

MULLS: Versatile LiDAR SLAM via Multi-metric Linear Least Square

Yue Pan¹, Pengchuan Xiao², Yujie He³, Zhenlei Shao² and Zesong Li²

Abstract— The rapid development of autonomous driving and mobile mapping calls for off-the-shelf LiDAR SLAM solutions that are adaptive to LiDARs of different specifications on various complex scenarios. To this end, we propose MULLS, an efficient, low-drift, and versatile 3D LiDAR SLAM system. For the front-end, roughly classified feature points (ground, facade, pillar, beam, etc.) are extracted from each frame using dual-threshold ground filtering and principal components analysis. Then the registration between the current frame and the local submap is accomplished efficiently by the proposed multi-metric linear least square iterative closest point algorithm. Point-to-point (plane, line) error metrics within each point class are jointly optimized with a linear approximation to estimate the ego-motion. Static feature points of the registered frame are appended into the local map to keep it updated. For the back-end, hierarchical pose graph optimization is conducted among regularly stored history submaps to reduce the drift resulting from dead reckoning. Extensive experiments are carried out on three datasets with more than 100,000 frames collected by seven types of LiDAR on various outdoor and indoor scenarios. On the KITTI benchmark, MULLS ranks among the top LiDAR-only SLAM systems with real-time performance.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) plays a key role in robotics tasks, including robot navigation [1], field surveying [2] and high-definition map production [3] for autonomous driving. Compared with vision-based [4]–[6] and RGB-D-based [7]–[9] SLAM systems, LiDAR SLAM systems are more reliable under various lighting conditions and are more suitable for tasks demanding for dense 3D map.

The past decade has witnessed enormous development in LiDAR odometry (LO) [10]–[28], LiDAR-based loop closure detection [29]–[36] and pose optimization [37]–[42]. Though state-of-the-art LiDAR-based approaches performs well in structured urban or indoor scenes with the localization accuracy better than most of the vision-based methods, the lack of versatility hinders them from being widely used in the industry. Most of methods are based on sensor dependent point cloud representations such as, ring [10], [20] and range image [14]–[18], which requires detailed LiDAR model parameters such as, scan-line distribution and the field of view. It's non-trivial to directly adapt them on recently developed mechanical LiDARs with unique configuration of beams and solid-state LiDARs with limited field of view [21]. Besides, some SLAM systems target on specific application scenarios

¹Y.Pan is with Department of Civil, Environmental and Geomatic Engineering, ETH Zurich, Switzerland yuepan@ethz.ch

²P.Xiao, Z.Shao and Z.Li are with Hesai Technology Co., Ltd, Shanghai, China {xiaopengchuan.intern, shaozhenlei, lizesong}@hesaitech.com

³Y.He is with the School of Engineering, EPFL, Lausanne, Switzerland yujie.he@epfl.ch

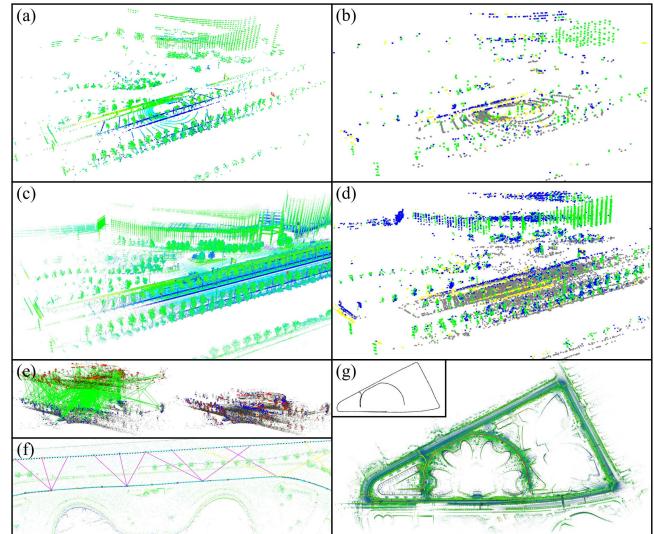


Fig. 1. Overview of the proposed MULLS-SLAM system: (a) the point cloud from a single scan of a multi-beam spinning LiDAR, (b) various types of feature points (ground, facade, pillar, beam) extracted from the scan, (c) registered point cloud of the local map, (d) feature points of the local map, (e) global registration between revisited submaps by TEASER [46], (f) loop closure edges among submaps, (g) generated consistent map and trajectory.

such as, urban road [43], highway [44], tunnel [45] and forest [22] using the ad-hoc framework. They tend to get stuck in degenerated corner-cases or a rapid switch of the scene.

In this work, we proposed a LiDAR-only SLAM system (as shown in Fig. 1) to cope with the aforementioned challenges. The continuous inputs of the system are sets of 3D coordinates (optionally with point-wise intensity and timestamp) without the conversion to rings or range images. Therefore the system is independent of the LiDAR's specification and without the loss of 3D structure information. Geometric feature points with plentiful categories such as ground, facade, pillar, beam, and roof are extracted from each frame, giving rise to better adaptability to the surroundings. The ego-motion is estimated efficiently by the proposed multi-metric linear least square (MULLS) iterative closest points (ICP) algorithm based on the classified feature points. Furthermore, a submap-based pose graph optimization (PGO) is employed as the back-end with loop closure constraints constructed via map-to-map global registration.

The main contributions of this work are listed as follows:

- A scan-line independent LiDAR-only SLAM solution named MULLS¹, with low drift and real-time performance on various scenarios. Currently, MULLS ranks top 10 on the competitive KITTI benchmark².

¹<https://github.com/YuePanEdward/MULLS>

²http://www.cvlibs.net/datasets/kitti/eval_odometry.php

- An efficient point cloud local registration algorithm named MULLS-ICP that realizes the linear least square optimization of point-to-point (plane, line) error metrics jointly in roughly classified geometric feature points.

II. RELATED WORKS

As the vital operation of LiDAR SLAM, point cloud registration can be categorized into local and global registration. Local registration requires good transformation initial guess to finely align two overlapping scans without stuck in the local minima while global registration can coarsely align them regardless of their relative pose.

Local registration based on ICP [47] has been widely used in LO to estimate the relatively small scan-to-scan(map) transformation. Classic ICP keeps alternately solving for dense point-to-point closest correspondences and rigid transformation. To guarantee LO's real-time performance (mostly at 10Hz) on real-life regularized scenarios, faster and more robust variants of ICP have been proposed in recent years focusing on the sampled points for matching and the error metrics [48]–[51] for optimization. As a pioneering work of sparse point-based LO, LOAM [10] selects the edge and planar points by sorting the curvature on each scan-line. The squared sum of point-to-line(plane) distance is minimized by non-linear optimization to estimate the transformation. As a follow-up, LeGO-LOAM [14] conducts ground segmentation and isotropic edge points extraction from the projected range image. Then a two-step non-linear optimization is executed on ground and edge correspondences to solve two sets of transformation coefficients successively. Unlike LOAM, SuMa [15] is based on the dense projective normal ICP between the range image of the current scan and the surfel map. SuMa++ [16] further improved SuMa by realizing dynamic filtering and multi-class projective matching with the assistance of semantic masks. These methods share the following limits. First, they require the LiDAR's model and lose 3D information due to the operation based on range image or scan-line. [11] [26] operate based on raw point cloud but lose efficiency. Second, ICP with different error metrics is solved by less efficient non-linear optimization. Though efficient linear least square optimization for point-to-plane metric has been applied on LO [52]–[54], the joint linear solution to point-to-point (plane, line) metrics has not been proposed. These limits are solved in our work.

