory and pointcloud are shown in Figure 11. The map overall shows a high consistency, with all three loops closed accurately. Geometry inside the building and outdoors on the scrapyard are sharp and the forest track is mapped consistently. Minor registration artifacts are notable in the lower left side of the DRZ Living Lab building indicating indicating a small angular offset in the submap registration.

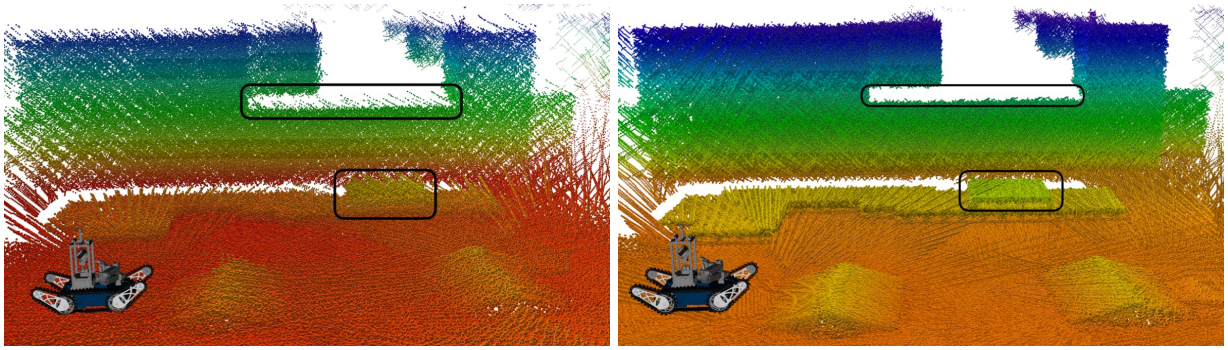
While setting up the configuration for the data set we noted the tendency of the lidar odometry to erroneous rotational motions in the forest part leading to a warping of the map. This issue could be resolved by increasing the weight of the wheel-inertial odometry rotation component in lidar-odometry optimization.

We performed the computations with an AMD Ryzen 7 3800X processor, taking 648.7 s wall time and 1832.6 s CPU time. In comparison the the RoboCup dataset, the increased pointcloud data lead to increased time of the map update, while the loop closure detection is executed in seperate threads and therby mainly increased the CPU time but not the wall time. The peak memory usage was 2.03 GB and the real-time factor 1.09. The approach is thus capable to operate in real-time with current hardware.

V. Conclusion and Future Work

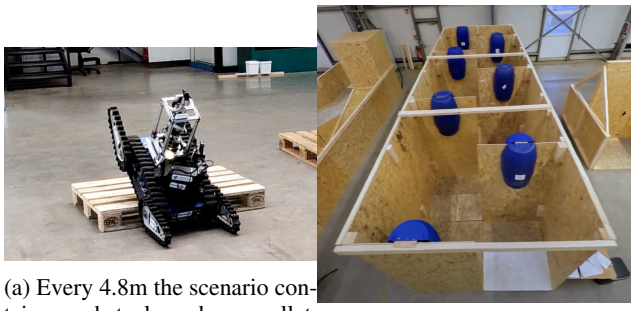
In this paper, we presented and evaluated a full 3D SLAM framework based on accurate localization by continuous-time pose estimation and robust scan registration based on multi-resolution signed distance functions. Our evaluation showed that the proposed approach achieves high accuracy in challenging USAR and large scale mixed indoor-outdoor environments with real-time performance. The proposed method has been published as open source.

For new lidar sensors, such as the Ouster OS-0 128, the map update is a performance bottleneck. Therefore, paral-



(a) Cartographer (b) HectorGrapher

Fig. 8: Comparison of the registered pointclouds (colored by height) in the "Continuous ramps" scenario. The Cartographer result shows shift artifacts at the pallet stack and the wall (marked by the black boxes), which are less notable in the HectorGrapher result.



(a) Every 4.8m the scenario contains an obstacle such as a pallet, ramp or wooden bar.

