

Efficient Continuous-time SLAM for 3D Lidar-based Online Mapping

David Droschel and Sven Behnke

Abstract—Modern 3D laser-range scanners have a high data rate, making online simultaneous localization and mapping (SLAM) computationally challenging. Recursive state estimation techniques are efficient but commit to a state estimate immediately after a new scan is made, which may lead to misalignments of measurements. We present a 3D SLAM approach that allows for refining alignments during online mapping. Our method is based on efficient local mapping and a hierarchical optimization back-end. Measurements of a 3D laser scanner are aggregated in local multiresolution maps by means of surfel-based registration. The local maps are used in a multi-level graph for allocentric mapping and localization. In order to incorporate corrections when refining the alignment, the individual 3D scans in the local map are modeled as a sub-graph and graph optimization is performed to account for drift and misalignments in the local maps. Furthermore, in each sub-graph, a continuous-time representation of the sensor trajectory allows to correct measurements between scan poses. We evaluate our approach in multiple experiments by showing qualitative results. Furthermore, we quantify the map quality by an entropy-based measure.

I. INTRODUCTION

Laser-based mapping and localization has been widely studied in the robotics community and applied to many robotic platforms [1], [2], [3], [4]. The variety of approaches that exists either focus on efficiency, for example when used for autonomous navigation, or on accuracy when building high-fidelity maps offline. Often, limited resources—such as computing power on a micro aerial vehicle—necessitate a trade-off between the two. A popular approach to tackle this trade-off is to leverage other sensor modalities to simplify the problem. For example, visual odometry from cameras and inertial measurement units (IMU) are used, to estimate the motion of the laser-range sensor over short time periods. The motion estimate is used as a prior when aligning consecutive laser scans, allowing for fast and relatively accurate mapping.

Often inaccuracies remain, for example caused by wrong data associations in visual odometry. These inaccuracies lead to misalignments and degeneration in the map and require costly reprocessing of the sensor data. To this end, graph-based optimization is popular to minimize accumulated errors [5], [6], [7]. However, depending on the granularity of the modeled graph, optimization is computationally demanding for large maps.

Another difficulty in laser-based SLAM is the sparseness and distribution of measurements in laser scans. As a result, pairwise registration of laser scans quickly accumulates

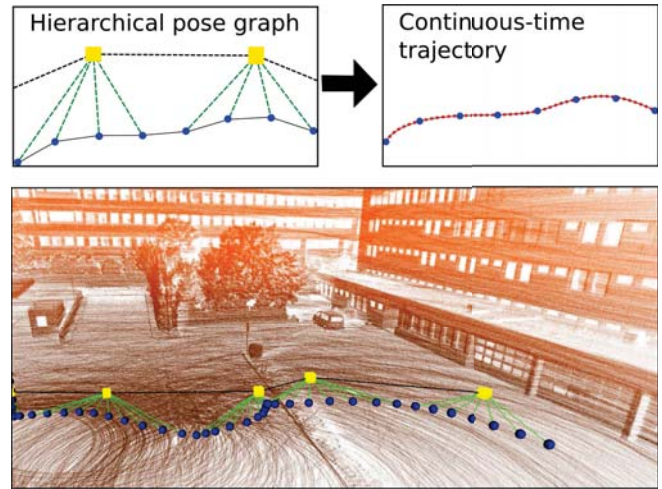


Fig. 1: We propose a hierarchical continuous-time SLAM method, allowing for online map refinement. It generates highly accurate maps of the environment from laser measurements. Yellow squares: coarse nodes; Blue circles: fine nodes; Red dots: continuous-time trajectory.

errors. Registering laser scans to a map, built by aggregating previous measurements, often minimizes accumulated error.

However, errors remain, e.g., due to missing information. For example, incrementally mapping the environment necessitates bootstrapping from sparse sensor data at the beginning—resulting in relatively poor registration accuracy, compared to aligning with a dense and accurate map.

In our previous work [8], we showed that local multiresolution in combination with a surfel-based registration method allows for efficient and robust mapping of sparse laser scans. The key data structure in our previous work is a robot-centric multiresolution grid map to recursively aggregate laser measurements from consecutive 3D scans, yielding local mapping with constant time and memory requirements. Furthermore, modeling a graph of local multiresolution maps allows for allocentric mapping of large environments [9]. While being efficient, the approach did not allow reassessing previously aggregated measurements in case of registration errors and poor or missing motion estimates from visual odometry.

In this paper, we extend our previous approach, allowing for reassessing the registration of previously added 3D scans. By modeling individual 3D scans of a local map as a sub-graph, we build a hierarchical graph structure, enabling refinement of the map in case misaligned measurements

All authors are with the Autonomous Intelligent Systems Group, Computer Science Institute VI, University of Bonn, 53115 Bonn, Germany {droschel, behnke}@ais.uni-bonn.de