Global registration is often required for the constraint construction in LiDAR SLAM's back-end, especially when the odometry drifts too much to close the loop by ICP. One common solution is to apply correlation operations on global features [31]–[35] encoded from a pair of scans to estimate relative azimuth. Other solutions solve the 6DOF transformation based on local feature matching and verification among the keypoints [55], [56] or segments [30]. Our system follows these solutions by adopting the recently proposed TEASER [46] algorithm. By applying truncated least square estimation with semi-definite relaxation [57], TEASER is more efficient and robust than methods based on RANSAC [55], [58] and branch & bound (BnB) [59], [60].

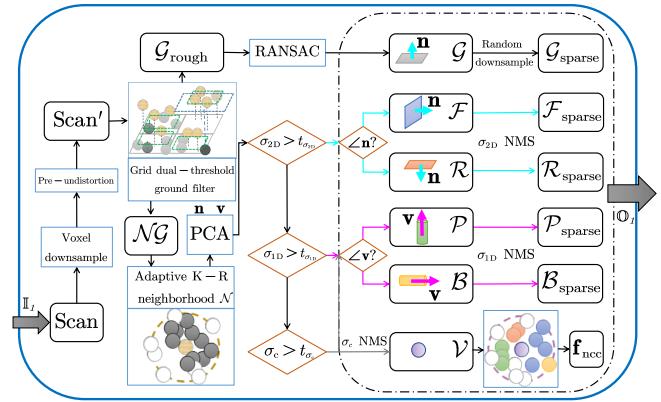


Fig. 2. Pipeline of geometric feature points extraction and encoding

III. METHODOLOGY

A. Motion compensation

Given the point-wise timestamp of a frame, the time ratio for a point \mathbf{p}_i with timestamp τ_i is $s_i = \frac{\tau_e - \tau_i}{\tau_e - \tau_b}$, where τ_b , τ_e are the timestamp at the start and the end of the frame. When IMU is not available, the transformation of \mathbf{p}_i to \mathbf{p}_i^e can be computed under the assumption of uniform motion:

$$\mathbf{p}_i^{(e)} = \mathbf{R}_{e,i}\mathbf{p}_i + \mathbf{t}_{e,i} \approx \text{slerp}(\mathbf{R}_{e,b}, s)\mathbf{p}_i + s\mathbf{t}_{e,b}, \quad (1)$$

where slerp represents the spherical linear interpolation, $\mathbf{R}_{e,b}$ and $\mathbf{t}_{e,b}$ stands for the rotation and translation estimation throughout the frame. The undistorted point cloud is achieved by packing all the points $\{\mathbf{p}_i^e\}$ at the end of the frame.

B. Geometric feature points extraction and encoding

The workflow of this section is summarized in Fig. 2. The input is the raw point cloud of each frame, and the outputs are six types of feature points with primary and normal vectors.

1) **Dual-threshold ground filtering:** The point cloud after optional preprocessing is projected to a reference plane, which is horizontal or fitted from last frame's ground points. For non-horizontal LiDAR, the initial orientation needs to be known. The reference plane is then divided into equal-size 2D grids. The minimum point height in each grid g_i and in its 3×3 neighboring grids, denoted as $h_{\min}^{(i)}$ and $h_{\text{neimin}}^{(i)}$, are recorded respectively. With two thresholds δh_1 , δh_2 , each point p_k in grid g_i is classified as a roughly determined ground point G_{rough} or a nonground point NG by:

$$p_k^{(i)} \in \begin{cases} NG, & \text{if } h_k - h_{\min}^{(i)} > \delta h_1 \text{ or } h_{\min}^{(i)} - h_{\text{neimin}}^{(i)} > \delta h_2 \\ G_{\text{rough}}, & \text{otherwise} \end{cases}. \quad (2)$$

To refine the ground points, RANSAC is employed in each grid to fit the grid-wise ground plane. Inliers are kept as ground points G and their normal vectors n are defined as the surface normal of the grid-wise best-fit planes.

2) **Nonground points classification based on PCA:** Non-ground points are downsampled to a fixed number and then fed into the principal components analysis (PCA) module in parallel. The point-wise K-R neighborhood \mathcal{N} is defined as the nearest K points within a sphere of radius R . The covariance matrix C for \mathcal{N} is calculated as $C = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} (\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{p}_i - \bar{\mathbf{p}})^T$, where $\bar{\mathbf{p}}$ is the center

of gravity for \mathcal{N} . Next, eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$ and the corresponding eigenvectors $\mathbf{v}, \mathbf{m}, \mathbf{n}$ are determined by the eigenvalue decomposition of \mathbf{C} , where \mathbf{v}, \mathbf{n} are the primary and the normal vector of \mathcal{N} . Then the local linearity σ_{1D} , planarity σ_{2D} , and curvature σ_c [61] are defined as $\sigma_{1D} = \frac{\lambda_1 - \lambda_2}{\lambda_1}$, $\sigma_{2D} = \frac{\lambda_2 - \lambda_3}{\lambda_1}$, $\sigma_c = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$.

According to the magnitude of local feature $\sigma_{1D}, \sigma_{2D}, \sigma_c$ and the direction of \mathbf{v}, \mathbf{n} , five categories of feature points can be extracted, namely facade \mathcal{F} , roof \mathcal{R} , pillar \mathcal{P} , beam \mathcal{B} , and vertex \mathcal{V} . To refine the feature points, non-maximum suppression (NMS) based on $\sigma_{1D}, \sigma_{2D}, \sigma_c$ are applied on linear points (\mathcal{P}, \mathcal{B}), planar points (\mathcal{F}, \mathcal{R}) and vertices \mathcal{V} respectively, followed by an isotropic downsampling. Together with \mathcal{G} , the roughly classified feature points are packed for registration.