(b) Aerial Photo of the EXP1 Arena with barrel fiducials. Image: Hartmut Surmann

Fig. 9: The RoboCup Rescue League 2021 scenario contains multiple small obstacles and the EXP1 exploration arena.

lelization schemes such as GPU-utilization appear promising. Currently, there is no feedback from laser odometry and pose-graph optimization to the wheel-inertial component. Aiming for a tighter coupling, similar to IMU integration in LIO-SAM[1], could further improve results as e.g. biases can be better compensated.

References

[1] T. Shan et al. "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 5135–5142.

[2] J. Quenzel et al. "Real-time multi-adaptive-resolution-surfel 6D LiDAR odometry using continuous-time trajectory optimization". In: *arXiv preprint arXiv:2105.02010* (2021).

[3] C. Park et al. "Elastic lidar fusion: Dense map-centric continuous-time slam". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1206–1213.

[4] M. Bosse et al. "Continuous 3D scan-matching with a spinning 2D laser". In: *2009 IEEE International Conference on Robotics and Automation*. IEEE. 2009, pp. 4312–4319.

[5] S. Lovegrove et al. "Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras." In: *BMVC*. Vol. 2. 5. 2013, p. 8.

[6] W. Hess et al. "Real-Time Loop Closure in 2D LIDAR SLAM". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 1271–1278.

[7] K. Daun et al. "Large scale 2d laser slam using truncated signed distance functions". In: *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE. 2019, pp. 222–228.

[8] J. Zhang et al. "LOAM : Lidar Odometry and Mapping in real-time". In: *Robotics: Science and Systems Conference (RSS)* (Jan. 2014), pp. 109–111.

[9] T. Shan et al. "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 4758–4765.

[10] C. Forster et al. "On-Manifold Preintegration for Real-Time Visual–Inertial Odometry". In: *IEEE Transactions on Robotics* 33.1 (2016), pp. 1–21.

[11] D. Droeschel et al. "Efficient continuous-time SLAM for 3D lidar-based online mapping". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 5000–5007.

[12] A. Nüchter et al. "Improving Google's Cartographer 3D mapping by continuous-time slam". In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 42 (2017), p. 543.

[13] M. Palieri et al. "LOCUS: A Multi-Sensor Lidar-Centric Solution for High-Precision Odometry and 3D Mapping in Real-Time". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 421–428.

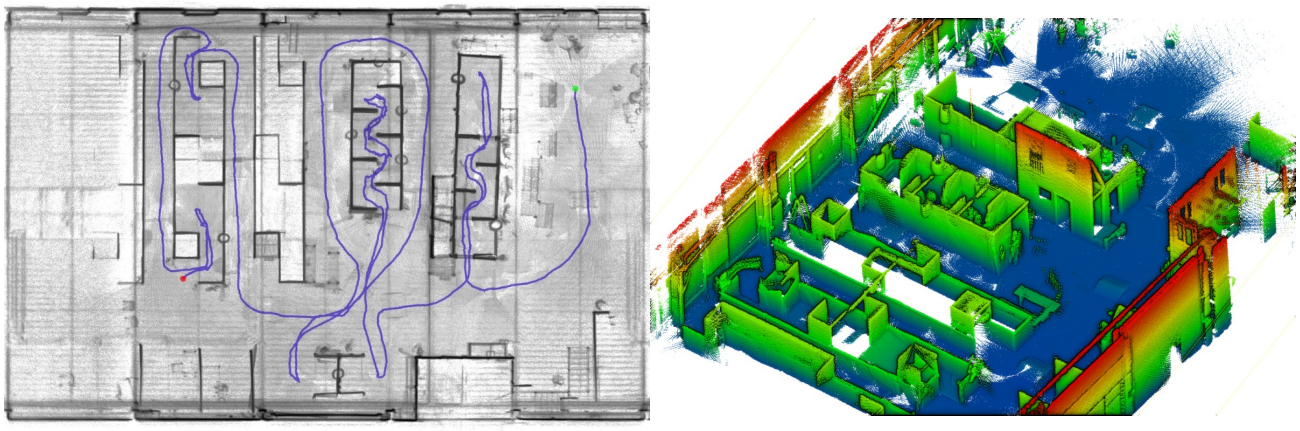
[14] B. Curless et al. "A volumetric method for building complex models from range images". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM. 1996, pp. 303–312.