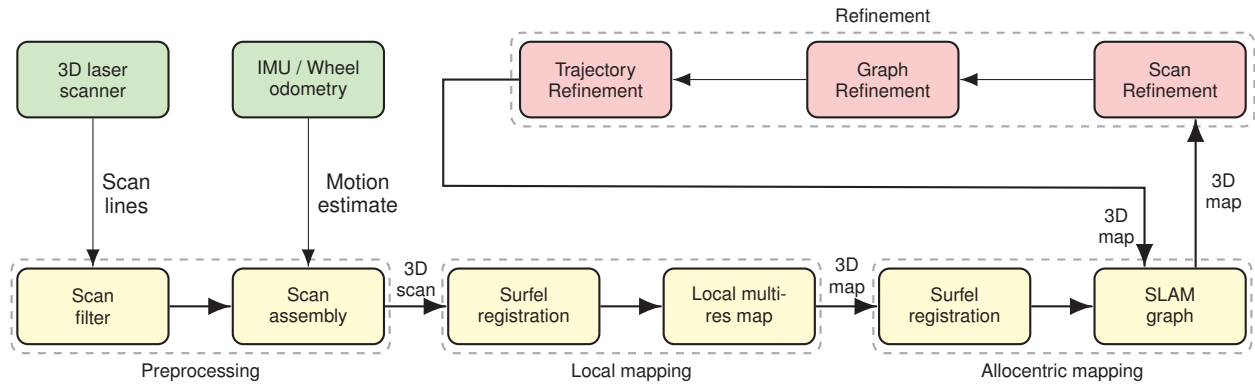


Fig. 2: Schematic illustration of our mapping system. Laser measurements are preprocessed and assembled to 3D point clouds. The resulting 3D point cloud is used to estimate the transformation between the current scan and map. Registered scans are stored in a local multiresolution map. Local multiresolution maps from different view poses are registered against each other in a SLAM graph. During mapping, parts of the graph are refined and misaligned 3D scans are corrected.

when more information is available. Furthermore, the approach preserves efficient local and allocentric mapping, as with our previous method. In summary, the contribution of our work is a novel combination of a hierarchical graph structure—allowing for scalability and efficiency—with local multiresolution maps to overcome alignment problems due to sparsity in laser measurements, and a continuous-time trajectory representation.

II. RELATED WORK

Mapping with 3D laser scanners has been investigated by many groups [1], [2], [3], [4]. While many methods assume the robot to stand still during 3D scan acquisition, some approaches also integrate scan lines of a continuously rotating laser scanner into 3D maps while the robot is moving [10], [11], [12], [13], [14]. The mentioned approaches allow creating accurate maps of the environment under certain conditions, but do not allow an efficient assessment and refinement of the map.

Measurements from laser scanners are usually subject to rolling shutter artifacts when the sensor is moving during acquisition. These artifacts are expressed by a deformation of the scan and, when treating laser scans as rigid bodies, these artifacts degrade the map quality and introduce errors when estimating the sensor pose. A common approach to address this problem is to model a deformation in the objective function of the registration approach. Non-rigid registration of 3D laser scans has been addressed by several groups [15], [16], [17], [18], [19].

Ruhnke et al. [17] jointly optimize sensor poses and measurements. They extract surface elements from range scans, and seek for close-by surfels from different scans. This data association contributes to the error term of the optimization problem but also results in a relatively high state space. Thus, their approach can build highly accurate 3D maps but does not allow for online processing.

Furthermore, rolling shutter effects can be addressed by modeling the sensor trajectory as a continuous function over

time, instead of a discrete set of poses. Continuous-time representations show great advantages when multiple sensors with different temporal behavior are calibrated [20] or fused [21], but also to compensate for rolling shutter effects, e.g., for data from a RGB-D camera [22]. Continuous-time trajectory representations have been used for laser-based mapping in different works [23], [24], [25]. While most of the continuous-time approaches use a spline to represent the trajectory, Anderson et al. [19] employ Sparse Gaussian Process Regression.

Kaul et al. [18], [25] present a continuous-time mapping approach using non-rigid registration and global optimization to estimate the sensor trajectory from a spinning laser scanner and an industrial-grade IMU. The trajectory is modeled as a continuous function and a spline is used to interpolate between the sensor poses.

Recently, Hess et al. presented *Google's Cartographer* [26]. By aggregating laser scans in local 2D grid maps and an efficient branch-and-bound approach for loop closure optimization. Results of Google's Cartographer have been improved by Nüchter et al. [27]. Their method refines the resulting trajectory by a continuous-time mapping approach, based on their previous work [28].

Grisetti et al. present a hierarchical graph-based SLAM approach [29]. Similar to our approach, a hierarchical pose graph represents the environment on different levels, allowing for simplifying the problem and optimizing parts of it independently.

Following Grisetti et al. [29], we model our problem as hierarchical graph, allowing us to optimize simplified parts of the problem independently. Compared to their approach we aggregate scans in local sub-maps to overcome sparsity in the laser scans. Furthermore, we augment the local sub-graphs with a continuous-time representation of the trajectory, allowing to address the mentioned rolling shutter effects.

