

# FasterGICP: Acceptance-Rejection Sampling Based 3D Lidar Odometry

Jikai Wang<sup>✉</sup>, Meng Xu<sup>✉</sup>, Farzin Foroughi, Deyun Dai, and Zonghai Chen<sup>✉</sup>

**Abstract**—Distribution-to-distribution-based lidar odometry is known for its good accuracy, while it cannot run in real-time when the number of points is large. To alleviate this problem, Faster Generalized Iterative Closest Point (FasterGICP) is proposed in this letter, in which an acceptance-rejection sampling-based two-step point filter is proposed to exclude the points that rarely benefit the lidar odometry performance. Specifically, the lidar point cloud is firstly filtered and only the points with high planarity tend to be preserved, which can reduce the distribution approximation errors when the GICP works as a plane-to-plane Iterative Closest Point (ICP). Secondly, during the pose estimation optimization process, the lidar points are further iteratively filtered according to their contributions to the optimization objective function, in which the point's matching error defines the contribution. The two-step filtering process is achieved by designing the target and proposal distributions in the acceptance-rejection sampling framework. With the help of the point filter, our odometry can work in a scan-to-model strategy while demonstrating both efficiency and accuracy improvements. Extensive validation experiments are conducted on the public and our datasets. The results demonstrate that our method can achieve competitive performance compared with the state-of-the-art lidar odometry and Simultaneously Localization and Mapping (SLAM) methods. Our code has been made public available at <https://github.com/SLAMWang/fasterGICP>.

**Index Terms**—3D lidar odometry, generalized-ICP, acceptance-rejection sampling, localization, mapping.

## I. INTRODUCTION

THE rapid development of autonomous mobile exploration and mapping calls for off-the-shelf odometry solutions that are adaptive to different specifications on various complex scenarios. Compared with vision-based and RGB-D-based odometry systems [1], [2], lidar odometry systems are more reliable under various lighting conditions and are more suitable for tasks demanding for dense 3D map [3]–[5].

The core of 3D lidar odometry is lidar scan matching. In recent years, 3D lidar odometry has been extensively studied, and the existing methods can be classified into three categories,

including feature point-based lidar odometry [5], dense surface representation-based lidar odometry [6], [7], and probabilistic distribution-based lidar odometry [8]. Generalized Iterative Closest Point (GICP) [9], a distribution-to-distribution version of the Iterative Closest Point method, has been recently introduced to 3D lidar odometry [8]. It applies Gaussian distribution to fit the local surface of each point, and then lidar odometry is achieved by minimizing the distribution-to-distribution distance. GICP-based lidar odometry has demonstrated high robustness and adaptability to the lidar points characteristic and various complex environments [8].

However, there remains challenges. Firstly, in GICP, all points are involved in computation, which makes the algorithm cannot run in real-time when the number of points is significant. Although it can adjust the running efficiency by changing the voxelization resolution, the pose tracking precision would be degraded by reducing the number of points evenly. Secondly, traditional GICP is suggested to be performed in plane-to-plane strategy by replacing the eigenvalues of the covariance matrix with  $(1, 1, \epsilon)$ . However, approximation errors exist for the points from the surface with less planarity. Thirdly, it is redundant to preserve all the point matches in the objective function during the pose optimization since some matches are outliers. Though some kernel functions are applied for outlier processing, the maintenance of these matches leads to computational costs. Only preserving the valuable points is an effective way to improve the efficiency and accuracy of lidar odometry concurrently. Acceptance-rejection sampling [10] is widely used for sampling from a target distribution. The motivation of this paper is that lidar odometry can be combined with an acceptance-rejection sampling mechanism to obtain the desirable points for pose tracking.

In this paper, FasterGICP is proposed to achieve both accuracy and efficiency improvements. To reduce the involved lidar points without degrading the pose tracking precision, acceptance-rejection sampling-based two-step point filter is introduced to refine the lidar points according to their covariance matrices and matching errors. A general acceptance-rejection sampling strategy is applied in the first-step filtering process to select the points with high planarity. We notice that the point correspondences whose matching errors being close to 0 have fewer contributions to the pose optimization, which can also be mathematically proved. Thus, an anti-acceptance-rejection sampling strategy is proposed to exclude these points for the sake of efficiency.

Manuscript received July 23, 2021; accepted October 20, 2021. Date of publication November 2, 2021; date of current version November 10, 2021. This letter was recommended for publication by Associate Editor Lin Zhang and Editor Javier Civera upon evaluation of the reviewers' comments. This work was supported by the National Natural Science Foundation of China under Grant 91848111. (Corresponding author: Zonghai Chen.)

The authors are with the Department of Automation, University of Science and Technology of China, Hefei 230027, China (e-mail: wangjk@ustc.edu.cn; xml1996@mail.ustc.edu.cn; farzin.foroughi2011@gmail.com; daideyun@mail.ustc.edu.cn; chenzh@ustc.edu.cn).

Digital Object Identifier 10.1109/LRA.2021.3124072

The main contributions of this paper are three-fold.

- First, we incorporate the acceptance-rejection sampling mechanism into the GICP method, which significantly increases the lidar odometry performance.
- Second, in the two-step point filtering process, different but effective target and proposal distributions are proposed according to the characteristic of lidar points and lidar odometry requirements.
- Third, extensive validation experiments are conducted on the public and our datasets. The proposed method is easy to follow and can be generalized to other lidar odometry methods.

The rest of this paper is organized as follows. Section II presents the related work. Section III gives the proposed lidar odometry in detail. The experimental evaluations are shown in Section IV. Section V concludes the paper.

## II. RELATED WORK

To extract geometrical features from LiDAR scans is widely used in the existing lidar odometry methods. These methods exclude the less informative points for the sake of efficiency and preserve the geometrical features for the sake of accuracy. LOAM [11] performs point feature to edge/plane scan-matching to find correspondences between scans. LeGO-LOAM [12] is another state-of-the-art lidar SLAM system that achieves reliable, real-time six-degree-of-freedom pose estimation for ground vehicles equipped with 3D lidar. LeGO-LOAM makes great contributions for efficient feature extraction and matching scans that have large quantities of points. Zhou *et al.* argue that redundant regions of point cloud should be removed during the scan-matching process [13]. In their work, a voxelized cube set, ROI-cloud, is proposed to solve this problem by exclusively reserving the regions of interest for better point set registration and pose estimation. Recently, the directions of geometrical points are exploited to improve the scan registration performance [5], [14]. In these methods, sophisticated mechanisms are exploited to extract geometrical feature points (ground, plane, pillar, edge) according to some criteria, including eigenvalues [5], [14] and smoothness [12]. Since the number of the feature points is far less than the whole point cloud, it is more efficient to perform scan matching based on the feature points. Furthermore, these points are more stable and less redundant, which can also benefit the lidar odometry performance. However, the extraction of feature points relies on the settings of hyper-parameters. It is required to adapt the hyper-parameters to different complex scenarios to achieve the best performance. Furthermore, the dependence of geometry feature points also decreases the generality of the lidar odometry to less structured environments.

