

# WiCRF: Weighted Bimodal Constrained LiDAR Odometry and Mapping With Robust Features

Dengxiang Chang , Runbang Zhang, Shengjie Huang, Manjiang Hu , Rongjun Ding, and Xiaohui Qin 

**Abstract**—Accurate localization is a fundamental capability of autonomous driving systems, and LiDAR has been widely used for localization systems in recent years due to its high reliability and accuracy. In this paper, we propose a robust and accurate LiDAR SLAM, which innovates feature point extraction and motion constraint construction. For feature extraction, the proposed adaptive point roughness evaluation based on geometric scaling effectively improves the stability and accuracy of feature points (plane, line). Then, outliers are removed with a dynamic threshold filter, which improves the accuracy of outlier recognition. For motion constraint construction, the proposed weighted bimodal least squares is employed to optimize the relative pose between current frame and point map. The map stores both 3D coordinates and vectors (principal or normal vectors). Using vectors in current frame and point map, bimodal reprojection constraints are constructed. And all constraints are weighted according to the neighboring vector distribution in the map, which effectively reduces the negative impact of vector errors on registration. Our solution is tested in multiple datasets and achieve better performance in terms of accuracy and robustness.

**Index Terms**—SLAM, localization, mapping.

## I. INTRODUCTION

AUTOMATIC localization is one of the key technologies for autonomous driving. Environment perception, path planning, and motion control all rely on vehicle localization results. To achieve robust and high-precision localization, autonomous vehicles are equipped with various sensors, such as GNSS/IMU, camera, LiDAR, etc. In recent years, localization solutions based on pre-built high precision maps have been proposed to solve the pose estimation under GNSS signal blockage or loss scenarios such as tree-lined roads, urban canyons and underground

Manuscript received 8 August 2022; accepted 21 December 2022. Date of publication 30 December 2022; date of current version 2 February 2023. This letter was recommended for publication by Associate Editor B. Englot and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported in part by the National Key R&D Program of China under Grant 2021YFB2501800, in part by the NSF of China under Grants 52222216 and 52172384, in part by the Changsha NSF of China under Grant kq202162, and in part by the State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body under Grant 32125011. (Corresponding author: Xiaohui Qin.)

Dengxiang Chang, Runbang Zhang, and Shengjie Huang are with the College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410006, China (e-mail: cdx@hnu.edu.cn; runbang@hnu.edu.cn; hovxb428@hnu.edu.cn).

Manjiang Hu, Rongjun Ding, and Xiaohui Qin are with the State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410006, China, and also with the Wuxi Intelligent Control Research Institute of Hunan University, Wuxi 214115, China (e-mail: manjiang\_h@hnu.edu.cn; dingrj@hnu.edu.cn; qxh880507@163.com).

Digital Object Identifier 10.1109/LRA.2022.3233229

garages [1], [2]. However, these approaches cannot be applied to environments with large changes. Simultaneous localization and mapping (SLAM) technology can provide pose information and construct high precision maps in unknown environments or environments with large changes. Therefore, SLAM is essential for building a comprehensive autonomous localization system.

In the past decade, numerous SLAM methods based on different sensors have been proposed, such as vision-based SLAM [3], [4], [5] and LiDAR-based SLAM [6], [7], [8]. Currently, vision-based methods have been used in robotics, but are still sensitive to illumination, depth estimation and initial values, making them unable to meet requirements of autonomous driving. Compared with camera, LiDAR is not affected by illumination conditions and can acquire accurate geometry information. In recent years, the large-scale application of LiDAR has greatly reduced the cost of hardware, while the rapid development of manufacturing technology has improved the reliability of the device.

LiDAR-SLAM can be divided into ICP-based methods and feature-based methods. The classical iterative closest point (ICP) directly uses raw point cloud for registration, which has high computational cost and cannot meet the real-time requirement. Normal distribution transformation (NDT) algorithm introduces voxel filtering together with normal distribution to reduce the number of point while retaining the original environmental information. Compared with the classical ICP algorithm, the computational efficiency is improved, but the registration accuracy is decreased. Feature-based methods extract key points from raw point cloud and construct geometric primitive constraints, such as point-to-line and point-to-plane constraints, which significantly reduce the computational resource consumption while retaining sufficient motion constraints. Although feature-based methods mostly outperform ICP-based methods, there are still some drawbacks: feature extraction process does not consider invariance to LiDAR viewpoint, resulting in unstable and inaccurate feature points; vector estimation of feature point is not always reliable, however it is difficult to eliminate the negative impact of vector errors on geometric primitive constraints.

To solve above problems, a novel weighted bimodal constrained LiDAR odometry and mapping with robust features (WiCRF) is proposed. First, the adaptive roughness algorithm scales the inter-point geometric configuration to enhance the robustness of roughness on LiDAR viewpoints and improve the stability of feature recognition. Then, we utilize dynamic distance threshold to check each feature correspondence, which significantly improves the accuracy of outlier rejection. Finally, based on feature vector in current frame and vector distribution

in point map, we propose a weighted bimodal reprojection constraint to reduce the impact of vector errors. Experimental results show that our method is more stable and accurate compared with existing state-of-the-art methods.

The main contributions of this paper are as follows:

- We propose a novel feature-based LiDAR SLAM system that employs stable feature points to construct robust geometric primitive constraints that improve the stability and accuracy of localization system.
- We use adaptive point roughness algorithm and dynamic outlier threshold to improve the stability and accuracy of feature points.
- We propose a pose estimation algorithm based on weighted bimodal geometric primitive constraints that effectively enhances the robustness of localization system to feature vector errors.
- We construct a point map containing feature vectors to display feature vector distributions and introduce them into geometric primitive constraints.

The paper is organized as follows. Section II presents a review of related works. The overview of the proposed LiDAR SLAM system is introduced in Section III. Section IV describes the detail of the proposed approach, including feature extraction, feature association, weighted bimodal pose estimation, and mapping. Experimental results are presented in Section V. Finally, Section VI summarizes this paper.

## II. RELATED WORKS

In recent years, many research groups have worked on improving the robustness and accuracy of LiDAR SLAM systems. According to the type of point cloud registration, they are classified into ICP-based and feature-based methods.