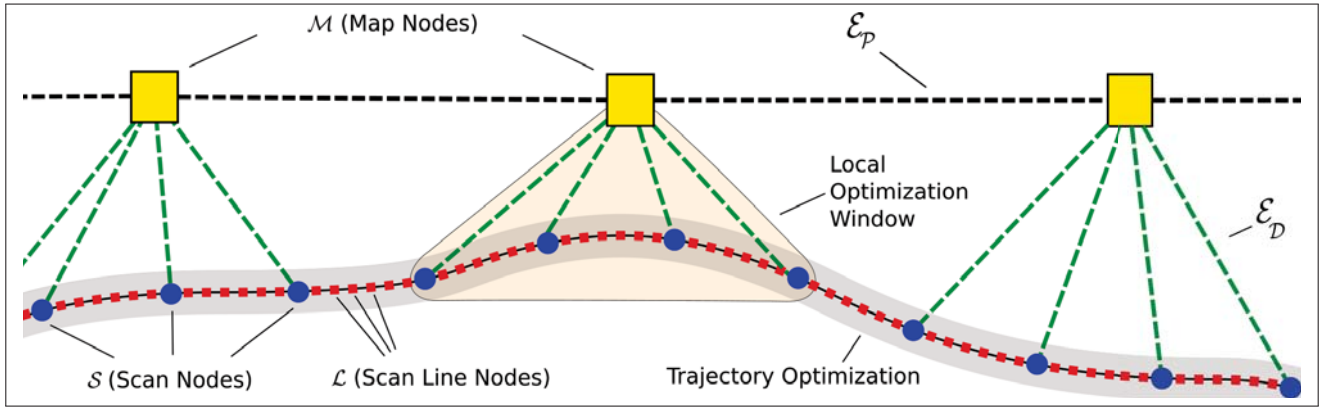


Fig. 3: Hierarchical graph representation of the optimization problem at hand: The vertex sets \mathcal{M} , \mathcal{S} , and \mathcal{L} correspond to the estimation variables, i.e., the poses of the local multiresolution maps (\mathcal{M}), the 3D scans (\mathcal{S}), and scan lines (\mathcal{L}) of the Velodyne VLP-16. The edge sets \mathcal{E}_P and \mathcal{E}_D represent constraints from registration: \mathcal{E}_P from aligning two local maps to each other and \mathcal{E}_D from aligning a 3D scan to a surfel map. From \mathcal{S} a continuous-time representation of the trajectory is estimated by a cubic B-spline, allowing to interpolate the pose for each measurement of the 3D scan (\mathcal{L}).

III. SYSTEM OVERVIEW

Our system aggregates measurements from a laser scanner in a robot-centric local multiresolution grid map—having a high resolution in the close proximity to the sensor and a lower resolution with increasing distance [30]. In each grid cell, individual measurements are stored along with an occupancy probability and a surface element (surfel). A surfel summarizes its attributed points by their sample mean and covariance.

If available, we incorporate information from other sensors, such as an inertial measurement unit (IMU) or wheel odometry, to account for motion of the sensor during acquisition. Furthermore, these motion estimates are used as prior for the registration.

To register acquired 3D scans to the so far aggregated map, we use our surfel-based registration method [30], [31]. The registered 3D scan is added to the local map, replacing older measurements. Similar to [32] we use a beam-based inverse sensor model and ray-casting to update the occupancy of a cell. For every measurement in the 3D scan, we update the occupancy information of cells on the ray between the sensor origin and the endpoint with an approximated 3D Bresenham algorithm [33]. Since local multiresolution maps consist of consecutive scans from a fixed period of time, they allow for efficient local mapping with constant memory and computation demands.

Local multiresolution maps from different view poses are aligned with each other by means of surfel-based registration and build an allocentric pose graph. The registration result from aligning two local maps constitutes an edge in this pose graph. Edges are added when the pose graph is extended by a new local map and between close-by local maps—e.g., when the robot revisits a known location. The later allows for loop closure and minimizes the drift accumulated by the local mapping.

Local maps that are added to the pose graph are subject to our refinement method, reassessing the alignment of 3D

scans when more information is available. After realigning selected 3D scans from a local map, the sensor trajectory is optimized: first for refined local maps, then for the complete pose graph. Figure 2 shows an overview of our mapping system. Since local mapping, allocentric mapping and refinement are independent from each other, our system allows for online mapping while refining previously acquired sensor data when more information is available.

IV. HIERARCHICAL REFINEMENT

We model our mapping approach in a hierarchical graph-based structure as shown in Figure 3. The coarsest level is a pose graph, representing the allocentric 6D pose of local maps $\mathcal{M} = \{m_1, \dots, m_M\}$ with nodes. Each local map aggregates multiple consecutive 3D scans and represents the robot's vicinity at a given view pose.

They are connected by edges \mathcal{E}_P imposing a spatial constraint from registering two local maps with each other by surfel-based registration. We denote edges $E = ((M_E, M'_E), T_E, I_E) \in \mathcal{E}_P$ as spatial constraint between the local maps M_E and M'_E with the relative pose T_E and the information matrix I_E , which is the inverse of the covariance matrix from registration.

The scan poses of a local map \mathcal{M}_j are modeled by vertices $\mathcal{S} = \{s_1, \dots, s_S\}$ in a sub-graph \mathcal{G}_j . They are connected by edges \mathcal{E}_D . Registering a scan S_E to a local map M_E poses a spatial constraints $E = ((S_E, M_E), T_E, I_E) \in \mathcal{E}_D$ with the relative pose T_E and the information matrix I_E .

The 3D scans of the local maps consist of a number of so-called scan lines. A scan line is the smallest element in our optimization scheme. Depending on the sensor setup, a scan line consists of measurements acquired in a few milliseconds. For the Velodyne VLP-16 used in the experiments, a scan line is a single firing sequence (1,33 ms). We assume the measurements of a scan line to be too sparse for robust registration. Thus, we interpolate the poses of scan line

acquisitions with a continuous-time trajectory representation for each sub-graph, as described later.

Optimization of the sub-graphs and the pose graph is efficiently solved using the g^2o framework by [5], yielding maximum likelihood estimates of the view poses \mathcal{S} and \mathcal{M} . On their local time scale, sub-graphs are independent from each other, allowing to minimize errors independent from other parts of the graph. Optimization results from sub-graphs are incorporated in the higher level pose graph, correcting the view poses of the local maps. Therefore, we define the last acquired scan node in a local sub-graph as reference node and update the pose of the map node according to it.