3) Neighborhood category context encoding: Based on the extracted feature points, the neighborhood category context (NCC) is proposed to describe each vertex keypoint with almost no other computations roughly. As shown in (3), the proportion of feature points with different categories in the neighborhood, the normalized intensity and height above ground are encoded. NCC is later used as the local feature for global registration in the SLAM system's back-end (III-E):

$$\mathbf{f}_i^{(ncc)} = \left[\frac{|\mathcal{F}_N|}{|\mathcal{N}|} \quad \frac{|\mathcal{P}_N|}{|\mathcal{N}|} \quad \frac{|\mathcal{B}_N|}{|\mathcal{N}|} \quad \frac{|\mathcal{R}_N|}{|\mathcal{N}|} \quad \frac{\bar{I}_N}{I_{\max}} \quad \frac{h_g}{h_{\max}} \right]_i^T. \quad (3)$$

C. Multi-metric linear least square ICP

This section's pipeline is summarized in Fig. 3. The inputs are the source and the target point cloud composed of multi-class feature points extracted in III-B, as well as the initial guess of the transformation from the source point cloud to the target point cloud $\mathbf{T}_{t,s}^{\text{guess}}$. After the iterative procedure of ICP [47], the outputs are the final transformation estimation $\mathbf{T}_{t,s}$ and its accuracy evaluation indexes.

1) Multi-class closest point association: For each iteration, the closest point correspondences are determined by nearest neighbor (NN) searching within each feature point category ($\mathcal{G}, \mathcal{F}, \mathcal{R}, \mathcal{P}, \mathcal{B}, \mathcal{V}$) under an ever-decreasing distance threshold. Direction consistency check of the normal vector \mathbf{n} and the primary vector \mathbf{v} are further applied on the planar points ($\mathcal{G}, \mathcal{F}, \mathcal{R}$) and the linear points (\mathcal{P}, \mathcal{B}) respectively. Constraints on category, distance and direction make the point association more robust and efficient.

2) Multi-metric transformation estimation: Suppose $\mathbf{q}_i, \mathbf{p}_i$ are corresponding points in the source point cloud and the target point cloud, the residual \mathbf{r}_i after certain rotation \mathbf{R} and translation \mathbf{t} of the source point cloud is defined as:

$$\mathbf{r}_i = \mathbf{q}_i - \mathbf{p}'_i = \mathbf{q}_i - (\mathbf{R}\mathbf{p}_i + \mathbf{t}). \quad (4)$$

With the multi-class correspondences, we expect to estimate the optimal transformation $\{\mathbf{R}^*, \mathbf{t}^*\}$ that jointly minimized the point-to-point, point-to-plane and point-to-line distance between the vertex \mathcal{V} , planar ($\mathcal{G}, \mathcal{F}, \mathcal{R}$) and linear (\mathcal{P}, \mathcal{B}) point correspondences. As shown in Fig. 4, the point-to-point (plane, line) distance can be calculated as $d_i^{\text{po} \rightarrow \text{po}} = \|\mathbf{r}_i\|_2$, $d_j^{\text{po} \rightarrow \text{pl}} = \mathbf{r}_j \cdot \mathbf{n}_j$, and $d_k^{\text{po} \rightarrow \text{li}} = \|\mathbf{r}_k \times \mathbf{v}_k\|_2$ respectively from the residual, normal and primary vector ($\mathbf{r}, \mathbf{n}, \mathbf{v}$). Thus, the transformation estimation is formulated

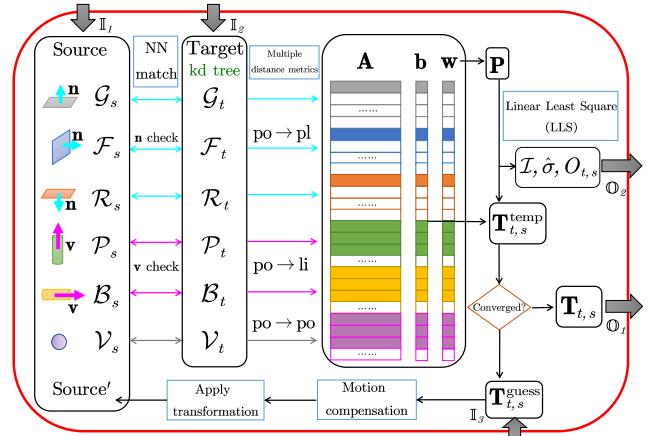
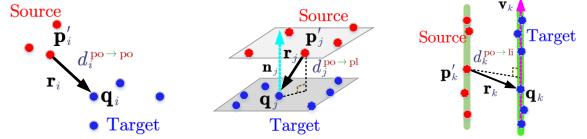


Fig. 3. Pipeline of the multi-metric linear least square ICP



(a) point to point (b) point to plane (c) point to line
Fig. 4. Overview of three different distance error metrics

as a weighted least square optimization problem with the following target function:

$$\{\mathbf{R}^*, \mathbf{t}^*\} = \underset{\{\mathbf{R}, \mathbf{t}\}}{\operatorname{argmin}} \sum_i w_i (d_i^{\text{po} \rightarrow \text{po}})^2 + \sum_j w_j (d_j^{\text{po} \rightarrow \text{pl}})^2 + \sum_k w_k (d_k^{\text{po} \rightarrow \text{li}})^2, \quad (5)$$

where w is the weight for each correspondence. Under the tiny angle assumption, \mathbf{R} can be approximated as:

$$\mathbf{R} \approx \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}_x + \mathbf{I}_3, \quad (6)$$

where α, β and γ are roll, pitch and yaw, respectively under $x - y' - z''$ Euler angle convention. Regarding the unknown parameter vector as $\xi = [t_x \ t_y \ t_z \ \alpha \ \beta \ \gamma]^T$, (5) becomes a weighted linear least square problem. It is solved by Gauss-Markov parameter estimation with the functional model $\mathbf{e} = \mathbf{A}\xi - \mathbf{b}$ and the stochastic model $\mathbf{D}\{\mathbf{e}\} = \sigma_0^2 \mathbf{P}^{-1}$, where σ_0 is a prior standard deviation, and $\mathbf{P} = \operatorname{diag}(w_1, \dots, w_n)$ is the weight matrix. Deduced from (5), the overall design matrix $\mathbf{A} \in \mathbb{R}^{n \times 6}$ and observation vector $\mathbf{b} \in \mathbb{R}^{n \times 1}$ is formulated as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_i^{\text{po} \rightarrow \text{po}^T} & \dots & \mathbf{A}_j^{\text{po} \rightarrow \text{pl}^T} & \dots & \mathbf{A}_k^{\text{po} \rightarrow \text{li}^T} & \dots \end{bmatrix}^T, \quad (7)$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_i^{\text{po} \rightarrow \text{po}^T} & \dots & \mathbf{b}_j^{\text{po} \rightarrow \text{pl}^T} & \dots & \mathbf{b}_k^{\text{po} \rightarrow \text{li}^T} & \dots \end{bmatrix}^T,$$

where the components of \mathbf{A} and \mathbf{b} for each correspondence under point-to-point (plane, line) error metric are defined as:

$$\begin{aligned} \mathbf{A}_i^{\text{po} \rightarrow \text{po}} &= [\mathbf{I}_3 \ [\mathbf{p}_i]_x] & \mathbf{b}_i^{\text{po} \rightarrow \text{po}} &= \mathbf{q}_i - \mathbf{p}_i \\ \mathbf{A}_j^{\text{po} \rightarrow \text{pl}} &= [\mathbf{n}_j^T \ (\mathbf{p}_j \times \mathbf{n}_j)^T] & \mathbf{b}_j^{\text{po} \rightarrow \text{pl}} &= \mathbf{n}_j^T (\mathbf{q}_j - \mathbf{p}_j) \\ \mathbf{A}_k^{\text{po} \rightarrow \text{li}} &= [[\mathbf{v}_k]_x \ (\mathbf{v}_k^T \mathbf{p}_k) \ \mathbf{I}_3 - \mathbf{p}_k \mathbf{v}_k^T] & \mathbf{b}_k^{\text{po} \rightarrow \text{li}} &= \mathbf{v}_k \times (\mathbf{q}_k - \mathbf{p}_k) \end{aligned} \quad (8)$$

Therefore, the estimation of unknown vector $\hat{\xi}$ is:

$$\begin{aligned}\hat{\xi} = \underset{\xi}{\operatorname{argmin}} (\mathbf{A}\xi - \mathbf{b})^T \mathbf{P} (\mathbf{A}\xi - \mathbf{b}) &= (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{P} \mathbf{b}) \\ &= \left(\sum_{i=1}^n \mathbf{A}_i^T \mathbf{w}_i \mathbf{A}_i \right)^{-1} \sum_{i=1}^n \mathbf{A}_i^T \mathbf{w}_i \mathbf{b}_i\end{aligned}, \quad (9)$$

where both $\mathbf{A}^T \mathbf{P} \mathbf{A}$ and $\mathbf{A}^T \mathbf{P} \mathbf{b}$ can be accumulated efficiently from each correspondence in parallel. Finally, the transformation matrix $\hat{\mathbf{T}}\{\hat{\mathbf{R}}, \hat{\mathbf{t}}\}$ is reconstructed from $\hat{\xi}$.

3) *Multi-strategy weighting function*: Throughout the iterative procedure, a multi-strategy weighting function based on residuals, balanced direction contribution and intensity consistency is proposed. The weight w_i for each correspondence is defined as $w_i = w_i^{(\text{residual})} \times w_i^{(\text{balanced})} \times w_i^{(\text{intensity})}$, whose multipliers are explained as follows.

First, to better deal with the outliers, we present a residual weighting function deduced from the general robust kernel function covering a family of M-estimators [62]:

$$w_i^{(\text{residual})} = \begin{cases} 1, & \text{if } \kappa = 2 \\ \frac{2\epsilon_i}{\epsilon_i^2 + 2}, & \text{if } \kappa = 0 \\ \epsilon_i \left(\frac{\epsilon_i^2}{|\kappa - 2|} + 1 \right)^{\frac{\kappa}{2} - 1}, & \text{otherwise} \end{cases}, \quad (10)$$

where κ is the coefficient for the kernel's shape, $\epsilon_i = d_i/\delta$ is the normalized residual and δ is the inlier noise threshold. Although an adaptive searching for the best κ is feasible [63], it would be time-consuming. Therefore, we fix $\kappa = 1$ in practice, thus leading to a pseudo-Huber kernel function.

Second, the contribution of the correspondences is not always balanced in x, y, z direction. The following weighting function considering the number of correspondences from each category is proposed to guarantee observability.

$$w_i^{(\text{balanced})} = \begin{cases} \frac{|\mathcal{F}| + 2|\mathcal{P}| + |\mathcal{B}|}{2(|\mathcal{G}| + |\mathcal{R}|)}, & i \in \mathcal{G} \text{ or } i \in \mathcal{R} \\ 1, & \text{otherwise} \end{cases}. \quad (11)$$

Third, since the intensity channel provides extra information for registration, (12) is devised to penalize the contribution of correspondences with large intensity inconsistency.

$$w_i^{(\text{intensity})} = e^{-\frac{|\Delta I_i|}{I_{\max}}}. \quad (12)$$

4) *Multi-index registration quality evaluation*: After the convergence of ICP, the posterior standard deviation $\hat{\sigma}$ and information matrix \mathcal{I} of the registration are calculated as:

$$\hat{\sigma}^2 = \frac{1}{n-6} (\mathbf{A}\hat{\xi} - \mathbf{b})^T \mathbf{P} (\mathbf{A}\hat{\xi} - \mathbf{b}), \mathcal{I} = \frac{1}{\hat{\sigma}^2} (\mathbf{A}^T \mathbf{P} \mathbf{A}). \quad (13)$$

where $\hat{\sigma}$, \mathcal{I} together with the nonground point overlapping ratio O_{ts} , are used for evaluating the registration quality.

D. MULLS front-end

As shown in Fig. 5(a), the front-end of MULLS is a combination of III-B, III-C and a map management module. With an incoming scan, roughly classified feature points are extracted and further downsampled into a sparser group of points for efficiency. Besides, a local map containing static feature points from historical frames is maintained with the

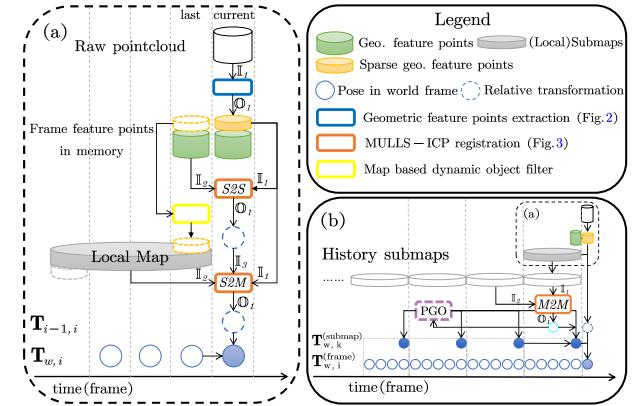


Fig. 5. Overall workflow of MULLS-SLAM: (a) front-end, (b) back-end.

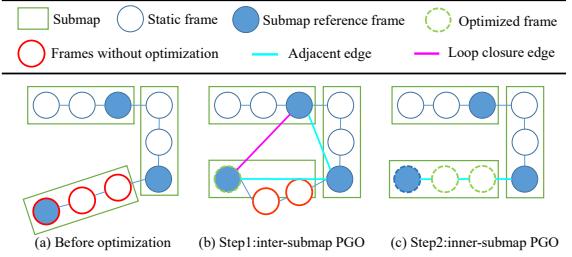


Fig. 6. Inter-submap and inner-submap pose graph optimization

reference pose of the last frame. Taking the last frame's ego-motion as an initial guess, the scan-to-scan MULLS-ICP is conducted between the sparser feature points of the current frame and the denser feature points of the last frame with only a few iterations. The estimated transformation is provided as a better initial guess for the following scan-to-map MULLS-ICP. It regards the feature points in the local map as the target point cloud and keeps iterating until the convergence ($\hat{\xi}$ smaller than a threshold). The sparser group of feature points are appended to the local map after the map-based dynamic object filtering. For this step, within a distance threshold to the scanner, those nonground feature points far away from their nearest neighbors with the same category in the local map are filtered. The local map is then cropped with a fixed radius. Only the denser group of feature points of the current frame are kept for the next epoch.