### A. ICP-Based Methods

ICP-based methods use raw or voxelized point cloud to match the target and source point clouds. Classical ICP algorithm [9] iteratively updates the transformation matrix by minimizing point-to-point distance. This method has well registration accuracy for dense point cloud, but it is computationally expensive and sensitive to initial values. For sparse point cloud, the registration accuracy is significantly reduced because there is less overlap between adjacent scans. GICP proposed by Segal [10] is a well-known variant of ICP, which adopts a distribution-to-distribution registration strategy. Compared with classical ICP, it is more robust to sparse point cloud. ICP and GICP all rely on K-d tree to provide strict nearest neighbor (NN) searching, which does not satisfy the real-time requirement. In order to reduce the time for K-d tree building, FAST-LIO2 [11] proposes a iK-d tree which supports incremental map update. It relies on an IMU to provide an initial pose to speed up convergence.

NDT algorithm [12] voxelizes target point cloud and introduces normal distribution to approximate the geometric information within each voxel. This method applies voxel association instead of K-d tree searching, which improves the computational efficiency. DLO [13] is built around the philosophy of speed-first, which reduces the computational cost of GICP by introducing voxel filtering. LiTAMIN [14], inspired by NDT and classical ICP, employs a locally approximated geometry

with clusters of normal distributions. Meanwhile, K-d tree is adopted to guarantee the NN searching. LiTAMIN2 [15] extends distribution-to-distribution matching. In addition, it introduces symmetric KL-divergence to maintain accuracy while further increasing voxel resolution to reduce computational cost, but the fixed voxel resolution is poorly adapted to environments. VGICP [16] also extends GICP algorithm with voxelization. In contrast to the work in [15], it employs voxel association to avoid costly NN search. Beside it utilizes distribution-to-multi-distribution matching to improve its robustness to voxel resolution. The recent work [17] uses GPU-accelerated VGICP to create a densely matching factor graph. Its preprocessing consists of costly NN search and covariance estimation for every point. In conclusion, the application of voxel filtering reduces computational cost, but registration accuracy is usually sensitive to voxel resolution. The application of distributions improves the robustness to voxel resolution, but still suffers from local geometric distortions and time-consuming estimation of complex distributions.

### B. Feature-Based Methods

Feature-based approaches extract representative features, such as plane, line, from raw point cloud and estimate transformation matrix based on effective feature correspondences. LOAM [18] introduces point roughness to extract plane and line features, which significantly reduces the number of points used for registration. LeGO-LOAM [19] introduces ground extraction and segmentation algorithms to filter out unreliable points. Then, plane features are extracted from ground points and line features from non-ground points, respectively. However, the feature extraction of LOAM and LeGO-LOAM uses only geometric information of consecutive points from the same row and does not eliminate the impact of LiDAR viewpoint on roughness estimation. As a follow-up, MULLS [20] first extracts ground points using double-threshold ground filtering, and then classifies non-ground points using principal component analysis (PCA) to extract five categories of features. Finally, each correspondence is weighted based on residuals, motion constraint contribution and intensity consistency, which effectively improves registration robustness. However, this method requires PCA for a large number of points, which greatly increases the computational cost. CT-ICP [21] combines simple grid sampling and planarity estimation to effectively simplify plane feature extraction. In addition, continuity in scan matching and discontinuity between scans are introduced to improve robustness to discontinuous high-frequency motion.

In recent years, some works have attempted to introduce intensity and semantics into SLAM systems to improve feature matching robustness and accuracy. Wang et al. [22] combines calibrated intensities and geometric distributions to extract features and estimate pose by minimizing geometric and intensity residuals. However, it is difficult to obtain accurate calibrated intensity. Segmap [23] uses 3D CNN to learn semantic information and combines it with LOAM to reject features associated with potentially dynamic objects. SA-LOAM [24] proposes a semantic-assisted ICP, which associates only features with the same semantic to reduce incorrect matching. In addition, different voxel resolutions are set for different semantic objects

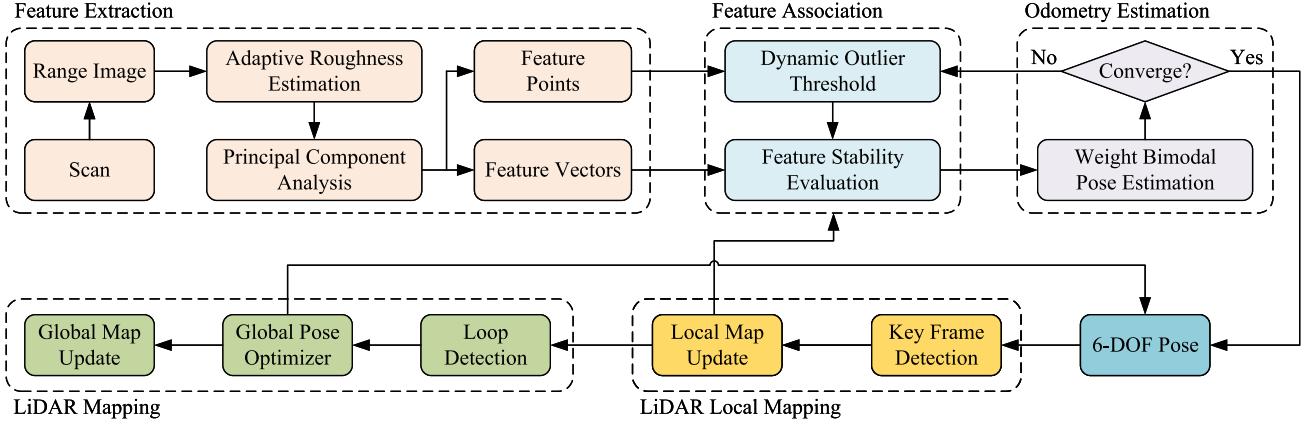


Fig. 1. System overview of proposed WiCRF-SLAM. In feature extraction, we use adaptive roughness function and PCA to extract stable features. In pose estimation, we combine feature stability scores and vectors in point map to construct weighted bimodal constraints.

to reduce point data while maintaining the information of important small objects. In practice, deep learning still suffers from defects, such as difficulties in data labeling and poor adaptation to environments.