During operation, we iteratively refine sub-graphs in parallel, depending on the available resources. Global optimization of the full graph is only performed when the local optimization has changed a sub-graph significantly or a loop closure constraint was added. Similarly, if global optimization was triggered by loop closure, sub-graphs are refined when the corresponding map node changed. To determine if optimization of a sub-graph necessitates global optimization or vice-versa, we compare the refined pose of the representative scan node s_r in a sub-graph to the view pose of the corresponding map node. For our experiments, we choose a threshold of 0.01 m in translation and 1° in rotation.

A. Local Sub-Graph Refinement

After a local map has been added to the pose graph, the corresponding sub-graph \mathcal{G}_M is refined by realigning selected 3D scans with its local map. Realigning only selected 3D scans, instead of all scans in a sub-graph, allows for fast convergence while resulting in similar map quality, as shown later in the experiments.

For a sub-graph, we determine a 3D scan s_k for refinement by the spatial constraints and their associated information matrix, which is the inverse of the covariance matrix Σ of the registration result. Following [34], we determine a scalar value for the uncertainty in the scan poses based on the entropy $H(T, \Sigma) \propto \ln(|\Sigma|)$. It allows to select the 3D scan with the largest expected alignment error.

Furthermore, the same measure is used to compare the spatial constraints that have been added to the local map after s_k , to determine if realigning s_k can decrease the alignment error. The selected 3D scan is then refined, by realigning it to its local map, resulting in a refined spatial constraint in the sub-graph. From the sub-graph of spatial constraints, we infer the probability of the trajectory estimate given all relative pose observations

$$p(\mathcal{G}_M | \mathcal{E}_D) \propto \prod_{e_{d_{ij}} \in \mathcal{E}_D} p(s_i^j | s_i, s_j). \quad (1)$$

We optimize the sensor trajectory for each local sub-graph independently. Results from sub-graph optimization are later incorporated when optimizing the allocentric pose graph.

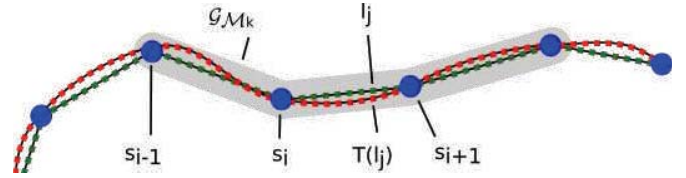


Fig. 4: Scan line poses (green squares) originated from odometry measures are refined (red squares) by interpolating with a continuous-time trajectory representation built from scan poses (blue dots) in a local sub-graph (gray).

B. Local Window Alignment

Registration errors are often originated from missing information in the map, e.g., due to occlusions or unknown parts of the environment. Thus, registration quality can only increase if the map has been extended with measurements that provide previously unknown information. In other words, realigning a 3D scans can only increase the map quality if more scans—in best case from different view poses—have been added to the map. Therefore, we increase the local optimization window by adding 3D scans to a local map from neighboring map nodes in the higher level. For example, when the robot revisits a known part of the environment, loop closure is performed and scan nodes from neighboring map nodes are added to a local map.

C. Continuous-Time Trajectory Representation

Acquiring 3D laser scans often involves mechanical actuation, such as rotating a mirror or a diode/receiver array, during acquisition of the scan. Especially for 3D laser scans—where the acquisition of measurements for a full scan can take multiple hundred milliseconds or seconds—a discretization of the sensor pose to the time where the scan was acquired, leads to artifacts degrading the map quality. However, since a finer discretization of the scan poses makes the state size intractable, temporal basis functions have been used to represent the sensor trajectory [35].

We represent the trajectory of the sensor as cubic B-spline in $SE(3)$ due to their smoothness and the local support property. The local support property allows to interpolate the trajectory from the discrete scan nodes in our local sub-graph. Following [36], we parameterize a trajectory by cumulative basis functions using the Lie algebra.

To estimate the trajectory spline, we use the scan nodes s_0, \dots, s_m with the acquisition times t_{s_0}, \dots, t_{s_m} as control points for the trajectory spline and denote the pose of a scan node s_i as T_{s_i} . In our system, scan poses follow a uniform temporal distribution. In other words, the difference between the acquisition times of consecutive scans can be assumed to be constant.

As illustrated in Figure 4, we use 4 control points to interpolate the sensor trajectory between two scan nodes s_i

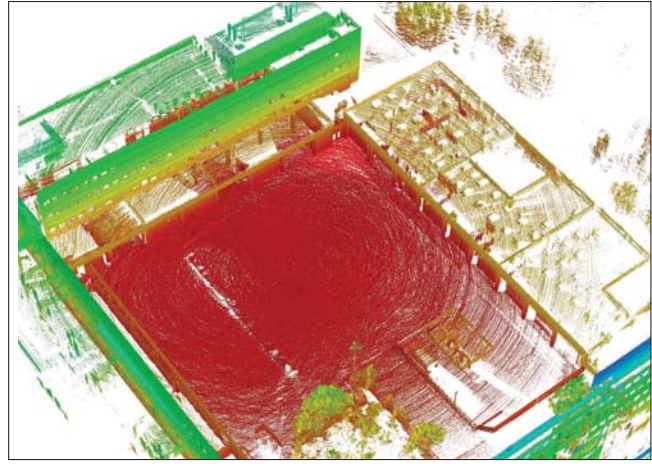


Fig. 5: The resulting 3D map from an out/in-door environment. Color encodes height from the ground.

and s_{i+1} . For time $t \in [t_{s_i}, t_{s_{i+1}})$ the pose along the spline is defined as

$$T(u(t)) = T_{s_{i-1}} \prod_{j=1}^3 \exp(\tilde{B}_j(u(t))\Omega_{i+j}). \quad (2)$$

Here, \tilde{B} is the cumulative basis, Ω is the logarithmic map, and $u(t) \in [0, 1)$ transforms time t in a uniform time [36]. Finally, the spline trajectory is used to update the scan line poses between two scan nodes.

