RF-LOAM: Robust and Fast LiDAR Odometry and Mapping in Urban Dynamic Environment

Jiong Li, Xudong Zhang, *Member, IEEE*, Yu Zhang, Yunfei Chang, and Kai Zhao

***Abstract*—In urban dynamic environment, most of the existing works on LiDAR simultaneous localization and mapping (SLAM) are based on static scene assumption and are greatly affected by dynamic obstacles. In order to solve this problem, this article is based on fast LiDAR odometry and mapping (F-LOAM) and adopts the FA-RANSAC algorithm, improved ScanContext algorithm, and global optimization to propose a robust and fast LiDAR odometry and mapping (RF-LOAM). First, the region-growing algorithm is used to cluster the fan-shaped grids. Then, we propose the FA-RANSAC algorithm based on feature information and adaptive threshold for dynamic object removal and extract the static edge and planar feature points for the first distortion compensation. Afterward, the estimated pose is calculated by the static feature points and is used to perform the second distortion compensation. Then, the height difference and adaptive distance threshold are used to improve the accuracy of ScanContext, and the efficiency of ScanContext is improved by deleting the loop closure historical matching frames and simplifying the feature matching. Finally, global optimization is used for keyframe. The experimental tests are carried out on the KITTI datasets, Urbanloco datasets, and our Extracted dataset. The results show that compared with the state-of-the-art SLAM methods, our method can not only accurately complete dynamic object removal and loop closure detection but also achieve more robust and faster localization and mapping in urban dynamic scenes.**

***Index Terms*— Autonomous vehicle, global optimization, LiDAR odometry, loop closure detection, simultaneous localization and mapping (SLAM).**

I. Introduction

SIMULTANEOUS localization and mapping (SLAM) has been a key technology in autonomous driving [1], [2]. Especially in the case of poor GPS signal or unknown dynamic scenes, this technology plays a vital role in autonomous driving. According to the type of sensor, the existing SLAM methods are mainly divided into based on LiDAR [3], [4], based on camera [5], and based on millimeter-wave radar [6], [7]. At present, two methods based on LiDAR (2-D and 3-D) [8], [9], [10] and camera (monocular, stereo, and RGB-D) [11], [12], [13] are the most common. Due to the fact that the camera is greatly affected by light and cannot be used all day long, this article adopts the more accurate and robust LiDAR in measurement for SLAM research of urban dynamic environment.

Although existing works on LiDAR SLAM have achieved good performance in public dataset evaluation, there are still some limitations in practical applications. The first limitation is poorly robustness from static to dynamic environment. For example, Hess et al. [14] have high accuracy in indoor or static outdoor environment, but the localization accuracy drops a lot in urban dynamic scenes. The second limitation is that it is difficult to balance computational cost and precision. For example, the low-drift and real-time LiDAR odometry and mapping (LOAM) [15] has high accuracy, but distortion compensation and mapping are computationally expensive. Moreover, although the fast LOAM (F-LOAM) [16] reduces time consumption as lightweight LiDAR SLAM, it lacks loop closure detection, resulting in large cumulative errors, and has low precision in urban dynamic scenes.

Therefore, this article proposes a robust and fast LiDAR SLAM method for dynamic urban scenes based on F-LOAM. The main framework of this proposed method is shown

Manuscript received 29 August 2023; revised 1 October 2023; accepted 7 October 2023. Date of publication 19 October 2023; date of current version 30 November 2023. This work was supported in part by the Natural Science Foundation of Beijing under Grant 3212013 and in part by the China Association for Science and Technology under Grant Young Elite Scientists Sponsorship Program by CAST. The associate editor coordinating the review of this article and approving it for publication was Prof. Yulong Huang. *(Corresponding author: Xudong Zhang.)*

Jiong Li is with the 95848 Army of PLA, Xiaogan 432019, China (e-mail: 17694911996@163.com).

Xudong Zhang is with the National Engineering Laboratory for Electric Vehicles, School of Mechanical Engineering, and the Collaborative Innovation Center of Electric Vehicles in Beijing, Beijing Institute of Technology, Beijing 100081, China (e-mail: xudong.zhang@bit.edu.cn).

Yu Zhang is with the Army Academy of Armored Forces, Beijing 100072, China (e-mail: zhangyuzgy2018@163.com).

Yunfei Chang is with Beijing North Vehicle Group Corporation, Beijing 100071, China (e-mail: chfei028@163.com).

Kai Zhao is with the Institute of Military Transportation, Army Military Transportation University, Tianjin 300181, China (e-mail: kzhao94@foxmail.com).

Digital Object Identifier 10.1109/JSEN.2023.3324429

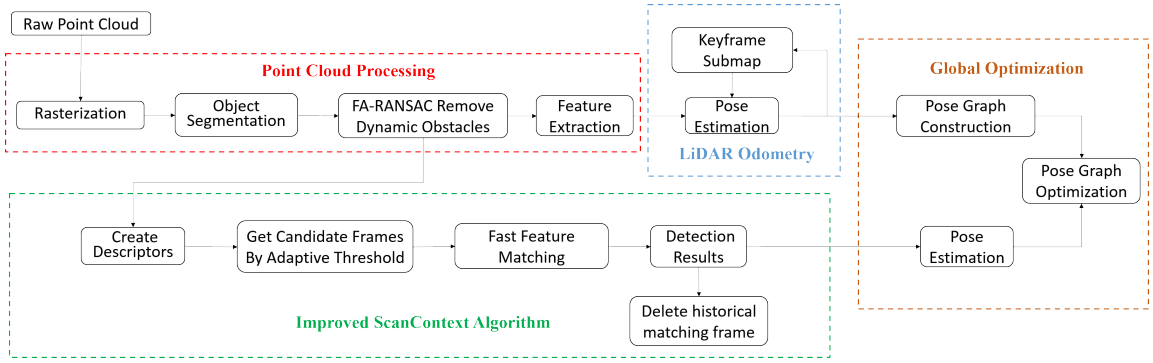


Fig. 1. Full pipeline block diagram of the proposed method.

in Fig. 1. First, the point clouds are divided into fan-shaped grids and the region growth is used to achieve object segmentation. Second, the FA-RANSAC algorithm is proposed to quickly remove dynamic obstacles based on feature information and adaptive threshold and obtain more accurate initial pose. Then, the static feature points are extracted for pose estimation. Afterward, height difference and adaptive distance are used to improve the loop closure detection accuracy of ScanContext, and simplifying feature matching and deleting loop historical matching frames are used to improve the loop closure detection efficiency of ScanContext. Finally, localization and mapping can be realized quickly and accurately by global optimization.

This article is organized as follows. The current state of SLAM approaches related to LiDAR is shown in Section II. Section III introduces the proposed method, including point cloud processing, LiDAR odometry, loop closure detection, and global optimization. Section IV shows the experimental results and analysis with details, followed by the conclusion in Section V.

II. Related Work

The important part of LiDAR SLAM is the matching between the consecutive frames. According to the existing works, it can be mainly divided into raw point cloud matching and feature point pairs matching [16].

For raw point cloud matching, the iterative closest points (ICP) [17] and its derivative methods are the most typical methods [18], [19], [20]. Their core idea is mainly to find the nearest point and iteratively minimize the distance residual to obtain the final estimated pose. Although these methods can achieve good localization in static scenes with high initial accuracy, it takes a long time. In addition, dynamic points can easily interfere with static point pair registration, leading to results falling into local optimization in a dynamic environment. Another common method for raw point cloud matching is the normal distribution transformation (NDT) [21] and its derivative methods [22], [23]. Their core idea is mainly to calculate the local normal distribution in each grid and iteratively minimize the probability residual to obtain the final estimated pose. Although these methods have been widely used in point cloud registration, its accuracy and time consumption are affected by the grid size.

