Feature- and Distribution-Based LiDAR SLAM With Generalized Feature Representation and Heuristic Nonlinear Optimization

Jikai Wang, Meng Xu, Guangpu Zhao, and Zonghai Chen, *Senior Member, IEEE*

***Abstract*— In this article, we propose a novel 3-D light detection and ranging (LiDAR) simultaneous localization and mapping (SLAM) method, which contains feature-based fast scan matching, distribution-based keyframe matching, and loop closure. First, by incorporating feature-based and distribution-based LiDAR scan matching into our SLAM system, we combine the advantages of the two mechanisms and formulate a more efficient and precise front end. Second, considering the local surface reconstructing errors caused by the existing feature representation mechanisms, we propose a generalized representation mechanism for geometry feature representation within the distribution-based framework. Third, due to the existence of noisy point correspondences during the pose estimation optimization process, we propose a novel heuristic knowledge to distinguish high-quality matching error items and put more weights on them during the iterative objective function construction process. Extensive experiments are conducted on public and custom datasets, including KITTI, Mulran, NCLT, Stevens, and USTC-VLP16 datasets collected by our mobile platform. The experimental results demonstrate that the proposed mechanisms can effectively promote the LiDAR SLAM system performance and demonstrate competitive performance compared with state-of-the-art methods. We will make our source code open to serve as a new baseline for the robotic community.**

***Index Terms*— 3-D light detection and ranging (LiDAR) simultaneous localization and mapping (SLAM), distribution model, geometry feature, heuristic nonlinear optimization, LiDAR odometry.**

I. Introduction

IN RECENT years, 3-D light detection and ranging (LiDAR) simultaneous localization and mapping (SLAM) has been intensively studied and many sophisticated SLAM systems have been proposed, including LOAM [1], LeGO-LOAM [2], MULLS [3], DGP-SLAM [4], SuMa++ [5], FasterGICP [6], and S4-SLAM [7]. Given the raw LiDAR point clouds, various mechanisms for feature extraction, association, and pose transformation optimization are studied and have come to the fore in pushing the boundaries of LiDAR SLAM systems in different environments.

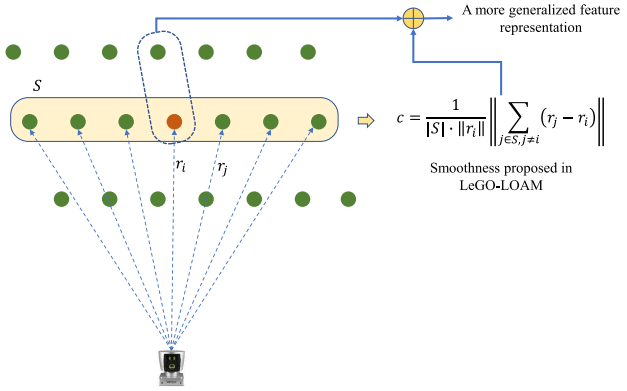


Fig. 1. Illustration of the smoothness calculation and our proposed enhanced feature representation.

Though the 3-D LiDAR sensor is well known for its robust and stable range perception, the LiDAR points demonstrate high sparsity and density variation. Performing feature extraction, point representation, and feature association on the raw point clouds remains challenges. Generally, the existing state-of-the-art LiDAR SLAM methods can be classified into two categories, including feature- and distribution-based LiDAR SLAM systems. The key motivation of the feature-based LiDAR SLAM systems is to extract geometry informative points and apply them to perform LiDAR scan matching. Specifically, LOAM and LeGO-LAOM define edge points and planar points according to smoothness values. MULLS and DGP-SLAM extract geometrical features using singular value decomposition (SVD) operation, and more sophisticated geometrical features are developed. Though SVD-based feature extraction mechanisms can obtain more regular features and identify dominant directions, smoothness-based mechanisms are more efficient. However, the widely used smoothness values are calculated along the xy plane in the LiDAR coordinates. The lack of geometry information description along the z-axis tends to cause feature alignment errors. An illustration of the smoothness value calculation is presented in Fig. 1. In this article, both geometry information on the xy plane and along the z-axis are considered and described.

Distribution-based SLAM systems introduce a probabilistic distribution model to represent the local geometry structure

Manuscript received 14 July 2022; revised 9 October 2022; accepted 1 November 2022. Date of publication 17 November 2022; date of current version 17 January 2023. This work was supported by the National Natural Science Foundation of China under Grant 62103393. The Associate Editor coordinating the review process was Dr. Guvenir Kaan Esen. *(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@mail.ustc.edu.cn; xm1996@mail.ustc.edu.cn; zhaoguangpu9807@mail.ustc.edu.cn; chenzh@ustc.edu.cn).

Digital Object Identifier 10.1109/TIM.2022.3223154

of each LiDAR point. These systems can achieve more accurate localization and environment mapping by performing point-to-distribution or distribution-to-distribution-based scan matching. However, neighbor points searching and SVD are involved, making the scan matching process time-consuming [8]. Among the existing methods, point cloud voxelization and parallel computation are widely used for system acceleration. However, distribution-based methods are still less efficient than feature-based methods. Furthermore, using Gaussian distribution to describe the local surface approximately exhibits inherent errors. The widely used Generalized-ICP [6] has been well proved that making GICP work as plane-ICP can effectively improve the matching performance, which indicates that the plane model is better than the raw covariance in terms of local structure description accuracy.

Besides feature representation, among the existing LiDAR SLAM systems, nonlinear optimization framework-based (including Gaussian–Newton method and Levenberg–Marquardt method) pose optimization module has been well developed and widely used. However, during the optimization process, the feature correspondences are obtained according to the initial pose guess, which is biased. Thus, the feature correspondence set contains true positive and false positive correspondences. Though kernel functions are applied in some systems, false positive correspondences cannot be effectively distinguished since their matching errors are small. Performing pose optimization with false positive correspondences significantly limits the pose estimation accuracy. The importance of false positive correspondences should be decreased during the optimization process.

In this article, we promote the LiDAR SLAM system in terms of precision and efficiency. In order to combine the high efficiency of feature-based mechanisms and the accuracy of distribution-based mechanisms, we propose a novel SLAM front-end module, in which fast pose tracking is performed using geometry feature and distribution-based keyframe matching is then implemented to correct the drift errors caused by the fast pose tracking. Furthermore, to alleviate the feature representation errors, a more generalized distribution-based geometry feature representation mechanism is proposed. Considering the deficiency of the nonlinear optimization mechanism, we develop a self-adaptive heuristic optimization mechanism based on a proposed prior qualitative knowledge, which can increase the importance of true positive correspondences during the objective function constructing process. Our method is developed toward a more balanced SLAM system and has been verified on various datasets. As an illustration, our mapping results and trajectory estimation on KITTI Seq00 are shown in Figs. 2 and 3, respectively.

The main contributions of this article are given as follows.

1) We propose a 3-D LiDAR SLAM method that incorporates feature-based odometry and distribution-based odometry into one framework, making it possible to combine both advantages of the two mechanisms.
2) Novel environmental local surface reconstructing algorithms are proposed for feature-based scan matching, which is more adaptable to the 3-D LiDAR scanning mode. We can effectively improve the feature-based scan-matching performance based on the proposed distribution-based representation models without sophisticated and complex mechanism design.
3) A heuristic nonlinear optimization mechanism is proposed and improves the GICP performance effectively.

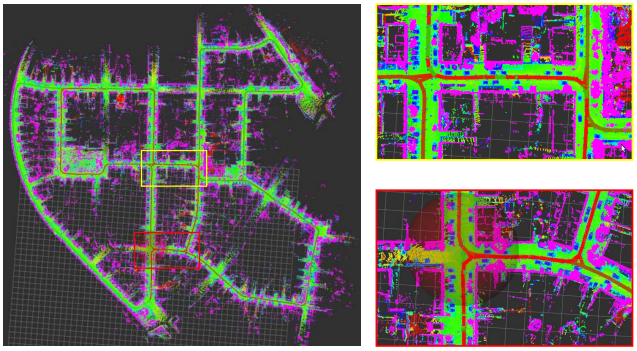


Fig. 2. Our SLAM system estimation results on the urban scenario (KITTI seq.00). The details of loop closure areas are also presented.

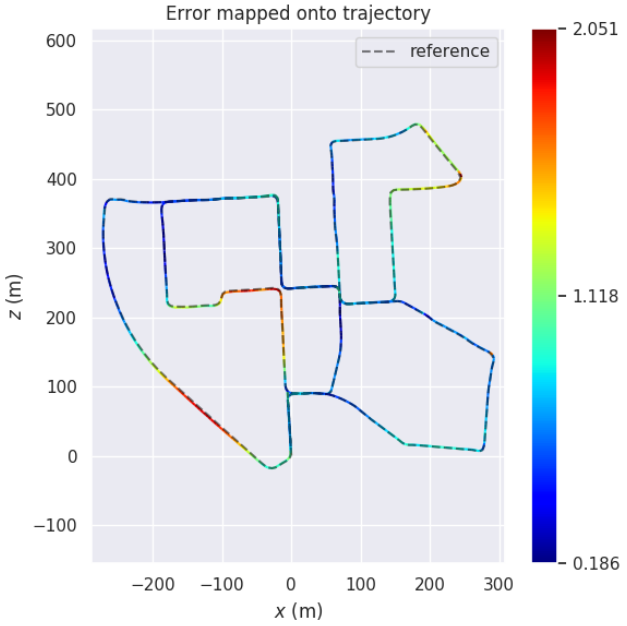


Fig. 3. Our SLAM system trajectory estimation results on the urban scenario (KITTI seq.00). Ground truth is also provided as a reference. Our trajectory estimation result is highly consistent with the ground truth.

The rest of this article is organized as follows. Section II presents related work. Section III presents the problem formulation. Section IV gives details of the proposed SLAM method. The experimental evaluations are shown in Section V. Section VI concludes this article.

II. Related Work

A. Feature-Based LiDAR SLAM Methods

LOAM [9] and its variants [2] perform point feature-to-edge/plane matching to align LiDAR scans. Smoothness-based

feature extraction mechanisms are proposed and exploited in their systems. These mechanisms are easy to implement and adaptable to multiple types of environments. Instead of representing geometry information at the point level, DGP [4] computes the key geometry factor (direction) using principal component analysis (PCA) and adds this factor to the geometry points. Their work is the most related to our odometry, while our point direction is derived from the LiDAR beam scanning direction. PCA is also directly applied to extract line and plane features in MULLS [3]. MULLS distinguishes more different geometry features, and the abundant classified features include ground, facade, pillar, beam, and so on. Combined with a multimetric linear least square-ICP, MULLS ranks top 10 on the competitive KITTI benchmark.