D. Loop-Closure and Global Optimization

After a new local map has been added to pose graph, we check for one new constraint between the current reference m_{ref} and other map nodes m_{cmp} . We determine a probability

$$p_{\text{chk}}(v_{\text{cmp}}) = \mathcal{N}(d(m_{\text{ref}}, m_{\text{cmp}}); 0, \sigma_d^2) \quad (3)$$

that depends on the linear distance $d(m_{\text{ref}}, m_{\text{cmp}})$ between the view poses m_{ref} and m_{cmp} . According to $p_{\text{chk}}(m)$, we choose a map node m from the graph and determine a spatial constraint between the nodes.

When a new spatial constraint has been added, the pose graph is optimized globally on the highest level. When the optimization modifies the estimate of a map node, the changes are propagated to the sub-graph. Similarly, when a sub-graph changed significantly, global optimization of the highest level is carried out.

V. EXPERIMENTS

We assess the accuracy of our refinement method on two different data sets with different sensor setups. The first data has been recorded with a MAV equipped with a Velodyne VLP-16 lidar sensor. The second data set has been recorded in the Deutsches Museum in Munich and is provided by the Google Cartographer team [26]. Throughout the experiments, we use a distance threshold of 5 m for adding new map nodes to the graph.

To measure map quality, we calculate the mean map entropy (MME) [30] from the points $\mathcal{Q} = \{q_1, \dots, q_Q\}$

of the resulting map. The entropy h for a map point q_k is calculated by

$$h(q_k) = \frac{1}{2} \ln |2\pi e \Sigma(q_k)|, \quad (4)$$

where $\Sigma(q_k)$ is the sample covariance of mapped points in a local radius r around q_k . We select $r = 0.5$ m in our evaluation. The mean map entropy $H(\mathcal{Q})$ is averaged over all points of the resulting map

$$H(\mathcal{Q}) = \frac{1}{Q} \sum_{k=1}^Q h(q_k). \quad (5)$$

It represents the crispness or sharpness of a map. Lower entropy measures correspond to higher map quality.

To examine the improvement of the map quality and the convergence behavior of our method, we first run the experiments without online refinement and perform the proposed refinement as a post-processing step. In each iteration, we refine one scan in every sub-graph and run local graph optimization. Local sub-graphs are refined in parallel and after refining all sub-graphs, global optimization is performed. To assess the number of iterations necessary for refinement, entropy measurements are plotted against the number of iterations. Afterwards, we run the proposed method with online refinement and compare the resulting map quality.

Evaluation was carried out on an Intel® Core™ i7-6700HQ quadcore CPU running at 2.6 GHz and 32 GB of RAM. For the reported runtime, we average over 10 runs for each data set.

A. Courtyard

The first data set has been recorded by a MAV during flight in a building courtyard. The MAV in this experiment is a DJI Matrice 600, equipped with a Velodyne VLP-16 lidar sensor and an IMU, measuring the attitude of the robot. The Velodyne lidar measures $\approx 300,000$ range measurements per second in 16 horizontal scan rings, has a vertical field of view of 30° and a maximum range of 100 m.

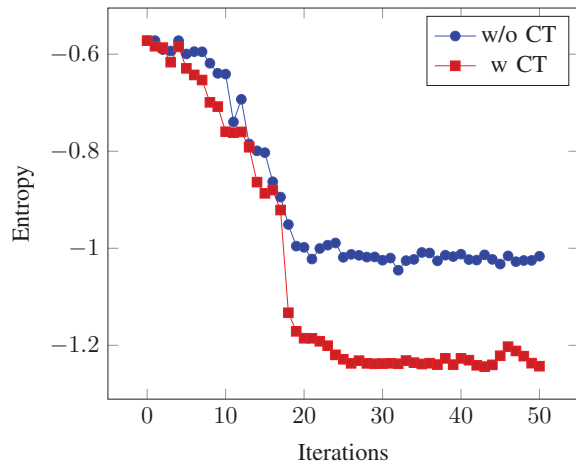


Fig. 6: The resulting map entropy with and without continuous-time trajectory interpolation (CT) Section IV-C.

It measures the environment with 16 emitter/detector pairs mounted on an array at different elevation angles from the horizontal plane of the sensor. The array is continuously rotated with up to 1200 rpm. In our experiments, a scan line corresponds to one data packet received by the sensor, i.e., 24 so-called firing sequences. During one firing sequence (1,33 ms) all 16 emitter/detector pairs are processed.

In total, 2000 scans were recorded during 200 s flight time. Controlled by a human operator, the MAV traversed a building front in different heights. The resulting graph consists of 16 map nodes with several loop closures, resulting in 27 edges between map nodes.

Figure 5 shows the environment and a resulting map. In a first experiment, we compare the method from our previous work with the proposed method. The resulting point clouds are shown in Figure 7. The figure shows, that the proposed method corrects misaligned 3D scans and increases the map quality. Figure 6 shows the resulting entropy plotted against the number of iterations when running the refinement as post-processing step. During one iteration, a single 3D scan in each map node is refined. We measure an average runtime of 54 ms per iteration for refining a single map node and 380 ms per iteration for refining all 16 map nodes in parallel.

