

DMSA - Dense Multi Scan Adjustment for LiDAR Inertial Odometry and Global Optimization

David Skuddis¹ and Norbert Haala¹

Abstract—We propose a new method for fine registering multiple point clouds simultaneously. The approach is characterized by being dense, therefore point clouds are not reduced to pre-selected features in advance. Furthermore, the approach is robust against small overlaps and dynamic objects, since no direct correspondences are assumed between point clouds. Instead, all points are merged into a global point cloud, whose scattering is then iteratively reduced. This is achieved by dividing the global point cloud into uniform grid cells whose contents are subsequently modeled by normal distributions. We show that the proposed approach can be used in a sliding window continuous trajectory optimization combined with IMU measurements to obtain a highly accurate and robust LiDAR inertial odometry estimation. Furthermore, we show that the proposed approach is also suitable for large scale keyframe optimization to increase accuracy. We provide the source code and some experimental data on https://github.com/davidskdds/DMSA_LiDAR_SLAM.git.

I. INTRODUCTION

The field of point cloud registration and simultaneous localization and mapping (SLAM) has been researched for several decades and there are already a large amount of mature approaches and implementations. The task here is to estimate the sensors trajectory using point clouds (and IMU data). Current challenges in this area are, in particular, achieving high accuracies (mm-range) and increasing robustness. In terms of robustness, especially narrow spaces, highly dynamic motion and dynamic objects are current challenges. In this paper, we present a new approach that aims to take us one step closer to solving these challenges.

The main contribution of this work is the proposal of a new method for dense multi scan adjustment. While other methods for point cloud registration already use multivariate normal distributions to separately model point clouds within a registration process e.g. [1], [2], [3], [4], our method considers the points from two or more point clouds in a common frame as one set of multivariate normal distributions. This approach has two main advantages: the normal distributions updated in each iteration serve as weights in the optimization, which is why non-matching shapes from e.g. dynamic objects are automatically weighted lower. An outlier rejection mechanism is therefore not necessary. The second advantage is that there are no direct correspondences between point clouds to be registered. Instead, there are only correspondences from each point to its respective normal distribution, thus number

of correspondences is largely reduced. As a result, multiple point clouds can be optimized efficiently at the same time. Furthermore, we have developed a LiDAR inertial odometry algorithm based the method that performs superior to other state-of-the-art methods in terms of robustness as well as accuracy in our experiments with a challenging selection of data sequences.

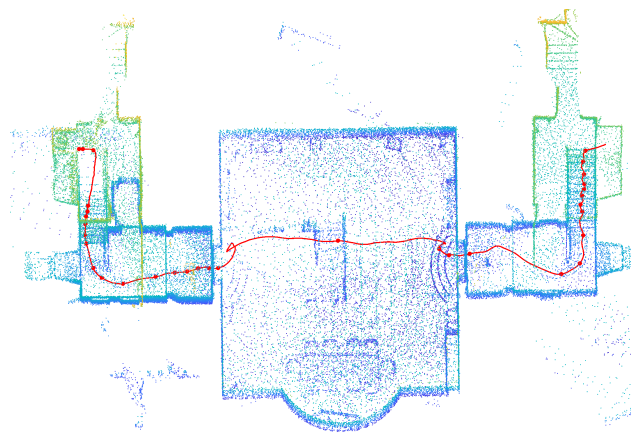


Fig. 1: Resulting sparse keyframe point cloud of sequence *exp14 Basement* of the Hilti-Oxford Dataset [5]. The estimated trajectory is marked with a red line and keyframe positions are outlined with red dots.

II. RELATED WORK

In this chapter we give a brief overview about point cloud registration, LiDAR (inertial) odometry and multi scan registration methods for global optimization.

A. Point Cloud Registration

The field of point cloud registration is very broad and there is an extensive number of methods. In our literature review, we restrict attention only to convergence-based approaches. Probably the most famous algorithm in the field of point cloud registration is the ICP algorithm [6], from which many other approaches are derived [7], [8]. There are still modern SLAM systems that are based on the ICP algorithm [9]. Another group of methods for point cloud registration transforms one of the two point clouds into sets of normal distributions and then optimizes the relative pose of the second point cloud by maximizing the likelihood, in some works represented as a score, of the points of the second point cloud [1], [2], [3]. There are also concepts, where both point clouds to be registered are converted to normal

¹Both authors are with Institute for Photogrammetry and Geoinformatics, University of Stuttgart, 70174 Stuttgart, Germany {david.skuddis, norbert.haala}@ifp.uni-stuttgart.de