B. Distribution-Based LiDAR SLAM Methods

3-D Haussian distribution

体素是三维空间中的最小单位，而像素是二维空间中的最小单位。

NDT-SLAM [10] is the most typical distribution-based LiDAR SLAM method. Their map consists of many voxels. In each voxel, a 3-D Gaussian distribution is fit from map points lying in the voxel and is used to register a new point by maximizing the likelihood of the point being taken from the distribution. LiTAMIN2 [11] further improves the accuracy of NDT. During their scan-to-map optimization process, the cost includes the distance between points and differences between distribution shapes. Though covariance computation is time-consuming, distribution-based methods can be well applied to voxelized LiDAR frames. FastGICP [12] and its variants [6] propose acceleration mechanisms within the GICP framework and aim to make the SLAM systems more practical. GP-SLAM [13] is another recently proposed typical distribution-based LiDAR SLAM system. Instead of modeling environmental surfaces using Gaussian distributions, they perform surface dense fitting based on a regionalized Gaussian process, making it feasible to obtain rigid point-to-point correspondences.

Surfel，Surface Element，是表面体素的缩写

C. Feature- and Distribution-Based LiDAR SLAM Methods

Surfel can be regarded as a combination of plane feature- and distribution-based representation. It is typically used in Suma [14] and Suma++ [5]. Though surfel-based representation can model environmental surfaces more accurately than Gaussian distributions, maintaining dense surfels is highly time-consuming. To alleviate such a problem, Yuan et al. [15] constructed plane features in voxel map with adaptive voxel sizes. This adaptive strategy makes their method suitable for environments with varying structures while ensuring high computation efficiency. Koide et al. [8] proposed a feature- and distribution-based LiDAR mapping system and they focused on performing real-time global matching cost minimization using GICP-based matching cost factor. They used MULLS as front end and GPU-based GICP in the backend. Their work is highly related to ours, while we focus on surface representation and pose optimization refinement within the two kinds of scan-matching frameworks.

III. Problem Formulation

In the LiDAR SLAM system, loop detection and global graph optimization have been well developed. The core of the SLAM system is LiDAR scan matching. Specifically, it can be formulated as

$$\hat{\mathbf{T}}_n^m = \arg\min_{\mathbf{T}_n^m} \sum d\left(\mathbf{T}_n^m \circ \mathcal{S}_n, \mathcal{S}_m\right) \tag{1}$$

where $\mathcal{S}_n$ and $\mathcal{S}_m$ are LiDAR scans for matching, $\mathbf{T}_n^m$ is the pose transformation matrix to be estimated, and $d(\cdot)$ is the user-defined point matching distance function. $\mathbf{T}_n^m \circ \mathcal{S}_n$ represents performing the following transformation for each point in $\mathcal{S}_n$:

$$\begin{aligned} \mathbf{T}_n^m \circ \mathbf{p}_n^i &= \mathbf{R}_n^m \cdot \mathbf{p}_n^i + \mathbf{t}_n^m \\ \mathbf{p}_n^i &= \left(x_n^i, y_n^i, z_n^i\right)^T \in \mathcal{S}_n \end{aligned} \tag{2}$$

where $\mathbf{R}_n^m$ and $\mathbf{t}_n^m$ are the rotational and translational part of $\mathbf{T}_n^m$, respectively. The widely used distance functions include point-to-point distance [16], point-to-edge distance [2], [17], point-to-plane distance [18], point-to-mesh distance [19], and point-to-distribution distance [3], [12].

Considering the point-to-edge distance, the specific formulation of (1) is denoted as

$$\hat{\mathbf{T}}_n^m = \arg\min_{\mathbf{T}_n^m} \sum d\left(\mathbf{p}_n^i, \text{edge}\left(\mathbf{p}_m^i\right)\right) \tag{3}$$

where $\mathbf{p}_m^i \in \mathcal{S}_m$ is the correspondence of $\mathbf{p}_n^i \in \mathcal{S}_n$ and $\text{edge}(\mathbf{p}_m^i)$ is the edge feature representation. The estimation of $\text{edge}(\mathbf{p}_m^i)$ can be formulated as $\hat{\text{edge}}(\mathbf{p}_m^i)$. The edge feature representation is generally derived from nearest neighbor searching or PCA in feature-based methods. Both kinds of representations require neighbor points searching. However, due to the sparsity of LiDAR points, the searched nearest neighbor points do not always satisfy the assumption that they are located on the same edge entity, which indicates the inherent geometry estimation error and thus leads to the following error:

feature-based

$$E_f = d\left(\mathbf{p}_n^i, \text{edge}\left(\mathbf{p}_m^i\right)\right) - d\left(\mathbf{p}_n^i, \hat{\text{edge}}\left(\mathbf{p}_m^i\right)\right) \tag{4}$$

which further leads to the distortion of the final pose estimation $\hat{\mathbf{T}}_n^m$.

Another kind of error exists in the distribution-based scan-matching process. Generalized-ICP [12] is the widely used classical distribution-based method. Its core formulation is given as

$$\mathbf{T} = \arg\min_{\mathbf{T}} \sum_i \mathbf{d}_i^T \left(\hat{\mathbf{C}}_i^m + \mathbf{T}\hat{\mathbf{C}}_i^n \mathbf{T}^T\right)^{-1} \mathbf{d}_i \tag{5}$$

where $\hat{\mathbf{C}}_i^m$ and $\hat{\mathbf{C}}_i^n$ are the estimated covariance matrices of $\mathbf{p}_i^m$ and $\mathbf{p}_i^n$ derived from their neighbor points, respectively, and $\mathbf{d}_i$ is $\mathbf{p}_i^m - \mathbf{T} \circ \mathbf{p}_i^n$. Such kinds of methods are based on one assumption that $\mathbf{p}_i^m$ and $\mathbf{p}_i^n$ are the true positive correspondence. However, due to the LiDAR points' sparsity and pose initialization errors, such an assumption cannot be well satisfied. Wrong correspondences tend to make the objective function distorted and thus lead to pose optimization errors. Though some methods [5], [20] introduce extra information, such as semantic information, to distinguish good correspondences, time efficiency is degraded and system complexity is increased. How to distinguish good correspondences by making full use of the existing information during the optimization process is important for efficient and accurate pose estimation.

distribution-based

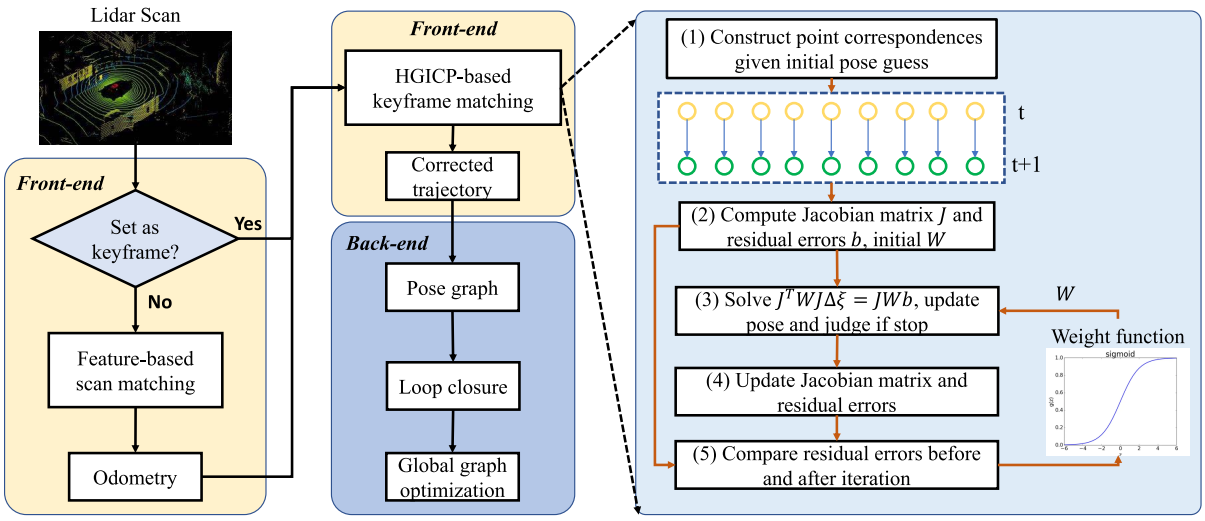


Fig. 4. Proposed feature and distribution-based LiDAR SLAM system framework.

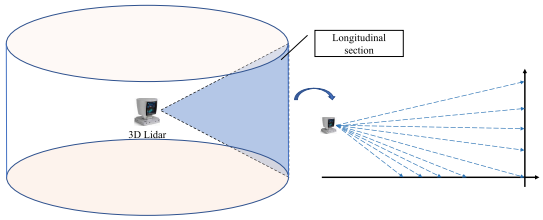


Fig. 5. Illustration of 3-D LiDAR observation and one longitudinal section.

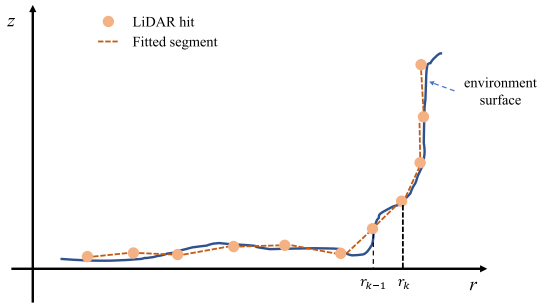


Fig. 6. Illustration of surface reconstruction on the longitudinal slice.

IV. Proposed Method

A. System Framework

Our 3-D LiDAR SLAM system consists of four modules: feature-based fast pose tracking, distribution-based keyframe matching, loop detection, and global graph optimization. The feature-and-distribution-based front end is the core of our system. The overall framework of our method is shown in Fig. 4.

B. LiDAR Feature-Based Fast Pose Tracking

1) Environment Surface Reconstruction Principle and Criteria: As shown in Fig. 5, the 3-D field of view (FOV) of LiDAR sensor can be represented as a cylinder. We first study the environmental surface representation from a longitudinal section of the cylinder FOV. The environment curve in the longitudinal section can be regarded as a function and denoted as

$$z = F(r, \theta) \tag{6}$$

where r is the distance to the center point and θ is the yaw angle that is fixed for a longitudinal slice. Essentially, the LiDAR points can be regarded as discrete function samples. Surface representation aims to regress $F(\cdot)$ based on the sparse LiDAR points, which can be defined as

$$\hat{F}(\cdot) = \mathcal{F}\left(\mathbf{p}_1^t, \ldots, \mathbf{p}_k^t\right) \tag{7}$$

and the evaluation criterion is defined as

$$\rho = \sum_{\mathbf{p}_i^{t+1} \in \mathcal{S}_l} d\left(\mathbf{p}_i^{t+1}, \hat{F}(\cdot)\right) \tag{8}$$

where $\mathcal{S}_l$ is the LiDAR point set that is located on the slice in the next LiDAR frame. The smaller the indicator ρ, the more accurate the surface reconstruction.