B. Deutsches Museum

For further evaluation, we compare our method on a data set that has been recorded at the Deutsches Museum in Munich. The data set is provided by Hess et al. [26]. Two Velodyne VLP-16 mounted on a backpack are carried through the museum. Parts of the data set contain dynamic objects, such as moving persons. The provided data set includes a calibration between the two laser scanners—one mounted horizontal, one vertical. We use the provided calibration between the two sensors as initial calibration guess and refine it by adding additional constraints to our pose graph and the local sub-graphs. Similar to the alignment of two local maps, we refine the calibration parameters by our surfel-based method, registering the scans of the graph

TABLE I: Resulting best mean map entropies (MME) for the data set recorded at Deutsches Museum.

Method	MME
Cartographer [26]	-2.04
Droeschel et al. [8]	-2.12
Nüchter et al. [27]	-2.34
Ours	-2.42

from the horizontal scanner to the scans of the graph from the vertical scanner.

Following [27], we select a part of the data set and run our method on it. Besides visual inspection of the resulting point cloud we compute the entropy as described before. Figure 8 shows the convergence behavior of our method with and without the covariance-based scan selection. It indicates that our covariance-based scan selection leads to faster convergence. Furthermore, we compare the presented method with the method from our previous work. We also compare our method to Google’s Cartographer [26] and the continuous-time slam method from [28] that has been evaluated in [27]. We summarize our results for each method in Table I.

VI. CONCLUSIONS

In this paper, we proposed a hierarchical, continuous-time approach for laser-based 3D SLAM. Our method is based on efficient local mapping and a hierarchical optimization backend. Measurements of a 3D laser scanner are aggregated in local multiresolution maps, by means of surfel based registration. The local maps are used in a graph-based structure for allocentric mapping. The individual 3D scans in the local map model a sub-graph to incorporate corrections when refining these sub-graphs. Graph optimization is performed to account for drift and misalignments in the local maps. Furthermore, a continuous-time trajectory representation allows to interpolate measurements between discrete scan poses. Evaluation shows that our approach increases map quality and leads to sharper maps.

ACKNOWLEDGMENTS

This work was supported by grant BE 2556/7 (Mapping on Demand) of the German Research Foundation (DFG), grant 01MA13006D (InventAIRy) of the German BMWi, and grant 644839 (CENTAURO) of the European Union’s Horizon 2020 Programme. We like to thank Andreas Nüchter, Michael Bleier and Andreas Schauer for providing datasets for the evaluation.

REFERENCES

- [1] A. Nuechter, K. Lingemann, J. Hertzberg, and H. Surmann, “6D SLAM with approximate data association,” in *Advanced Robotics (ICAR), IEEE International Conference on*, 2005.
- [2] M. Magnusson, T. Duckett, and A. J. Lilienthal, “Scan registration for autonomous mining vehicles using 3D-NDT,” *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.
- [3] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation,” in *Safety, Security, and Rescue Robotics (SSRR), IEEE International Symposium on*, 2011, pp. 155–160.

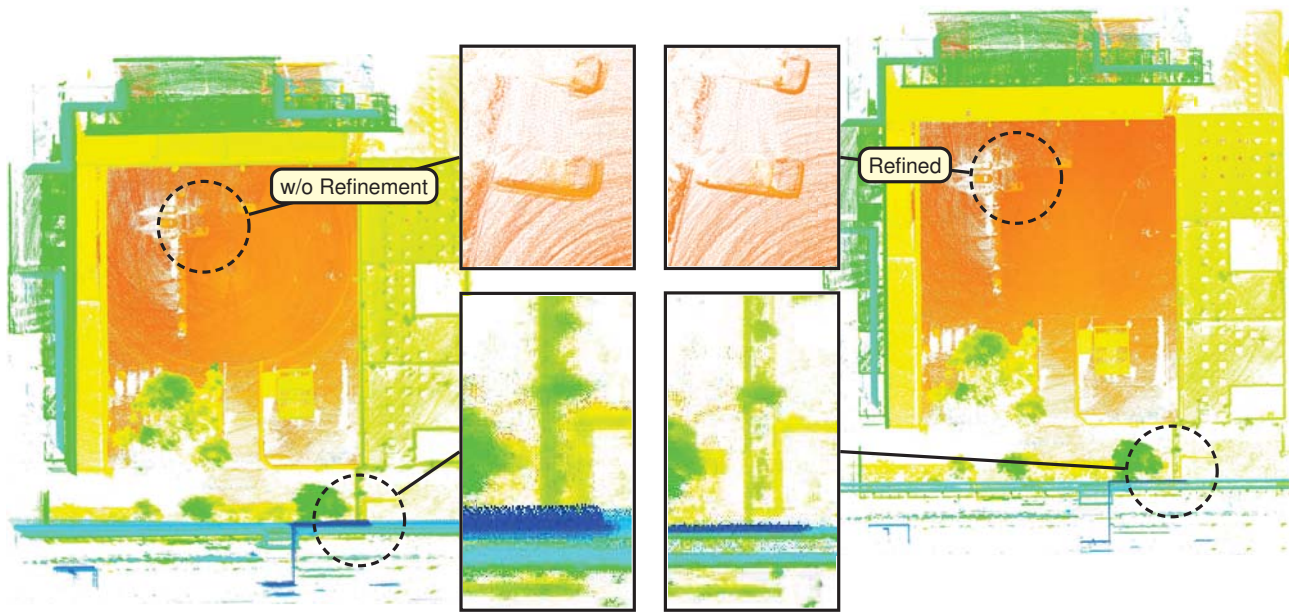


Fig. 7: Resulting point clouds from the Courtyard data set. Left: results from our previous method [8]. Right: results from the presented method. Color encodes the height.