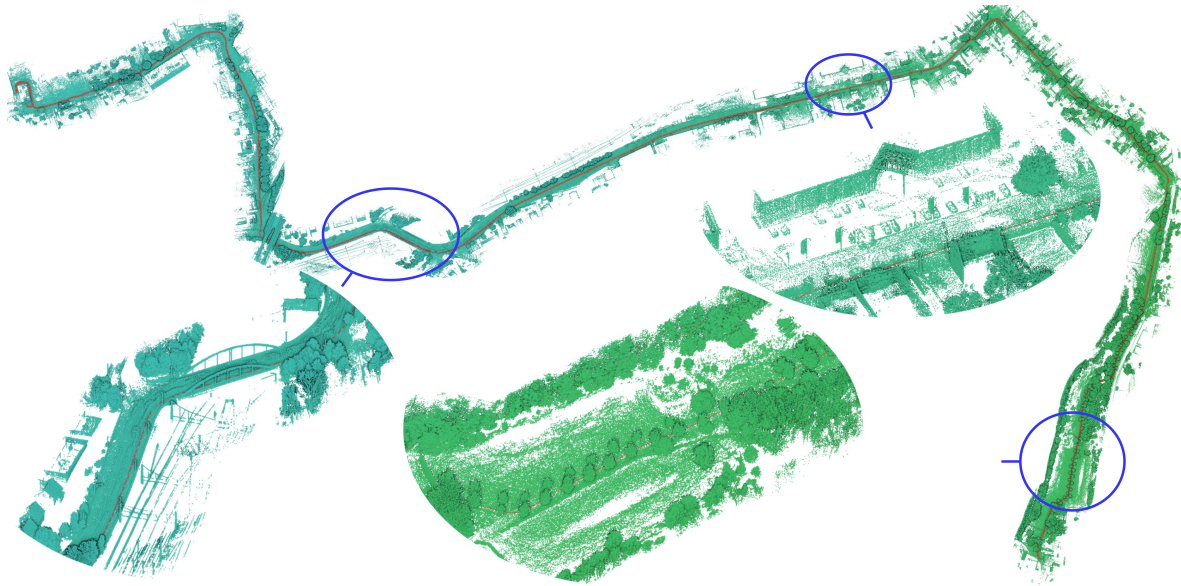


Fig. 2: Resulting keyframe point cloud from sequence *Bicycle Street* of our data recording. The scene contains urban areas, field paths and dynamic objects. The trajectory of the sensor platform mounted on a bicycle trailer is marked in red. For areas marked in blue, detail views are provided.

distributions and the error between the Normal distributions is minimized using metrics like KL divergence [4]. All these approaches assume correspondences between nearest points or nearest normal distributions between the point clouds to be registered. This can result in erroneous correspondences due to small overlaps or dynamic objects, which reduce the accuracy of the solution found. One way to deal with this is an advanced weighting of correspondences during optimization to reduce the influence of outliers [10], [9]. Other methods avoid the formation of explicit correspondences by minimizing a cost function that is independent of correspondences for the determination of relative poses. [11] proposes a correspondenceless method that utilizes eigenvalue-based geometric features for automatic extrinsic calibration. Although extrinsic calibration is a different type of problem, this work inspired us to develop the proposed approach.

B. LiDAR (Inertial) Odometry

LiDAR Odometry deals explicitly with the application and adaptation of point cloud registration methods to data from mobile laser scanners in order to estimate the trajectory of the sensor. LiDAR *Inertial* Odometry describes corresponding LiDAR odometry methods that additionally include IMU data to support the estimation. One of the first robust and accurate LiDAR odometry methods with real-time capability was LOAM [12]. Here, plane and edge features are used to estimate the trajectory. There are various extensions to this method, e.g. for planar moving robots [13] or for higher processing speeds [14]. Other LiDAR odometry methods use surface elements (surfels) as features [15], which can represent a point cloud relatively densely in most cases. While these approaches reduce the point clouds to a set

of features, there are also dense methods that do not rely on the presence of specific shapes. Examples are CT-ICP [16] and KISS-ICP [9]. The main advantages of LiDAR inertial odometry, enabled by IMU measurements, are higher robustness in fast motions, stabilization in feature poor locations and the observability of the gravitational direction to stabilize the drift of orientation. Wildcat SLAM [17] extracts surfels from the point clouds and optimizes a continuous trajectory defined by control poses and spline interpolation in a sliding window manner together with the IMU data over several seconds to obtain highly accurate submaps. For each submap, a gravitation direction is determined and submaps are registered relative to each other. LIO-SAM [18] extracts edge and planar features from the point clouds and performs a factor graph optimization along with the IMU data. LiLi-OM [19], which supports both spinning LiDARs and solid-state LiDARs with comparatively small fields of view, also extracts planar and edge features and performs scan-to-map matching. Keyframes are optimized both in a sliding window manner and after loop closure detections including IMU data. FAST-LIO2 [20] and Point-LIO [21] perform scan-to-map matching with the LiDAR point clouds and use Kalman filters for joint state estimation with the IMU data.