E. MULLS back-end

As shown in Fig. 5(b), the periodically saved submap is the processing unit. The adjacent and loop closure edges among the submaps are constructed and verified by the certificated and efficient TEASER [46] global registration. Its initial correspondences are determined according to the cosine similarity of NCC features encoded in III-B among the vertex keypoints. Taking TEASER's estimation as the initial guess, the map-to-map MULLS-ICP is used to refine inter-submap edges with accurate transformation and information matrices calculated in III-C. Those edges with higher $\hat{\sigma}$ or lower O_{ts} than thresholds would be deleted. As shown in Fig. 6, once a loop closure edge is added, the pose correction of free submap nodes is achieved by the inter-submap PGO. Hierarchically, the inner-submap PGO fixes each submap's reference frame and adjusts the other frames' pose.

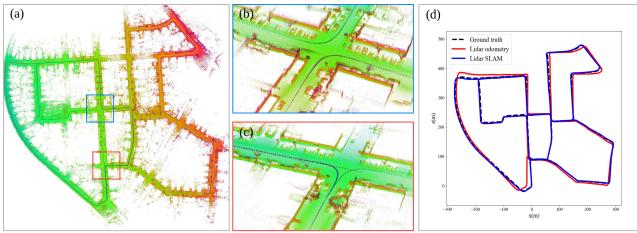


Fig. 7. MULLS’s result on urban scenario (KITTI seq.00): (a) overview, (b,c) map in detail of loop closure areas, (d) trajectory comparison.

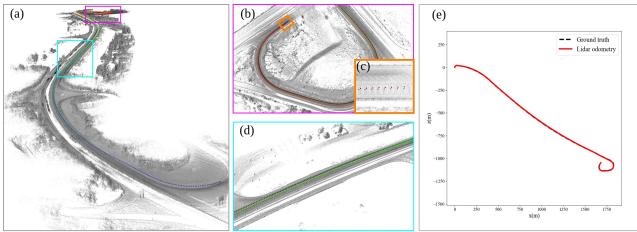


Fig. 8. MULLS’s result on highway scenarios (KITTI seq.01): (a) overview, (b) map and trajectory (black: ground truth) at the end of the sequence, (c) lane-level final drift, (d) map and trajectory of a challenging scene with high similarity and few features, (e) trajectory comparison (almost overlapped).

IV. EXPERIMENTS

The proposed MULLS system is evaluated qualitatively and quantitatively on KITTI, MIMAP and HESAI datasets, covering various outdoor and indoor scenes using seven different types of LiDAR, as summarized in Table I. All the experiments are conducted on a PC equipped with an Intel Core i7-7700HQ@2.80GHz CPU for fair comparison.

A. Quantitative experiment

The quantitative evaluations are done on the KITTI Odometry/SLAM dataset [64], which is composed of 22 sequences of data with more than 43k scans captured by a Velodyne HDL-64E LiDAR. KITTI covers various outdoor scenarios such as urban road, country road, and highway. Seq. 0-10 are provided with GNSS-INS ground truth pose while seq. 11-21 are used as the test set for online evaluation and leaderboard without provided ground truth. For better performance, an intrinsic angle correction of 0.2° is applied [11], [66]. Following the odometry evaluation criterion in [64], the average translation & rotation error (ATE & ARE) are used for localization accuracy evaluation. The performance of the proposed MULLS-LO (w/o loop closure) and MULLS-SLAM (with loop closure), as well as 10 state-of-the-art LiDAR SLAM solutions (results taken from their original papers and KITTI leaderboard), are reported in Table II. MULLS-LO achieves the best ATE (0.49%) on seq. 00-10 and runner-up on online test sets. With loop closure and PGO, MULLS-SLAM gets the best ARE ($0.13^\circ/100m$) but worse ATE. This is because ATE & ARE can’t fully reflect the gain of global positioning accuracy as they are measured locally within a distance of 800m. Besides, the maps and

TABLE I: The proposed method is evaluated on three datasets collected by various LiDARs in various scenarios.

Dataset	LiDAR	Scenario	#Seq.	#Frame
KITTI [64]	HDL64E	urban, highway, country	22	43k
MIMAP [65]	VLP32C, HDL32E	indoor	3	35k
HESAI	PandarQTLite, XT, 64, 128	residential, urban, indoor	8	30k



Fig. 9. MULLS’s result on MIMAP dataset: (a) the front view of the point cloud map of a 5-floor building generated by MULLS, (b) point cloud of the building (floor 1&2) collected by TLS, (c) point to point distance ($\mu = 6.7\text{cm}$) between the point cloud generated from TLS and MULLS.

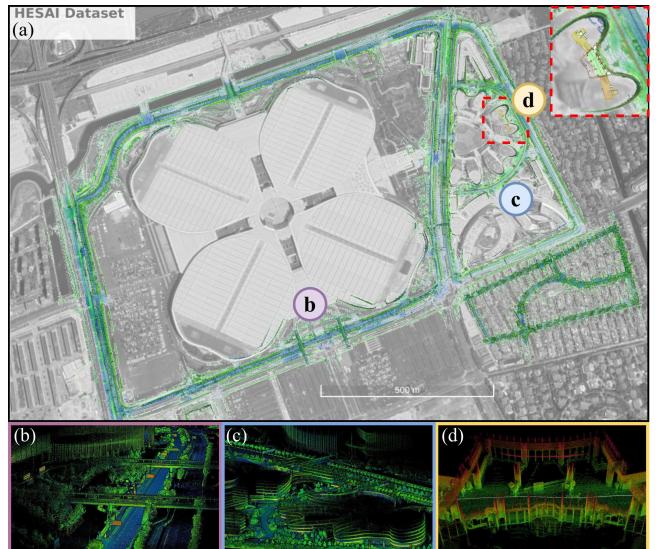


Fig. 10. Overview of MULLS’s results on HESAI dataset: (a) MULLS SLAM map aligned with Google Earth, (b-d) example of the generated point cloud map of the urban expressway, industry park, and indoor scenario.

trajectories constructed from MULLS for two representative sequences are shown in Fig. 7 and 8. Specifically, our method surpasses state-of-the-art methods with a large margin. It has only a lane-level drift in the end on the featureless and highly dynamic highway sequence, as shown in Fig. 8(c).