For the feature point pairs matching, LOAM [15] and its derivative methods [24], [25], [26] are the most typical methods. Their core idea is mainly to gain feature points and iteratively minimize feature distance to achieve state estimation. Although feature points are used for iterative calculation to improve the efficiency and avoid the feature loss caused by grid, dynamic feature points can affect the construction of error functions when in dynamic scenes. The more dynamic objects, the worse the performance of these methods. Therefore, they are difficult to achieve accurate state estimation in dynamic scenes.

In recent years, there has also been some research on deep learning matching. Velas et al. [27] projected LiDAR points to the 2-D plane and adopted three channels to encode information. Then, the convolution and full connection layers are used for the pose regression. It shows better performance when only estimating the translation, but its performance is poor when estimating the 6-DOF pose. Li and Wang [28] proposed a network DMLO to generate feature point pairs with high confidence. Then, SVD is used to obtain the estimated pose. Although it can get the 6-DOF pose, dynamic objects can easily interfere with feature point pairs and affect localization accuracy in urban scenes. Wang et al. [29] proposed a network PWCLO-Net to estimate and refine the pose based on pyramid, warping, and cost volume structure. Although it can show superior performance on KITTI datasets, it is not robust in some dynamic scenes and its generalization ability is poor.

Most of the above LiDAR SLAM technologies are based on static scenes. However, we are faced with more dynamic scenes in our real life [30]. Therefore, some scholars have also carried out some research on how to remove dynamic objects and improve the robustness of localization in dynamic scenes. Most of them adopt detection and tracking methods to remove dynamic objects [31], [32]. Although these methods can realize localization in dynamic scenes, their real-time performance is difficult to meet the requirements. Then, some scholars try to study some dynamic object removal methods that do not depend on the above technologies. Kim and Kim [33] proposed the Removert method to retain static points from falsely removed points. However, it cannot filter out the dynamic points correctly when occlusion occurs. Lim et al. [34] proposed a static map building method ERASOR based on region-wise ground plane fitting. It relies on the ground

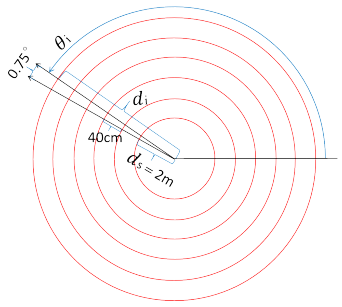


Fig. 2. Fan-shaped grid model.

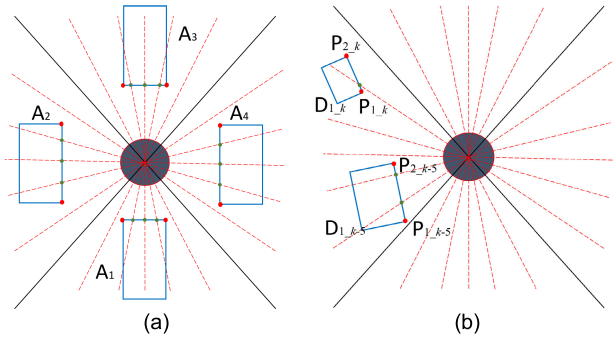


Fig. 3. Schematic of dynamic selection. (a) Feature point selection in sectors. (b) Robustness description when yaw angle changes.

fitting to remove dynamic points above the ground. Therefore, it shows poor performance in some slope scenarios. Due to the fact that works [33] and [34] are implemented offline that cannot be run online, Yang and Wang [35] proposed a fast LiDAR SLAM algorithm based on RANSAC. Although it can achieve an online registration effect, it shows poor performance when there are many dynamic objects. Wang et al. [36] proposed a multilayer RANSAC (ML-RANSAC) algorithm. Based on the segmentation of point cloud, it uses iterative associated object matching to achieve registration and removes dynamic objects at the same time. This method can be used in more complex scenes and large initial errors, but it uses the poor robustness of single matching point and has a large number of iterations.

The above methods have some issues with computational cost or localization accuracy in an urban dynamic environment. Therefore, we propose the RF-LOAM method that is efficient and fits well to urban dynamic scenes by combining the FA-RANSAC algorithm and improved ScanContext algorithm.

III. Methodology

The entire framework of the proposed method in this article consists of four modules: point cloud processing, LiDAR odometry, loop closure detection, and global optimization. In this section, the four modules will be elaborated in detail.

A. Point Cloud Processing

1) Point Cloud Rasterization: In this article, HDL-64 LiDAR is used for SLAM research. Due to the rotation characteristics of the LiDAR, the density of point clouds will decrease with increasing distance. In order to better realize object segmentation and facilitate the subsequent loop closure feature extraction, we adopt the fan-shaped grid, which is more consistent with the distribution of point cloud to cluster objects. The index is calculated according to (1) and (2) within the range of [2, 40] m around the LiDAR, and the fan-shaped grid model is constructed, as shown in Fig. 2

$$r_i = \frac{d_i - d_s}{T_r} \tag{1}$$

$$s_i = \frac{\theta_i}{T_s} \tag{2}$$

where d_i and θ_i denote the linear distance and angle from the point i to the origin of the LiDAR coordinate, respectively, and T_r and T_s denote the ray resolution and circumferential resolution ($T_r = 0.4$ and $T_s = 0.75°$), respectively.

2) Object Segmentation: After fan-shaped rasterization, in order to cluster objects quickly and accurately, we adopt a region growth algorithm to achieve object segmentation. The specific process is given as follows.

First, the height difference Δh of each fan-shaped grid is calculated, and the seed grids are selected by height difference threshold $T_{\Delta h}$ ($T_{\Delta h} = 0.3$ m) and point cloud number threshold T_{np} ($T_{\text{np}} = 5$).

Then, the untraversed seed grids of eight neighborhoods around the current seed grid are judged, and the neighborhood seed grids with the maximum height close to the maximum height of the current seed grid are added to the same object list. Next, the untraversed grid in the list is repeated 8 neighborhood searches until there is no untraversed grid in the object list, and the single object segmentation is ended.

Finally, after traversing all untraversed grids in turn, all object segmentation can be completed.

3) FA-RANSAC Remove Dynamic Objects: Most of the existing methods judge dynamic objects by detection and tracking, which often takes a long time. However, we do not need to obtain the accurate object speed in SLAM and just need to know whether it is moving. Therefore, this article skips the detection and tracking process and proposes an FA-RANSAC algorithm to quickly remove dynamic objects based on feature information and adaptive threshold.

a) Dynamically select feature points: After the object segmentation is completed, we obtain a rectangular box of object through convex polygon according to the method in [37]. Considering that the distance from object to the LiDAR changes continuously in a realistic scene, the rectangular box of object will also change, resulting in the fluctuation of the barycentre. At this time, the ML-RANSAC algorithm [36] with single point matching has poor accuracy. Therefore, this article uses the scanning characteristics of the LiDAR to divide the area around the LiDAR into four sectors and proposes a more robust feature point selection method, as shown in Fig. 3(a). When the object is located in the front or rear sector of the LiDAR, it will retain a relatively complete width feature, as shown in the green points in Fig. 3(a). When the object is located in the left or right sectors of the LiDAR, it will retain a relatively complete length feature. Therefore, by judging the area where the barycentre of the object is located, the stable red border points are selected as feature points to match with

that of the historical frame, so as to improve the robustness and accuracy of dynamic object removal. Even if the object D_1 changes in yaw angle from frame $k-5$ to frame k and its relatively stable length feature also changes as shown in Fig. 3(b), interior point calculation in Section III-A3c can still select relatively stable feature point pair P_{1_k-5} and P_{1_k} from feature points for judging interior points. Therefore, changes in yaw angle do not affect the dynamic object removal.

b) Query corresponding points based on multiple features: The transform between two frames usually involves six parameters. Therefore, at least two sets of corresponding points are required to obtain the corresponding transform matrix. In order to get the transform matrix through a small number of iterations for removing dynamic objects, we compare current frame m with the frame $m-5$ and propose a fast and accurate corresponding points search method by multiple features.