ICP [15] and its variants [16], [17] are widely used to perform dense point cloud matching. Since the registration constraints provided by point-to-point distances tend to contain much matching noise, the ICP variants apply local structural information to provide more robust constraints for transformation optimization, such as point-to-plane ICP [6], [17], [18], Generalized-ICP [9], and point-to-surface ICP [19], [20]. Recently, point-to-mesh features-based lidar odometry methods are also proposed [7], [21], which demonstrate that the dense

and accurate mapping of the environment surface can significantly improve the lidar odometry performance. LiTAMIN2 [22] is the most recently proposed probabilistic distribution-based lidar odometry method. It significantly reduces the number of points used for point cloud registration using a symmetric KL-divergence based ICP metric to speed up the registration process while maintaining accuracy. Among these methods, all the points are involved in the pose optimization process.

## III. PROPOSED METHOD

The overall framework of our method is presented in Fig. 1. The points are firstly filtered according to their covariance matrices SVD results. The lidar points are further iteratively filtered during the pose estimation optimization process according to their matching errors. The two-step point filter is achieved by the acceptance-rejection sampling mechanism.

### A. GICP Formulation

We consider the estimation of the transformation  $\mathbf{T}_{K_{t-1}K_t} \in \mathbb{SE}_3$ . For each input point cloud  $K_t = \{\mathbf{p}_i^t \in \mathbb{R}^3, i = 0, 1, \dots, N\}$ , we first perform voxelization, which is provided by the PCL library,<sup>1</sup> and we define the voxelization function as:

$$K'_t = V(K_t, \gamma), \quad (1)$$

where  $\gamma$  is the voxelization resolution. Then, we align the current lidar scan  $\bar{K}_t = \{\mathbf{p}_i^t \in \mathbb{R}^3, i = 0, 1, \dots, N\}$  (source point cloud) with respect to the last lidar scan  $K'_{t-1} = \{\mathbf{p}_i^{t-1} \in \mathbb{R}^3, i = 0, 1, \dots, N\}$  (target point cloud). The correspondences between  $K'_t$  and  $K'_{t-1}$  are given by nearest neighbor search:  $\mathbf{p}_i^{t-1} = \mathbf{T} \circ \mathbf{p}_i^t$ , where the operation  $\circ$  is defined as

$$\mathbf{T} \circ \mathbf{p}_i^t = [\mathbf{T} \cdot [\mathbf{p}_i^t, 1]^T]_{1:3}. \quad (2)$$

Following the GICP, we model the surface from which a point was sampled as a Gaussian distribution:  $\mathbf{p}_i^t \sim \mathcal{N}(\hat{\mathbf{p}}_i^t, \mathbf{C}_i^t)$ ,  $\mathbf{p}_i^{t-1} \sim \mathcal{N}(\hat{\mathbf{p}}_i^{t-1}, \mathbf{C}_i^{t-1})$ . Then, the transformation error is defined as follows:

$$\hat{\mathbf{d}}_i = \hat{\mathbf{p}}_i^{t-1} - \mathbf{T} \circ \hat{\mathbf{p}}_i^t. \quad (3)$$

The distribution of  $\hat{\mathbf{d}}_i$  is given by the reproductive property of the Gaussian distribution as:

$$\begin{aligned} \mathbf{d}_i &\sim \mathcal{N}(\hat{\mathbf{p}}_i^{t-1} - \mathbf{T} \circ \hat{\mathbf{p}}_i^t, \mathbf{C}_i^{t-1} + \mathbf{T}\mathbf{C}_i^t\mathbf{T}^T) \\ &= \mathcal{N}(\mathbf{0}, \mathbf{C}_i^{t-1} + \mathbf{T}\mathbf{C}_i^t\mathbf{T}^T) \end{aligned} \quad (4)$$

Typically, the lidar odometry uses nonlinear optimization (such as the Gaussian-Newton method) to find the  $\mathbf{T}$  that maximizes the log-likelihood of the distribution of  $\hat{\mathbf{d}}_i$  as follows:

$$\begin{aligned} \mathbf{T} &= \arg \max_{\mathbf{T}} \sum_i \log(p(\mathbf{d}_i)) \\ &= \arg \max_{\mathbf{T}} \sum_i \mathbf{d}_i^T (\mathbf{C}_i^{t-1} + \mathbf{T}\mathbf{C}_i^t\mathbf{T}^T)^{-1} \mathbf{d}_i \\ &= \arg \max_{\mathbf{T}} \sum_i \mathbf{d}_i^T \Omega_i \mathbf{d}_i \end{aligned} \quad (5)$$