From above discussions, we can see that feature-based methods are generally less computationally expensive than ICP-based methods. However, relying on geometric information to extract stable features has not been well addressed. In addition, vector error of feature point is one of the main causes of localization drift, but no effective solution has been proposed to solve this problem. In this paper, we aim to extract stable features and reduce adverse influences of vector errors on localization accuracy.

### III. SYSTEM OVERVIEW

This section describes the proposed LiDAR system, WiCRF, as shown in Fig. 1. The overall system consists of five modules. Feature extraction module receives raw point cloud and outputs robust features. First, the raw point cloud is projected into a range image. The roughness of each point is estimated using an adaptive roughness evaluation function that is robust to LiDAR viewpoints. Then, points in each sub-image are sorted based on roughness. Finally, we perform PCA on neighboring point set searched by range image indexes to estimate normal vector of valid plane point or principal vector of valid line point.

Feature association module evaluates feature correspondences between current frame and local map. First, nearest points are searched in corresponding local map according to feature category. Then, outlier detection unit removes incorrect correspondences based on distance threshold and point-to-plane or point-to-line distance. Unlike LOAM, which uses a fixed distance threshold, we calculate dynamic distance threshold based on point range and pose optimization information. Feature stability evaluation unit uses vectors stored in local map to estimate the consistency of vector distribution.

Odometry estimation module uses effectively associated features to find relative transformation between current frame and local map. To reduce the negative influence of unstable feature vectors on pose estimation, pose optimization unit weights

reprojection constraints with feature stability parameters obtained from feature association module. In addition, bimodal reprojection constraints are constructed based on vectors in local map and vectors of current frame to improve robustness to vector errors.

LiDAR local mapping module performs keyframe detection and registers features in keyframe to corresponding local maps according to feature categories. LiDAR global mapping module performs loop-closure detection between keyframes and constructs pose graph optimization based on loop-closure constraint to refine relative poses between historical keyframes. It is worth mentioning that both local and global maps consist of feature points and corresponding vectors. The vector distribution in maps is used for feature stability evaluation and bimodal constraint construction.

## IV. METHODOLOGY

### A. Feature Extraction

Typical LiDAR single scan consists of tens of thousands of points and can generate tens of scans per second. The task of feature extraction is to reduce the number of point clouds while removing noisy data and retaining stable plane and line features.

*1) Adaptive Roughness Estimation:* In this paper, point roughness with invariance to LiDAR viewpoint is obtained by scaling inter-point geometric configuration. Local sparsity of point cloud is closely related to LiDAR viewpoint. The further away from LiDAR viewpoint, the sparser point cloud in its neighborhood, and the greater difference in geometric information between neighboring points. Both LOAM and LeGO-LOAM do not consider the influence of viewpoint on roughness. The tendency of distant point to have greater roughness compared to close point makes it difficult to extract stable features.

In neighboring point search, we use distance threshold  $\delta_d$  instead of point number threshold to obtain consecutive neighboring points with variable numbers from the same row, which significantly weakens the adverse impact of point cloud sparsity on roughness estimation. In addition, the geometric configuration between neighboring points  $p_{i-n}$  and  $p_{i+n}$  on each side of point  $p_i$  is scaled to make the roughness evaluation function

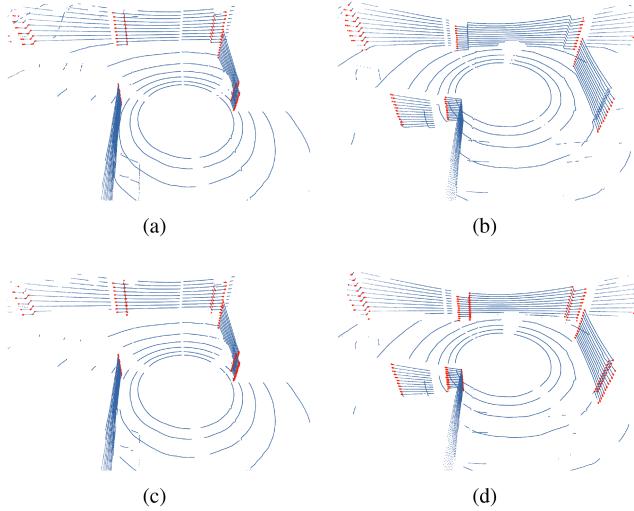


Fig. 2. The figure shows the comparison of point cloud classification results based on adaptive roughness and roughness used in LOAM. Blue points indicate candidate plane points and red points indicate candidate line points. (a) and (b) are derived from LOAM. (c) and (d) are derived from WiCRF. Comparing (a) and (b), it can be seen that the roughness used in LOAM is sensitive to LiDAR viewpoints. Under the same conditions, both (c) and (d) can accurately identify point categories.

insensitive to point range. The roughness  $c_i$  that is robust to LiDAR viewpoint is defined by:

$$c_i = \frac{1}{N_i} \sum_{n=1}^{N_i} \frac{\delta_d \|(\mathbf{p}_{i+n} - \mathbf{p}_i) - (\mathbf{p}_{i-n} - \mathbf{p}_i)\|}{\min(\|\mathbf{p}_{i+n} - \mathbf{p}_i\|, \|\mathbf{p}_{i-n} - \mathbf{p}_i\|)}. \quad (1)$$

Where  $N_i$  is the number of neighboring points on each side of  $\mathbf{p}_i$ , which is calculated as:

$$\begin{aligned} N_i^L &= \min(n) \text{ s.t. } \|(\mathbf{p}_{i-n} - \mathbf{p}_i)\| \geq \delta_d, \\ N_i^R &= \min(n) \text{ s.t. } \|(\mathbf{p}_{i+n} - \mathbf{p}_i)\| \geq \delta_d, \\ N_i &= \max(N_i^L, N_i^R). \end{aligned} \quad (2)$$

Then, point cloud is categorized using roughness threshold  $\delta_c$ . As shown in Fig. 2, points with roughness less than  $\delta_c$  are marked as candidate plane points and vice versa as candidate line points. Note that since we use distance threshold  $\delta_d$  to scale geometric configuration between each neighboring point pair, making roughness  $c_i$  consistent with  $\delta_d$ . Thus roughness threshold  $\delta_c$  is set by  $\delta_c = w_c \delta_d$ .  $w_c$  is the coefficient for  $\delta_d$ . In practice, we fix  $w_c = 1$ .