2) Surface Reconstruction on the Longitudinal Slice: Piecewise linear functions are most widely used for fast nonlinear curve fitting. In this article, we perform interpolating linearly between the LiADR points to approximate the environmental curve. A general curve fitting is shown in Fig. 6, which can be formulated as

$$\mathcal{F}(\mathbf{p}_1, \ldots, \mathbf{p}_k) = \begin{cases} a_1 \cdot r + b_1, & r_0 < r < r_1 \\ \cdots\cdots & \\ a_{k-1} \cdot r + b_{k-1}, & r_{k-1} < r < r_k \end{cases} \tag{9}$$

where a_i and b_i are the corresponding slope and intercept derived from the endpoints $\{\mathbf{p}_i^t, \mathbf{p}_{i+1}^t\}$, respectively. The proposed surface reconstruction model is represented as

$$\mathcal{F}(\mathbf{p}_1, \ldots, \mathbf{p}_k) = \begin{cases} U(a_1 \cdot r + b_1), & r_0 < r < r_1 \\ \cdots\cdots & \\ U(a_k \cdot r + b_k), & r_{k-1} < r < r_k \end{cases} \tag{10}$$

where $U(\cdot)$ is defined as

$$U(a_i \cdot r + b_i) = \mathcal{N}(0.5 \cdot (\mathbf{p}_k + \mathbf{p}_{k-1}), \Sigma_{k-1}) \tag{11}$$

and

$$\Sigma_{k-1} = [\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \cdot [\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]^T. \tag{12}$$

We define the vectors as follows:

$$\begin{aligned} \mathbf{v}_0 &= (\mathbf{p}_k - \mathbf{p}_{k-1})/||\mathbf{p}_k - \mathbf{p}_{k-1}||_2 \\ \mathbf{v}_1 &= \mathbf{v}_0 \times \mathbf{v}_c, \quad c \in \text{Coor} = \{x, y, z\} \\ \text{s.t. } c &= \arg\min_{c \in \text{Coor}} \mathbf{v}_0 \cdot \mathbf{v}_c \\ \mathbf{v}_2 &= \mathbf{v}_0 \times \mathbf{v}_1 \end{aligned} \tag{13}$$

where $\mathbf{v}_c, c \in \text{Coor} = \{x, y, z\}$ corresponds to the unit vectors along the x-axis, y-axis, and z-axis, respectively, and $\mathbf{v}_0$ represents the dominant direction of the linear distribution. For each small line segment, we construct a Gaussian representation to describe its linear feature. Since we derive the covariance matrix in a forward way and no SVD operation is involved, less computational cost is introduced. For the sake of expression simplification, we use

$$U(\mathbf{p}_{k-1}) = (\mu_{k-1}, \Sigma_{k-1}) \tag{14}$$

to denote $\mathcal{N}(0.5 \cdot (\mathbf{p}_k + \mathbf{p}_{k-1}), \Sigma_{k-1})$.

3) Feature Selection Based on Smoothness: We further filter the raw LiDAR point cloud based on the defined smoothness to generate a less redundant point cloud. We apply the same smoothness formulation proposed in LeGO-LOAM. Then, one feature point is formulated as

$$\mathbf{\Theta}_i = [\mathbf{p}_i, U(\mathbf{p}_i), c_i] \tag{15}$$

where c_i is the smoothness value of $\mathbf{p}_i$.

According to the smoothness, the extracted planar point set $\mathbf{P}_t$ is represented as

$$\mathbf{P}_t = \{\mathbf{\Theta}_i | c_i < p_{\text{th}}\} \tag{16}$$

where p_{th} is the smoothness threshold for distinguishing planar points. The extracted edge point set $\mathbf{L}_t$ is represented as

$$\mathbf{L}_t = \{\mathbf{\Theta}_i | c_i > e_{\text{th}}\} \tag{17}$$

where e_{th} is the smoothness threshold for distinguishing edge points.

4) Fast Pose Tracking Based on the Feature Point Cloud: The fast pose tracking is achieved by aligning $\mathbf{L}_{t-1}$ and $\mathbf{L}_t$, $\mathbf{P}_{t-1}$ and $\mathbf{P}_t$. For points in $\mathbf{L}_t$ and $\mathbf{P}_t$, we need to find their corresponding points from feature sets $\mathbf{L}_{t-1}$ and $\mathbf{P}_{t-1}$.

Specifically, for each local edge representation model denoted as

$$\mathbf{\Theta}_i^t = [\mathbf{p}_i^t, U(\mathbf{p}_i^t), c_i^t] \in \mathbf{L}_t \tag{18}$$

we first transform $\mathbf{p}_i^t$ into the coordinates of $\mathbf{L}_{t-1}$ using $\hat{\mathbf{T}}_{t-1}^t$, which is initialized according to the uniform motion model. The closest point of the transformed point in $\mathbf{L}_{t-1}$ is defined as its correspondence, called $\mathbf{p}_i^{t-1}$. Its corresponding edge representation model is

$$\mathbf{\Theta}_i^{t-1} = [\mathbf{p}_i^{t-1}, U(\mathbf{p}_i^{t-1}), c_i^{t-1}] \in \mathbf{L}_{t-1}. \tag{19}$$

Then, the edge-to-edge matching error is defined based on the representation model

$$\mathbf{e}_i^l = \left(\mathbf{T}_t^{t-1} \circ \mu_i^t - \mu_i^{t-1}\right)^T \cdot \left(\mathbf{T}_t^{t-1} \cdot \Sigma_i^t \cdot \left(\mathbf{T}_t^{t-1}\right)^T + \Sigma_i^{t-1}\right)^{-1} \times \left(\mathbf{T}_t^{t-1} \circ \mu_i^t - \mu_i^{t-1}\right). \tag{20}$$

For each local surface representation model denoted as

$$\mathbf{\Theta}_i^t = \left[\mathbf{p}_i^t, U\left(\mathbf{p}_i^t\right), c_i^t\right] \in \mathbf{P}_t \tag{21}$$

we transform it into the coordinates of $\mathbf{P}_{t-1}$ using $\hat{\mathbf{T}}_t^{t-1}$, and two closest points in $\mathbf{P}_{t-1}$ are defined as its correspondences, denoted as $\mathbf{p}_{i1}^{t-1}$ and $\mathbf{p}_{i2}^{t-1}$. Their corresponding surface representation models are

$$\mathbf{\Theta}_{i1}^{t-1} = \left[\mathbf{p}_{i1}^{t-1}, U\left(\mathbf{p}_{i1}^{t-1}\right), c_{i1}^{t-1}\right] \in \mathbf{P}_{t-1} \tag{22}$$

and

$$\mathbf{\Theta}_{i2}^{t-1} = \left[\mathbf{p}_{i2}^{t-1}, U\left(\mathbf{p}_{i2}^{t-1}\right), c_{i2}^{t-1}\right] \in \mathbf{P}_{t-1}. \tag{23}$$

Then, the point-to-plane distance is defined as

$$\mathbf{e}_i^p = \left(\mathbf{T}_{t-1}^t \circ \mu_i^t - \mu_{i12}^{t-1}\right) \cdot \left(\mathbf{T}_t^{t-1} \cdot \Sigma_i^t \cdot \left(\mathbf{T}_t^{t-1}\right)^T + \Sigma_{12}\right)^{-1} \cdot \left(\mathbf{T}_{t-1}^t \circ \mu_i^t - \mu_{i12}^{t-1}\right)^T \tag{24}$$

where Σ_{12} is the covariance matrix of $U(\mathbf{p}_{i1}^{t-1}) + U(\mathbf{p}_{i2}^{t-1})$ and μ_{i12}^{t-1} is the mean of μ_{i1}^{t-1} and μ_{i2}^{t-1}.

With all the valid data associations, the relative pose is optimized by minimizing the objective function

$$\arg\min_{\mathbf{T}_{t-1}^t} \sum_{i=1}^{|\mathbf{L}_t|} h\left(\left\|\mathbf{e}_i^l\right\|_2^2\right) + \sum_{i=1}^{|\mathbf{P}_t|} h\left(\left\|\mathbf{e}_i^p\right\|_2^2\right) \tag{25}$$

where $h(\cdot)$ is the Huber loss function [21] and $|\cdot|$ returns the cardinality. Then, we use the Gaussian–Newton iteration method to optimize the objective function. The solution is then used to update the next initial pose guess according to the uniform motion model.

GICP，即Generalized-ICP，是一种点云配准算法，它通过引入概率信息来统一和拓展传统的ICP（Iterative Closest Point）算法。

C. Distribution-Based Keyframe-to-Keyframe Matching

To reduce the drift errors introduced by the feature-based odometry, we perform distribution-based keyframe-to-keyframe matching using the GICP framework. 生成式？

1) Keyframe Selection Strategy: In our method, given the scan-to-scan odometry, we construct a keyframe sequence as follows:

$$\left\{\text{KF}_t, \mathbf{T}_t^w, \ t = 1, 2, \ldots, K\right\} \tag{26}$$

where KF_t is the voxelized keyframe and $\mathbf{T}_t^w$ represents the pose transformation in the global world coordinates derived from the odometry. We set the first frame as the initial keyframe. Then, one LiDAR frame is designated as a new keyframe during the running process if its distance to the last keyframe is beyond a threshold.

2) Distribution-Based Keyframe-to-Keyframe Pose Optimization: In a general GICP framework [12], the objective function is revisited as

$$F = \sum_i \mathbf{d}_i^T \left(\mathbf{C}_i^{t-1} + \mathbf{T}\hat{\mathbf{C}}_i^t \mathbf{T}^T\right)^{-1} \mathbf{d}_i = \sum_i \mathbf{d}_i^T \mathbf{M}_i \mathbf{d}_i. \quad (27)$$

In this article, we use the Gaussian–Newton method as the framework. The optimization process is to iteratively update the pose to decrease the objective function value. In the nonlinear optimization framework, we use the representation as twist coordinate $\boldsymbol{\xi}$ given by the Lie algebra $\mathfrak{se}(3)$ associated with the group SE(3), which is widely applied in the SLAM community [22]. $\boldsymbol{\xi} = [\boldsymbol{\omega}; \boldsymbol{\upsilon}]$ is a six-vector, where $\boldsymbol{\omega}$ is called the angular part and $\boldsymbol{\upsilon}$ is called the linear part. The transformation matrix $\mathbf{T}$ can be calculated from the increments $\boldsymbol{\xi}$ using the matrix exponential

$$\mathbf{T} = \exp(\boldsymbol{\xi}^\wedge) \quad (28)$$

where

$$\boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\omega}^\wedge & \boldsymbol{\upsilon} \\ \mathbf{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4} \quad (29)$$

and $\boldsymbol{\omega}^\wedge$ denotes the skew-symmetric matrix derived from $\boldsymbol{\omega}$.