C. Multi Scan Registration for Global Optimization

When registering multiple point clouds, the goal is to specifically model and minimize the error between multiple point clouds to obtain a consistent global point cloud. Of course, the previously mentioned procedures for point cloud registration (compare II-A) can also be applied sequentially to point cloud pairs of a larger scene to obtain a merged point cloud. However, this approach can lead to inconsistencies and is not suitable for finding the optimal solution in many cases.

Most of the LiDAR (inertial) odometry methods mentioned above build incremental maps the current scan is aligned to. While some map the environment without optimizing the consistency within the map [20], others perform feature-based optimization only in a sliding time window [12], [17]. After loop closure detections, a pose graph optimization is usually performed. In a pose graph optimization, the poses of keyframes are typically optimized on the basis of pairwise alignments. An early work that uses this principle is [22]. While these types of procedures are very efficient and improve consistency it is also an indirect concept without specifically modeling global consistency. An intuitive approach for multi scan registration is to define an error function that represents the correspondences between all point cloud pairs with overlap and then minimize them. While previous works have studied this approach [23], it is computationally expensive, since the possible number of correspondences grows quadratically with the number of keyframes. An early approach uses range image to optimize a set of planar points from different scans [24]. Other works extract planes from point clouds and perform global optimization based on the planes within the SLAM context [25], [26]. In [27] BALM is presented, an efficient analytical solution of an adapted bundle adjustment for LiDAR based on plane and edge features. Using features such as planes and edges for keyframe optimization can, as shown in [27], improve the accuracy of the resulting trajectory as well as global consistency in many cases. At the same time, when reducing the point clouds to a set of features, information may be lost and there may be corner cases where corresponding features are not available. With our work we present a method that can optimize dense keyframes without a reduction to features. In addition, our approach is efficient because no direct correspondences need to be formed between all the point sets.

III. METHODOLOGY

In this chapter, we describe the proposed approach in detail. First, we explain the basic algorithm. Then, we explain how the DMSA approach can be combined with IMU measurements to obtain accurate and robust LiDAR inertial odometry. Subsequently, we show how the approach can be further used to optimize keyframes, both after loop closures and in a sliding window manner to further increase accuracy.

A. Dense Multi Scan Adjustment (DMSA)

The basic idea behind the DMSA algorithm is a landmark-based graph optimization. Instead of using pre-selected landmark types like planes and edges, the environment is modeled with multivariate normal distributions. More precisely, the space is divided into voxels hierarchically with two levels. For better illustration, only one grid is shown in each of the following sketches. Each voxel containing more than n_{min} points is considered a landmark. It should be noted that due to the two different sized voxel grids, a single LiDAR point may be represented in two landmarks. In the following

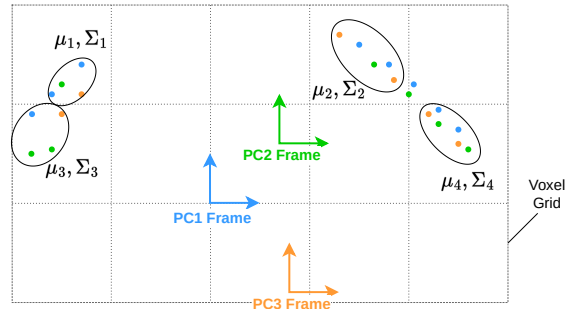


Fig. 3: Exemplary two-dimensional setup with three point clouds. In this case, point cloud 1 (PC1, blue) is used as reference and the relative transformations of point cloud 2 and 3 (PC2 and PC3) to point cloud 1 are optimized. Four voxels contain enough points to be considered as landmark.

we call a set of points, that form a landmark, L_j and the number of points of a landmark $n(L_j)$. The position of the j^{th} landmark L_j is defined by the mean μ_j of all points in the cell and the shape distribution is described by the points covariance Σ_j . An exemplary setup is outlined in Fig. 3.