First, two objects O_1 and O_2 with proper distance [38] are randomly selected from current frame m, and their barycentres are $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. Then, the prediction pose of current frame m is calculated by the pose of the last frame $m-1$ and transformed between two consecutive frames $m-1$ and $m-2$.

Second, considering that the integrity of point cloud features in length and width directions is different, we adaptively adjust the weights of length and width according to the barycentre of object in the current frame m and calculate the shape deviation between the object in the current frame m and the object in frame $m-5$, as shown in (3) and (4). Then, the weighted distance D_f between the object in current frame m and the object in frame $m-5$ is calculated by combining the barycentre deviation D_g, the shape deviation D_b, and the number deviation of point cloud D_n, as shown in the following:

$$a_l = \frac{p_{n\_x}}{p_{n\_x}+p_{n\_y}}, \quad a_w = \frac{p_{n\_y}}{p_{n\_x}+p_{n\_y}} \tag{3}$$

$$D_b = a_l \times \frac{|l_n - l_t|}{\max(l_n, l_t)} + a_w \times \frac{|w_n - w_t|}{\max(w_n, w_t)} \tag{4}$$

$$D_f = D_g \times W_g + D_b \times W_b + D_n \times W_n \tag{5}$$

where p_{n_x} and p_{n_y} denote the barycenter value of the object n in the current frame; a_l and a_w denote the weights of length and width, respectively; l_n and w_n denote the length and width of rectangular box of the object n in the current frame m, respectively; l_t and w_t denote the length and width of rectangular box of the object t in frame $m-5$, respectively; and W_g, W_b, and W_n denote the weights corresponding to the variations D_g, D_b, and D_n, respectively.

Next, the $N_{\min}$ ($N_{\min}=5$) objects with the minimum weighted distance to O_1 are extracted from frame m-5, and their barycentres are used as candidate point set $\mathbf{M}_1$ of O_1. In the same way, the candidate point set $\mathbf{M}_2$ of O_2 is obtained.

Finally, $\boldsymbol{q}_{1\_1}$ is selected from $\mathbf{M}_1$, and the points that meet (6) are selected from $\mathbf{M}_2$, as shown in Fig. 4. Then, the corresponding relationship pair set $\mathbf{R}_1$ between $\boldsymbol{q}_{1\_1}$ and $\boldsymbol{q}_{2\_i}$ can be obtained. Next, all points in $\mathbf{M}_1$ are traversed in turn, and the corresponding relationship pair set $\mathbf{R}_i$ is accumulated

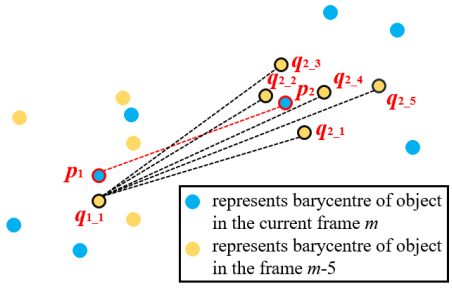


Fig. 4. Schematic of corresponding point query.

to obtain all relationship pairs set $\mathbf{R}_{1\_2}$ between $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, as shown in the following equation:

$$\mathbf{R}_1 = \left\{ (\mathbf{q}_{1\_1}, \mathbf{q}_{2\_i}) \,\middle|\, \left| |\mathbf{p}_1 - \mathbf{p}_2| - |\mathbf{q}_{1\_1} - \mathbf{q}_{2\_i}| \right| < \varepsilon, \mathbf{q}_{2\_i} \in \mathbf{M}_2 \right\} \tag{6}$$

$$\mathbf{R}_{1\_2} = \sum_{i=1}^{N_{\min}} \mathbf{R}_i. \tag{7}$$

c) Adaptive filter interior points and set termination condition: Due to the fast scanning frequency of LiDAR, the similarity of point clouds between adjacent frames is relatively high. Therefore, the average distance and the optimal interior ratio of the previous frame can be used to guide current frame matching. Among them, when judging the inner and outer points, the current distance needs to be calculated in two situations. If the object in previous frame is judged to be moving by the FA-RANSAC algorithm at previous time, the current distance refers to the maximum distance between two pairs of feature points after transform of each iteration. Otherwise, the current distance refers to the minimum distance between two pairs of feature points. Also, the interior rate refers to the ratio of object pairs with distance less than the distance threshold to all object pairs. The specific process is given as follows.

First, the distance d_i of the optimal matching interior pairs of previous frame is introduced. After that, the average distance d_L of previous frame is adaptively calculated by (8) as the distance threshold of current frame to filter the interior points of current frame

$$d_L = \frac{\lambda}{N} \sum_{i=1}^{N} d_i \tag{8}$$

where N denotes the number of the interior points in the previous frame and λ denotes the interior point factor.

Then, in order to avoid a large number of sampling times and reduce dispensable time consumption, we adaptively set the prediction interior ratio r_c (termination condition) of current frame through the optimal interior ratio r_l of previous frame, so as to obtain the optimal transform matrix that meets r_c after a small number of iterations

$$r_c = \mu r_l \tag{9}$$

where μ denotes the interior ratio factor.

d) FA-RANSAC removes dynamic objects: Based on the above contents, the FA-RANSAC algorithm is proposed.

The algorithm framework is shown in Algorithm 1, and the specific process is given as follows.

Algorithm 1 FA-RANSAC

1. intput: objects A for current frame m; objects B for frame m-5; empty feature list F_A, F_B; random sampling times S; the optimal interior ratio r_b, the average distance d_b for last frame
2. output: Mov, R_{best}, T_{best}, r_{best}
3. F_A = Get_ feature_points(A), F_B = Get_feature_points(B);
4. C_A = Get_barycenter(A), C_B = Get_barycenter(B);
5. $d = \lambda \times d_b$, $r = \mu \times r_b$;
6. for i = 0 to S do
7. (p1, p2) = Random_sample(C_A);
8. M1 = Weighted_Search(p1, C_B, R_{best}, T_{best}), M2 = Weighted_Search(p2, C_B, R_{best}, T_{best});
9. R_{1_2} = Query_based_on_features(M1, M2);
10. R_t, T_t, r_t, M_t = RANSAC (R_{1_2}, F_A, F_B, d);
11. if ($r < r_t$) then
12. $r_{best} = r_t$, $R_{best} = R_t$, $T_{best} = T_t$, Mov = M_t, break;
13. else if ($r_{best} < r_t$) then
14. $r_{best} = r_t$, $R_{best} = R_t$, $T_{best} = T_t$, Mov = M_t;
15. end if
16. end for

First, the dynamic selection of feature points in Section III-A3a is used to obtain multiple stable feature points of objects in current frames m and $m-5$.

Second, two objects with proper distance are randomly selected from the current frame. Then, all relationship pairs set between the selected two objects in the current frame and the objects in frame $m-5$ is obtained according to the corresponding point query method in Section III-A3b.