2) *Feature Filtering Based on PCA*: Considering that roughness-based feature categorization only utilizes geometric information among neighboring points from a single row, we introduce PCA to check validity of each candidate feature using neighboring points from multiple rows. Meanwhile, feature vectors are calculated which are used for feature stability evaluation in Section IV-B and bimodal pose constraint construction in Section IV-C.

Points with minimum roughness in candidate plane point set or maximum roughness in candidate line point set are input to PCA module sequentially. In this process, combining range

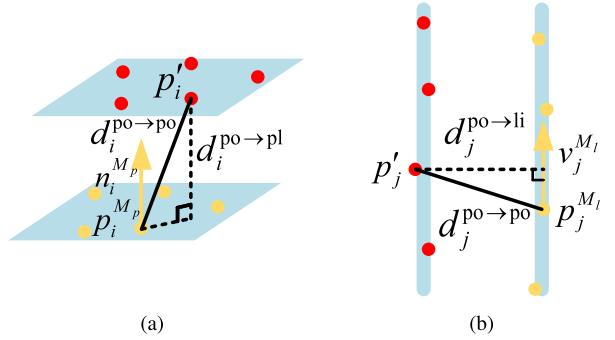


Fig. 3. The figure shows the differences between point-to-point distance  $d^{po \rightarrow po}$ , point-to-plane distance  $d^{po \rightarrow pl}$  and point-to-line distance  $d^{po \rightarrow li}$ . Due to the sparse property of point cloud, outlier recognition based on point-to-point distance is prone to mistakenly reject features with correct associations.

image index, category of point  $p$  and distance threshold, we can quickly obtain neighboring point set  $S$  with the same category as  $p$ . Eigenvalues  $\lambda_1 > \lambda_2 > \lambda_3$  and corresponding eigenvectors  $v$ ,  $m$ ,  $n$  are calculated by performing eigenvalue decomposition of covariance matrix of  $S$ . For each candidate plane point, if its planar scalar  $\sigma_{2D} = (\lambda_2 - \lambda_3)/\lambda_1$  is smaller than pre-set threshold  $\delta_{2D}$ , the point is added to plane feature set  $F^p$ , while the corresponding normal vector  $n$  is saved. For each candidate line point, if its linear scalar  $\sigma_{1D} = (\lambda_1 - \lambda_2)/\lambda_2$  is greater than pre-set threshold  $\delta_{1D}$ , the point is added to line feature set  $F^l$ , while the corresponding primary vector  $v$  is saved. Note that in order to obtain features with uniform spatial distribution and reduce the computational cost of PCA, the spatial sampling strategy as LeGO-LOAM is introduced, i.e., the range image is divided into several sub-images, and a maximum number of features is preset for each sub-image (see more detail in [19]).

## B. Feature Association

The task of feature association module is to reject incorrect feature correspondences and estimate the stability of feature points that associated correctly.

1) *Dynamic Threshold-Based Outlier Detection*: The reprojection geometry error is positively related to the point range. In order to better identify outlier points, a dynamic distance threshold for outlier detection is introduced as follows:

$$\delta_o = e_\theta \|\mathbf{p}_i\| + e_t. \quad (3)$$

Where  $e_\theta$  and  $e_t$  are the pre-set rotation error and translation error, which are continuously reduced with the iterative optimization.

LOAM uses point-to-point reprojection distance to reject outliers. Due to the limitation of LiDAR resolution, it is difficult to repeatedly scan the same point during the motion. As shown in Fig. 3, the point-to-point reprojection distance is naturally large, which hardly reflects the true reprojection distance, resulting in some features with correct correspondences being recognized as outliers. In this paper, based on already saved vectors in local map, point-to-plane reprojection distance  $d^p$  and point-to-line

reprojection distance  $d^l$  are introduced respectively:

$$d_i^p = \left\| \left( \widehat{\mathbf{T}} \mathbf{p}_i^p - \mathbf{p}_i^{M_p} \right) \mathbf{n}_i^{M_p} \right\|^2, \quad (4)$$

$$d_j^l = \left\| \left( \widehat{\mathbf{T}} \mathbf{p}_j^l - \mathbf{p}_j^{M_l} \right) \times \mathbf{v}_j^{M_l} \right\|^2. \quad (5)$$

Where  $\widehat{\mathbf{T}}$  is an estimation of transformation matrix between current frame and local map,  $\mathbf{p}_i^p$  and  $\mathbf{p}_j^l$  are plane feature and line feature in current frame, respectively,  $\mathbf{p}_i^{M_p}$  is the nearest point of transformed point  $\widehat{\mathbf{T}} \mathbf{p}_i^p$  in local plane map,  $\mathbf{n}_i^{M_p}$  is the normal vector of  $\mathbf{p}_i^{M_p}$ ,  $\mathbf{p}_j^{M_l}$  is the nearest point of transformed point  $\widehat{\mathbf{T}} \mathbf{p}_j^l$  in local line map,  $\mathbf{v}_j^{M_l}$  is the primary vector of  $\mathbf{p}_j^{M_l}$ . For plane feature,  $d^p$  greater than  $\delta_o$  is considered as incorrect association. For line feature,  $d^l$  greater than  $\delta_o$  is considered as incorrect association.

2) *Feature Stability Evaluation*: The consistency of vector distribution among neighboring features reveals the stability of feature. Stable feature vectors are especially important to construct robust point-to-plane constraints or point-to-line constraints. We use neighborhood vector distribution in map to estimate the stability of each feature. The stability function of plane feature is defined as:

$$w_i^p = e^{\frac{\sum_{k=2}^K \mathbf{n}_i^{M_p} \mathbf{n}_{i,k}^{M_p}}{K-1} - 1}. \quad (6)$$

Where  $\mathbf{n}_i^{M_p}$  and  $\mathbf{n}_{i,k}^{M_p}$  are normal vectors of  $K$  nearest points associated with  $\mathbf{p}_i^p$  in local plane map. Similarly, the stability function of line feature is defined as:

$$w_j^l = e^{\frac{\sum_{k=2}^K \mathbf{v}_j^{M_l} \mathbf{v}_{j,k}^{M_l}}{K-1} - 1}. \quad (7)$$

Where  $\mathbf{v}_j^{M_l}$  and  $\mathbf{v}_{j,k}^{M_l}$  are primary vectors of  $K$  nearest points associated with  $\mathbf{p}_j^l$  in local line map, respectively.