The target is now to minimize the distances of all measured points $p_{L_j,k}$ to the respective landmark means μ_j of their cells, taking into account the landmark shape distributions. The approach can be summarized with formula 1:

$$\min\left\{\sum_{j=1}^{n_L} \frac{1}{n(L_j)} \sum_{k=1}^{n(L_j)} (p_{L_j,k} - \mu_j)^T \Sigma_j^{-1} (p_{L_j,k} - \mu_j)\right\} \quad (1)$$

Weighting with the inverse number of points per landmark prevents overfitting of locations with high point densities. At this point, the coupling of the error weighting by the inverse covariance matrix Σ_j^{-1} with the errors to be minimized should be noted. The smaller the distances of the points $p_{L_j,k}$ to the center of the respective landmarks μ_j become, the stronger the errors are subsequently weighted. The described error function is minimized iteratively using Levenberg-Marquardt [28] optimization steps. During each optimization step, the covariance matrices of the landmarks are assumed to be constant and the errors of the points to their respective landmarks are minimized. Here landmarks are not distinguished between inliers and outliers.

B. DMSA LiDAR Inertial Odometry

We have developed a LiDAR inertial odometry system based on the previously described DMSA algorithm. Fig. 4 gives an overview of the structure of the approach. New scans are first adaptively downsampled (see section III-B.1) and then stored in a ring buffer in the same manner as IMU measurements. After the arrival of a new scan, all points within a sliding time window are extracted from the ring buffer. The trajectory is parameterized by control poses within the time window. To support the subsequent optimization, IMU measurements are preintegrated as proposed in [29] between two control poses. Then, a continuous trajectory is optimized based on the points within the time window, the preintegrated

Algorithm 1 The DMSA Algorithm

Input: n_P point clouds $P_{s,i}$ in sensor frame with initial transforms set $\mathbf{T}_{s,\text{init}}^w = (T_{s,1}^w, \dots, T_{s,n_P}^w)$ with $T_{s,i}^w = (r_1, r_2, r_3, x, y, z), i \in \{1, \dots, n_P\}$ from sensor frame (index s) to reference frame (index w)

- 1: $\mathbf{T}_s^w = \mathbf{T}_{s,\text{init}}^w$
- 2: **while** not converged **do**
- 3: Transform point clouds to a common reference frame
- 4: Split all points in the reference frame into voxels twice, once with a coarse grid size and once with a fine grid size.
- 5: Remove voxels that contain less than n_{min} points
- 6: Init error vector $E \in \mathbb{R}^{n(L_j) \times 1}$
- 7: **for all** voxels with their respective point sets L_j **do**
- 8: Calculate mean μ_j and covariance Σ_j of points
- 9: Calculate weight $w_j = \frac{1}{n(L_j)}$
- 10: **for all** points $p_k, k \in \{1, \dots, n(L_j)\}$ in L_j **do**
- 11: $E(j) += w_j(p_k - \mu_j)\Sigma_j^{-1}(p_k - \mu_j)$
- 12: **end for**
- 13: **end for**
- 14: Calculate numeric Jacobian $\mathbf{J}_E(\mathbf{T}_s^w)$ while keeping the covariances unchanged
- 15: Update \mathbf{T}_s^w with Levenberg–Marquardt algorithm [28]
- 16: **end while**
- 17: **return** \mathbf{T}_s^w

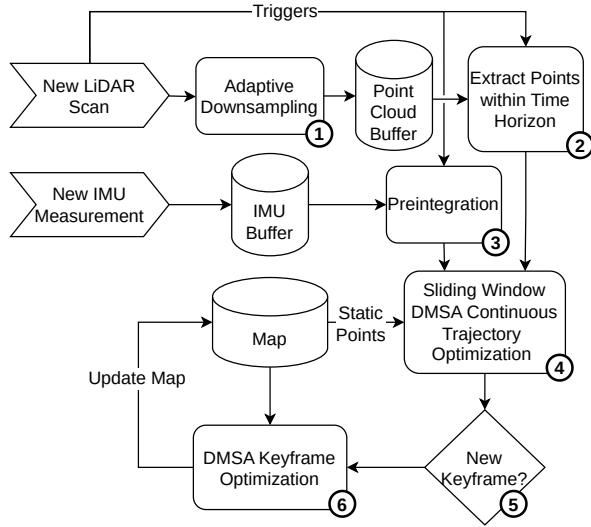


Fig. 4: DMSA LiDAR Inertial Odometry overview. Numbers indicate the processing order.

IMU measurements, and static points derived from the map. This process is the core of DMSA LiDAR Inertial Odometry and is described in more detail in section III-B.2.