Next, the transform matrix is calculated by the feature points matching of selected objects under each corresponding relationship. Then, the current distance is calculated based on the two pairs of feature points in the transformed object, and the adaptive distance threshold obtained in Section III-A3c is used to filter the interior points and get the interior rate corresponding to each transform matrix. Considering that the number of static objects in a frame is usually greater than that of dynamic objects, the transform matrix with the highest interior rate is finally selected as the RANSAC registration result of this sampling.

Finally, the sampling and registration steps are repeated until the interior rate is higher than the adaptive interior rate threshold obtained in Section III-A3c or the number of sampling reaches the fixed value. At this time, the transform matrix with the highest interior rate is retained, and the corresponding exterior points will be directly removed as dynamic objects.

4) Feature Extraction: After removing dynamic objects, there are still a large number of point clouds per frame. If these point clouds are directly processed, it will take a long time. Therefore, in order to improve the matching efficiency, this article extracts the planar features set S_L and edge features set E_L based on local point smoothness according to the method in [16]. Then, the optimal transform matrix obtained by the FA-RANSAC algorithm is used as the initial transform matrix, and S_L and E_L are transformed from the current LiDAR coordinate to the global coordinate for preliminary distortion compensation according to the method in [16].

B. LiDAR Odometry

1) Pose Estimation: The core idea of pose estimation is to match the current undistorted planar features set S_W and edge features set E_W with the global keyframe feature submap in Section III-B2. For each planar feature point p_S of S_W and edge feature point p_E of E_W, the planar feature points closest to p_S are first selected from global planar feature submap by KD-trees. Similarly, the edge feature points closest to p_E are also selected. Then, the line and plane are estimated by collecting nearby points from the edge and planar feature submap. Next, the weighted distance [16] of the point-to-edge and point-to-planar is calculated. After that, the Gauss–Newton method is used to minimize the weighted distance sum of the current undistorted features, and the current estimated pose is obtained by iterative optimization until it converges. For details of the above steps, please refer to [16].

2) Keyframe Submap Update: First, translation and rotation are calculated between the current frame pose obtained by pose estimation and the previous keyframe pose. If translation and rotation are less than their corresponding thresholds, the current frame is not considered as a keyframe and is skipped directly. Otherwise, the current frame is considered as a keyframe. The distortion compensation is performed on current planar features and edge features according to the method in [16]. Then, the recomputed undistorted edge features and planar features will be updated to global keyframe edge submap and planar submap, respectively, and simultaneously passed into the global optimization to build the pose graph.

C. Loop Closure Detection

The ScanContext algorithm [39] can achieve loop closure detection, but there are many problems, such as difficulty in distinguishing between bumpy roads, false loop closure detection of far-away similar scenes, and long time-consuming of candidate frame search and matching. Based on the above problems, we improve the ScanContext algorithm to achieve faster and more accurate loop closure detection after removing dynamic objects. Specifically, the robustness of scene recognition is improved by using the height difference. Then, an adaptive distance threshold is used to filter invalid similar candidate frames. Finally, the efficiency of candidate frame matching and search is improved by optimizing distance calculation and deleting loop historical matching frames.

1) Create Descriptors: Considering the deceleration zones and slopes on urban roads, and the small inclination of installation, it is difficult to judge the same bumpy roads only by the maximum height. Therefore, after dynamic object removal, the height difference Δh in each grid is selected as the value in the feature matrix of the ScanContext algorithm.

In order to avoid repeated division of the fan-shaped grid as much as possible and quickly obtain the descriptors of

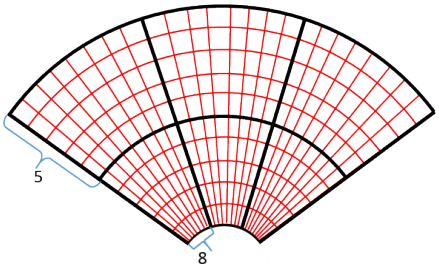


Fig. 5. Schematic of fan-shaped grid merge.

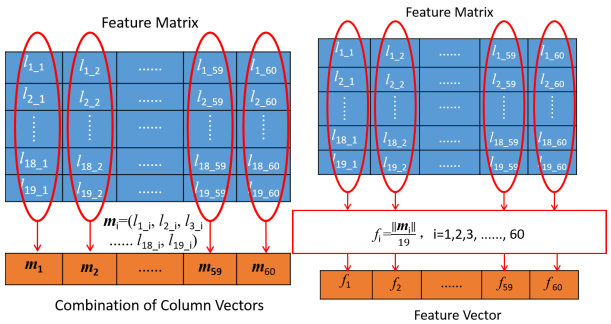


Fig. 6. Simplified model of feature matrix. (a) ScanContext algorithm. (b) Proposed method.

the current keyframe, we combine 480 × 95 grids used for segmentation in Section III-A2 and the grids marked as dynamic objects in Section III-A3 to quickly build a new $N_s \times N_r$ feature extraction grid model. Since N_s is 60 and N_r is 19 in this article, it is only needed to enclose five original red fan-shaped grids in the ray direction and eight original red fan-shaped grids in the circumferential direction into a new black fan-shaped grid, as shown in Fig. 5. Then, the height difference of the original nondynamic object grids in the new black fan-shaped grid is compared in turn, as shown in (10). Next, the maximum height difference of the new fan-shaped grid is used as its feature value. Finally, the feature matrix of the current keyframe is obtained as descriptors by iterating all the new fan-shaped grids

$$\Delta h = \max_{k\in\{1,2,\ldots,40\}\ \&\ k\notin \boldsymbol{Mov}} \left(\Delta h_{m\_k}\right) \tag{10}$$

where Δh_{m_k} denotes the height difference of original nondynamic object grid k in the new fan-shaped grid and $\boldsymbol{Mov}$ denotes the grids set marked as dynamic objects in Section III-A3.

2) Get Candidate Frames: After the feature matrix of current frame is obtained, the traditional ScanContext algorithm adopts a fixed distance threshold to determine the candidate frames, which is easy to cause false loop closure detections of far-away similar scenes. Therefore, we introduce an adaptive distance threshold to accurately select the candidate frames. Specifically, we first calculate the norm of each row vector of the feature matrix to obtain a rotation-insensitive description vector according to the method in [39]. Then, the vector deviation defined in [39] is used to find out N_m ($N_m = 10$) initial candidate frames with the smallest deviation from the historical keyframes more than ten frames apart from the current frame. Next, since each loop closure optimization will greatly reduce the cumulative error, we design a distance threshold d_T that changes synchronously with the cumulative error to further filter out the false loop closure detections, as shown in (11) and (12). Finally, the candidate frames of the current frame are accurately obtained.

$$d_T = 15 + \frac{k_{\text{cur}} - k_{\text{last\_loop}}}{40} \tag{11}$$

$$d = \sqrt{T_{\text{cur}}^{\text{can}} \cdot x^2 + T_{\text{cur}}^{\text{can}} \cdot y^2 + T_{\text{cur}}^{\text{can}} \cdot z^2} \tag{12}$$

where k_{cur} and $k_{\text{last\_loop}}$ denote the number of current frame and last loop frame in all keyframes, respectively; $T_{\text{cur}}^{\text{can}}$ denotes the transform matrix between the current frame and the candidate frame; and d denotes the linear distance between the current frame and the candidate frame.

3) Fast Loop Closure Detection of Candidate Frames: After obtaining the candidate frames, in order to finally confirm whether the vehicle enters loop closure, the traditional ScanContext algorithm [39] regards the feature matrix as a combination of column vectors as shown in Fig. 6(a), and the feature matrix is translated to judge the similarity. This method needs to calculate the cosine value of column vectors 60 times per translation, which takes a long time. Therefore, in order to reduce the complexity of similarity calculation of the feature matrix, we improve the method of translation column vectors in [39] and use the column vector norm to simplify the feature matrix for the translation similarity judgment. The details are given as follows.