<sup>1</sup><https://pointclouds.org/>

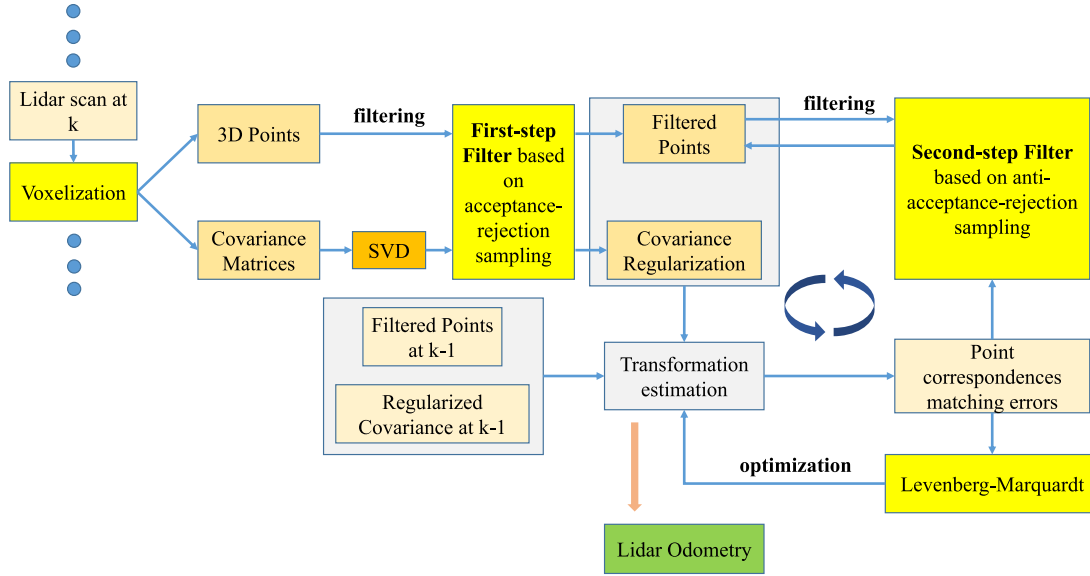


Fig. 1. The framework of our lidar odometry.

As is suggested by the authors of GICP, each covariance matrix is regularized by replacing its eigenvalues with  $(1, 1, \epsilon)$ , which makes GICP work as a plane-to-plane ICP.

### B. First-Step Point Filtering

1) *Covariance Matrix Regularization Error*: Mathematically, by performing Singular Value Decomposition (SVD),  $\mathbf{C}_i^t$  can be represented as

$$\mathbf{C}_i^t = [\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}] \cdot \begin{bmatrix} \lambda_0 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix} \cdot [\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}]^T, \quad (6)$$

where  $\lambda_0, \lambda_1$  and  $\lambda_2$  are the eigenvalues and are sorted decreasingly.  $\hat{\mathbf{a}}, \hat{\mathbf{b}}$  and  $\hat{\mathbf{c}}$  are the corresponding eigenvectors. After the regularization,  $\mathbf{C}_i^t$  is approximated as

$$\hat{\mathbf{C}}_i^t = [\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \epsilon \end{bmatrix} \cdot [\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}]^T = \mathbf{C}_i^t + \Delta \mathbf{C}_i^t, \quad (7)$$

where  $\Delta \mathbf{C}_i^t$  is named as regularization error. Then, the corresponding information matrix to construct objective function is denoted as

$$\begin{aligned} \hat{\Omega}_i &= (\hat{\mathbf{C}}_i^{t-1} + \mathbf{T} \hat{\mathbf{C}}_i^t \mathbf{T}^T)^{-1} \\ &= (\mathbf{C}_i^{t-1} + \Delta \mathbf{C}_i^{t-1} + \mathbf{T} \mathbf{C}_i^t \mathbf{T}^T + \mathbf{T} \Delta \mathbf{C}_i^t \mathbf{T}^T)^{-1}, \end{aligned} \quad (8)$$

The objective function is then modified to

$$\hat{\mathbf{T}} = \arg \max_{\mathbf{T}} \sum_i \mathbf{d}_i^T \hat{\Omega}_i \mathbf{d}_i, \quad (9)$$

The bias between  $\mathbf{T}$  in Eq. (5) and  $\hat{\mathbf{T}}$  in Eq. (9) is caused by the regularization errors. To decrease  $\|\mathbf{T} - \hat{\mathbf{T}}\|_2$ , we filter the points that are associated with large regularization errors in the first-step filter.

2) *Acceptance-Rejection Sampling Based First-Step Filtering*: Since we have computed the covariance matrix and its eigenvalues of each point in  $K'_t$ , we perform acceptance-rejection sampling on these points. Specifically, we first normalize the third eigenvalues

$$\bar{\lambda}_2 = \frac{\lambda_2}{\lambda_0}. \quad (10)$$

Then, the acceptance-rejection sampling steps to filter the points are as follows:

- 1) **Target distribution**: The motivation of our method is to only preserve the points whose  $\bar{\lambda}_2$  satisfy  $\bar{\lambda}_2 \ll 1$ . We define the target distribution as

$$f(\mathbf{p}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\bar{\lambda}_2 - 0)^2}{2\sigma^2}\right), \quad (11)$$

where  $\bar{\lambda}_2$  is the normalized minimum eigenvalue of the covariance matrix of  $\mathbf{p}$ . The target distribution is a Gaussian distribution whose mean value is 0.

- 2) **Proposal distribution**: In this paper, the lidar points in  $K'_t$  are directly regarded as the proposal distribution

$$g(\mathbf{p}) = \frac{1}{N'}, \quad (12)$$

where  $N'$  is the number of the total lidar points.

- 3) **Acceptance-rejection sampling**: For each point  $\mathbf{p}_i^t$  in  $K'_t$ , we preserve it if

$$\mu \leq \frac{f(\mathbf{p}_i^t)}{c \cdot g(\mathbf{p}_i^t)}, \quad (13)$$

where  $\mu$  is a value sampled from a uniform distribution:  $\mathbf{U}(0, 1)$ , and  $c$  is the constant coefficient. In this paper, we define  $c$  as

$$c = \frac{N'}{\sqrt{2\pi}\sigma}. \quad (14)$$

If Eq. (13) does not hold, the point will not be involved in the following optimization process. After the filtering process, the left point set is denoted as  $\bar{K}'_t$ , where we have  $|\bar{K}'_t| < |K'_t|$ .  $|\cdot|$  returns the cardinality of the input set.

### C. Second-Step Point Filtering for Pose Optimization

1) *Matching Error-Based Point Redundancy Derivation*: To find the optimal solution of Eq. (9), Gauss-Newton method is widely used. During the optimization process, the key step is to construct the iterative updating equation

$$\mathbf{J}^T \Omega_i \mathbf{J} \Delta \xi_i = -\mathbf{J}^T \Omega_i \mathbf{d}_i, \quad (15)$$

where  $\mathbf{J}$  is the Jacobian Matrix of the objective function and  $\xi$  is the pose representation in Lie-algebra. The computed  $\Delta \xi_i$  is then used to update the pose estimation. Then, considering  $\mathbf{J}^T \Omega_i \mathbf{J}$  is nonsingular matrix, we have the following qualitative corollary:

*Corollary 1*: If  $\mathbf{d}_i$  consists of small values that are close to zero, then the elements in the computed  $\Delta \xi_i$  are also close to zero, which means that it has little contribution to the pose estimation update.