Some variants of MULLS are also compared in Table II. It is shown that scan-to-map registration can boost accuracy a lot. With only one iteration, MULLS(m1) already ranks 5th among the listed methods. With five iterations, MULLS(m5) get ATE of about 0.6%, which is close to the converged MULLS (with about 15 iterations on average)’s performance. The results indicate that we can sacrifice a bit of accuracy for faster operation on platforms with limited computing power. Besides, scan-to-scan registration before scan-to-map registration is not necessary once the local map is constructed as (s5m5) has similar performance to (m5). We thus use scan-to-map registration till convergence (mc) in MULLS.

B. Qualitative experiments

1) **MIMAP:** The ISPRS Multi-sensor Indoor Mapping and Positioning dataset [65] is collected by a backpack mapping system with Velodyne VLP32C and HDL32E in a 5-floor building. High accuracy terrestrial laser scanning (TLS) point cloud of floor 1 & 2 scanned by Rigel VZ-1000 is regarded as the ground truth for map quality evaluation. Fig. 9 demonstrates MULLS’s indoor mapping quality with a 6.7cm mean mapping error, which is calculated as the mean distance between each point in the map point cloud and the corresponding nearest neighbor in TLS point cloud.

TABLE II: Quantitative evaluation and comparison on KITTI dataset.

All errors are represented as ATE[%] / ARE[°/100m] (the smaller the better). **Red** and **blue** fonts denote the first and second place, respectively.

Denotations: U: urban road, H: highway, C: country road, *: with loop closure, -: not available, (*mc*):scan-to-map registration till convergence, (*sI*):scan-to-scan registration for 1 iteration, (*m5*):scan-to-map registration for 5 iterations, (*s5m5*):scan-to-scan for 5 iterations before scan-to-map for 5 iterations

Method	00 U*	01 H	02 C*	03 C	04 C	05 C*	06 U*	07 U*	08 U*	09 C*	10 C	00-10mean	11-21mean	time(s)/frame
LOAM [10]	0.78/-	1.43/-	0.92/-	0.86/-	0.71/-	0.57/-	0.65/-	0.63/-	1.12/-	0.77/-	0.79/-	0.84/-	0.55/0.13	0.10
IMLS-SLAM [11]	0.50 /-	0.82/-	0.53 /-	0.68/-	0.33 /-	0.32/-	0.33/-	0.33/-	0.80 /-	0.55/-	0.53 /-	0.52 /-	0.69/0.18	1.25
MC2SLAM [13]	0.51 /-	0.79 /-	0.54 /-	0.65 /-	0.44/-	0.27 /-	0.31/-	0.34/-	0.84/-	0.46 /-	0.52 /-	0.52 /-	0.69/ 0.16	0.10
S4-SLAM [26]*	0.62/-	1.11/-	1.63/-	0.82/-	0.95/-	0.50/-	0.65/-	0.60/-	1.33/-	1.05/-	0.96/-	0.92/-	0.93/0.38	0.20
PSF-LO [27]	0.64/-	1.32/-	0.87/-	0.75/-	0.66/-	0.45/-	0.47/-	0.46/-	0.94/-	0.56/-	0.54/-	0.74/-	0.82/0.32	0.20
SUMA ⁺ [16]*	0.64/0.22	1.60/0.46	1.00/0.37	0.67/0.46	0.37/ 0.26	0.40/0.20	0.46/0.21	0.34/0.19	1.10/0.35	0.47 /0.23	0.66/0.28	0.70/0.29	1.06/0.34	0.10
LiTAMIN2 [51]*	0.70/0.28	2.10/0.46	0.98/0.32	0.96/0.48	1.05/0.52	0.45/0.25	0.59/0.34	0.44/0.32	0.95/0.29	0.69/0.40	0.80/0.47	0.85/0.33	-/-	0.01
LO-Net [18]	0.78/0.42	1.42/0.40	1.01/0.45	0.73/0.59	0.56/0.54	0.62/0.35	0.55/0.35	0.56/0.45	1.08/0.43	0.77/0.38	0.92/0.41	0.83/0.42	1.75/0.79	0.10
FALO [25]	1.28/0.51	2.36/1.35	1.15/0.28	0.93/ 0.24	0.98/0.33	0.45/0.18	-/-	0.44/0.34	-/-	0.64/ 0.13	0.83/ 0.17	1.00/0.39	-/-	0.10
LoDoNet [28]	1.43/0.69	0.96/ 0.28	1.46/0.57	2.12/0.98	0.65/0.45	1.07/0.59	0.62/0.34	1.86/1.64	2.04/0.97	0.63/0.35	1.18/0.45	1.27/0.66	-/-	-
MULLS-LO(<i>mc</i>)	0.51/0.18	0.62/0.09	0.55/0.17	0.61/0.22	0.35/0.08	0.28/0.17	0.24/0.11	0.29/0.18	0.80/0.25	0.49/0.15	0.61/0.19	0.49/0.16	0.65/0.19	0.08
MULLS-SLAM(<i>mc</i>)*	0.54/0.13	0.62/0.09	0.69/0.13	0.61/0.22	0.35/0.08	0.29/0.08	0.27/0.11	0.83/0.17	0.51/0.12	0.61/0.19	0.52/0.13	-/-	0.10	
MULLS-LO(<i>sI</i>)	2.36/1.06	2.76/0.89	2.81/0.95	1.26/0.67	5.72/1.15	2.19/1.01	1.12/0.51	1.65/1.27	2.73/1.19	2.14/0.96	3.61/1.65	2.57/1.03	-/-	0.03
MULLS-SLAM(<i>mI</i>)*	0.79/0.28	1.01/0.25	1.46/0.39	0.79/0.23	1.02/0.15	0.40/0.14	0.38/0.13	0.48/0.34	0.94/0.29	0.61/0.22	0.57/0.29	0.77/0.25	-/-	0.05
MULLS-SLAM(<i>m5</i>)*	0.65/0.14	0.79/0.16	1.08/0.27	0.59/0.25	0.50/0.11	0.32/0.08	0.36/0.09	0.36/0.24	0.75/0.18	0.58/0.23	0.68/0.23	0.60/0.18	-/-	0.07
MULLS-SLAM(<i>s5m5</i>)*	0.66/0.14	0.82/0.19	1.01/0.26	0.59/0.27	0.63/0.12	0.34/0.08	0.34/0.08	0.40/0.29	0.81/0.19	0.56/0.20	0.58/0.20	0.61/0.18	-/-	0.08

TABLE III: Ablation study w.r.t. geometric feature points
(-: LiDAR odometry failed)

po → pl	po → li	po → po	KITTI 00 U	KITTI 01 H
\mathcal{G}	\mathcal{F}	\mathcal{P}	\mathcal{B}	\mathcal{V}
✓	✓	✓	✗	0.53 / 0.20
✓	✓	✓	✗	0.62 / 0.09
✓	✓	✗	✗	1.02 / 0.16
✓	✓	✗	✗	0.54 / 0.21
✓	✗	✓	✗	0.92 / 0.28
✓	✗	✓	✗	0.70 / 0.33
✓	✓	✓	✓	0.57 / 0.23

weighting function	KITTI 00 U	KITTI 01 H
$w^{(\text{bal.})}$	$w^{(\text{res.})}$	$w^{(\text{int.})}$
✓	✓	✓
✗	✗	✗
✗	✓	✗
✓	✗	✓
✓	✓	✗

2) **HESAI**: HESAI dataset is collected by four kinds of mechanical LiDAR (Pandar128, 64, XT, and QT Lite). Since the ground truth pose or map is not provided, the qualitative result of the consistent maps built by MULLS-SLAM indicate MULLS's ability for adapting to LiDARs with different number and distribution of beams in both the indoor and outdoor scenes, as shown in Fig. 10.