According to the Gaussian–Newton optimization method [22], the pose estimation is updated according to

$$\sum_i \mathbf{J}_i^T \mathbf{M}_i \mathbf{J}_i \cdot \delta\boldsymbol{\xi} = \sum_i \mathbf{J}_i^T \mathbf{M}_i \mathbf{d}_i \quad (30)$$

where $\mathbf{J}_i$ is the Jacobian matrix and $\delta\boldsymbol{\xi}$ is the increment vector. $\mathbf{J}_i$ is computed as

$$\mathbf{J}_i = \frac{\partial \mathbf{d}_i}{\partial \delta\boldsymbol{\xi}} = \begin{bmatrix} -\mathbf{I} & (\mathbf{R}\mathbf{p}_i + \mathbf{t})^\wedge \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (31)$$

启发式优化 Then, in a general optimization framework, $\delta\boldsymbol{\xi}$ is computed as a linear equation solution. In this article, we provide a different viewpoint of (27). Specifically, for each $\mathbf{p}_i^t \in \mathrm{KF}_t$, it has a correspondence $\mathbf{p}_i^{t-1} \in \mathrm{KF}_{t-1}$, and the projection error is denoted as $\mathbf{d}_i$. Then, a function can be constructed as

$$D = f(\mathrm{KF}_t, \boldsymbol{\xi}), \quad D = \{||\mathbf{d}_i||_2, i = 1, \ldots, N\} \quad (32)$$

where $\boldsymbol{\xi}$ is the Lie algebra representation of pose transformation $\mathbf{T}$. Then, these correspondences can be classified into the following two classes.

1) *True Positive Correspondences:* If $\mathbf{p}_i^{t-1} \in \mathrm{KF}_{t-1}$ is the correct correspondence of $\mathbf{p}_i^t \in \mathrm{KF}_t$, then $\{\mathbf{p}_i^{t-1}, \mathbf{p}_i^t\}$ is defined as the true positive correspondence and should make a positive contribution to the objective function. However, it is impractical to directly obtain true positive correspondences due to the lack of ground truth during the SLAM process. Thus, in this article, we propose the following definition.
Definition 1: $\{\mathbf{p}_i^{t-1}, \mathbf{p}_i^t\}$ is defined as true positive if

$$f\left(\mathbf{p}_i^t, \boldsymbol{\xi} \boxplus \delta\boldsymbol{\xi}\right) < f\left(\mathbf{p}_i^t, \boldsymbol{\xi}\right) \quad (33)$$

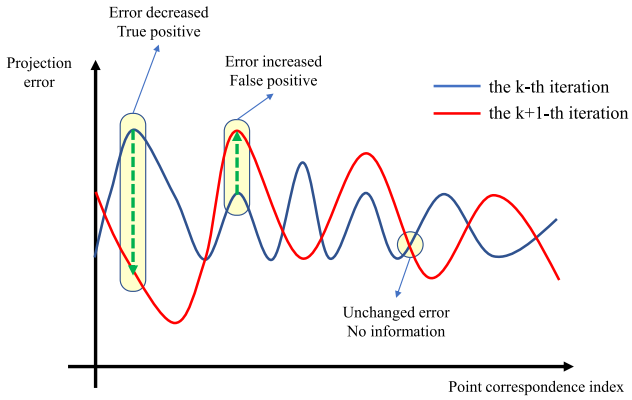


Fig. 7. Illustration of the point correspondence matching distance function. Notice that the projection errors of some correspondences demonstrate less changes during the iteration optimization process, which means that these correspondences provide less information gain for the pose optimization.

where $\boldsymbol{\xi} \boxplus \delta\boldsymbol{\xi}$ is defined as the Lie-algebra representation of the updated pose transformation matrix denoted as

$$\mathbf{T} = \exp(\delta\boldsymbol{\xi}^\wedge) \cdot \exp(\boldsymbol{\xi}^\wedge). \quad (34)$$

Equation (33) means that the projection error is decreased when the optimization iteration proceeds, which is consistent with the global objective function's change, since the optimization step aims to decrease the objective function.

2) *False Positive Correspondences:* If $\mathbf{p}_i^{t-1} \in \mathrm{KF}_{t-1}$ is not the correct correspondence of $\mathbf{p}_i^t \in \mathrm{KF}_t$, then $\{\mathbf{p}_i^{t-1}, \mathbf{p}_i^t\}$ is defined as the false positive correspondence and should make less contribution to the objective function. Then, we have the following definition.
Definition 2: $\{\mathbf{p}_i^{t-1}, \mathbf{p}_i^t\}$ is defined as false positive if

$$f\left(\mathbf{p}_i^t, \boldsymbol{\xi} \boxplus \delta\boldsymbol{\xi}\right) > f\left(\mathbf{p}_i^t, \boldsymbol{\xi}\right). \quad (35)$$

It means that the projection error is increased during the optimization process, which indicates that the constraints provided by the correspondence are contrary to the global objective function.

An illustration of matching distance function iteration for distinguishing true positive and false positive correspondences is shown in Fig. 7.

Thus, based on the proposed two definitions, we modify (30) as

$$\sum_i (W(\triangle d_i) \cdot \mathbf{J}_i)^T \mathbf{M}_i (W(\triangle d_i) \cdot \mathbf{J}_i) \cdot \delta\boldsymbol{\xi} = \sum_i (W(\triangle d_i) \cdot \mathbf{J}_i)^T \mathbf{M}_i \mathbf{d}_i \quad (36)$$

where $\triangle d_i$ is defined as

$$\triangle d_i = f\left(\mathbf{p}_i^t, \boldsymbol{\xi}\right) - f\left(\mathbf{p}_i^t, \boldsymbol{\xi} \boxplus \delta\boldsymbol{\xi}\right). \quad (37)$$

Also, $W(\triangle d_i)$ is defined as

$$W(\triangle d_i) = \frac{1}{1 + \exp(-\gamma \cdot \triangle d_i)} \quad (38)$$

which is a sigmoid function and $\gamma > 0$ is used for tunning the weights. If $\triangle d_i > 0$, then $W(\triangle d_i) > 0.5$, and vice versa.

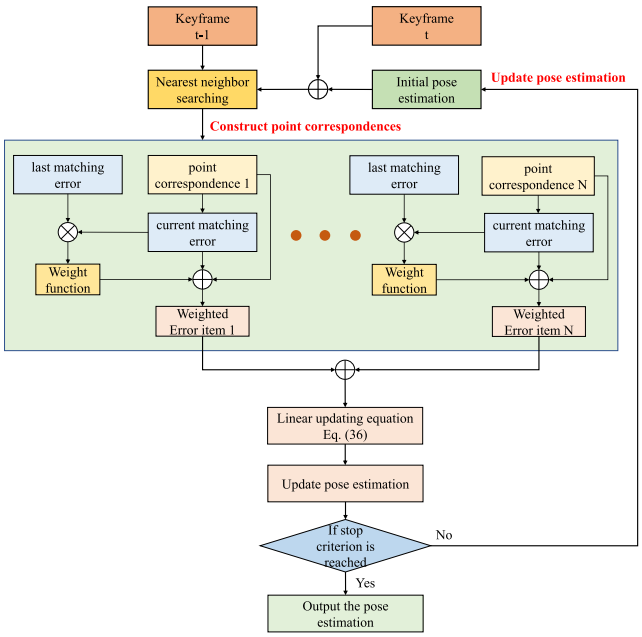


Fig. 8. Flowchart of the proposed nonlinear optimization method.

Such a function can ensure that we put more weights on the defined true positive correspondences and less weights on the defined false positive correspondences. In this article, the proposed heuristic GICP is termed HGICP.

To summarize, the flowchart is presented in Fig. 8. The proposed modified nonlinear optimization process is shown in Algorithm 1. When the maximum value of the updated vector $\delta\boldsymbol{\xi}$ is smaller than a threshold ϵ, the iteration process is stopped.

Algorithm 1 Proposed Nonlinear Iterative Optimization Method

Input: KF_t, KF_{t-1}, and the initial pose guess $\boldsymbol{\xi}_0$
Output: the final pose estimation $\boldsymbol{\xi}$

Given $\boldsymbol{\xi}_0$, construct the objective function, Jacobian matrices, residual errors, and the iterative pose update equation as Eq. (30).
Compute the first update pose $\boldsymbol{\xi}_1 = \boldsymbol{\xi}_0 \boxplus \delta\boldsymbol{\xi}_0$.
for $k = 1 : maxiterationstep$ **do**
  Compute $\triangle d_i = f(\mathbf{p}_i^t, \boldsymbol{\xi}_{k-1}) - f(\mathbf{p}_i^t, \boldsymbol{\xi}_k)$ of each point correspondence.
  Compute $W(\triangle d_i)$ for each point correspondence.
  Construct iterative update equation as Eq. (36).
  Solve Eq. (36) and obtain $\boldsymbol{\xi}_{k+1}$.
  if $\max(\boldsymbol{\xi}_{k+1}) < \epsilon$ is achieved **then**
    stop optimization and output $\boldsymbol{\xi}_{k+1}$.
  else
    continue.
return $\boldsymbol{\xi}_{k+1}$

3) Stability Analysis and Proof: The stability of the proposed optimization method is illustrated and proved as follows. Let $\mathbf{T}^g$ denote the ground-truth pose transformation between KF_t and KF_{t-1}. Assuming that we have $\mathbf{T}^g$, then for each point in KF_t, we can obtain its optimal correspondence in KF_{t-1}. Then, at the kth optimization step, using the current pose estimation $\mathbf{T}$ and the optimal point correspondences, an ideal pose updating formulation can be derived as

$$\sum_i \mathbf{J}_i^{gT}\mathbf{M}_i\mathbf{J}_i^g \cdot \delta\boldsymbol{\xi} = \sum_i \mathbf{J}_i^{gT}\mathbf{M}_i\mathbf{d}_i^g \tag{39}$$

where $\mathbf{d}_i^g$ is computed as

$$\mathbf{d}_i^g = {}^o\mathbf{p}_i^{t-1} - \mathbf{T} \circ \mathbf{p}_i^t \tag{40}$$

${}^o\mathbf{p}_i^{t-1} \in KF_{t-1}$ is the optimal correspondence of $\mathbf{p}_i^t \in KF_t$. The difference between (30) and (39) is that the point correspondences in (30) are constructed according to the last pose estimation $\mathbf{T}$ instead of $\mathbf{T}^g$, and the point correspondence set in (30) exhibits noisy and wrong correspondences. Actually, it is impractical to obtain $\mathbf{T}^g$ during the optimization process and the assumption of the availability of $\mathbf{T}^g$ is only for the analytical proof. Given the ideal pose updating formulation, we can find the optimal pose estimation after several iteration steps. However, it is impractical to obtain (39) and we can only obtain its approximation formulation, which is (30). For the sake of brevity, we use the following representation:

$$\begin{aligned} \mathbf{H}_i &= \mathbf{J}_i^T\mathbf{M}_i\mathbf{J}_i, \quad \mathbf{b}_i = \mathbf{J}_i^T\mathbf{M}_i\mathbf{d}_i \\ \mathbf{H}_i^g &= \mathbf{J}_i^{gT}\mathbf{M}_i\mathbf{J}_i^g, \quad \mathbf{b}_i^g = \mathbf{J}_i^{gT}\mathbf{M}_i\mathbf{d}_i^g. \end{aligned} \tag{41}$$

$\mathbf{H}$ is called the Hessian matrix in the SLAM community. Then, (30) is reformulated as

$$\sum_i \left(\mathbf{H}_i^g + \triangle\mathbf{H}_i\right) \cdot \delta\boldsymbol{\xi} = \sum_i \left(\mathbf{b}_i^g + \triangle\mathbf{b}_i\right). \tag{42}$$

For each error term, if ${}^o\mathbf{p}_i^{t-1} \in KF_{t-1}$ is consistent with the searched correspondence according to $\mathbf{T}$, then we have

$$\triangle\mathbf{H}_i = \mathbf{0}^{6\times6}, \quad \triangle\mathbf{b}_i = \mathbf{0}^{6\times1}. \tag{43}$$

However, if the constructed point correspondence is low-quality, we have

$$\triangle\mathbf{H}_i \neq \mathbf{0}^{6\times6}, \quad \triangle\mathbf{b}_i \neq \mathbf{0}^{6\times1}. \tag{44}$$

The existences of $\triangle\mathbf{H}_i$ and $\triangle\mathbf{b}_i$ make the optimial $\delta\boldsymbol{\xi}$ biased and unstable.