First, the column vector norms of current frame and candidate frames are calculated, and their feature matrix is simplified into a single feature vector, as shown in Fig. 6(b). Then, the selected candidate frame feature vector $\boldsymbol{k}$ is continuously translated and its distance from the current frame feature vector is calculated according to the method in [39]. Among them, the vector cosine value is only calculated once per translation and the distance of the feature matrix can be obtained quickly and accurately. Next, the minimum distance value is selected from all translations as the distance between $\boldsymbol{k}$ and the current frame feature vector. Finally, according to this translation method, all candidate frames are calculated in turn. The minimum distance value is selected from them to compare with the distance threshold, and the loop closure detection can be completed.

4) Optimize Candidate Frames Search: To avoid the problem that the accumulation of historical data leads to an increase of time spent on searching candidate frames, this article further improves the method in [39]. After the current frame enters loop closure, the history loop closure frame is cleared continuously. Our method can ensure that the number of historical frames remains stable with the continuous increase of point cloud and the search speed of candidate frames is improved.

D. Global Optimization

After the loop closure detection of current keyframe, we use the global optimization algorithm commonly used in SLAM

to further eliminate the cumulative error, specifically including pose graph construction and pose graph optimization.

1) Pose Graph Construction: First, the module receives the pose of current keyframe in the global coordinate from the LiDAR odometry. Then, the pose can be added as a node in pose graph to complete the pose graph construction, and the feature set of current keyframe can be viewed as a measurement of this node.

2) Pose Graph Optimization: If the current keyframe does not enter loop closure after the construction of the pose graph, the following steps will be skipped directly. Otherwise, the feature submap of loop closure frame is constructed from feature set of its neighbors in pose graph. Then, the transform between the current frame and the loop closure frame is calculated according to the method in Section III-B1. Next, the transform is added as the edge corresponding to current pose node and loop closure pose node to pose graph. At this time, due to the deviation between the current pose obtained by loop closure and the current pose obtained by odometry, the L-M algorithm in [24] is used to optimize the global pose of pose graph. Finally, the optimized poses update the poses of submap in odometry and the poses of loop closure historical frames.

IV. Experimental Results and Analysis

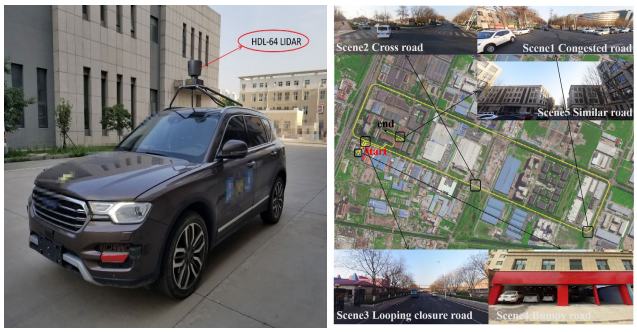
A. Experimental Setup

1) Datasets: To validate the proposed method, we evaluate RF-LOAM on the KITTI datasets [40], Urbanloco datasets [41], and urban dynamic dataset collected by ourselves.

KITTI datasets are recorded by an HDL-64 LiDAR, a set of RTK-GPS/IMU device, and a binocular camera. Its trajectories are long enough to test outdoor localization methods. Urbanloco datasets are recorded by an HDL-32E LiDAR, a set of GPS/IMU device, and six 360° view cameras. It contains a large number of dynamic objects. In order to test the localization and loop closure detection effect of our method in urban scenes, we select three sequence urban data with the highest loop closure repetition rate: 00, 05, and 08 (08 only contains the reverse loops and 00 and 05 only contain loop events with the same direction) from KITTI datasets and two sequence urban data: CA_20190828155828 (CA-1) and CA_20190828190411 (CA-2) with more dynamic objects from Urbanloco datasets for experiments.

Extracted urban dataset is recorded by unmanned ground vehicle (UGV). HDL-64 LiDAR is installed on top of the UGV, as shown in Fig. 7(a). This dataset obtained in dynamic urban environment contains a large number of pedestrians and vehicles about 5.2 km, as shown in Fig. 7(b). From the local typical pictures of Fig. 7(b), it can be seen that the dataset includes a traffic congested road, a cross road, a loop closure road, an urgent bumpy road, and a similar road for better validation algorithms.

2) Implementation Details: All tests are performed in C++ based on the robot operating system (ROS) and run on a computer with i7-8700 CPU, 16-GB RAM, and Ubuntu 18.04. Then, RF-LOAM uses the detailed constant parameters shown in Table I, according to [42] and practical experience.



(a) (b)

Fig. 7. Extracted urban dataset. (a) Collection platform. (b) Satellite map.

TABLE I
Detailed Constant Parameters Used in RF-LOAM

Parameter	W_g	W_b	W_n	λ	μ
Value	0.5	0.3	0.2	1.3	1.2

B. Performance Evaluation

1) KITTI Datasets: The point cloud maps built by our method under the specified KITTI datasets are shown in Fig. 8(a-1)–(c-1). We can see that our method can accurately build point cloud map. Then, the typical local dynamic scenes are selected from KITTI datasets for analysis, as shown in Fig. 8(a-3)–(c-3). From Fig. 8(a-2) to (c-2), it can be seen that our method can successfully remove these dynamic objects and achieve accurate dynamic scene construction.

In order to further illustrate the advantages of our method, RF-LOAM is compared with the state-of-the-art SLAM methods, such as optimized-SC-F-LOAM, LeGO-LOAM, F-LOAM, A-LOAM, PWCLO-Net, and ML-RANSAC. In addition, to show the benefits of dynamic object removal before scan matching, we extract point cloud processing and odometry of our method as Our-Front for further comparison in Section IV-C. Then, average translation error (ATE) and average rotation error (ARE) are used as the evaluation metrics, which are defined by KITTI datasets [40]. The results are shown in Table II and the comparison of the typical trajectories is shown in Fig. 8(d)–(f). It can be seen that our method outperforms the compared methods, and the trajectory of our method is closest to ground truth with the smallest error under these KITTI datasets. Due to the lack of loop closure detection in A-LOAM and F-LOAM and the interference of dynamic objects, the average error is relatively large, as shown in Table II. In this article, we adopt the FA-RANSAC algorithm to quickly remove dynamic objects. Thus, the localization accuracy of Our-Front is higher than that of A-LOAM and F-LOAM. Also, due to the fact that it uses multiple feature point matching, Our-Front is also better than ML-RANSAC with single point matching. However, PWCLO-Net cannot rely on geometry correspondence and achieve better LiDAR odometry performance through large-scale training than Our-Front. LeGO-LOAM and optimized-SC-F-LOAM have loop closure detection, so their cumulative errors are also smaller than those of Our-Front. However, the improved scan context