[15] R. A. Newcombe et al. "KinectFusion: Real-time dense surface mapping and tracking". In: *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE. 2011, pp. 127–136.

[16] E. Bylow et al. "Real-time camera tracking and 3D reconstruction using signed distance functions." In: *Robotics: Science and Systems*. Vol. 2. 2013, p. 2.

[17] M. Slavcheva et al. "Sdf-2-sdf registration for real-time 3d reconstruction from RGB-D data". In: *International Journal of Computer Vision* (2018), pp. 1–22.

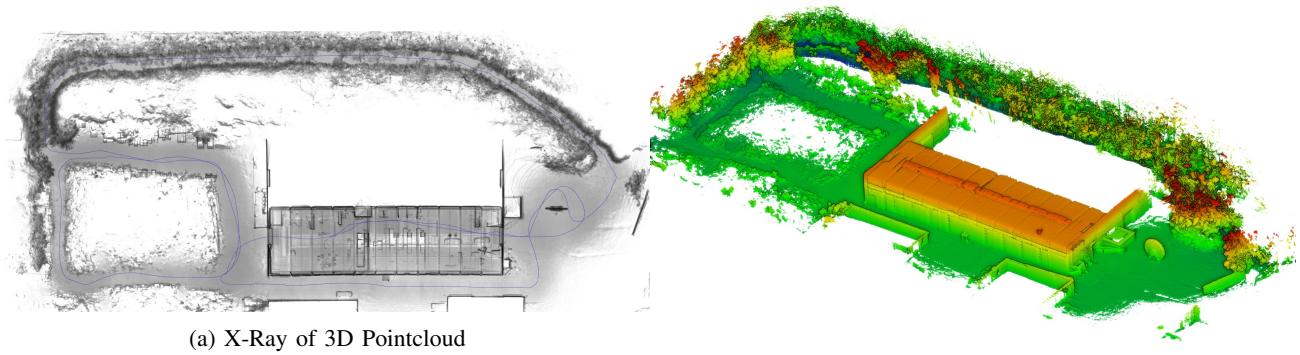
[18] J.-D. Fossel et al. "2D-SDF-SLAM: A signed distance function based SLAM frontend for laser scanners". In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE. 2015, pp. 1949–1955.



(a) X-Ray of 3D Pointcloud

(b) 3D Pointcloud - Roof removed for better visibility

Fig. 10: Map, trajectory and pointcloud for the RoboCup 2021 scenario.



(a) X-Ray of 3D Pointcloud

(b) 3D Pointcloud

Fig. 11: Map, trajectory and pointcloud for the DRZ Living Lab Loops scenario.

[19] M. Splietker et al. “Directional TSDF: Modeling Surface Orientation for Coherent Meshes”. In: Nov. 2019.

[20] S. Kohlbrecher et al. “A flexible and scalable slam system with full 3d motion estimation”. In: *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*. IEEE. 2011, pp. 155–160.

[21] J. Chen et al. “Scalable real-time volumetric surface reconstruction”. In: *ACM Transactions on Graphics (ToG)* 32.4 (2013), pp. 1–16.

[22] E. Vespa et al. “Adaptive-resolution octree-based volumetric SLAM”. In: *2019 International Conference on 3D Vision (3DV)*. IEEE. 2019, pp. 654–662.

[23] S. Agarwal et al. *Ceres Solver*.

[24] A. Jacoff et al. “Using competitions to advance the development of standard test methods for response robots”. In: *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*. ACM. 2012, pp. 182–189.

[25] M. Schnaubelt et al. “Entwicklung eines autonomiefokussierten hochmobilen Bodenrobotersystems für den Katastrophenschutz”. In: *Digital-Fachtagung VDI-MECHATRONIK 2021*. Ed. by T. Bertram et al. Universitäts- und Landesbibliothek Darmstadt, Mar. 2021, pp. 20–25.

[26] R. Kümmerle et al. “On measuring the accuracy of SLAM algorithms”. In: *Autonomous Robots* 27 (2009), pp. 387–407.