1) *Adaptive Downsampling:* To reduce the number of points we use a grid random filter. Here, the point cloud is divided into uniform grid cells. Then one point is randomly selected from each grid cell. The downsampled point cloud is the set of all selected points and the filtered point count

corresponds to the set of grid cells containing points. We decided to use an adaptive grid size to be able to use the system in different environments without adjustments. The system automatically switches between four different resolution levels. The goal is to reach a minimum number of points $N_{\text{min_pts}}$. The system starts with the coarsest resolution level. If the number of points after downsampling is smaller than $N_{\text{min_pts}}$, the next finer resolution is used. If the number of points greater or equal to $N_{\text{min_pts}}$, the corresponding resolution is used. After downsampling, the most distant points are step by step removed until the number of points $N_{\text{min_pts}}$ is reached or the current point falls below a distance threshold.

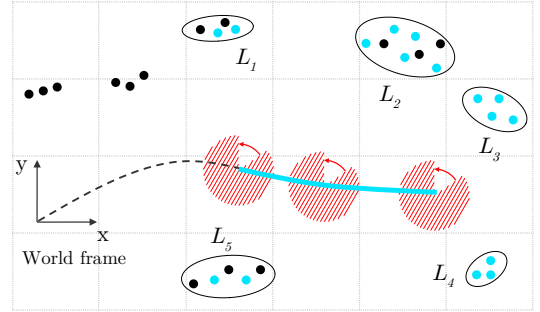


Fig. 5: Outline of the sliding window continuous trajectory optimization. The teal line represents the continuous trajectory within the sliding time window to be optimized. Teal dots symbolize LiDAR points acquired within the time window. Black dots represent *Static Points* from the map. The points belonging to the landmarks L_1 to L_5 are marked with black ellipses.

2) *Sliding Window DMSA Continuous Trajectory Optimization:* In this step, a continuous trajectory of a sliding time window of duration d_{window} is optimized. The modeling of the trajectory was inspired by [17]. The trajectory is represented by a set of control poses uniformly distributed in time. Cubic Hermitian spline interpolation is performed between the control poses for the positions to obtain a temporal resolution of 1 ms. Between the orientations of the control poses spherical linear interpolation is used. The parameters of the control poses are then optimized using the DMSA algorithm with two extensions (see section III-A). In addition to the downsampled points within the sliding time window, *Static Points* are extracted from the map and added as fixed points within DMSA optimization. Furthermore, preintegrated IMU measurements (compare [29]) between the control poses are added as a cost term at the Levenberg–Marquardt [28] optimization step.

3) *Keyframe Selection and Representation:* New keyframes are selected based on the overlap with the map and the distance to the nearest keyframe (not the last). Using the distance to the nearest keyframe as a condition results in fewer redundancies in the global point cloud, what enables faster optimization. If a certain location is already densely mapped, the system implicitly switches to a localization

TABLE I: Results of our experiments: APE w.r.t. translation part in meters (with SE(3) Umeyama alignment) using [30] of selected sequences from the Hilti-Oxford Dataset [5], the Newer College Dataset [31] and own data recordings. The lowest rmse value and lowest maximum error of each sequence are bold. All procedures were performed with the same functional parameter settings for all sequences. Each approach had identical functional parameter settings for all sequences.

| Dataset | Sequence | Duration [s] | Length [m] | FAST-LIO2 [20] | | LiLi-OM [19] | | KISS-ICP [9] | | DMSA-LIO | |
|---------------|--------------------|--------------|------------|------------------|--------------|-------------------|-------|------------------|---------|--------------|--------------|
| | | | | rmse | max | rmse | max | rmse | max | rmse | max |
| Hilti-Oxford | exp04 Construction | 125 | 44.8 | 0.024 | 0.091 | N/A | | 5.217 | 11.618 | 0.022 | 0.070 |
| | exp06 Construction | 150 | 54.2 | 0.049 | 0.258 | N/A | | 69.961 | 263.500 | 0.047 | 0.228 |
| | exp14 Basement | 74 | 24.9 | 0.200 | 0.369 | N/A | | fails after 9 s | | 0.068 | 0.158 |
| | exp18 Corridor | 109 | 33.6 | fails after 87 s | | N/A | | fails after 5 s | | 0.378 | 0.732 |
| Newer College | Parkland Mound | 501 | 443.2 | 31.242 | 45.524 | fails after 52 s | | 0.254 | 1.384 | 0.170 | 1.365 |
| | Stairs | 118 | 38.6 | fails after 38 s | | fails after 31 s | | fails after 11 s | | 0.047 | 0.109 |
| | Cloister | 279 | 288.5 | 0.077 | 0.315 | 0.766 | 1.958 | 0.632 | 3.338 | 0.052 | 0.144 |
| Own data | Bicycle Street | 1000 | 1907.7 | 5.870 | 9.588 | fails after 104 s | | 10.241 | 17.248 | 15.986 | 24.060 |

mode, since no new keyframes are generated. Downsampled point clouds from sliding time window optimization are used as keyframes. The planarity [32] is then calculated for each point of the downsampled point cloud as well as a normal vector based on their local neighborhood. The normal vectors are used once to remove non-visible points from the *Static Points*, which are included in the sliding window optimization (compare section III-B.2). In addition, the normal vectors are used in the subsequent keyframe optimization to split landmark point sets with opposite normal vectors. For each keyframe, a gravity direction is estimated based on the estimated trajectory and IMU measurements within the optimized trajectory period. This gravity direction is included in the subsequent keyframe optimization.