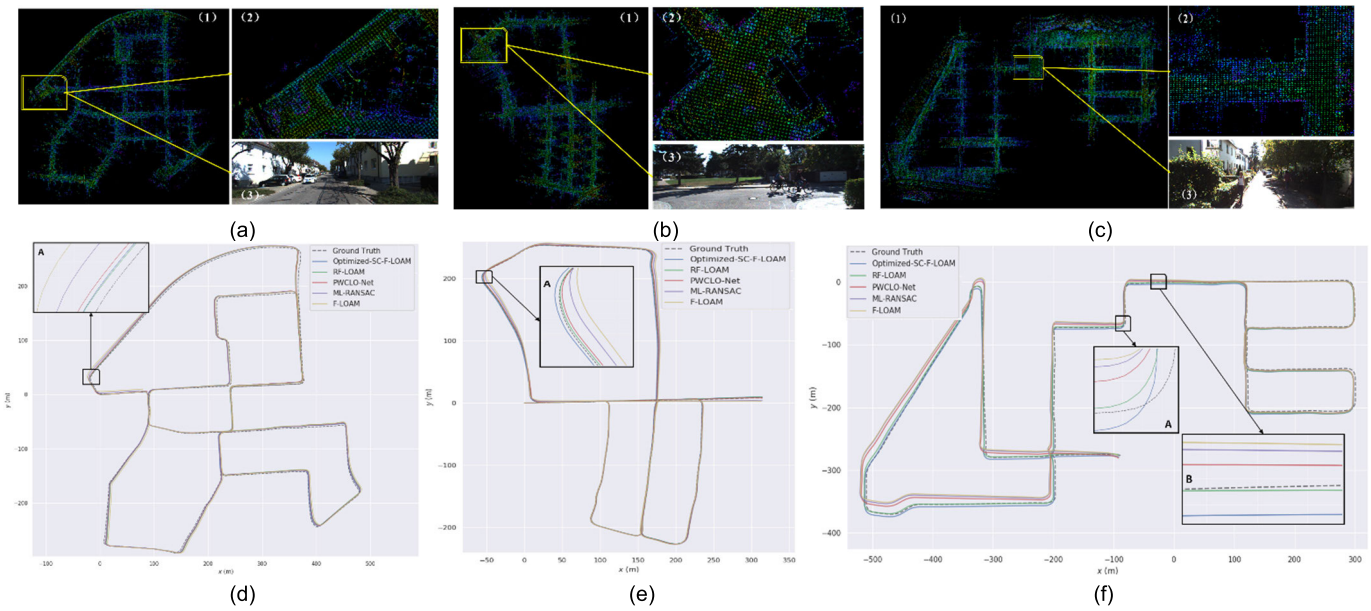


Fig. 8. Sample results on KITTI datasets. The left sides of (a)–(c) show the mapping results of our method on KITTI datasets sequence 00, 05, and 08, and the right sides of (a)–(c) show the image and submap of local dynamic scene. (d)–(f) Comparison of trajectories on the specified KITTI datasets.

TABLE II
RESULTS ON THE SPECIFIED KITTI DATASETS IN DIFFERENT ALGORITHMS

	Datasets	A-LOAM	F-LOAM	LeGO-LOAM (with ICP)	Optimized-SC-F-LOAM	PWCLO-Net	ML-RANSAC	Our-Front	Ours
ATE (%)	00	1.01	0.92	0.80	0.76	0.85	0.89	0.84	**0.72**
	05	0.68	0.63	0.51	0.44	0.46	0.58	0.54	**0.39**
	08	1.56	1.54	1.23	1.12	1.36	1.45	1.38	**0.97**
ARE (deg/100m)	00	0.45	0.42	0.38	0.29	0.40	0.44	0.39	**0.25**
	05	0.41	0.39	0.33	0.26	0.29	0.38	0.35	**0.18**
	08	0.58	0.57	0.52	0.44	0.52	0.56	0.54	**0.36**

is used for fast and accurate loop closure detection in dynamic scenes. Therefore, after global optimization, our method can be more accurate than other compared methods. It can achieve 0.69% ATE and 0.27°/100 m ARE over three sequences.

2) Urbanloco Datasets: To better evaluate the SLAM performance of our method in an urban dynamic environment, we implement RF-LOAM and the state-of-the-art SLAM methods on Urbanloco datasets. It can be seen that the dynamic objects in Urbanloco datasets are significantly more than those in KITTI datasets from the right of Fig. 9(a) and (b). However, our method can still remove these dynamic objects and build accurate dynamic point cloud maps as shown in the left and middle of Fig. 9(a) and (b).

Then, mean translation error (MTE) and mean rotation error (MRE) are used as the evaluation metrics, which are defined by Urbanloco datasets [41]. The results and comparison of typical trajectories are shown in Table III and Fig. 9(c) and (d), respectively. It can be seen that due to the more complex dynamic scenes of Urbanloco datasets than those of KITTI datasets, the errors have significantly increased. However, our method can still achieve the best SLAM performance. The results are consistent with the results on KITTI datasets, but our method has more obvious advantages. Compared with LeGO-LOAM (with ICP) and optimized-SC-F-LOAM, our method improves trajectory accuracy by 31.2% and 24.8%, respectively. After dynamic object removal, the advantage of Our-Front is also more obvious compared to A-LOAM, F-LOAM, and ML-RANSAC in complex dynamic scene, as shown in Table III. In addition, the density of point clouds collected by HDL-32E LiDAR is relatively sparse and the scenes are more complex in the Urbanloco datasets. Therefore, after using these point clouds to train PWCLO-Net, the trajectory error is relatively large and its accuracy is not as good as that of Our-Front.

3) Extracted Urban Dataset: To further demonstrate the superiority of our method in practical dynamic environment, we select the Extracted urban dataset for testing, which is collected in urbanized scenes with many dynamic objects. Some typical pictures of the urban dataset are shown in Fig. 7(b), and the mapping results of our method are shown in Fig. 10. We can see that our method can accurately build point cloud map.

Then, the ATE and ARE are used as the evaluation metrics. The results of the compared methods are shown in Table IV, and the comparison of typical trajectories is shown in Fig. 11. Due to the fact that the urban environment is more complex than KITTI, our method has more obvious advantages and the results are similar to the results