In our method, we distinguish the quality of the point correspondence according to the proposed prior knowledge. Then, we put different weights on the error terms according to the estimated point correspondence quality. For the high-quality correspondence, it is not required to consider $\triangle\mathbf{H}_i$ and $\triangle\mathbf{b}_i$. For the low-quality correspondence, since the given weight is a small value, we have

$$W^2(\triangle d_i)\triangle\mathbf{H}_i \to \mathbf{0}^{6\times6}, \quad W(\triangle d_i)\triangle\mathbf{b}_i \to \mathbf{0}^{6\times1} \tag{45}$$

where $\to$ means being close to. Thus, the application of the matching error term weights can make the linear equation less distorted. In other words, it can decrease the difference between the constructed linear updating equation (30) and the ideal linear updating equation (39). Notice that the proposed method and the proof are based on the assumption that a good pose estimation initialization is performed. In our method,

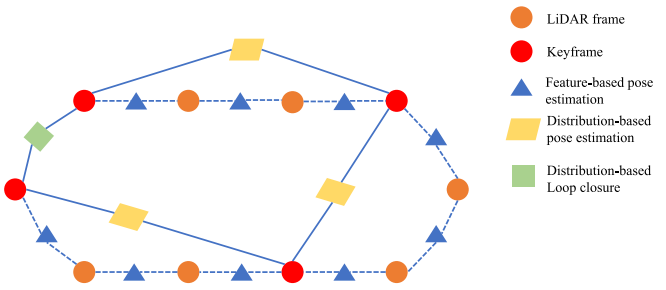


Fig. 9. Illustration of the constructed pose graph in our system. G2O is used to perform global pose graph optimization.

the keyframe pose transformation estimation initialization is performed according to the fast feature-based scan-matching results.

Since our method is based on the Gaussian–Newton method, another issue related to the stability is the singularity of the Hessian matrix. In (30), if

$$\left|\sum_i \mathbf{H}_i\right| = 0 \text{ or } \left|\sum_i \mathbf{H}_i\right| \to 0 \tag{46}$$

where $|\cdot|$ returns the determinant of matrix, and then, the computed solution tends to become highly unstable. This will happen when there are less informative structures, and the LiDAR point correspondences cannot provide enough constraints for the six-freedom pose estimation, which is called degenerate problem. However, since the 3-D sensor is omnidirectional and the perception range is large, such a problem hardly happens in a man-made environment. Our method can also be directly adapted to the Levenberg–Marquardt optimization method.

D. Loop Detection and Backend Optimization

Each new keyframe searches for loop candidates in the historical keyframe sequence within a certain distance during the localization and mapping process. The proposed HGICP method estimates the pose transformation between the current keyframe and the candidates. The fitness score in HGICP is used for loop closure decision. Then, the pose graph is updated, in which the keyframe poses are set as vertexes and edges represent the relative transformation between keyframes. An illustration is presented in Fig. 9. In our system, we apply the G2O library[1] to optimize the global pose graph.

V. Experiments

We evaluate the proposed LiDAR SLAM system on public datasets and in our campus scenarios. The computing hardware arrangement is a laptop with an Intel i7 CPU at 3.66 GHz and 24 GB of memory. The methods are implemented in C++ and ROS. To demonstrate our method's performance, we qualitatively and quantitatively analyze our method in terms of accuracy and efficiency. The absolute trajectory error (RMSE ATE) and the ATE averaged on 100 m are used for localization accuracy evaluation [23]. We first perform the ablation study of the proposed feature-based pose tracking, distribution-based keyframe-matching modules, and the weight function in the pose optimization module. Then, we compare our SLAM system with the baselines on the public and custom datasets. Furthermore, to verify the robustness to viewpoint change, we implement verification experiments about the keyframe distance.

A. Baselines

Among the state-of-the-art LiDAR SLAM systems, we compare our method with hdl-graph-SLAM, LeGO-LOAM, and MULLS. Hdl-graph-SLAM[2] [24] is a closed pure distribution-based LiDAR SLAM system and its core scan-matching method is GICP. LeGO-LOAM is the most typical geometry feature-based LiDAR SLAM system. In this article, LeGO-LOAM with loop closure based on scan context [25] is used. MULLS achieves the best performance among the existing methods through sophisticated feature extraction and matching mechanisms. Since the MULLS system depends on numerous hyperparameters and the opened source code only provides configured files for the KITTI dataset, we only perform MULLS on the KITTI and USTC-VLP16 datasets. The provided configured files have been experimentally proved that they are unsuitable for Mulran and NCLT datasets.

B. Datasets

1) KITTI-Dataset [26]: The KITTI odometry dataset provides 11 sequences (00–10) with ground-truth trajectories. A Velodyne HDL-64E 3-D laser scanner (10 Hz, 64 laser beams, 360° FOV, range: 100 m) is used to collect data in multiple environments, including the urban (Seq. 00, 06, 07, and 08), country (Seq. 02, 03, 04, 05, 09, and 10), and highway scene (Seq. 01). This dataset is widely used for SLAM system evaluation.

2) USTC-VLP16-Dataset: We collect the dataset in the University of Science and Technology of China (USTC). Some scenes are shown in Fig. 10. A Velodyne VLP-16 3-D laser scanner is mounted on our mobile robot. We use the robot to collect LiDAR data in campus scenarios, which contain structural and unstructured scenes. An RTK is mounted on the platform shown in Fig. 11. We have made this dataset public.[3] Three LiDAR sequences and the corresponding ground-truth trajectories are contained in the dataset. The LiDAR scans are recorded by the rosbag[4] tool. The ground-truth poses are recorded following the same format as the TUM dataset.[5] The trajectories of the three sequences and the constructed map according to the ground-truth poses are shown in Fig. 12.

3) Mulran Dataset [27]: This dataset provides a multienvironment, multisession, and multimodal range (i.e., radar-LiDAR) dataset. It has both radar and 3-D LiDAR (Ouster OS1-64) and includes multisession sequences along a repeated trajectory within a changing city with month-level

[1]https://github.com/RainerKuemmerle/g2o

[2]https://github.com/koide3/hdl_graph_slam

[3]https://rec.ustc.edu.cn/share/d4624e00-bad1-11ec-9e41-bdb9001e20d4 password: USTC

[4]http://wiki.ros.org/rosbag

[5]https://vision.in.tum.de/data/datasets/rgbd-dataset/download

Fig. 10. Some representative scenes in the USTC dataset. Dynamic vehicles, pedestrians, unstructured and structured scenes, and repetitive scenes are exhibited in this dataset.

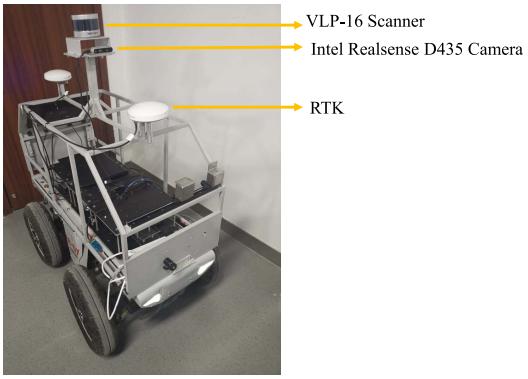


Fig. 11. Our mobile platform.

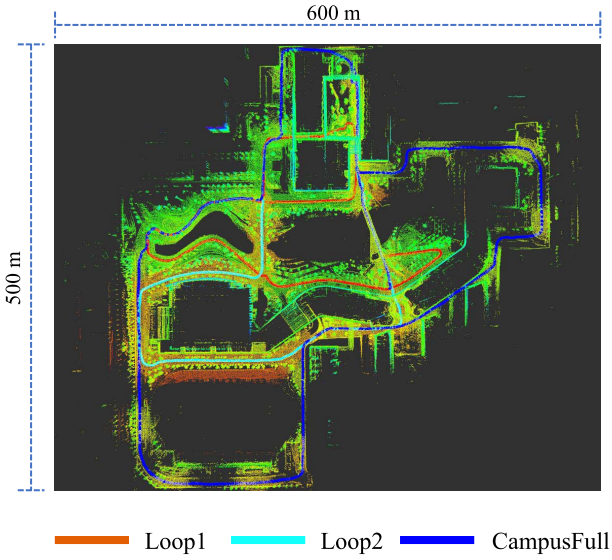


Fig. 12. Trajectories of the mobile robot and the constructed LiDAR map according to the RTK results.

temporal gaps. Furthermore, this dataset has various types of revisit events. Three representative sequences are used for evaluation in this article.

TABLE I

ATE [%] (THE SMALLER, THE BETTER) EVALUATION AND COMPARISON BETWEEN OUR ODOMETRY AND LEGO-LOAM ODOMETRY ON THE KITTI DATASET

	Our Odometry	LeGO-LOAM Odometry	MULLS Odometry	Length [m]
00	**0.31**	0.97	0.51	3742
01	1.63	10.47	**0.62**	2406
02	1.18	1.67	**0.55**	5071
03	**0.45**	1.04	0.61	566
04	**0.30**	0.52	0.35	392
05	**0.26**	0.67	0.28	2216
06	0.44	1.15	**0.24**	1234
07	**0.21**	0.58	0.29	697
08	**0.34**	0.96	0.80	3239
09	0.81	0.94	**0.49**	1709
10	**0.43**	0.71	0.61	924

4) NCLT Dataset[6]*:* This is a large-scale, long-term autonomy dataset for robotics research collected on the University of Michigan's North Campus. The dataset consists of omnidirectional imagery, 3-D LiDAR (Velodyne HDL-32E), planar LiDAR, GPS, and proprioceptive sensors for odometry collected using a Segway robot. Four representative sequences are used for evaluation.