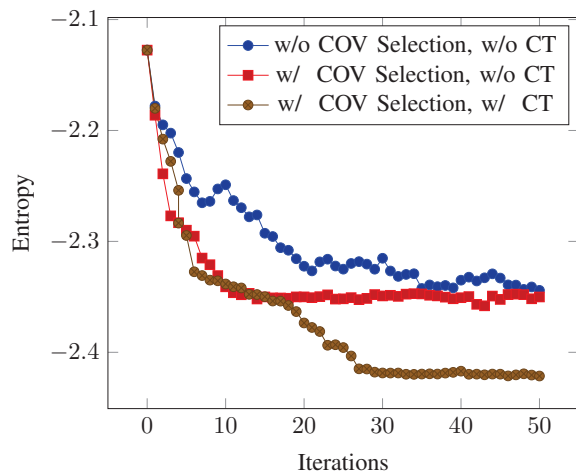


Fig. 8: The resulting map entropy with and without our covariance based scan selection (COV Selection) described in Section IV-A and continuous-time trajectory interpolation (CT) Section IV-C. Scan selection leads to faster convergence of our method.

- [4] D. Cole and P. Newman, "Using laser range data for 3D SLAM in outdoor environments," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2006, pp. 1556–1563.
- [5] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2011.
- [6] U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous localization and mapping," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 196–207, April 2005.
- [7] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Robotics and Automation (ICRA), IEEE International Conference on*. IEEE, 2006, pp. 2262–2269.

- [8] D. Droschel, M. Schwarz, and S. Behnke, "Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner," *Robotics and Autonomous Systems*, vol. 88, pp. 104 – 115, 2017.
- [9] D. Droschel, M. Nieuwenhuisen, M. Beul, D. Holz, J. Stückler, and S. Behnke, "Multilayered mapping and navigation for autonomous micro aerial vehicles," *Journal of Field Robotics (JFR)*, vol. 33, no. 4, pp. 451–475, 2016.
- [10] M. Bosse and R. Zlot, "Continuous 3D scan-matching with a spinning 2D laser," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2009.
- [11] J. Elseberg, D. Borrmann, and A. Nuechter, "6DOF semi-rigid SLAM for mobile scanning," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 2012.
- [12] T. Stoyanov and A. Lilienthal, "Maximum likelihood point cloud acquisition from a mobile platform," in *Int. Conf. on Advanced Robotics (ICAR)*, 2009.
- [13] W. Maddern, A. Harrison, and P. Newman, "Lost in translation (and rotation): Fast extrinsic calibration for 2D and 3D LIDARs," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2012.
- [14] S. Anderson and T. D. Barfoot, "Towards relative continuous-time SLAM," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.
- [15] B. J. Brown and S. Rusinkiewicz, "Global non-rigid alignment of 3-d scans," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, ACM, 2007, p. 21.
- [16] J. Elseberg, D. Borrmann, K. Lingemann, and A. Nüchter, "Non-rigid registration and rectification of 3D laser scans," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1546–1552.
- [17] M. Ruhnke, R. Kümmerle, G. Grisetti, and W. Burgard, "Highly accurate 3D surface models by sparse surface adjustment," in *Robotics and Automation (ICRA), IEEE International Conference on*. IEEE, 2012, pp. 751–757.
- [18] R. Zlot and M. Bosse, "Efficient large-scale three-dimensional mobile mapping for underground mines," *Journal of Field Robotics*, vol. 31, no. 5, pp. 758–779, 2014.
- [19] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch non-linear continuous-time trajectory estimation as exactly sparse gaussian process regression," *Auton. Robots*, vol. 39, pp. 221–238, 2015.
- [20] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Intelligent Robots and Systems*