Based on the motivation that the preservation of the redundant points whose corresponding matching error items have little contribution, we seek to exclude these points for the sake of efficiency in the second-setp filter.

2) *Anti-Acceptance-Rejection Sampling-Based Second-Step Filtering*: Let  $d_i$  denote  $\mathbf{d}_i^T (\mathbf{C}_i^{t-1} + \mathbf{T} \mathbf{C}_i^t \mathbf{T}^T)^{-1} \mathbf{d}_i$ . The anti-acceptance-rejection sampling process is detailed as follows:

- 1) **Target distribution**: We define the target distribution of  $d_i$  as

$$f(d_i) = \frac{1}{\sqrt{2\pi}\sigma_d} \exp\left(-\frac{(d_i - 0)^2}{2\sigma_d^2}\right), \quad (16)$$

The target distribution is a Gaussian distribution whose mean value is 0 and variance is  $\sigma_d$ .

- 2) **Proposal distribution**: Similarly, the lidar points  $\bar{K}'_t$  can be directly regarded as the proposal distribution

$$g(d_i) = \frac{1}{\bar{N}'}, \quad (17)$$

where  $\bar{N}'$  is the number of the total lidar points.

- 3) **Anti-acceptance-rejection sampling**: The objective function is modified as

$$\mathbf{T} = \arg \max_{\mathbf{T}} \sum_i \Gamma(d_i) \cdot d_i, \quad (18)$$

where

$$\Gamma(d_i) = \begin{cases} 1, & \text{if } \mu_d \geq \frac{f(d_i)}{c_d \cdot g(d_i)} \\ 0, & \text{if } \mu_d < \frac{f(d_i)}{c_d \cdot g(d_i)} \end{cases} \quad (19)$$

where  $\mu_d$  is a value sampled from a uniform distribution:  $\mathbf{U}(0, 1)$ , and  $c_d$  is the constant coefficient. In the anti-acceptance-rejection sampling process, the acceptance condition is contrary to that of the acceptance-rejection

sampling process. In this paper, we define  $c_d$  as

$$c_d = \frac{\bar{N}'}{\sqrt{2\pi}\sigma_d}. \quad (20)$$

With the sampling process, the number of point correspondences involved in the transformation optimization process is reduced.

### D. The Optimization Process

The steps are given as follows:

- 1) **Initialization and First-step Filter**: initialize the transformation estimation according to the uniform motion model, implement the first-setp filter and then construct the objective function given in Eq. (9);
- 2) **Anti-acceptance-rejection sampling-based point filter**: perform the sampling process to refine the point correspondences used for constructing the objective function Eq. (18);
- 3) **Nonlinear Optimization and Update**: perform transformation estimation optimization. Then, update the point correspondences and cost function according to the updated estimation of  $\mathbf{T}$ . Then, continue to **Anti-acceptance-rejection sampling-based point filter**.

### E. Scan-to-Model Based Lidar Odometry

We perform lidar odometry in a scan-to-model manner to reduce the drift errors. To formulate the local map  $\mathcal{M}$ , the most recent  $n$  frames are aggregated and transformed into the coordinates of  $K'_{t-1}$ , which can be represented as

$$\mathcal{M} = V \left( \bigcup_{i=t-1}^{t-n} \mathbf{T}_i^{t-1} \circ \bar{K}'_i, \gamma \right). \quad (21)$$

Then,  $\mathbf{T}_{t-1}^t$  is estimated by aligning  $K'_t$  with  $\mathcal{M}$ .

## IV. EXPERIMENTS

We evaluate the proposed FasterGICP lidar odometry on the public datasets and our collected dataset. The hardware arrangement is a laptop with an Intel i7 CPU at 3.66 GHz and 24 GB memory. The algorithms are implemented in C++. To demonstrate our method's performance, we qualitatively and quantitatively analyze our method regarding the accuracy and efficiency of the lidar odometry. Following the odometry evaluation criterion [23], the average translational error (ATE) is used for localization accuracy evaluation. We first perform an ablation study of our method on the sequences of the KITTI dataset.<sup>2</sup> Then, we thoroughly validate our method on the KITTI dataset and compare our method with the state-of-the-art lidar SLAM systems. We also perform our method on our dataset, USTC-VLP16-Dataset<sup>3</sup> and a public dataset, Stevens-VLP-16-Dataset.<sup>4</sup> Efficiency analysis and discussion are also presented in this section.

<sup>2</sup>[http://www.cvlibs.net/datasets/kitti/eval\\_odometry.php](http://www.cvlibs.net/datasets/kitti/eval_odometry.php)

<sup>3</sup><http://rec.ustc.edu.cn/share/62224f00-eb56-11eb-83a5-a167e10a5405>  
password: frws

<sup>4</sup><https://github.com/TixiaoShan/Stevens-VLP16-Dataset>



TABLE I  
THE 3D LIDAR SLAM BASELINES

Baselines	Characteristic	Proposed Year
LOAM [11]	Geometric points extraction and scan-to-map matching framework	2014
IMLS [19]	Specific points sampling and using of Implicit Moving Least Squares base scan-matching	2018
LeGO-LOAM [12]	Faster and similar accuracy as LOAM, and has better global mapping accuracy	2018
SuMa++ [6]	Surfel-based mapping and integrating semantic information	2019
ROI-SLAM [13]	A voxelized cube set is proposed to reserve the regions of interest exclusively	2020
POU-SLAM [20]	A global environment model with Partition of Unity implicit surface representation	2020
DGP-SLAM [14]	Using directed geometric points and sparse frame	2021
MULLS [5]	More sophisticated feature extraction and matching mechanism	2021
LiTAMIN2 [22]	symmetric KL-divergence is introduced to the ICP cost that reflects the difference between two probabilistic distributions.	2021
T-LOAM	A computationally efficient and robust lidar-only odometry framework based on truncated least squares	2021

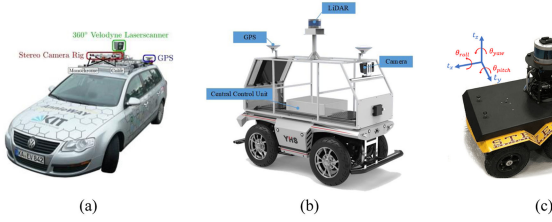


Fig. 2. The platforms of the datasets. (a) KITTI Dataset; (b) Our-VLP16-Dataset; (c) Stevens-VLP16-Dataset.