C. Ablation studies

Furthermore, in-depth performance evaluation on the adopted categories of geometric feature points and the weighting functions are conducted to verify the proposed method's effectiveness. Table III shows that vertex point-to-point correspondences undermine LO's accuracy on both sequences because points from regular structures are more reliable than those with high curvature on vegetations. Therefore, vertex points are only used as the keypoints in back-end global registration in practice. Besides, linear points (pillar and beam) are necessary for the supreme performance on highway scene. Points on guardrails and curbs are extracted as linear points to impose cross direction constraints. It also indicates MULLS may encounter problem in tunnels, where structured features are rare. As shown in Table IV, all the three functions presented in III-C are proven effective on both sequences. Specifically, translation estimation realizes obvious improvement on featureless highway scenes with the consideration of intensity consistency.

TABLE V: Runtime analysis per frame in detail (unit: ms)

(Typically, there're $\approx 2k$ and $20k$ feature points in current frame and local map)

Module	Submodule	t_{submod}	#Iter.	t_{mod}	T
Feature extraction	ground points	8	1	29	80
	non-ground feature points	20			
	NCC encoding	1			
Map updating	dynamic object removal	2	1	3	
	local map updating	1			
Registration	closest point association	3	15	48	
	transform estimation	0.2			

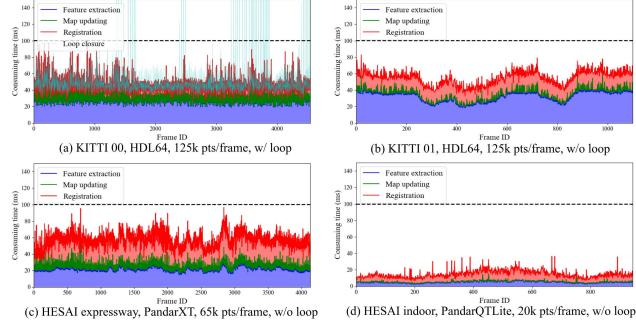


Fig. 11. Run time analysis of the proposed MULLS using different types of LiDAR with or without loop closure

D. Runtime analysis

A detailed runtime analysis for each frame is shown in Table V. MULLS transformation estimation only costs 0.2 ms per ICP iteration with about 2k and 20k feature points in the source and target point cloud, respectively. Moreover, Fig. 11 shows MULLS's timing report of four typical sequences from KITTI and HESAI dataset with different point number per frame. MULLS operates faster than 10Hz on average for all the sequences, though it sometimes runs beyond 100ms on sequences with lots of loops such as Fig. 11(a). Since the loop closure can be implemented on another thread, MULLS is guaranteed to perform in real-time on a moderate PC.

V. CONCLUSIONS

In this work, we present a versatile LiDAR-only SLAM system named MULLS, which is based on the efficient multi-metric linear least square ICP. Experiments on more than 100 thousand frames collected by 7 different LiDARs show that MULLS achieves real-time performance with low drift and high map quality on various challenging outdoor and indoor scenarios regardless of the LiDAR specifications.

REFERENCES

- [1] H. Temeltas and D. Kayak, "SLAM for robot navigation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 23, no. 12, pp. 16–19, 2008.
- [2] K. Ebadi, Y. Chang, M. Palieri, A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood, *et al.*, "LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 80–86.
- [3] W.-C. Ma, I. Tartavull, I. A. Bârsan, S. Wang, M. Bai, G. Mattus, N. Homayounfar, S. K. Lakshminarayanan, A. Pokrovsky, and R. Urtasun, "Exploiting Sparse Semantic HD Maps for Self-Driving Vehicle Localization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5304–5311.
- [4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [5] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE international conference on robotics and automation (ICRA)*, 2014, pp. 15–22.
- [6] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [7] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, *et al.*, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559–568.
- [8] T. Whelan, M. Kaess, H. Johansson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense rgbd slam with volumetric fusion," *The International Journal of Robotics Research*, vol. 34, no. 4–5, pp. 598–626, 2015.
- [9] M. Labb   and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.
- [10] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.
- [11] J.-E. Deschaud, "Imls-slam: scan-to-model matching based on 3d data," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2480–2485.
- [12] J. Graeter, A. Wilczynski, and M. Lauer, "Limo: Lidar-monocular visual odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7872–7879.
- [13] F. Neuhaus, T. Ko  , R. Kohnen, and D. Paulus, "Mc2slam: Real-time inertial lidar odometry using two-scan motion compensation," in *German Conference on Pattern Recognition*, 2018.
- [14] 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)*, 2018, pp. 4758–4765.
- [15] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments," in *Robotics: Science and Systems*, 2018.
- [16] X. Chen, A. Milioto, E. Palazzolo, P. Gigu  re, J. Behley, and C. Stachniss, "Suma++: Efficient lidar-based semantic slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4530–4537.
- [17] D. Kovalenko, M. Korobkin, and A. Minin, "Sensor aware lidar odometry," in *2019 European Conference on Mobile Robots (ECMR)*, 2019, pp. 1–6.
- [18] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li, "LO-Net: Deep Real-time Lidar Odometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8473–8482.
- [19] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3144–3150.
- [20] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-Fusion: LiDAR-Inertial-Camera Odometry," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5848–5854.
- [21] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3126–3131.
- [22] S. W. Chen, G. V. Nardari, E. S. Lee, C. Qu, X. Liu, R. A. F. Romero, and V. Kumar, "SLOAM: Semantic lidar odometry and mapping for forest inventory," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 612–619, 2020.
- [23] T. Qin and S. Cao, "A-loam: Advanced implementation of loam," 2019. [Online]. Available: <https://github.com/HKUST-Aerial-Robotics/A-LOAM>
- [24] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020.
- [25] I. Garc  a Daza, M. Rentero, C. Salinas Maldonado, R. Izquierdo Gonzalo, and N. Hern  ndez Parra, "Fail-aware lidar-based odometry for autonomous vehicles," *Sensors*, vol. 20, no. 15, p. 4097, 2020.
- [26] B. Zhou, Y. He, K. Qian, X. Ma, and X. Li, "S4-SLAM: A real-time 3D LIDAR SLAM system for ground/watersurface multi-scene outdoor applications," *Autonomous Robots*, pp. 1–22, 2020.
- [27] G. Chen, B. Wang, X. Wang, H. Deng, B. Wang, and S. Zhang, "PSF-LO: Parameterized Semantic Features Based Lidar Odometry," *arXiv preprint arXiv:2010.13355*, 2020.
- [28] C. Zheng, Y. Lyu, M. Li, and Z. Zhang, "Lodonet: A deep neural network with 2d keypoint matching for 3d lidar odometry estimation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2391–2399.
- [29] L. He, X. Wang, and H. Zhang, "M2DP: A novel 3D point cloud descriptor and its application in loop closure detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 231–237.
- [30] R. Dub  , A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: Segment-based mapping and localization using data-driven descriptors," *The International Journal of Robotics Research*, vol. 39, no. 2–3, pp. 339–355, 2020.
- [31] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4802–4809.
- [32] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2095–2101.
- [33] J. Jiang, J. Wang, P. Wang, P. Bao, and Z. Chen, "LiPMatch: LiDAR Point Cloud Plane Based Loop-Closure," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6861–6868, 2020.
- [34] F. Liang, B. Yang, Z. Dong, R. Huang, Y. Zang, and Y. Pan, "A novel skyline context descriptor for rapid localization of terrestrial laser scans to airborne laser scanning point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 165, pp. 120–132, 2020.
- [35] X. Chen, T. L  be, A. Milioto, T. R  hling, O. Vysotska, A. Haag, J. Behley, C. Stachniss, and F. Fraunhofer, "OverlapNet: Loop closing for LiDAR-based SLAM," in *Proc. Robot.: Sci. Syst.*, 2020.
- [36] A. Zaganidis, A. Zerntev, T. Duckett, and G. Cielniak, "Semantically Assisted Loop Closure in SLAM Using NDT Histograms," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4562–4568.
- [37] "isAM: Incremental smoothing and mapping, author=Kaess, Michael and Ranganathan, Ananth and Dellaert, Frank." *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.
- [38] G. Grisetti, R. K  mmerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2D and 3D mapping," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 273–278.
- [39] K. Ni and F. Dellaert, "Multi-level submap based slam using nested dissection," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2558–2565.
- [40] R. K  mmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.
- [41] G. Grisetti, R. K  mmerle, and K. Ni, "Robust optimization of factor graphs by using condensed measurements," in *2012 IEEE/RSJ Inter-*