4) *DMSA Keyframe Optimization*: Adding a new keyframe triggers keyframe optimization. The first step is to analyze which part of the map should be optimized. To obtain a consistent map, the oldest keyframe is identified, whose distance to the current keyframe is smaller than a threshold value and whose overlap with the current keyframe is larger than a second threshold value. All keyframes that lie temporally between the identified oldest keyframe and the current keyframe are then optimized. For keyframe optimization, the DMSA approach presented in section III-A is used with two modifications. The first modification is that the planarity and the normal vectors of points within a landmark point set are analyzed. Here, planar point sets with opposing normal vectors are split into two sets based on the directions of the normal vectors, which are then processed as separate landmarks. The second modification is that an additional error for the gravitation term is added during the optimization. This error term ensures that the gravitational directions estimated per keyframe point in the same direction in the global frame.

IV. EXPERIMENTS

To investigate the accuracy and robustness of our approach, we conducted experiments with sequences from the Hilti-Oxford Dataset [5], the Newer College Dataset [31]

and own datasets. In selecting the sequences, the goal was to cover a wide range of conditions. Our selection includes construction sites, narrow staircases, long corridors, park landscapes and streets with dynamic objects. An important boundary condition for the selection was the presence of ground truth poses. While the Hilti-Oxford and the Newer College datasets contain groundtruth poses in centimeter range, we use a GNSS-based inertial navigation system as groundtruth for the evaluation of our own recording. Due to the length of the sequence, it is assumed here that the accuracy of the GNSS-based inertial navigation system is sufficient. In total, the sequences include data from three different LiDAR scanner types: Hesai PandarXT-32, Ouster OS1-64 and Ouster OS0-128.

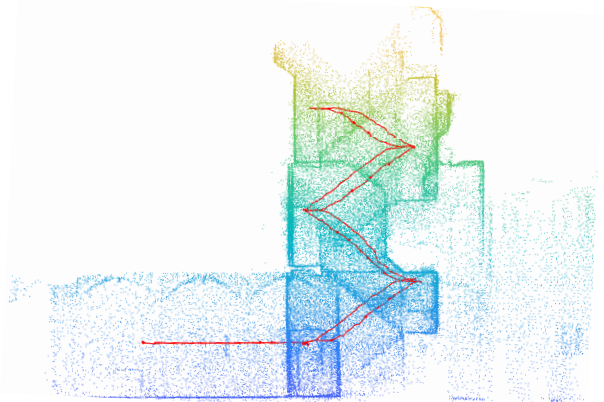


Fig. 6: Resulting sparse keyframe point cloud of sequence *Stairs* of the Newer College Dataset [31]. The estimated trajectory is marked with a red line and keyframe positions are outlined with red dots.

A. Selection of Comparing Approaches and Configuration

For comparison, we also processed the sequences using other open source state of the art LiDAR-SLAM approaches. Here we selected FAST-LIO2 [20], LiLi-OM [19] and KISS-ICP [9]. While the first two combine LiDAR and IMU

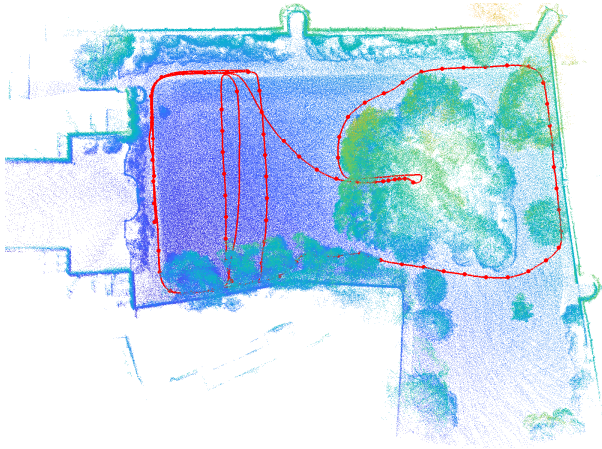


Fig. 7: Resulting sparse keyframe point cloud of sequence *Parkland Mound* of the Newer College Dataset [31]. The estimated trajectory is marked with a red line and keyframe positions are outlined with red dots.