### A. Baselines

We compare our method with the state-of-the-art LiDAR SLAM systems, including feature-based and probabilistic distribution-based lidar odometry. Their SLAM systems are also included. The baselines and their characteristics are shown in Table I.

### B. Datasets

**KITTI-Dataset [24]:** KITTI odometry dataset provides 11 sequences (00-10) with ground truth trajectories. A Velodyne HDL-64E 3D laser scanner (10 Hz, 64 laser beams, 360° Field of View (FOV), range: 100 m) is used to collect data in multiple environments, including the urban (Seq. 00, 06, 07, 08), country (Seq. 02, 03, 04, 05, 09, 10), and highway scene (Seq. 01). This dataset is widely used for SLAM system evaluation.

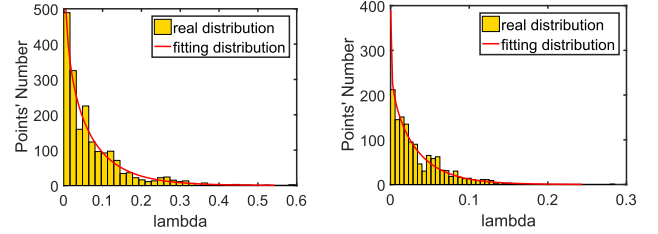
**USTC-VLP16-Dataset:** The dataset is gathered by us in the west campus of the University of Science and Technology of China (USTC). A Velodyne VLP-16 3D laser scanner is mounted on our mobile robot. We use the robot to collect lidar data in the campus scenarios, which contain structural and unstructured scenes.

**Stevens-VLP16-Dataset:** This dataset is captured using a Velodyne VLP-16, which is mounted on a UGV - Clearpath Jackal on the Stevens Institute of Technology campus.

The used mobile platforms of these datasets are shown in Fig. 2. Among the three datasets, only the KITTI-Dataset provides pose ground truth.

### C. Experimental Settings

Some hyper-parameters are involved in our lidar odometry and their settings are as follows: the default voxelization resolution  $\gamma = 0.15$  m, the variances of the defined target distributions in Eq. (11) and (16) are  $\sigma^2 = 0.01$ ,  $\sigma_d^2 = 0.25$ .

Fig. 3. The planarity ( $\bar{\lambda}_2$ ) distribution of the raw point cloud and the filtered point cloud by acceptance-rejection sampling. We can see that the points satisfying  $\bar{\lambda}_2 > 0.15$  are excluded in the filtered point cloud.

### D. Ablation Study

In this section, to validate the performance proposed modules, we perform an ablation study of our method, in which FastGICP [8] is used for comparison. The implementations of FastGICP are available from a public repository.<sup>5</sup>

1) *First-Step Point Filter Validation:* We perform three comparative experiments, including (1)GICP-N, the classic GICP method without covariance matrix regularization; (2) FastGICP, the GICP method that works as plane-to-plane ICP; (3) FasterGICP-1, the GICP method embeded with the first-step point filter. To illustrate the first-step filtering process, we randomly select one lidar scan and perform this filtering. We consider the distribution of  $N_m^1$ , which is defined as

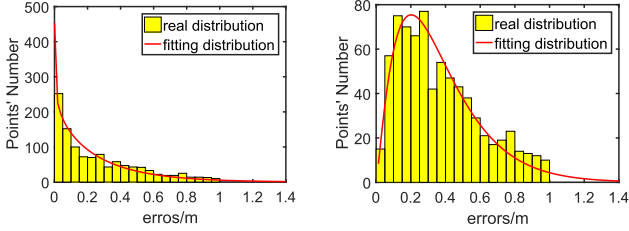
$$N_m^1 = |S|, \text{ where } S = \{\mathbf{p}_i | \bar{\lambda}_2^i \in [l_{m-1}, l_m]\}, \quad (22)$$

where  $[l_{m-1}, l_m]$  is a small interval and  $N_m^1$  is the number of the points whose  $\bar{\lambda}_2$  are within  $[l_{m-1}, l_m]$ . The distributions of  $N_m$  of the raw point cloud and filtered point cloud are shown in Fig. 3. We can tell that most points with less planarity (corresponding to large  $\bar{\lambda}_2$ ) are excluded in the filtered point cloud. The evaluation results are presented in Table II. Firstly, we can see that GICP with covariance matrix regularization is superior to GICP-N. The reason is as follows. Due to the sparsity and density variation of LiDAR points, the computed covariance matrix cannot represent the local geometry information accurately. In addition, it is appropriate to model the environment as a set of local plane surfels. In the last row, the ratio of the number of filtered points to total points are provided. In FasterGICP-1, less points are involved in the optimization process, thus the LiADR odometry efficiency is

<sup>5</sup>[https://github.com/SMRT-AIST/fast\\_gicp](https://github.com/SMRT-AIST/fast_gicp)

TABLE II  
THE VALIDATION RESULTS OF THE FIRST-STEP POINT FILTER ON THE KITTI SEQUENCES

	ATE [%]			Frequency [Hz]			Filtered Points/ Total Points
	GICP-N	GICP	FasterGICP-1	GICP-N	GICP	FasterGICP-1	
00	0.92	0.15	<b>0.11</b>	7.31	11.38	<b>13.17</b>	0.08
01	39.17	1.13	<b>0.85</b>	3.71	7.32	<b>9.72</b>	0.14
02	2.98	0.22	<b>0.21</b>	5.07	11.73	<b>14.34</b>	0.08
03	7.80	0.22	<b>0.22</b>	5.47	9.58	<b>11.62</b>	0.07
04	16.26	0.09	<b>0.08</b>	4.55	8.25	<b>10.71</b>	0.11
05	1.27	0.16	<b>0.16</b>	7.02	11.09	<b>12.64</b>	0.09
06	1.77	0.14	<b>0.10</b>	6.52	10.16	<b>12.33</b>	0.06
07	1.16	0.11	<b>0.10</b>	8.13	11.08	<b>11.88</b>	0.08
08	1.47	<b>0.16</b>	0.23	5.14	9.62	<b>11.69</b>	0.07
09	3.90	0.14	<b>0.13</b>	4.37	8.39	<b>10.11</b>	0.20
10	6.27	0.22	<b>0.19</b>	9.90	12.71	<b>13.20</b>	0.13
Avg.	7.54	0.24	<b>0.21</b>	6.10	10.11	<b>11.94</b>	0.1



(a) The matching error distribution of the raw point cloud. (b) The matching error distribution of the filtered point cloud.