- national Conference on Intelligent Robots and Systems*, 2012, pp. 581–588.
- [42] J. L. Blanco-Claraco, “A modular optimization framework for localization and mapping,” in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.
- [43] S. Li, G. Li, L. Wang, and Y. Qin, “Slam integrated mobile mapping system in complex urban environments,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, pp. 316–332, 2020.
- [44] S. Zhao, Z. Fang, H. Li, and S. Scherer, “A robust laser-inertial odometry and mapping method for large-scale highway environments,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1285–1292.
- [45] M. Palieri, B. Morrell, A. Thakur, K. Ebadi, J. Nash, A. Chatterjee, C. Kanellakis, L. Carbone, C. Guaragnella, and A.-a. Aghamohammadi, “LOCUS: A Multi-Sensor Lidar-Centric Solution for High-Precision Odometry and 3D Mapping in Real-Time,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 421–428, 2020.
- [46] H. Yang, J. Shi, and L. Carlone, “TEASER: Fast and Certifiable Point Cloud Registration,” *IEEE Trans. Robotics*, 2020.
- [47] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Sensor fusion IV*, vol. 1611, 1992, pp. 586–606.
- [48] S. Rusinkiewicz and M. Levoy, “Efficient variants of the icp algorithm,” in *Proceedings third international conference on 3-D digital imaging and modeling*, 2001, pp. 145–152.
- [49] A. Segal, D. Haehnel, and S. Thrun, “Generalized-ICP,” in *Robotics: science and systems*, vol. 2, no. 4, 2009, p. 435.
- [50] A. Censi, “An icp variant using a point-to-line metric,” in *IEEE International Conference on Robotics and Automation*, 2008, pp. 19–25.
- [51] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, “LiTAMIN2: Ultra Light LiDAR-based SLAM using Geometric Approximation applied with KL-Divergence,” *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [52] 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.
- [53] F. Pomerleau, F. Colas, and R. Siegwart, “A review of point cloud registration algorithms for mobile robotics,” *Foundations and Trends in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.
- [54] T. Kühner and J. Kümmerle, “Large-scale volumetric scene reconstruction using lidar,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 6261–6267.
- [55] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (FPFH) for 3D registration,” in *2009 IEEE international conference on robotics and automation*, 2009, pp. 3212–3217.
- [56] S. Huang, Z. Gojcic, M. Usyatsov, A. Wieser, and K. Schindler, “PREDATOR: Registration of 3D Point Clouds with Low Overlap,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [57] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, “Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.
- [58] N. Mellado, D. Aiger, and N. J. Mitra, “Super 4pcs fast global pointcloud registration via smart indexing,” in *Computer Graphics Forum*, vol. 33, no. 5, 2014, pp. 205–215.
- [59] J. Yang, H. Li, D. Campbell, and Y. Jia, “Go-ICP: A globally optimal solution to 3D ICP point-set registration,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2241–2254, 2015.
- [60] Z. Cai, T.-J. Chin, A. P. Bustos, and K. Schindler, “Practical optimal registration of terrestrial LiDAR scan pairs,” *ISPRS journal of photogrammetry and remote sensing*, vol. 147, pp. 118–131, 2019.
- [61] T. Hackel, J. D. Wegner, and K. Schindler, “Fast semantic segmentation of 3D point clouds with strongly varying density,” *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, vol. 3, pp. 177–184, 2016.
- [62] J. T. Barron, “A general and adaptive robust loss function,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4331–4339.
- [63] N. Chebrolu, T. Läbe, O. Vysotska, J. Behley, and C. Stachniss, “Adaptive robust kernels for non-linear least squares problems,” *arXiv preprint arXiv:2004.14938*, 2020.
- [64] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [65] C. Wang, Y. Dai, N. Elsheimy, C. Wen, G. Retscher, Z. Kang, and A. Lingua, “Isprs benchmark on multisensory indoor mapping and positioning,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 5, pp. 117–123, 2020.
- [66] D. Yin, Q. Zhang, J. Liu, X. Liang, Y. Wang, J. Maanpää, H. Ma, J. Hyppä, and R. Chen, “CAE-LO: LiDAR Odometry Leveraging Fully Unsupervised Convolutional Auto-Encoder for Interest Point Detection and Feature Description,” *arXiv preprint arXiv:2001.01354*, 2020.