5) Stevens-VLP16-Dataset[7]*:* This dataset is captured using Velodyne VLP-16, which is mounted on a UGV in the Stevens Institute of Technology campus.

Except for the Stevens-VLP16-Dataset, the other datasets provide pose ground truth.

C. Experimental Settings

The hyperparameters used in our system are presented as follows. p_{th} and e_{th} in (16) and (17) are 0.3 and 1, respectively. The distance threshold for setting a new keyframe is 5 m. The voxelization resolution used in the KITTI, Mulran, and NCLT datasets is set to 0.5 m. In the USTC-VLP16 dataset, the voxelization resolution is set to 1 m. The number of searched neighbor points in GICP is 20. γ in (38) is set to 1 as default.

D. Ablation Study

1) Feature-Based LiDAR Odometry Evaluation: We first validate our odometry and compare it against LeGO-LOAM since our odometry can be regarded as a modified version of LeGO-LOAM odometry. Specifically, smoothness values are used for extracting edge and planar feature points in both methods. However, we use Gaussian models to depict these features and more structural information can be encoded into the models. In LeGO-LOAM odometry, the edge feature or plane feature is constructed by searching for neighbor points online during the point-to-feature distance computing process, in which wrong searched neighbors tend to cause feature representation and association errors.

The experimental results on the KITTI dataset are presented in Table I. According to the results presented in this table, our odometry is superior to LeGO-LOAM odometry, especially on Seq01. As is concluded in MULLS, linear points

[6]http://robots.engin.umich.edu/nclt/

[7]https://github.com/TixiaoShan/Stevens-VLP16-Dataset

TABLE II

ATE [%] (THE SMALLER, THE BETTER) EVALUATION AND COMPARISON BETWEEN OUR ODOMETRY AND LEGO-LOAM ODOMETRY ON THE USTC-VLP16 DATASET

	Our Odometry	LeGO-LOAM Odometry	MULLS Odometry	Length [m]
Loop1	5.66	5.96	**0.87**	1012
Loop2	3.01	5.49	**2.87**	942
CampusFull	**0.68**	1.73	1.27	2177

TABLE III

ATE [%] (THE SMALLER, THE BETTER) EVALUATION AND COMPARISON BETWEEN OUR ODOMETRY AND LEGO-LOAM ODOMETRY ON THE MULRAN DATASET

	Our Odometry	LeGO-LOAM Odometry	Length [m]
DCC01	1.73	**1.50**	5036
KAIST01	**1.80**	2.70	6165
RIVERSIDE01	**1.79**	3.08	6455

TABLE IV

ATE [%] (THE SMALLER, THE BETTER) EVALUATION AND COMPARISON BETWEEN OUR ODOMETRY AND LEGO-LOAM ODOMETRY ON THE NCLT DATASET

	Our Odometry	LeGO-LOAM Odometry	Length[m]
2013-04-05	**4.55**	8.57	2416
2012-06-15	**1.81**	4.14	1930
2012-11-16	**1.89**	4.52	1620
2012-05-26	**1.96**	4.70	1946

(pillar and beam) are necessary for supreme performance in such highway scene. Instead of extracting the linear points explicitly, linear representation is encoded in all the extracted feature points in our odometry. In the point cloud feature extracting paradigm, raw points are regarded as the basic components for multiple features. In our method, each point is assigned with a Gaussian model according to its located surface trend along the z-axis. Thus, our features can provide more geometry restrictions. The experimental results on the USTC-VLP16, Mulran, and NCLT datasets are presented in Tables II–IV, respectively. These datasets exhibit structured scenes, less-structured scenes, vehicles turning at corners, variant scales, and moving objects. We can tell that our odometry is competitive compared with LeGO-LOAM and MULSS on these datasets. Though MULLS demonstrates much better performance on some sequences, complex hyperparameters tuning is required, while our feature-based odometry relies on less hyperparameters and can be generalized to various environments.

2) Ablation Study of the Proposed Mechanisms for Distribution-Based Scan Matching: In this section, the proposed heuristic optimization mechanism is studied and verified. Since it is based on the GICP framework, we use the FastGICP [12] as the baseline. Two groups of experiments are set: 1) FastGICP: the raw version of FastGICP and 2) Heuristic-FastGICP: using the proposed heuristic optimization mechanism in FastGICP. The trajectory estimation precision evaluation results on the datasets are presented in Tables V–VIII. Notice that we directly use the ATE for evaluation for the sake of clear comparison and improvement

TABLE V

ATE [m] (THE SMALLER, THE BETTER) EVALUATION AND COMPARISON BETWEEN FASTGICP AND FASTGICP-HEURISTIC ON THE KITTI DATASET

	FastGICP	FastGICP_Heuristic	Improvements
00	8.23	**6.23**	+24%
01	**25.22**	25.87	-2%
02	17.93	**13.72**	+23%
03	2.10	2.10	+0%
04	0.76	**0.75**	+1%
05	3.78	**3.25**	+14%
06	1.38	**1.37**	+0%
07	0.69	**0.64**	+7%
08	**5.51**	5.62	-2%
09	2.55	**2.48**	+3%
10	1.79	**1.70**	+5%

TABLE VI

ATE [m] (THE SMALLER, THE BETTER) EVALUATION AND COMPARISON BETWEEN FASTGICP AND FASTGICP-HEURISTIC ON THE USTC-VLP16 DATASET

	FastGICP	FastGICP_Heuristic	Improvements
Loop1	2.95	**2.43**	+17%
Loop2	3.24	**2.16**	+33%
CampusFull	16.74	**15.18**	+9%

TABLE VII

ATE [m] (THE SMALLER, THE BETTER) EVALUATION AND COMPARISON BETWEEN FASTGICP AND FASTGICP-HEURISTIC ON THE MULRAN DATASET

	FastGICP	FastGICP_Heuristic	Improvements
DCC	30.50	**23.97**	+21%
KAIST	39.60	**28.58**	+28%
RIVERSIDE	**29.90**	29.97	-0%

TABLE VIII

ATE [m] (THE SMALLER, THE BETTER) EVALUATION AND COMPARISON BETWEEN FASTGICP AND FASTGICP-HEURISTIC ON THE NCLT DATASET

	FastGICP	FastGICP_Heuristic	Improvements
2013-04-05	12.55	**5.20**	+59%
2012-06-15	**4.67**	6.27	-34%
2012-11-16	9.46	**5.61**	+41%
2012-05-26	**3.93**	5.47	-39%

demonstration. From these tables, we can tell that the proposed heuristic optimization mechanism can achieve significant improvements on most sequences. In the proposed mechanism, the definitions of true positive and false positive correspondences assume that pose transformation estimation initialization is good. Such assumption can be generally satisfied using the uniform motion model under most localization tasks. However, if the uniform motion model is not consistent with the mobile platform movements, especially at the crossroads, then the two kinds of correspondence definitions tend to introduce errors in the weighted iterative equation. Thus, on some sequences, such as NCLT2012-06-15 and NCLT2012-05-26, less performance improvements are achieved even though our mechanism is still effective. In the future, IMU observations

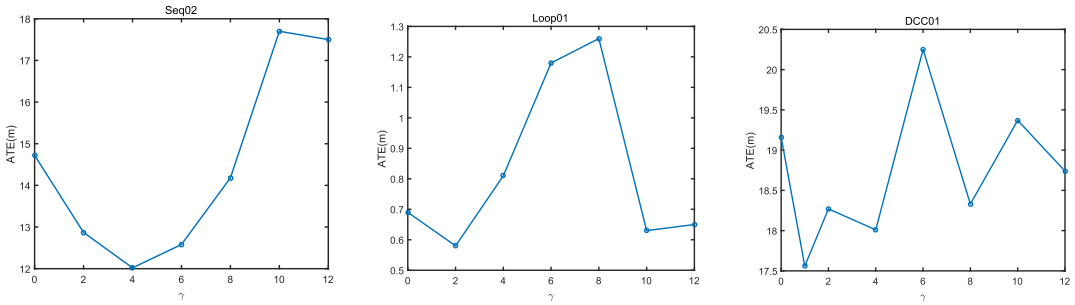


Fig. 13. Trajectory estimation results on the representative sequences when γ takes different values. (a) Results on Seq02. (b) Results on Loop01. (c) Results on DCC01.

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

Seq	*00*	*01*	*02*	*03*	*04*	*05*	*06*	*07*	*08*	*09*	*10*
LOAM [9]	0.78	1.43	0.92	0.86	0.71	0.57	0.65	0.63	1.12	0.77	0.80
A-LOAM[8]	0.79	1.96	4.57	0.95	0.77	0.50	0.62	0.45	1.11	0.74	1.01
LeGO-LOAM [2]	0.17	7.56	0.51	0.17	0.16	0.13	0.07	0.12	0.11	0.13	0.22
SUMA++ [5]	0.64	1.60	1.00	0.67	0.37	0.40	0.46	0.34	1.10	0.47	0.66
LiTAMIN2 [11]	0.70	2.10	0.98	0.96	1.05	0.45	0.59	0.44	0.95	0.69	0.80
MULLS-LO [3]	0.51	0.62	0.55	0.61	0.35	0.28	0.24	0.29	0.80	0.49	0.61
MULLS-SLAM [3]	0.54	0.62	0.69	0.61	0.35	0.29	0.29	0.27	0.83	0.51	0.61
DGP-SLAM [4]	0.85	1.94	0.96	0.79	0.71	0.42	0.44	0.44	1.00	0.73	1.09
ROI-SLAM [28]	0.03	0.59	**0.13**	0.20	0.09	0.03	0.07	0.15	0.11	**0.07**	0.15
POU-SLAM [29]	0.64	0.90	0.74	0.59	0.49	0.43	0.36	0.35	0.84	0.53	0.83
Hdl-Graph-SLAM	0.02	1.14	0.22	**0.12**	0.18	0.04	0.03	0.05	0.05	0.29	0.23
PCA-SLAM[30]	0.07	0.42	0.18	0.14	0.10	0.08	0.07	**0.01**	0.13	0.11	0.16
FD-SLAM-GICP	0.04	0.23	0.29	0.18	0.06	0.04	0.04	0.05	0.05	0.27	0.21
FD-SLAM-HGICP	**0.02**	**0.20**	0.20	0.18	**0.05**	**0.03**	**0.04**	0.04	**0.05**	0.08	**0.15**

can be introduced and fused with our mechanism to alleviate the problem in the odometry framework.