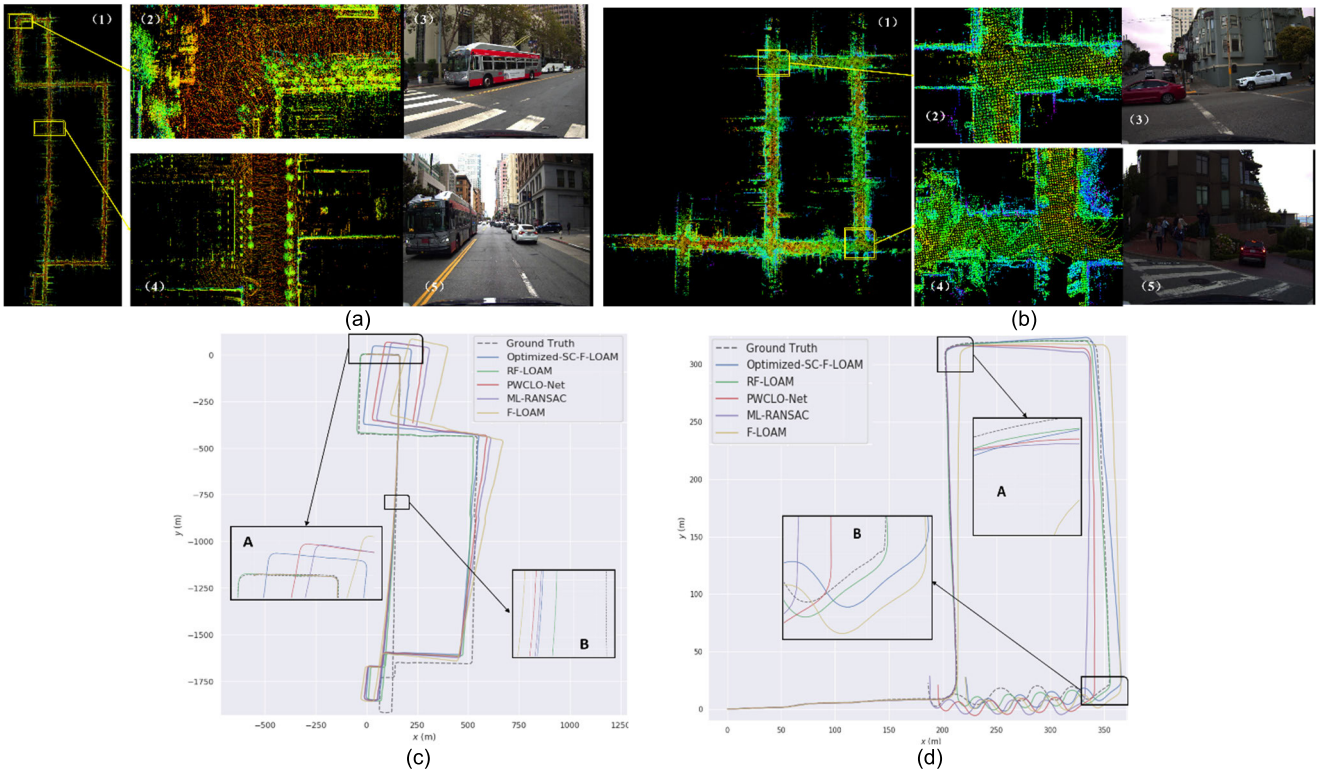


Fig. 9. Sample results on Urbanloco datasets. The left sides of (a) and (b) show the mapping results of our method on Urbanloco datasets sequence CA-1 and CA-2, the middle of (a) and (b) shows the submap of local dynamic scene, and the right sides of (a) and (b) show the image of local dynamic scene. (c) and (d) Comparison of trajectories on the specified Urbanloco datasets.

TABLE III
RESULTS ON THE SPECIFIED URBANLOCO DATASETS IN DIFFERENT ALGORITHMS

	Datasets	A-LOAM	F-LOAM	LeGO-LOAM (with ICP)	Optimized-SC-F-LOAM	PWCLO-Net	ML-RANSAC	Our-Front	Ours
MTE (m)	CA-1	93.98	85.44	41.80	32.76	56.83	61.89	52.84	**22.72**
	CA-2	22.3	20.63	14.51	13.94	16.24	17.17	15.57	**10.32**
MRE (deg)	CA-1	3.87	3.62	2.98	2.69	3.47	3.44	3.17	**1.95**
	CA-2	5.57	5.49	3.52	3.44	4.81	4.96	4.54	**2.86**

TABLE IV
RESULTS ON THE EXTRACTED URBAN DATASET IN DIFFERENT ALGORITHMS

	A-LOAM	F-LOAM	LeGO-LOAM (with ICP)	Optimized-SC-F-LOAM	PWCLO-Net	ML-RANSAC	Our-Front	Ours
ATE（%）	15.26	12.84	7.72	2.84	8.87	10.89	8.24	**1.38**
ARE（deg/100m	2.38	1.94	1.47	0.94	1.61	1.86	1.54	**0.62**

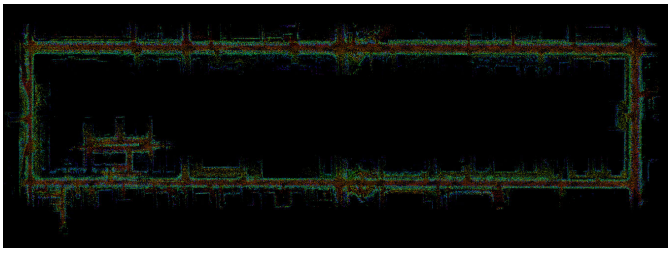


Fig. 10. Example of our method on extracted urban dataset.

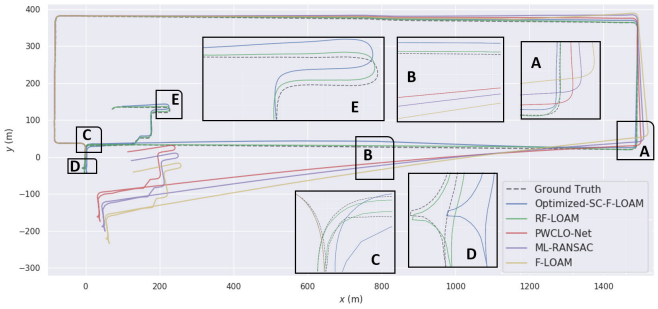


Fig. 11. Comparison of trajectories on extracted urban dataset.

on Urbanloco datasets. Our method adopts the FA-RANSAC algorithm to remove dynamic objects. Therefore, Our-Front has better accuracy than A-LOAM and F-LOAM, as shown in Table IV. Considering that PWCLO-Net trained on KITTI datasets has insufficient generalization ability in our Extracted urban dataset and the robustness of ML-RANSAC using single point matching in complex scene is limited, Our-Front

TABLE V
REMOVAL RATE OF DYNAMIC OBJECT POINTS

Algorithms	Datasets	Removal Rate (%)
Removert	CA-1	78.6
	CA-2	85.3
	Extracted	80.5
ERASOR	CA-1	**83.4**
	CA-2	92.3
	Extracted	**87.9**
ML-RANSAC	CA-1	77.2
	CA-2	88.7
	Extracted	81.3
Our-Front	CA-1	81.8
	CA-2	**93.3**
	Extracted	85.1

TABLE VI
RELATIVE LOOP RATE ON THE ALL DATASETS

Datasets	IMSC	RM-IMSC	Relative Loop Rate（%）
KITTI	847	862	98.3
Urbanloco	284	292	97.2
Extracted	628	642	97.8

TABLE VII
ABLATION STUDY OF LOOP CLOSURE ON THE EXTRACTED DATASET

Algorithms	ATE（%）	ARE（deg/100m）
OF-ICP	6.63	1.37
OF-OPSC	2.64	0.81
Ours	**1.38**	**0.62**

performs also better than PWCLO-Net and ML-RANSAC. LeGO-LOAM and optimized-SC-F-LOAM have loop closure detection that helps reduce cumulative errors, so their trajectory errors are smaller than those of Our-Front. However, they are disturbed by dynamic objects and bumpy roads, resulting in unstable loop closure detection and larger fluctuation of trajectory error. We improve the feature extraction and adaptive distance threshold of ScanContext algorithm to accurately detect loop closures in bumps and similar roads. Although there are still errors between our trajectory and ground truth, the trajectory obtained by our method is closest to ground truth, and our method has the best SLAM performance compared with the other methods. Compared with the optimized-SC-F-LOAM, our method reduces ATE and ARE by 48% and 33%, respectively, as shown in Table IV.

C. Ablation Study

In order to analyze the effectiveness of each module, we remove or change components of our method to do ablation studies.

1) Impact of Dynamic Object Removal: In this study, the removal rate proposed in [43] is first used to evaluate the effectiveness of dynamic object removal. We compare Our-Front with the state-of-the-art methods, namely, Removert, ERASOR, and ML-RANSAC, and adopt Urbanloco datasets and Extracted urban dataset with many dynamic objects for testing. The results are shown in Table V. The removal rate of Removert dramatically decreases in these dynamic scenes due to the visibility issues. ML-RANSAC adopts a single matching point, which can easily cause dynamic object omission. Our-Front uses multiple points matching to quickly remove dynamic objects and can achieve comparable performance to that of ERASOR that is not limited by the points existing in the invalid range of visibility. Note that Removert and ERASOR are offline in which a prebuilt map is needed. However, Our-Front can remove dynamic points online.