### C. Weight Bimodal Pose Estimation

This module uses features in current frame and corresponding points in the local map to construct geometric primitive constraints, and then iteratively minimizes point-to-plane and point-to-line residuals and finally output the optimal transformation estimation. Note that although the adaptive roughness and PCA filter out unstable features, the feature vectors vary from frame to frame due to the variation of LiDAR viewpoints and the sparse nature of the point cloud. For the point-to-plane and point-to-line residuals, the gradient decreases along the normal direction of the plane and the orthogonal vector direction of the line vector, respectively. We present a pose optimization function based on bimodal geometric primitive constraints, which is constructed with feature vectors in current frame and corresponding vectors in local map. The gradient direction is given by the sum vector of the current frame feature vector and the map feature vector, which improves the robustness of the pose optimization function to feature vector errors. For each plane feature  $\mathbf{p}_i^p$  in current frame, the bimodal point-to-plane residual can be defined as:

$$\begin{aligned} r_{i,1}^p &= \left\| \left( \mathbf{T} \mathbf{p}_i^p - \mathbf{p}_i^{M_p} \right) \mathbf{n}_i^{M_p} \right\|^2, \\ r_{i,2}^p &= \left\| \left( \mathbf{T}^{-1} \mathbf{p}_i^{M_p} - \mathbf{p}_i^p \right) \mathbf{n}_i \right\|^2. \end{aligned} \quad (8)$$

Similarly, for each line feature  $\mathbf{p}_j^l$  in current frame, the bimodal point-to-line residual can be defined as:

$$\begin{aligned} r_{j,1}^l &= \left\| \left( \mathbf{T} \mathbf{p}_j^l - \mathbf{p}_j^{M_l} \right) \mathbf{v}_j^{M_l} \right\|^2, \\ r_{j,2}^l &= \left\| \left( \mathbf{T}^{-1} \mathbf{p}_j^{M_l} - \mathbf{p}_j^l \right) \mathbf{v}_j \right\|^2. \end{aligned} \quad (9)$$

In addition, feature stability parameters  $w_i^p$  and  $w_j^l$  are applied to penalize corresponding constraints with large feature vector inconsistencies. Thus, the pose estimation can be formulated as a weighted least square optimization problem with bimodal geometric primitive constraints:

$$\mathbf{T}^* = \min_{\mathbf{T}} \sum_i w_i^p (r_{i,1}^p + r_{i,2}^p) + \sum_j w_j^l (r_{j,1}^l + r_{j,2}^l). \quad (10)$$

The Levenberg-Marquardt algorithm [25] is applied to solve the optimal pose estimation with the initial pose provided by constant velocity model.

### D. Local Mapping & Global Mapping

The local map is used for feature matching of LiDAR odometry. The global map provides historical environmental measurements for loop-closure detection. Both local and global maps contain 3D coordinates of features and corresponding vectors. Furthermore, plane map and line map are constructed separately according to feature categories. To reduce map data while decreasing the number of loop-closure detection, we extract key frames based on translation increment and rotation increment. The local map is updated by appending latest keyframe and cropping earliest keyframe. For keyframe cropping, we must guarantee that the number of keyframes is greater than a pre-set number threshold and that the distance between the latest keyframe and the earliest keyframe is greater than a pre-set distance threshold.

The global map stores all keyframes. To reduce odometry drift, we use Scan Context [26] to identify candidate loop-closure frame. Since the pose drift between successive frames in a short period of time is small, a local map of loop-closure scene is formed by selecting a group of neighboring keyframes of candidate loop-closure frame. Then the relative transformation matrix of loop-closure is solved by minimizing the proposed weighted bimodal residuals. To check the reliability of loop-closure constraint and remove spurious closed-loop, a percentage threshold of correctly associated features occupying all features is exploited. Finally, loop-closure constraint and odometry constraints are added to pose graph optimization module. And the consistent global map is constructed with refined keyframe poses.

## V. EXPERIMENTS

In this section we present experiments to validate the proposed WiCRF system, including experiments based on public dataset and private dataset. For public dataset tests, the proposed method is evaluated in KITTI with GNSS/IMU measurements as ground truths. For private dataset tests, the evaluation is performed on an autonomous driving platform (ADP). The localization accuracy of our method is compared with literature works using relative average translation error (RATE) and relative average