3) Ablation Study of the Weight Function: We use a sigmoid function as the weight function and γ is one hyperparameter. Then, we perform experiments to validate the influence of γ on the distribution-based scan-matching performance. In the experiments, γ takes value from $\{0, 2, 4, 6, 8, 10, 12\}$. We can tell that a larger value of γ leads to higher weights. Then, the ATE evaluations on some representative sequences under different γ's are shown in Fig. 13. Notice that $\gamma = 0$ means that all the weights are the same. From these figures, we can tell that the performance achieves the best when γ is around 2. When γ increases and more weights are put on the defined true positive correspondences, large estimation errors are obtained. It means that there exist point correspondences misclassifications. Thus, our mechanism is heuristic instead of a certain mechanism.

E. SLAM System Evaluation

We perform an SLAM system comparison on the datasets, and the results are shown in Table IX–XII. We compare against HDL-graph-SLAM, LeGO-LOAM, MULLS SLAM, and other state-of-the-art SLAM systems. Since the KITTI dataset is widely used for LiDAR SLAM system evaluation, the trajectories evaluation results of the state-of-the-art SLAM systems on KITTI datasets are directly reported in Table IX, which are referred from their published work. Our method is termed FD-SLAM. In the experiments, we provide two versions of FD-SLAM, including FD-SLAM-GICP and

TABLE X
ATE [%] (THE SMALLER, THE BETTER) EVALUATION AND COMPARISON OF THE SLAM SYSTEMS ON THE USTC-VLP16 DATASET

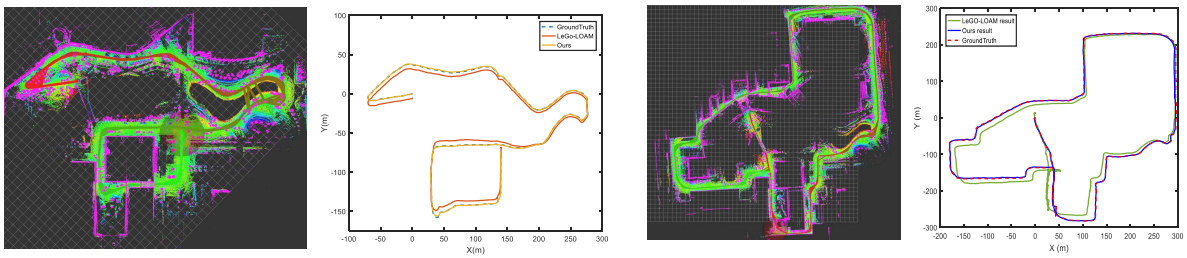
	MULLS SLAM	LeGO-LOAM	Hdl-Graph -SLAM	FD-SLAM -GICP	FD-SLAM -HGICP	Length [m]
Loop1	0.87	0.59	**0.08**	0.24	0.20	1012
Loop2	2.87	1.06	**0.16**	0.29	0.28	942
CampusFull	0.23	0.25	0.18	0.18	**0.06**	2177

TABLE XI
ATE [%] (THE SMALLER, THE BETTER) EVALUATION AND COMPARISON OF THE SLAM SYSTEMS ON THE MULRAN DATASET

	LeGO-LOAM	Hdl-Graph -SLAM	FD-SLAM -GICP	FD-SLAM -HGICP	Length [m]
DCC01	0.28	0.33	0.13	**0.12**	5036
KAIST01	0.35	0.15	0.07	**0.04**	6165
Riverside01	0.87	0.14	0.21	**0.13**	6455

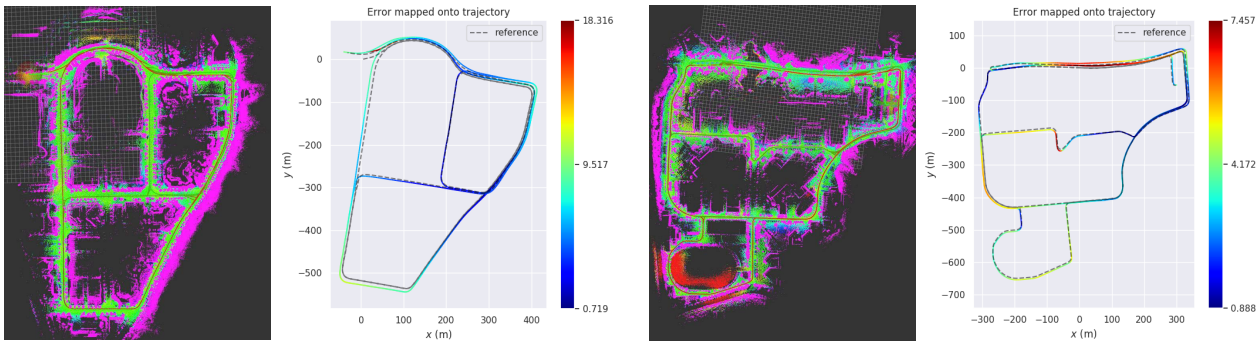
TABLE XII
ATE [%] (THE SMALLER, THE BETTER) EVALUATION AND COMPARISON OF THE SLAM SYSTEMS ON THE NCLT DATASET

	LeGO-LOAM	Hdl-Graph-SLAM	FDSLAM -GICP	FDSLAM -HGICP	Length [m]
2013-04-05	0.56	0.27	0.24	**0.24**	2416
2012-06-15	2.12	0.18	0.19	**0.15**	1930
2012-11-16	1.40	0.31	0.28	**0.24**	1620
2012-05-26	0.39	**0.14**	0.18	0.16	1946



(a) (b)

Fig. 14. Our SLAM system estimation results on sequences in the USTC dataset. (a) Results on Loop01. (b) Results on CampusFull.



(a) (b)

Fig. 15. Our SLAM system estimation results on sequences in the Mulran dataset. (a) Results on DCC01. (b) Results on KAIST01.

FD-SLAM-HGICP. The difference is whether the heuristic mechanism is applied. We can tell that our method is competitive with the state-of-the-art SLAM systems. Furthermore, three conclusions can be drawn from these results. First, HDL-graph-SLAM demonstrates stable and acceptable performance on these datasets. However, it also shows less adaptability on some sequences, such as Seq01 and DCC01. These sequences exhibit repetitive geometry distributions. Second, FD-SLAM-HGICP can achieve better performance than FD-SLAM-GICP on most sequences. Third, in LeGO-LOAM, the odometry drift errors are corrected by scan-to-model matching and optimization. Submaps are required to construct and maintain. However, in our method, the distribution-based keyframe matching is easy to implement and can provide better drift error reduction performance. Some mapping results of our system are presented in Figs. 14 and 15. The mapping result on the Stevens dataset is shown in Fig. 16. We can tell that our system can achieve global consistent mapping results.

We also perform the ablation study of the keyframe distance. On the public and custom datasets, we tune the keyframe distance and our method demonstrates various performances. Among the sequences, the representative changing patterns are shown in Fig. 17. On Seq00 in KITTI, the trajectory estimation evaluation error is large when the keyframe distance is small, which indicates that performing keyframe matching frequently does not benefit the drift error reduction. When the keyframe distance increases, the ATE decreases at first and increases gradually, which is due to that the keyframe matching error increases with the viewpoint distance. However, on Seq01 in KITTI, which is a representative structural repetitive

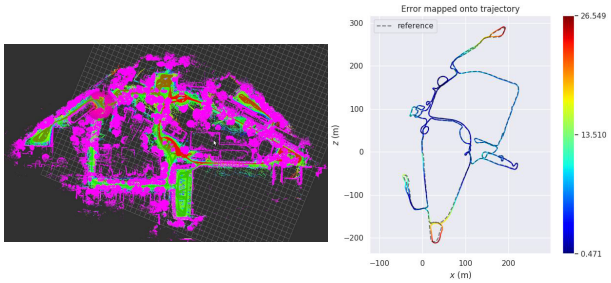


Fig. 16. Our mapping results on the Stevens-VLP16-Dataset. The trajectory estimated from SC-LeGO-LOAM is presented for comparison since this dataset does not provide ground truth.

environment, the keyframe matching can effectively reduce the drift errors and the performance becomes better when the keyframe distance increases, which is due to that distribution-based keyframe matching can perform better alignment when the viewpoints are significantly different, since more specific alignment information can be captured. Such a pattern only occurs when the environment is highly geometry repetitive. On the other sequences, the precision performance roughly becomes worse when the keyframe distance increases. Generally, the drift errors obtained by the feature-based pose tracking module tend to increase when the keyframe distance becomes large. The precision performance of the whole SLAM system depends on the keyframe matching performance.

The trajectory estimations of our method and the baselines are also presented in Fig. 18. From Fig. 18(a), we can tell that both LeGO-LOAM and HDL-graph-SLAM, which are the

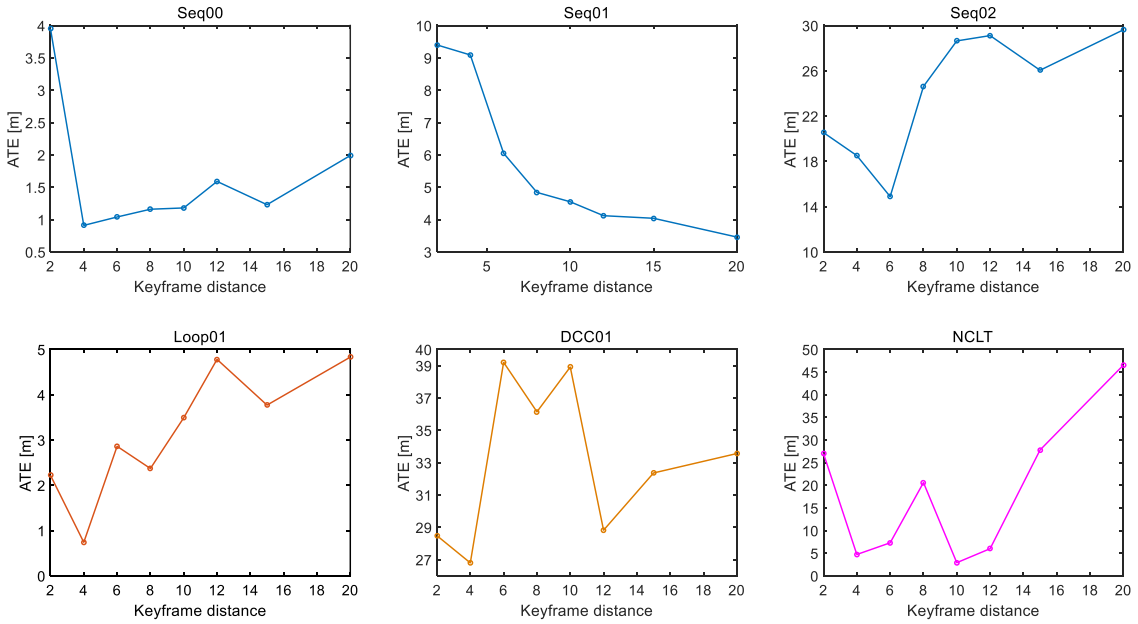


Fig. 17. Ablation study of the keyframe distance threshold.