Fig. 4. The matching error distribution of the raw point cloud and filtered point cloud.

TABLE III  
THE VALIDATION RESULTS OF THE SECOND-STEP POINT FILTER ON THE KITTI SEQUENCES

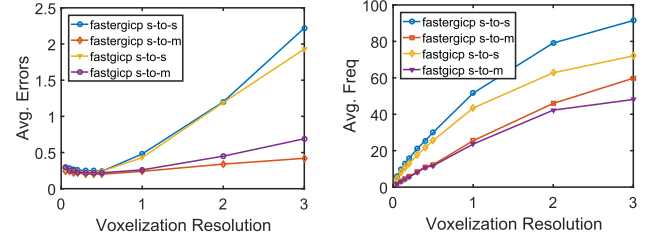
	ATE [%]		Frequency [Hz]		Improvements	
	FastGICP	FasterGICP-2	FastGICP	FasterGICP-2	APE	Frequency
00	<b>0.15</b>	0.19	11.38	<b>14.12</b>	-26%	+24%
01	1.13	<b>1.08</b>	7.32	<b>10.84</b>	+4%	+48%
02	<b>0.22</b>	0.25	11.73	<b>15.03</b>	-13%	+28%
03	0.22	<b>0.21</b>	9.58	<b>12.46</b>	+4%	+30%
04	0.09	<b>0.09</b>	8.25	<b>10.99</b>	0%	+33%
05	<b>0.16</b>	0.17	11.09	<b>12.30</b>	-6%	+10%
06	0.14	<b>0.13</b>	10.16	<b>12.52</b>	+7%	+23%
07	0.11	<b>0.09</b>	11.08	<b>13.70</b>	+18%	+23%
08	<b>0.16</b>	0.17	9.62	<b>12.13</b>	-6%	+26%
09	0.14	<b>0.12</b>	8.39	<b>10.55</b>	+14%	+25%
10	0.22	<b>0.20</b>	12.71	<b>15.51</b>	+9%	+22%
Avg.	0.24	<b>0.24</b>	10.11	<b>12.74</b>	0%	+26%

significantly increased. Furthermore, since the excluded points are outliers according to the regularization error, the LiADR odometry precision is not degraded by the first-step filter. The regions that cannot be represented by the plane surfels are excluded.

2) *Second-Step Point Filter Validation*: Compared with FastGICP, only the second-step point filter is used in FasterGICP, named FasterGICP-2. For one lidar scan, we consider the distribution of  $N_m^2$ , which is defined as

$$N_m^2 = |S|, \text{ where } S = \{\mathbf{p}_i | d_i \in [l_{m-1}, l_m]\}, \quad (23)$$

where  $d_i$  is the corresponding matching error of  $\mathbf{p}_i$ . The distributions of  $N_m^2$  of the raw point cloud and the filtered point cloud are shown in Fig. 4. After the second-step point filtering, the distribution of  $N_m^2$  can be fitted with the chi square distribution. We can tell that the points with minor matching errors tend to be excluded in the filtered point cloud. The lidar odometry results of FastGICP and FasterGICP-2 are shown in Table III. We can tell that the application of the second-step point filter can achieve



(a) The matching errors (m) comparison results. (b) The running efficiency (Hz) comparison results.

Fig. 5. The performance evaluation results of FasterGICP and FastGICP using different voxelization resolutions, where s-to-s means scan-to-scan matching strategy and s-to-m means scan-to-model matching strategy.

efficiency improvements without degrading the pose estimation accuracy.

3) *Validation of the Scan-to-Model Strategy*: In this paper, we also perform the experimental validation of the scan-to-model strategy. The performance evaluation results of the FasterGICP with the two mechanisms are shown in Fig. 5. Three conclusions can be drawn from the results.

- Firstly, considering the performance of (1) fastergicp s-to-s, and (2) fastergicp s-to-m, the scan-to-model strategy can greatly reduce the drift errors when the voxel size increases.
- Secondly, though the scan-to-model version of FasterGICP can also achieve real-time performance.
- Thirdly, when scan-to-model strategy is applied, our method can achieve better performance compared with fastgicp in terms of both efficiency and precision.

4) *Analysis of the Voxelization Resolution*: We compare the FasterGICP with FastGICP in terms of the performance under different voxel sizes, including 0.05 m, 0.1 m, 0.15 m, 0.2 m, 0.3 m, 0.4 m, 0.5 m, 1 m, 2 m, and 3 m. The results are shown in Fig. 5. We can tell the precision is inversely proportional to the resolution while the running frequency is oppsite. When using scan-to-model strategy, the increase of resolution leads to insignificant precision loss. Thus, if we put more importance on the efficiency, the resolution can be set to a larger value. Furthermore, when larger voxelization resolution is used, our method can achieve much more performance improvement compared with FastGICP.

#### E. Performance Evaluation on the Datasets

1) *Performance Evaluation on KITTI*: The precision of FasterGICP is evaluated on the KITTI odometry benchmark [23]. The competitors are the state-of-the-art methods presented in Section IV-A. The evaluation results of all sequences are presented in Table IV. Our method has two versions, including scan-to-scan FasterGICP and scan-to-model FasterGICP. Concerning pure odometry without corrections through loop closed, it can be concluded that our method can achieve superior precision than others. Compared with the scan-to-scan version, the scan-to-model version of our method can achieve more accuracy improvements. The total length of the trajectory of each sequence is also presented. The estimated trajectories by our lidar odometry method are also given in Fig. 6.