TABLE I  
PERFORMANCE EVALUATION OF EACH LOCALIZATION METHOD ON KITTI DATASET

Method	00*-U	01-H	02*-C	03-C	04-C	05*-C	06*-U	07*-U	08*-U	09*-C	10-C	00-10 Average	11-21 Average
LOAM	0.78/-	1.43/-	0.92/-	0.86/-	0.71/-	0.57/-	0.65/-	0.63/-	1.12/-	0.77/-	0.79/-	0.84/-	<b>0.55/0.13</b>
CT-ICP	<b>0.49</b> /-	0.76/-	0.52/-	0.72/-	0.39/-	<b>0.25</b> /-	<b>0.27</b> /-	0.31/-	<b>0.81</b> /-	0.49/-	<b>0.48</b> /-	0.53/-	<b>0.59/0.14</b>
LITAMIN2*	0.70/0.28	2.10/0.46	0.98/0.32	0.96/0.48	1.05/0.52	0.45/0.25	0.59/0.34	0.44/0.32	0.95/0.29	0.69/0.40	0.80/0.47	0.85/0.33	-/-
SUMA++*	0.64/0.22	1.60/0.46	1.00/0.37	0.67/0.46	0.37/0.26	0.40/0.20	0.46/0.21	0.34/0.19	1.10/0.35	<b>0.47</b> /0.23	0.66/0.28	0.70/0.29	1.06/0.34
GPU-GICP	<b>0.49/0.16</b>	<b>0.65/0.10</b>	<b>0.50/0.12</b>	<b>0.62/0.19</b>	<b>0.41/0.10</b>	<b>0.24/0.10</b>	<b>0.29/0.07</b>	<b>0.30/0.11</b>	<b>0.80/0.18</b>	<b>0.46/0.11</b>	<b>0.54/0.15</b>	<b>0.52/0.14</b>	<b>0.59/0.15</b>
LO-Net	0.78/0.42	1.42/0.40	1.01/0.45	0.73/0.59	0.56/0.54	0.62/0.35	0.55/0.35	0.56/0.45	1.08/0.43	0.77/0.38	0.92/0.41	0.83/0.42	1.75/0.79
MULLS	<b>0.51/0.18</b>	<b>0.62/0.09</b>	0.55/0.17	<b>0.61/0.22</b>	<b>0.35/0.08</b>	0.28/0.17	<b>0.24/0.11</b>	<b>0.29/0.18</b>	<b>0.80/0.25</b>	0.49/0.15	0.61/0.19	<b>0.49/0.16</b>	0.65/0.19
WiCRF-LO	0.62/0.28	1.00/0.39	0.53/0.18	0.64/0.25	<b>0.36/0.14</b>	0.45/0.25	0.44/0.21	<b>0.30/0.19</b>	0.88/0.24	<b>0.47/0.10</b>	<b>0.53/0.18</b>	0.56/0.22	0.68/0.28
WiCRF-SLAM*	<b>0.50</b> /0.20	1.00/0.39	<b>0.49/0.14</b>	0.64/0.25	<b>0.36/0.14</b>	0.28/0.15	0.43/0.19	<b>0.30/0.19</b>	0.85/0.22	<b>0.46/0.10</b>	<b>0.53/0.18</b>	0.53/0.19	-/-

Relative average pose errors over different lengths from 100 M to 800 M: rate is represented as % and rare is represented as deg/100 m. The mark \* represents with loop-closure. Capital letters U, H and C denote urban, highway and country, respectively. Results of all comparison methods are from the corresponding original papers. Bold font indicates the top two.

rotation error (RARE) (see more details in [27]). All experiments are conducted on a PC equipped with an Intel Core i7-10750H@2.60 GHz CPU.

### A. Experiments on Public Dataset

The KITTI dataset is captured from a vehicle equipped with Velodyne HDL-64E LiDAR. It contains 22 sequences covering multiple complex scenarios, such as urban, highway and country. We use RATE and RARE to evaluate LiDAR odometry performance. In addition, LiDAR SLAM performance is evaluated using sequences with closed-loop scenarios. Comparison results of the proposed WiCRF system with existing state-of-the-art LiDAR-based SLAM solutions are shown in Table I. Sequences 00-10 provide GNSS/IMU ground truth. WiCRF-LO obtains the average RATE of about 0.56% and the average RARE of 0.22 deg/100 m over Sequences 00-10. With the loop closure, WiCRF-SLAM gets better accuracy on both RATE and RARE, 0.53% and 0.19 deg/100 m respectively, which is comparable to the performance of current state-of-the-art methods. Our localization system achieves the top two RATEs or RAREs in Sequence 00, 02, 04, 05, 07, 08, 09 and 10, which account for about 70% of Sequences 00-10. In addition, WiCRF-LO enters the top 12% of the leaderboard with RATE of 0.68% and RARE of 0.28 deg/100 m over Sequences 11-21 that are used as online assessment without ground truth. The results verify the robustness and accuracy of WiCRF system. Some of trajectories and maps output by WiCRF covering three representative scenarios are shown in Fig. 4. It can be seen that our method can build precise point maps with low-drift poses even in challenging featureless environments with highly dynamic objects.

### B. Experiments on Private Dataset

As shown in Fig. 5, the vehicle is equipped with RS-LiDAR-16 and CGI-610 GNSS/IMU mounting on the top side. Both point clouds and GNSS/IMU measurements are captured at 10 Hz, and RTK-GNSS/IMU is used as ground truth in outdoor environments. We collect ADP datasets from three kinds of environments including park, urban and suburban. Similar to evaluations on KITTI, we use RATE and RARE to verify the performance of our system and compare it with open-source algorithms. The results of WiCRF-LO, WiCRF-SLAM and other comparison methods are shown in Table II. It can be seen

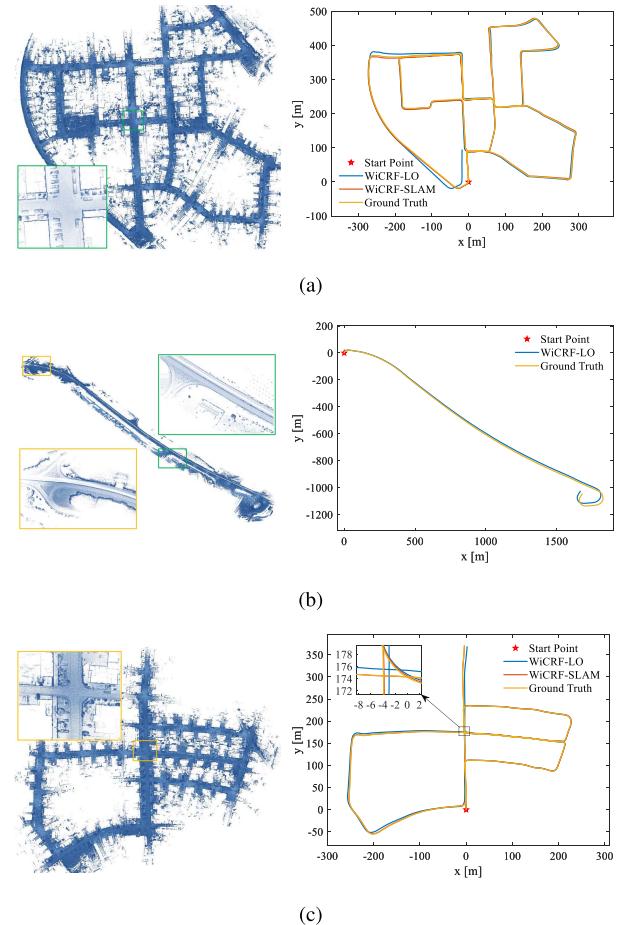


Fig. 4. Results of WiCRF on KITTI dataset. (a) Sequence 00 is urban scenario. (b) Sequence 01 is highway scenario. (c) Sequence 05 is country scenario.

that our method is competitive in terms of both RATE and RARE. WICRF-LO achieves the top two RATEs or RAREs in Sequence 00, 01 and 02. WICRF-SLAM gets the best RATE over all sequences. Specially, the performance of Sequence 00 in park improved more significantly compared to other sequences because the park is planted with numerous trees that are not conducive to localization accuracy, while we can extract more stable and accurate features compared to other open-source algorithms. Fig. 6(a) shows trajectories of WiCRF and ground truth.