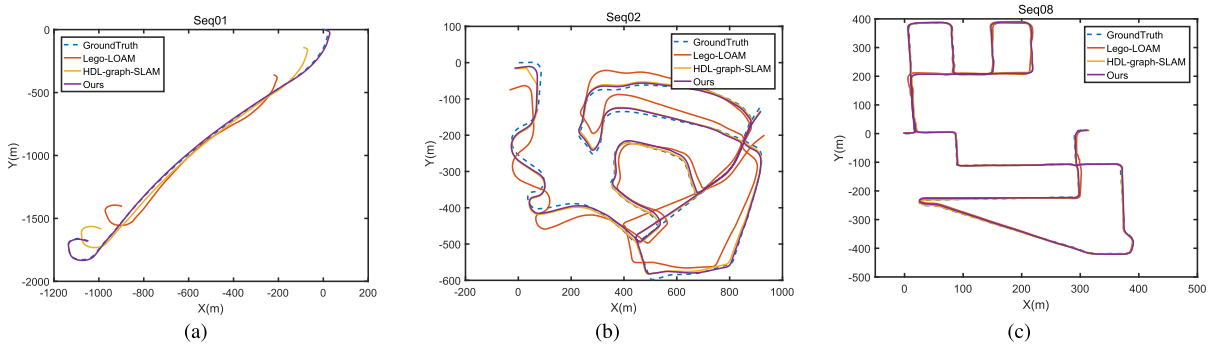


Fig. 18. Trajectory estimation results on the representative sequences in the KITTI dataset. (a) Results on Seq01. (a) Results on Seq02. (a) Results on Seq08.

most typical feature- and distribution-based SLAM systems, fail to perform accurate localization. Our method can provide accurate pose trajectory estimation. The feature-based scan matching can perform accurate pose estimation at a short distance while tending to generate large errors at a long distance. The distribution-based scan matching can provide effective pose estimation accuracy at a long distance, while it requires a good initial pose guess. Thus, the two kinds of modules are compatible with each other. On Seq02 in KITTI, rotational movement is frequent and the trajectory estimation results exhibit more localization challenges. We can tell that our system's performance is superior to the baselines.

F. Efficiency Analysis

The averaged computational costs of each module on the datasets are presented in Table XIII. The feature extraction process takes the most computational time in the odometry module. Compared with the scan-to-model matching module in LeGO-LOAM, GICP-based keyframe-matching is a time-consuming process, which is the motivation that we avoid performing scan-to-scan matching based on GICP. Compared with GICP, HGICP does not introduce too much efficiency loss. Specifically, considering that the loop closure and global pose optimization are running in parallel, for a LiDAR sequence with N frames, and the keyframes are set every k_0 frames around, the averaged SLAM running time per frame is computed as

TABLE XIII
COMPUTATIONAL COSTS OF EACH MODULE ON THE DATASETS

Dataset	Odometry			Drift error correction		
	FD-SLAM odometry	LeGO-LOAM odometry	MULLS odometry	FD-SLAM keyframe-matching w.HGICP	LeGO-LOAM scan-to-model matching	FD-SLAM keyframe-matching w.GICP
KITTI	28 ms	35 ms	33 ms	180 ms	150 ms	160 ms
Mulran	30 ms	26 ms	32 ms	190 ms	80 ms	170 ms
NCLT	15 ms	16 ms	–	185 ms	42 ms	170 ms
USTC-VLP16	6 ms	9 ms	–	65 ms	25 ms	80 ms

$$t_{\text{SLAM}} = \frac{N \cdot t_{sm} + \frac{N}{k_0} \cdot t_{km}}{N} \tag{47}$$

where t_{sm} is the computing time of feature-based LiDAR odometry per scan and t_{sm} is the computing time of distribution-based keyframe matching per keyframe. Generally, t_{km} is the five times of t_{sm}. Taking the KITTI dataset as an example, we perform one keyframe-matching every five

frames around. Thus, our system can achieve 20 Hz around. Our system can be further accelerated by making scan-to-scan matching and keyframe-matching work in parallel.

G. Discussion

In this article, we only perform heuristic optimization in the keyframe matching process. The reasons are given as follows. First, in the feature-based pose tracking process, most features can find correct correspondences since the relative movement is small and the uniform motion model can be well satisfied. Thus, it is not required to further filter the point correspondences. Second, in the keyframe matching process, viewpoint changing, occlusion, and dynamic objects, combined with pose initialization errors, lead to a large amount of nonrigid correspondences. The constructed objective function contains more false positive correspondences, which degrades the keyframe matching performance. Thus, the heuristic optimization is only used for large pose transformation optimization. The proposed mechanism can be well generalized to other state optimization tasks. More prior knowledge can be developed and fused to define true positive and false positive correspondences.

VI. Conclusion

This article proposes a loose coupling of feature- and distribution-based LiDAR scan-matching modules and formulates a full SLAM system. This system can take advantage of geometry feature- and distribution-based surface representations, and the experimental results demonstrate that the proposed method can achieve competitive performance with state-of-the-art methods. The proposed distribution-based geometry feature representation can be well generalized to different kinds of geometry features. These mechanisms can also be adapted to other LiDAR SLAM systems. In the future, we will further investigate more weight functions and heuristic mechanisms. We will also incorporate visual and IMU perception functions to make our SLAM system more adaptive to dynamic environments.

References

[1] X. Ji, L. Zuo, C. Zhang, and Y. Liu, "LLOAM: LiDAR odometry and mapping with loop-closure detection based correction," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2019, pp. 2475–2480.

[2] 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. (IROS)*, Oct. 2018, pp. 4758–4765.

[3] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "MULLS: Versatile LiDAR SLAM via multi-metric linear least square," 2021, *arXiv:2102.03771*.

[4] S. Liang, Z. Cao, C. Wang, and J. Yu, "A novel 3D LiDAR SLAM based on directed geometry point and sparse frame," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 374–381, Apr. 2021.

[5] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4530–4537.

[6] J. Wang, M. Xu, F. Foroughi, D. Dai, and Z. Chen, "FasterGICP: Acceptance-rejection sampling based 3D lidar odometry," *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 255–262, Jan. 2022.

[7] 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," *Auto. Robots*, vol. 45, no. 1, pp. 77–98, Jan. 2021.

[8] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Globally consistent 3D LiDAR mapping with GPU-accelerated GICP matching cost factors," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 8591–8598, Oct. 2021.

[9] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, USA: Univ. of California, Berkeley, Jul. 2014, pp. 1–9.

[10] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, Oct. 2007.

[11] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "LiTAMIN2: Ultra light LiDAR-based SLAM using geometric approximation applied with KL-divergence," 2021, *arXiv:2103.00784*.

[12] 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. Autom. (ICRA)*, May 2021, pp. 11054–11059.

[13] B. Li, Y. Wang, Y. Zhang, W. Zhao, J. Ruan, and P. Li, "GP-SLAM: Laser-based SLAM approach based on regionalized Gaussian process map reconstruction," *Auto. Robots*, vol. 44, no. 6, pp. 947–967, Jul. 2020.

[14] T. Schops, 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.

[15] C. Yuan, W. Xu, X. Liu, X. Hong, and F. Zhang, "Efficient and probabilistic adaptive voxel mapping for accurate online LiDAR odometry," 2021, *arXiv:2109.07082*.

[16] M. Barczyk and S. Bonnabel, "Towards realistic covariance estimation of ICP-based Kinect v1 scan matching: The 1D case," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 4833–4838.

[17] M. Velas, M. Spanel, and A. Herout, "Collar line segments for fast odometry estimation from velodyne point clouds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 4486–4495.

[18] J. Jiang, J. Wang, P. Wang, P. Bao, and Z. Chen, "LiPMatch: LiDAR point cloud plane based loop-closure," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6861–6868, Oct. 2020.

[19] M. Young, C. Pretty, J. McCulloch, and R. Green, "Sparse point cloud registration and aggregation with mesh-based generalized iterative closest point," *J. Field Robot.*, vol. 38, no. 8, pp. 1078–1091, 2021.

[20] M. Oelsch, M. Karimi, and E. Steinbach, "RO-LOAM: 3D reference object-based trajectory and map optimization in LiDAR odometry and mapping," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6806–6813, Jul. 2022.

[21] G. P. Meyer, "An alternative probabilistic interpretation of the Huber loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5261–5269.

[22] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 2100–2106.

[23] Y. Liu, Y. Fu, F. Chen, B. Goossens, W. Tao, and H. Zhao, "Simultaneous localization and mapping related datasets: A comprehensive survey," 2021, *arXiv:2102.04036*.

[24] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, p. 1729881419841532, 2019.

[25] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4802–4809.

[26] 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.

[27] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "MulRan: Multimodal range dataset for urban place recognition," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 6246–6253.

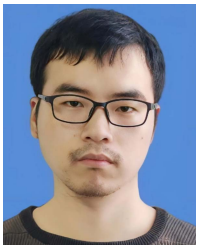
[28] Z. Zhou, M. Yang, C. Wang, and B. Wang, "ROI-cloud: A key region extraction method for LiDAR odometry and localization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3312–3318.

[29] 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.

[30] S. Guo, Z. Rong, S. Wang, and Y. Wu, "A LiDAR SLAM with PCA-based feature extraction and two-stage matching," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

Jikai Wang received the B.S. degree from the University of Yanshan, Qinhuangdao, China, in 2014, and the Ph.D. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2020.

He currently holds a post-doctoral position at the Department of Automation, USTC. His research interests include small sample processing, mobile robots' simultaneous localization and mapping (SLAM), and knowledge representation.



Meng Xu received the B.S. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2018, where he is currently pursuing the Ph.D. degree with the Department of Automation.

His research interests include 3-D reconstruction, robotics, and visual simultaneous localization and mapping (SLAM).

Guangpu Zhao received the B.S. degree from the Hefei University of Technology, Hefei, China, in 2020. He is currently pursuing the master's degree with the Department of Automation, University of Science and Technology of China (USTC), Hefei.

His research interests include machine learning and 3-D light detection and ranging (LiDAR) simultaneous localization and mapping (SLAM).

Zonghai Chen (Senior Member, IEEE) was born in Anhui, China, in 1963. He received the B.S. degree in automation and the M.E. degree in control theory and control engineering from the University of Science and Technology of China (USTC), Hefei, China, in 1988 and 1991, respectively.

He has been a Professor with the Department of Automation, USTC, since 1998. His research interests include modeling and control of complex systems, intelligent robotic and information processing, energy management technologies for electric vehicles, and smart microgrids.

Prof. Chen is a member of the Robotics Technical Committee and the Modelling, Identification and Signal Processing Technical Committee of the International Federation of Automation Control (IFAC). He was a recipient of special allowances from the State Council of the People's Republic of China.