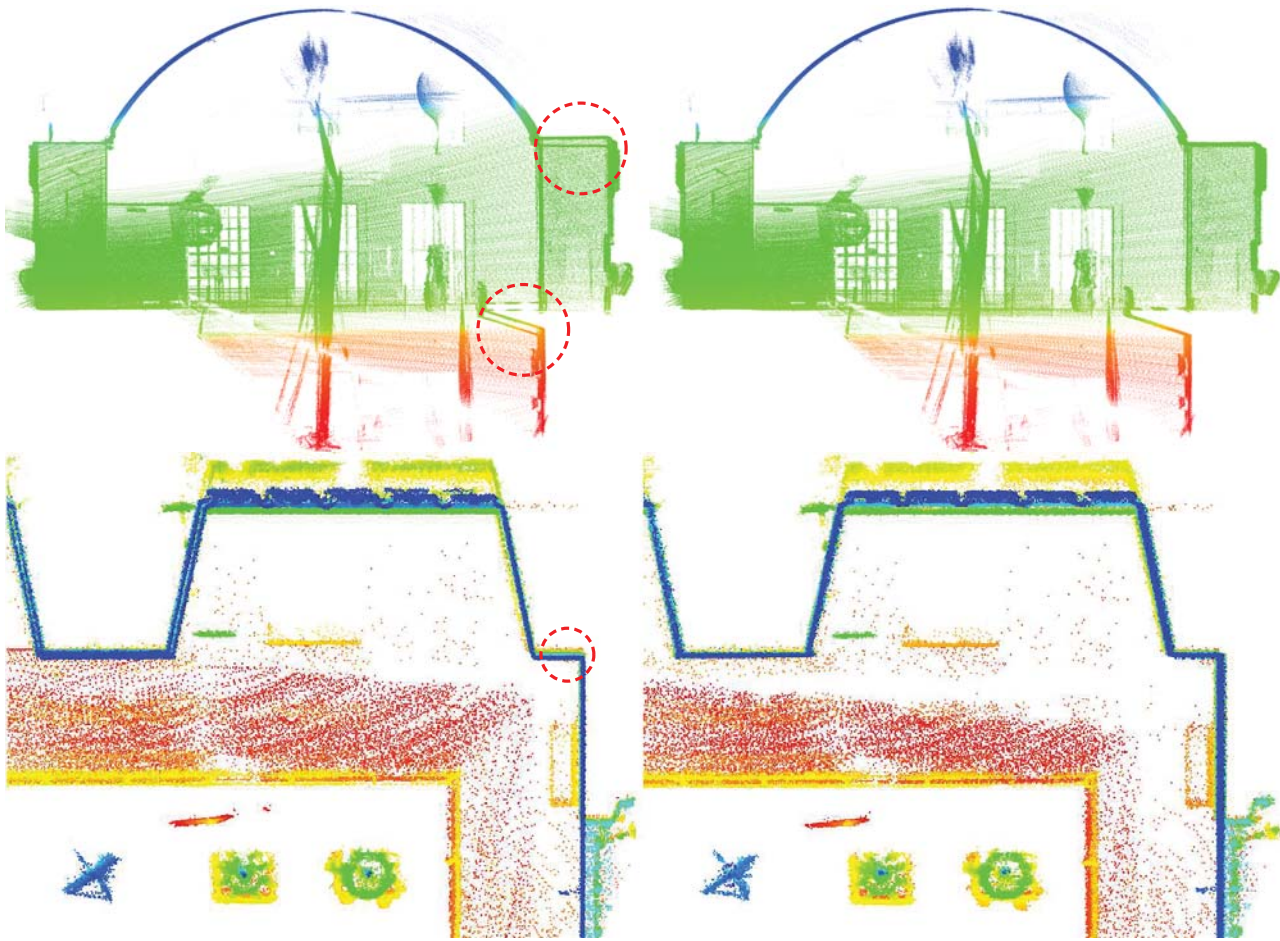


Fig. 9: Resulting point clouds from a part of the trajectory of the Deutsches Museum data set. Left: results from [27]. Right: results from our method. Red (dashed) circles highlight distorted parts of the map. Color encodes the height.

- (IROS), *IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1280–1286.
- [21] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, “Continuous-time visual-inertial trajectory estimation with event cameras,” *arXiv preprint arXiv:1702.07389*, 2017.
- [22] C. Kerl, J. Stückler, and D. Cremers, “Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras,” in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2264–2272.
- [23] H. Alismail, L. D. Baker, and B. Browning, “Continuous trajectory estimation for 3D SLAM from actuated lidar,” in *Robotics and Automation (ICRA), IEEE International Conference on*. IEEE, 2014, pp. 6096–6101.
- [24] A. Patron-Perez, S. Lovegrove, and G. Sibley, “A spline-based trajectory representation for sensor fusion and rolling shutter cameras,” *International Journal of Computer Vision*, vol. 113, no. 3, pp. 208–219, 2015.
- [25] L. Kaul, R. Zlot, and M. Bosse, “Continuous-time three-dimensional mapping for micro aerial vehicles with a passively actuated rotating laser scanner,” *Journal of Field Robotics*, vol. 33, no. 1, pp. 103–132, 2016.
- [26] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2D lidar slam,” in *Robotics and Automation (ICRA), IEEE International Conference on*, 2016, pp. 1271–1278.
- [27] A. Nüchter, M. Bleier, J. Schauer, and P. Janotta, “Improving Google’s Cartographer 3D mapping by continuous-time slam,” *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 543–549, 2017.
- [28] J. Elseberg, D. Borrmann, and A. Nüchter, “Algorithmic solutions for computing precise maximum likelihood 3D point clouds from mobile laser scanning platforms,” *Remote Sensing*, vol. 5, no. 11, pp. 5871–5906, 2013.
- [29] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg, “Hierarchical optimization on manifolds for online 2D and 3D mapping,” in *Robotics and Automation (ICRA), IEEE International Conference on*, 2010, pp. 273–278.
- [30] D. Droschel, J. Stückler, and S. Behnke, “Local multi-resolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner,” in *Robotics and Automation (ICRA), IEEE International Conference on*, 2014.
- [31] J. Stückler and S. Behnke, “Multi-resolution surfel maps for efficient dense 3D modeling and tracking,” *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 137–147, 2014.
- [32] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: an efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, vol. 34, pp. 189–206, 2013.
- [33] J. Amanatides and A. Woo, “A fast voxel traversal algorithm for ray tracing,” in *In Eurographics 87*, 1987, pp. 3–10.
- [34] C. Kerl, J. Sturm, and D. Cremers, “Dense visual slam for rgb-d cameras,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2100–2106.
- [35] P. Furgale, T. D. Barfoot, and G. Sibley, “Continuous-time batch estimation using temporal basis functions,” in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2012, pp. 2088–2095.
- [36] S. Lovegrove, A. Patron-Perez, and G. Sibley, “Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras,” in *British Machine Vision Conference*, 2013.