Fig. 5. Autonomous vehicle platform equipped with RS-LiDAR-16 and CGI-610 GNSS/IMU.

TABLE II  
PERFORMANCE EVALUATION OF EACH LOCALIZATION METHOD ON ADP DATASETS

Method	00*-P	01*-U	02-S	03*-S	Average
LOAM	2.63/2.20	0.70/0.44	0.77/0.20	0.79/0.23	1.22/0.76
LeGO-LOAM*	0.62/0.56	0.69/0.38	0.72/0.22	0.85/0.27	0.72/0.35
CT-ICP	<b>0.46/0.34</b>	<b>0.38/0.27</b>	0.48/0.21	<b>0.59/0.21</b>	<b>0.47/0.25</b>
MULLS	0.52/0.37	0.46/0.35	<b>0.37/0.19</b>	0.68/0.23	0.50/0.28
WiCRF-LO	<b>0.38/0.36</b>	<b>0.42/0.37</b>	<b>0.27/0.18</b>	0.63/0.24	<b>0.42/0.28</b>
WiCRF-SLAM*	<b>0.34/0.33</b>	<b>0.38/0.35</b>	<b>0.27/0.18</b>	<b>0.56/0.19</b>	<b>0.38/0.26</b>

Pose errors are represented as rate[%]/rare[DEG/100M]. Capital letters P, U and S denote park, urban, and suburban, respectively.

TABLE III  
ABLATION STUDIES OF WiCRF-LO ON KITTI AND ADP DATASETS

Method	KITTI-00	KITTI-01	ADP-00	ADP-01
WiCRF-LO <sup>1</sup>	0.66/0.30	1.13/0.43	0.41/0.40	0.50/0.42
WiCRF-LO <sup>2</sup>	0.67/0.31	1.20/0.44	0.40/0.41	0.46/0.40
WiCRF-LO <sup>3</sup>	0.67/0.29	1.73/0.43	0.41/0.39	0.61/0.40
WiCRF-LO	<b>0.62/0.28</b>	<b>1.00/0.39</b>	<b>0.38/0.35</b>	<b>0.42/0.37</b>

Pose errors are represented as ATE[m]/ARE[deg].

Fig. 6(b) demonstrates WiCRF’s mapping result of Sequence 00. Clear trees and vehicles in the map show that our approach can build consistent map in complex outdoor environments.

### C. Ablation Studies

The adaptive roughness evaluation, dynamic outlier threshold and weighted bimodal constraints are ablated to verify the effectiveness of the proposed method. The performance of WiCRF-LO<sup>1</sup> using the same roughness estimation as LOAM (without adaptive roughness), WiCRF-LO<sup>2</sup> using a point-to-point constant distance threshold (without dynamic outlier threshold), and WiCRF-LO<sup>3</sup> using unimodal point-to-plane and point-to-line constraints (without weighted bimodal constraints) were compared with WiCRF-LO on KITTT sequence 00-01 and ADP sequence 00-01, respectively. Table III shows WiCRF-LO outperforms WiCRF-LO<sup>1</sup>, WiCRF-LO<sup>2</sup> and WiCRF-LO<sup>3</sup> on all sequences, indicating the effectiveness of the method presented in Section IV. Among them, the performance improvement of

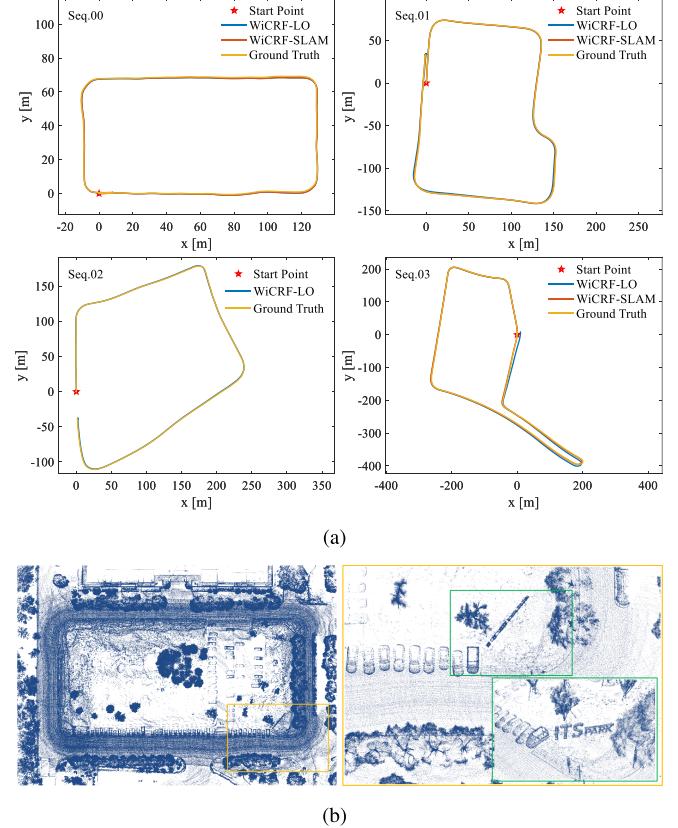


Fig. 6. Results of WiCRF on autonomous driving platform. (a) Trajectories of four sequences, (b) overview and details of mapping result with Sequence 00.