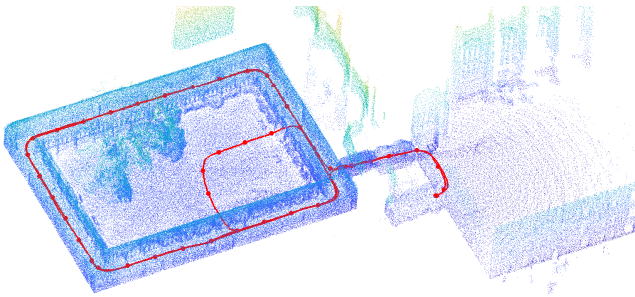


Fig. 8: Resulting sparse keyframe point cloud of sequence *Cloister* of the Newer College Dataset [31]. The estimated trajectory is marked with a red line and keyframe positions are outlined with red dots.

data like our approach, KISS-ICP [9] uses only the LiDAR data. We did not perform parameter optimization for any of the selected methods. Instead, we used the default settings provided. For LiLi-OM [19], in addition to the rostopics and extrinsics, an adjustment had to be made in the source code to generate the correct LiDAR ring IDs during preprocessing. For the sensor setup of the Hilti-Oxford Dataset [5], we could not find a successful adaptation, so we noted corresponding values in Table I with N/A. In order to have a reasonable comparison, we also used for all sequences identical functional parameter settings in our approach.

B. Processing Speed

The processing time per sequence of DMSA-LIO varied between 1.5 and 4 times the recording time using a 12th Gen Intel Core i9-12900K. For indoor scenes the processing speed is significantly higher than for outdoor scenes. The current bottle neck is the numerical calculation of the Jacobian (see algorithm 1).

C. Results

Fig. 2 shows the keyframe point cloud and the estimated trajectory of our approach for the *Bicycle Street* sequence. Despite the challenges due to dynamic objects and low structure locations, our method could be successfully applied here. Fig. 6, 7, and 8 show our results of the selected sequences from the Newer College Dataset [31]. Like the sequence *exp14 Basement* from the Hilti-Oxford Dataset [5] shown in Fig. 1, these also include narrow spaces as a challenge. A quantitative evaluation of the results is provided in table I. Fig. 9 shows the estimated trajectories of the methods in the xy plane for the *Cloister* sequence, where reasonable estimates were available from all approaches. Our method was able to provide a reasonable estimate for all selected sequences in the experiments. In contrast, FAST-LIO [20] failed in two sequences, LiLi-OM [19] failed in three sequences and KISS-ICP [9] also failed in three sequences. In addition, our approach was able to achieve the lowest rmse as well as lowest maximum error in seven from eight sequences.

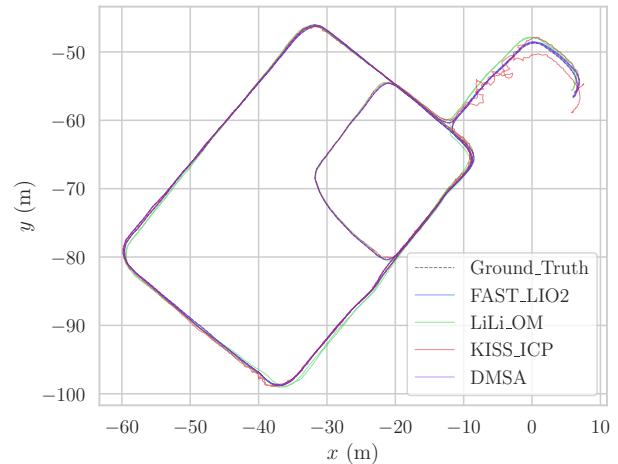


Fig. 9: Estimated trajectories and ground truth for the *Cloister* Sequence (with SE(3) Umeyama alignment) using [30].

V. CONCLUSIONS

This article presents a new method for dense multi scan adjustment. We also propose an extension of the method for application to continuous trajectories and show how this enables robust and accurate trajectory estimation. In the future, we aim to optimize the computational speed and further increase the accuracy. In addition, we also want to investigate a version of the method that works without an IMU.

ACKNOWLEDGMENT

We would like to thank the German Federal Ministry for Economic Affairs and Climate Action for funding our project through the ZIM funding program (identifier KK5375601GR1). We would also like to thank the company IGI mbH for the successful cooperation and the recording of sensor data.