TABLE IV  
ATE [%] EVALUATION AND COMPARISON OF THE LIDAR ODOMETRY METHODS AND SLAM SYSTEMS ON THE KITTI DATASET

	00	01	02	03	04	05	06	07	08	09	10
LOAM [11]	0.78	1.43	0.92	0.86	0.71	0.57	0.65	0.63	1.12	0.77	0.80
A-LOAM	0.79	1.96	4.57	0.95	0.77	0.50	0.62	0.45	1.11	0.74	1.01
LeGO-LOAM [12]	1.38	28.03	2.14	1.21	1.27	0.91	0.80	0.74	1.40	1.25	1.70
LeGO-LOAM wo loop closure	1.35	27.43	2.09	1.24	1.25	0.89	0.87	0.77	1.34	1.28	1.68
SUMA++ [6]	0.64	1.60	1.00	0.67	0.37	0.40	0.46	0.34	1.10	0.47	0.66
LiTAMIN2 [22]	0.70	2.10	0.98	0.96	1.05	0.45	0.59	0.44	0.95	0.69	0.80
MULLS-LO [5]	0.51	0.62	0.55	0.61	0.35	0.28	0.24	0.29	0.80	0.49	0.61
MULLS-SLAM [5]	0.54	<b>0.62</b>	0.69	0.61	0.35	0.29	0.29	0.27	0.83	0.51	0.61
DGP-SLAM [14]	0.85	1.94	0.96	0.79	0.71	0.42	0.44	0.44	1.00	0.73	1.09
DGP-SLAM wo loop closure	0.89	1.94	0.99	0.79	0.71	0.43	0.49	0.58	0.97	0.66	1.09
ROI-SLAM [13]	1.23	14.02	7.01	1.18	0.36	0.76	0.86	1.09	3.68	1.28	1.47
POU-SLAM [20]	0.64	0.90	0.74	0.59	0.49	0.43	0.36	0.35	0.84	0.53	0.83
Ours (Scan-to-scan)	0.15	0.95	0.24	0.24	<b>0.07</b>	0.24	0.13	0.12	0.30	0.27	<b>0.18</b>
Ours (Scan-to-model)	<b>0.11</b>	0.92	<b>0.23</b>	<b>0.20</b>	0.11	<b>0.11</b>	<b>0.07</b>	<b>0.07</b>	<b>0.12</b>	<b>0.13</b>	0.22
Total Length [m]	3742.91	2406.59	5071.16	566.363	392.47	2216.83	1234.30	697.54	3239.80	1709.04	924.95

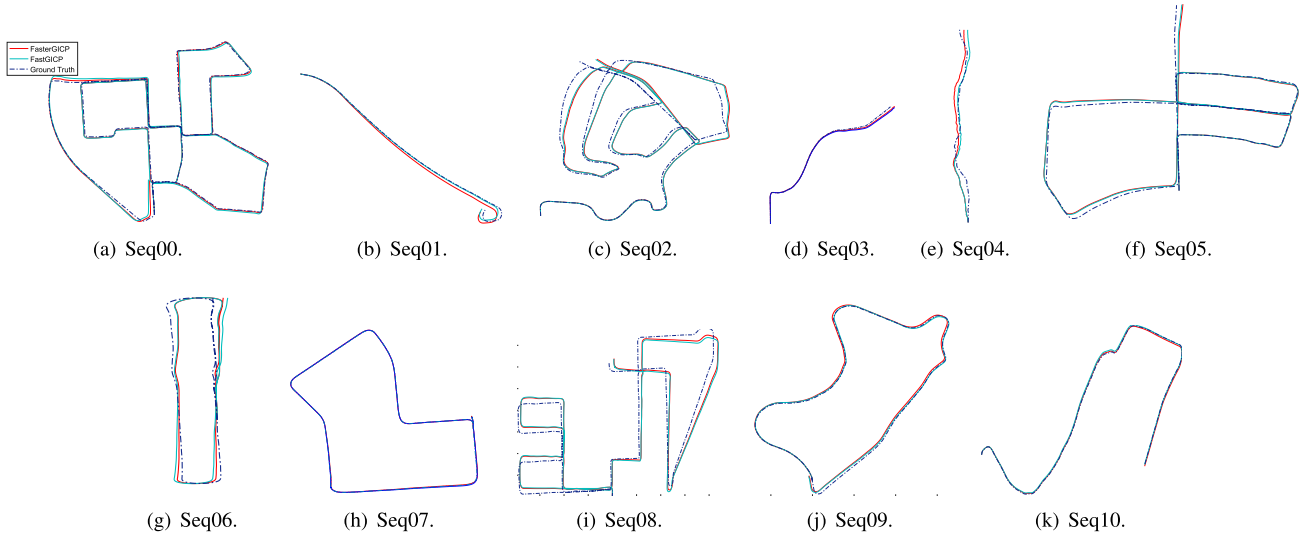
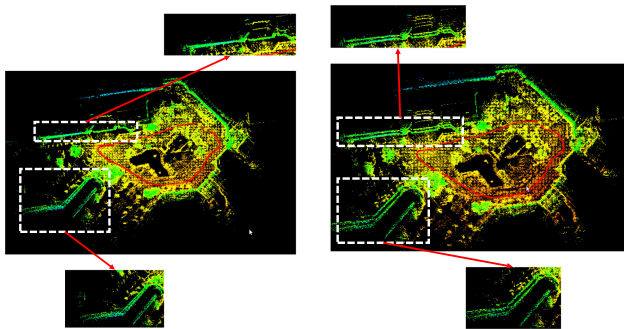


Fig. 6. The estimated trajectories of the FastGICP and FasterGICP on the KITTI dataset. The ground truth trajectories are also presented.



(a) The mapping and lidar odometry results of FasterGICP. (b) The mapping and lidar odometry results of FastGICP.

Fig. 7. The mapping results of our campus. The estimated trajectories are shown in red.

2) *Performance Evaluation on USTC-VLP16-Dataset:* The mapping results of one area in our campus are shown in Fig. 7. In this paper, the map is constructed by aggregating the lidar

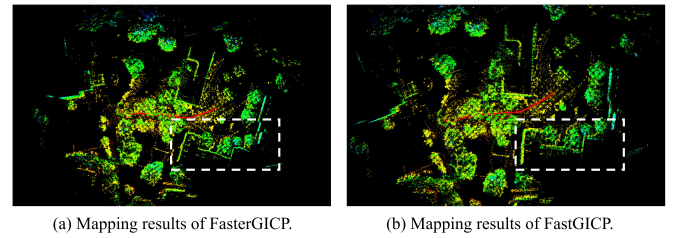


Fig. 8. The mapping results of the 02 sequence in the Stevens-VLP16-Dataset. The estimated trajectories are shown in red.

scans according to the lidar odometry outputs without bundle adjustment and loop closure. We can tell that our mapping results demonstrate less distortions compared with that of the FastGICP.