TABLE IV  
RUNTIME ANALYSIS PER FRAME FOR KITTI AND ADP (UNIT[MS])

Method	KITTI-00	KITTI-01	ADP-02	ADP-03
Feature extraction	34	26	10	9
Feature association	6	6	5	4
Pose estimation	7	6	4	4
Local map update	14	12	10	9
WiCRF-LO	61	50	30	26

the weighted bimodal constraint is the most obvious, especially for the highway scene. Because there are fewer structured objects in this scene, it is difficult to extract stable features, while the weighted bimodal constraint increases the importance of a few stable features, such as pillars and guardrails.

### D. Time Analysis

The time analysis for datasets collected from two different LiDAR is shown in the Table IV. The total runtime per frame of KITTI is about 55 ms. 2 K features are extracted from 100 K raw point clouds in about 30 ms, of which roughness estimation and PCA analysis are about 15 ms each. The feature association, including K-d tree NN search, dynamic threshold-based outlier filtering and feature stability analysis, takes about 6 ms for 5 iterations. Motion estimation is performed for about 1 ms per iteration. 100 K historical feature points of the local map are updated for about 13 ms. For tests on ADP datasets, each frame

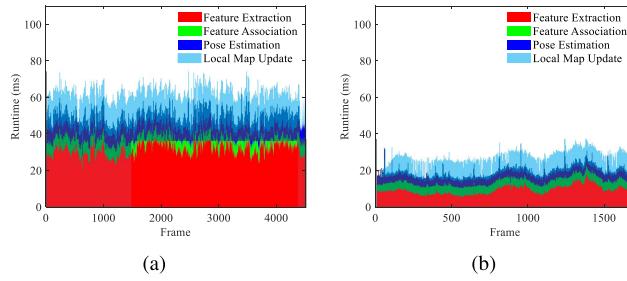


Fig. 7. Runtime per frame of WiCRF-LO. (a) Runtimes of WiCRF-LO on KITTI-00, (b) runtimes of WiCRF-LO on ADP-02.

of 25 K raw point clouds takes about 30 ms in total. Compared to KITTI, the runtime of feature extraction is significantly reduced to about 10 ms. Runtimes of feature association, motion estimation and local map update are slightly reduced to about 4 ms, 4 ms and 10 ms respectively. Furthermore, the runtime per frame for KITTI-00 and ADP-02 is shown in Fig. 7. Since the number of raw points varies per frame and the local map is updated only when a new keyframe is detected, the runtime varies per frame, but the maximum runtime per frame is still less than 100 ms.

## VI. CONCLUSION

In this paper, we present a real-time, robust and accurate WiCRF-SLAM system. Existing feature extraction methods based on geometric information are sensitive to LiDAR viewpoints. We introduce an adaptive roughness function to extract features that are invariant to LiDAR viewpoints. In addition, the proposed dynamic threshold for feature association enhances the accuracy of correspondences. Experimental results show that our method can extract more robust features in various challenging environments, which can significantly improve the system performance. Moreover, we construct point maps containing feature vectors to evaluate feature stabilities and introduce them into weighted bimodal geometric constraints. Experiment results of WiCRF achieve lower translation and rotation errors than the state-of-the-art LiDAR-based SLAM.

## REFERENCES

- [1] G. Wan et al., “Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 4670–4677.
- [2] W. Ding, S. Hou, H. Gao, G. Wan, and S. Song, “LiDAR inertial odometry aided robust LiDAR localization system in changing city scenes,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4322–4328.
- [3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
- [4] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
- [5] T. Qin, P. Li, and S. Shen, “VINS-Mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.
- [6] 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.*, 2019, pp. 4530–4537.
- [7] Q. Li et al., “LO-Net: Deep real-time LiDAR odometry,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8465–8474.
- [8] J. Lin and F. Zhang, “Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3126–3131.
- [9] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,” *Proc. SPIE*, vol. 1611, 1992, pp. 586–606.
- [10] A. Segal, D. Haehnel, and S. Thrun, “Generalized-ICP,” *Proc. Robot.: Sci. Syst.*, Seattle, WA, USA, vol. 2, no. 4, 2009, Art. no. 435.
- [11] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “FAST-LIO2: Fast direct LiDAR-inertial odometry,” *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.
- [12] P. Biber and W. Strasser, “The normal distributions transform: A new approach to laser scan matching,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3, 2003, pp. 2743–2748.
- [13] K. Chen, B. T. Lopez, A.-a. Agha-mohammadi, and F. Mehta, “Direct LiDAR odometry: Fast localization with dense point clouds,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2000–2007, Apr. 2022.
- [14] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, “LiTAMIN: LiDAR-based tracking and mapping by stabilized ICP for geometry approximation with normal distributions,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5143–5150.
- [15] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, “LiTAMIN2: Ultra light LiDAR-based SLAM using geometric approximation applied with kl-divergence,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11619–11625.
- [16] 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. 11054–11059.
- [17] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, “Globally consistent 3D LiDAR mapping with GPU-accelerated GICP matching cost factors,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8591–8598, Oct. 2021.
- [18] J. Zhang and S. Singh, “Low-drift and real-time LiDAR odometry and mapping,” *Auton. Robots*, vol. 41, no. 2, pp. 401–416, 2017.
- [19] 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.
- [20] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, “MULLS: Versatile LiDAR SLAM via multi-metric linear least square,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11633–11640.
- [21] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, “CT-ICP: Real-time elastic LiDAR odometry with loop closure,” in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 5580–5586.
- [22] H. Wang, C. Wang, and L. Xie, “Intensity-SLAM: Intensity assisted localization and mapping for large scale environment,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1715–1721, Apr. 2021.
- [23] R. Dube et al., “SegMap: Segment-based mapping and localization using data-driven descriptors,” *Int. J. Robot. Res.*, vol. 39, no. 2/3, pp. 339–355, 2020.
- [24] L. Li et al., “SA-LOAM: Semantic-aided LiDAR SLAM with loop closure,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 7627–7634.
- [25] J. J. Moré, “The Levenberg-Marquardt algorithm: Implementation and theory,” in *Numerical Analysis*. Berlin, Heidelberg: Springer, 1978, pp. 105–116.
- [26] 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.*, 2018, pp. 4802–4809.
- [27] A. Geiger, P. Lenz, and R. Urtasum, “Are we ready for autonomous driving The KITTI vision benchmark suite,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.