REFERENCES

- [1] P. Biber and W. Straßer, “The normal distributions transform: A new approach to laser scan matching,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*(Cat. No. 03CH37453), vol. 3. IEEE, 2003, pp. 2743–2748.
- [2] M. Magnusson, “The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection,” Ph.D. dissertation, Örebro universitet, 2009.
- [3] A. Myronenko and X. Song, “Point set registration: Coherent point drift,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.
- [4] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, “Litamin2: Ultra light lidar-based slam using geometric approximation applied with kl-divergence,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 619–11 625.
- [5] L. Zhang, M. Helmberger, L. F. T. Fu, D. Wisth, M. Camurri, D. Scaramuzza, and M. Fallon, “Hilti-oxford dataset: A millimeter-accurate benchmark for simultaneous localization and mapping,” *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 408–415, 2023.
- [6] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.
- [7] K.-L. Low, “Linear least-squares optimization for point-to-plane icp surface registration,” *Chapel Hill, University of North Carolina*, vol. 4, no. 10, pp. 1–3, 2004.
- [8] A. Segal, D. Haehnel, and S. Thrun, “Generalized-icp,” in *Robotics: Science and Systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.
- [9] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, “KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 2, pp. 1029–1036, 2023.
- [10] N. Chebrou, T. Låbe, O. Vysotska, J. Behley, and C. Stachniss, “Adaptive Robust Kernels for Non-Linear Least Squares Problems,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, pp. 2240–2247, 2021.
- [11] M. Hillemann, M. Weinmann, M. S. Mueller, and B. Jutzi, “Automatic extrinsic self-calibration of mobile mapping systems based on geometric 3d features,” *Remote Sensing*, vol. 11, no. 16, p. 1955, 2019.
- [12] J. Zhang and S. Singh, “Loam: Lidar odometry and mapping in real-time,” in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.
- [13] T. Shan and B. Englot, “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.
- [14] H. Wang, C. Wang, C.-L. Chen, and L. Xie, “F-loam: Fast lidar odometry and mapping,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4390–4396.
- [15] J. Behley and C. Stachniss, “Efficient surfel-based slam using 3d laser range data in urban environments,” in *Robotics: Science and Systems*, vol. 2018, 2018, p. 59.
- [16] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, “Ct-icp: Real-time elastic lidar odometry with loop closure,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5580–5586.
- [17] M. Ramezani, K. Khosoussi, G. Catt, P. Moghadam, J. Williams, P. Borges, F. Pauling, and N. Kottege, “Wildcat: Online continuous-time 3d lidar-inertial slam,” *arXiv preprint arXiv:2205.12595*, 2022.
- [18] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, “Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping,” in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.
- [19] K. Li, M. Li, and U. D. Hanebeck, “Towards high-performance solid-state-lidar-inertial odometry and mapping,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5167–5174, 2021.
- [20] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “Fast-lio2: Fast direct lidar-inertial odometry,” *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [21] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, “Point-lio: Robust high-bandwidth light detection and ranging inertial odometry,” *Advanced Intelligent Systems*, p. 2200459, 2023.
- [22] F. Lu and E. Milios, “Globally consistent range scan alignment for environment mapping,” *Autonomous robots*, vol. 4, pp. 333–349, 1997.
- [23] R. Benjema and F. Schmitt, “A solution for the registration of multiple 3d point sets using unit quaternions,” in *Computer Vision—ECCV’98: 5th European Conference on Computer Vision Freiburg, Germany, June 2–6, 1998 Proceedings, Volume II 5*. Springer, 1998, pp. 34–50.
- [24] Y. Chen and G. Medioni, “Object modeling by registration of multiple range images,” in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 2724–2729 vol.3.
- [25] M. Kaess, “Simultaneous localization and mapping with infinite planes,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4605–4611.
- [26] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, “Keyframe-based dense planar slam,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Ieee, 2017, pp. 5110–5117.
- [27] Z. Liu and F. Zhang, “Balm: Bundle adjustment for lidar mapping,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3184–3191, 2021.
- [28] J. J. Moré, “The levenberg-marquardt algorithm: implementation and theory,” in *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977*. Springer, 2006, pp. 105–116.
- [29] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-manifold preintegration for real-time visual–inertial odometry,” *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.
- [30] M. Grupp, “evo: Python package for the evaluation of odometry and slam.” <https://github.com/MichaelGrupp/evo>, 2017.
- [31] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, “The newer college dataset: Handheld lidar, inertial and vision with ground truth,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4353–4360.
- [32] K. F. West, B. N. Webb, J. R. Lersch, S. Pothier, J. M. Triscari, and A. E. Iverson, “Context-driven automated target detection in 3d data,” in *Automatic Target Recognition XIV*, vol. 5426. SPIE, 2004, pp. 133–143.