3) *Performance Evaluation on the Stevens-VLP16-Dataset:* The mapping and lidar odometry results of the FasterGICP and FastGICP on one sequence in the Stevens Dataset are shown in Figs. 8 and 9. We can tell that our method can achieve more consistent mapping result.

TABLE V  
THE EFFICIENCY COMPARISON RESULTS OF THE LIDAR ODOMETRY METHODS [HZ]

Method	00	01	02	03	04	05	06	07	08	09	10
LiTAMIN2	69.9	21.18	61.56	51.56	25.46	<b>74.15</b>	38.19	<b>76.92</b>	<b>49.30</b>	47.59	<b>73.80</b>
LiTAMIN	12.40	21.18	11.14	11.59	6.37	12.88	8.26	13.38	11.07	9.86	13.18
SuMa(Scan-to-Model)	12.89	12.75	16.64	17.52	12.12	13.27	13.30	15.64	14.97	13.80	13.85
LeGO-LOAM	16.36	20.97	16.34	16.78	15.16	16.46	15.59	17.48	17.01	16.62	16.77
LOAM	2.57	4.07	2.63	2.77	2.38	2.50	2.54	1.41	1.46	1.13	1.45
FasterGICP(scan-to-model)	<b>71.5</b>	<b>55.82</b>	<b>76.4</b>	<b>58.89</b>	<b>43.07</b>	67.35	<b>60.87</b>	61.01	46.33	<b>51.01</b>	65.21

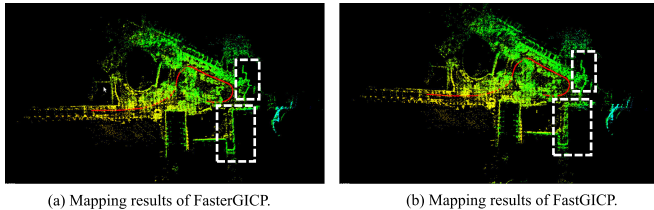


Fig. 9. The mapping results of the 06 and 07 sequences in the Stevens-VLP16-Dataset. The estimated trajectories are shown in red.

### F. Efficiency Analysis

Then, we compare our method with the state-of-the-art methods in terms of the efficiency. The voxel size of FasterGICP is set to 3 m. The results are shown in Table V. We can see that the FasterGICP can achieve competitive performance compared with the existing state-of-the-art methods in terms of efficiency. We also notice that the LiTAMIN2 achieves higher running speed than our method on some sequences. They reduce the number of involved points through voxelization and promote the precision by introducing symmetric KL-divergence. Among the existing methods, distribution-based LiADR odometry combined with large voxelization resolution can effectively achieve high running speed.

## V. CONCLUSION

In this paper, we propose qualitative definitions of outliers and redundant points in LiDAR odometry. Based on these definitions, probabilistic filters are proposed to select and remove these points. Experimental results on the public and collected datasets demonstrate that our method can achieve significantly efficiency improvements without reducing accuracy. In addition to the proposed filtered points, we believe that more filtering conditions can be added to FasterGICP, which are determined by the users and specific requirements.

## REFERENCES

- [1] A. Li, J. Wang, M. Xu, and Z. Chen, "DP-SLAM: A visual SLAM with moving probability towards dynamic environments," *Inf. Sci.*, vol. 556, pp. 128–142, 2021.
- [2] J. Wang, P. Wang, D. Dai, M. Xu, and Z. Chen, "Regression forest based RGB-D visual relocalization using coarse-to-fine strategy," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4431–4438, Jul. 2020.
- [3] J. Wang, P. Wang, and Z. Chen, "A novel qualitative motion model based probabilistic indoor global localization method," *Inf. Sci.*, vol. 429, pp. 284–295, 2018.
- [4] Y. Cong, C. Chen, J. Li, W. Wu, S. Li, and B. Yang, "Mapping without dynamic: Robust LIDAR-SLAM for ugv mobile mapping in dynamic environments," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 515–520, 2020.
- [5] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "MULLS: Versatile LiDAR SLAM via multi-metric linear least square," *Proc. IEEE Int. Conf. Robot. Automat.*, 2021.
- [6] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "SuMa: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2019, pp. 4530–4537.
- [7] M. Oelsch, M. Karimi, and E. Steinbach, "R-LOAM: Improving LiDAR odometry and mapping with point-to-mesh features of a known 3D reference object," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2068–2075, Apr. 2021.
- [8] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for fast and accurate 3D point cloud registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11 054–11059.
- [9] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," *Robot.: Sci. Syst.*, vol. 2, no. 4, 2009, pp. 435–442.
- [10] B. D. Flury, "Acceptance-rejection sampling made easy," *SIAM Rev.*, vol. 32, no. 3, pp. 474–476, 1990.
- [11] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Auton. Robots.*, vol. 41, no. 2, pp. 401–416, 2016.
- [12] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.
- [13] Z. Zhou, M. Yang, C. Wang, and B. Wang, "ROI-cloud: A key region extraction method for LIDAR odometry and localization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3312–3318.
- [14] S. Liang, Z. Cao, C. Wang, and J. Yu, "A novel 3D LIDAR SLAM based on directed geometry point and sparse frame," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 374–381, Apr. 2021.
- [15] P. J. Besl and N. D. McKay, "A Method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [16] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. IEEE 3rd Int. Conf. 3-D Digit. Imag. Model.*, 2001, pp. 145–152.
- [17] J. Serafini and G. Grisetti, "NICP: Dense normal based point cloud registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 742–749.
- [18] T. Schöps, T. Sattler, and M. Pollefeys, "Surfelmeshing: Online surfel-based mesh reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2494–2507, Oct. 2020.
- [19] J.-E. Deschaud, "IMLS-SLAM: Scan-to-model matching based on 3D data," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2480–2485.
- [20] J. Jiang, J. Wang, P. Wang, and Z. Chen, "Pou-slam: Scan-to-model matching based on 3D voxels," *Appl. Sci.*, vol. 9, no. 19, pp. 4147–4160, 2019.
- [21] I. Vizzo, X. Chen, N. Chebrolu, J. Behley, and C. Stachniss, "Poisson surface reconstruction for LiDAR odometry and mapping," in *Proc. IEEE Intl. Conf. Robot. Automat.*, 2021.
- [22] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "LiTAMIN2: Ultra light lidar-based slam using geometric approximation applied with KL-divergence," *Proc. IEEE Int. Conf. Robot. Automat.*, 2021.
- [23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
- [24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.