Then, to further clarify the impact of dynamic objects on front odometry, we compare Our-Front with the other odometry methods (ML-RANSAC, A-LOAM, and F-LOAM) and obtain the corresponding trajectory error from Tables II to IV for analysis. It can be seen that Our-Front has better localization accuracy than those of the other odometry methods after dynamic object removal. For example, the traffic congestion Scene 1 and cross road Scene 2 are selected from our Extracted dataset for the detailed explanation. There are a large number of dynamic vehicles and pedestrian, as shown in Figs. 12(a) and 13(a). Due to the fact that A-LOAM and F-LOAM are disturbed by dynamic objects, there are a large number of ghost tracks in the local map, as shown in Figs. 12(d) and (e) and 13(d) and (e), and the localization has also a large deviation, as shown in Table IV. Specifically, it can be seen that the boundary overlap effects are poor from regions A and B in Fig. 12(d) and (e). Although ML-RANSAC has reduced the ghost tracks by removing dynamic objects, single matching point easily causes dynamic object omission in complex dynamic scenes. Our-Front uses a more robust FA-RANSAC algorithm to remove dynamic objects to get an accurate static map, as shown in Figs. 12(b) and 13(b). Therefore, the trajectory error of Our-Front is relatively small and the boundary overlap effects obtained by Our-Front are better as shown in regions A and B in Fig. 12(b). In summary, Our-Front can improve the SLAM performance in dynamic environments by removing dynamic objects.

2) Impact of Improved ScanContext: In this study, the effectiveness of feature matrix similarity is first evaluated. For this purpose, the improved ScanContext method is referred to as IMSC, and we remove the column norm simplification step from IMSC as RM-IMSC. Then, we count the number of the loop closure frames obtained by RM-IMSC and IMSC and calculate the relative loop rate. The results are shown in Table VI. It can be seen that, after simplifying feature matrix, we can achieve loop closure detection performance comparable to that of RM-IMSC. Therefore, when calculating the similarity score of the feature matrix, column norms are used instead of column vectors, and this simplification method does not result in the loss of column vector information.

To further clarify the overall performance of our improved ScanContext method, we combine Our-Front with optimized-SC as OF-OPSC and add ICP to Our-Front as OF-ICP (Our-Front only provides poses and raw point clouds are used by optimized-SC and ICP for loop closure detection and mapping). Next, our method is tested on our Extracted dataset compared with these methods, and the trajectory error is shown in Table VII. It can be seen that our improved ScanContext outperforms optimized-SC and ICP in dynamic scenes. Then, the typical scenes are selected from our

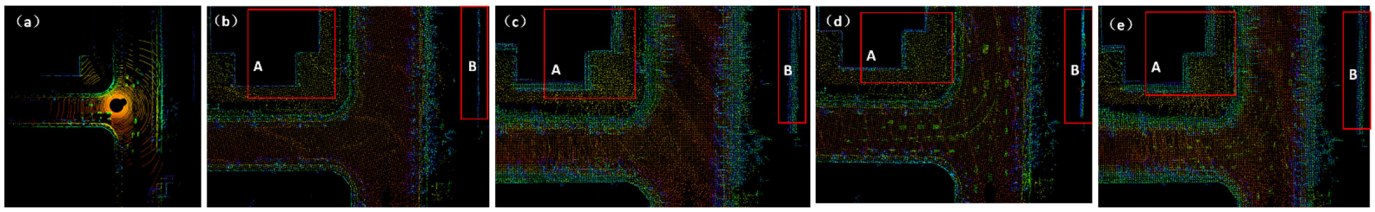


Fig. 12. Comparison of the mapping results in Scene 1. (a) Raw data. (b) Local mapping by Our-Front. (c) Local mapping by ML-RANSAC. (d) Local mapping by F-LOAM. (e) Local mapping by A-LOAM.

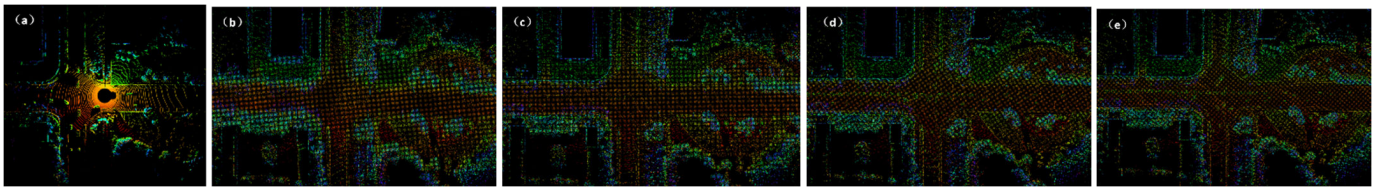


Fig. 13. Comparison of the mapping results in Scene 2. (a) Raw data. (b) Local mapping by Our-Front. (c) Local mapping by ML-RANSAC. (d) Local mapping by F-LOAM. (e) Local mapping by A-LOAM.

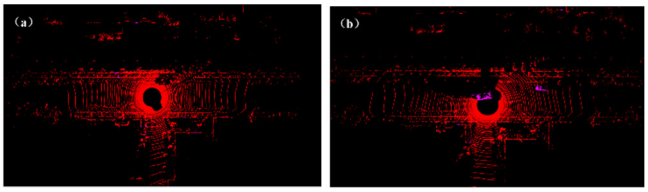


Fig. 14. Leaving and returned point cloud in Scene 3. (a) Leaving scene. (b) Returned scene.

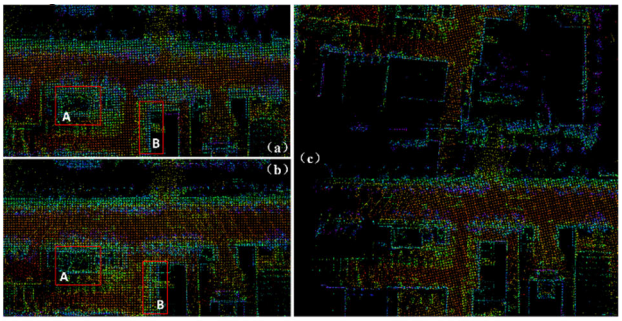


Fig. 15. Comparison of the mapping results in Scene 3. (a) Local mapping by RF-LOAM. (b) Local mapping by OF-OPSC. (c) Local mapping by OF-ICP.

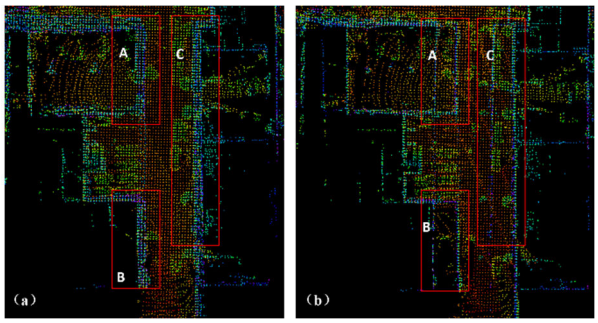


Fig. 16. Comparison of the mapping results in Scene 4. (a) Local mapping by RF-LOAM. (b) Local mapping by OF-OPSC.

Extracted dataset for detailed analysis, such as loop closure road Scene 3, bumpy road Scene 4, and similar road Scene 5.

First, for loop closure road Scene 3, the point clouds during the vehicle leaves and returns Scene 3 are shown in Fig. 14. The static points are represented by red, and the dynamic points detected by our method are represented by purple. Due to the fact that OF-OPSC and OF-ICP use raw point clouds for loop closure detection, there are many ghost tracks in the local map, as shown in Fig. 15(b) and (c). Next, due to the significant differences in loop closure features of the leaving and returned frames affected by dynamic objects, OF-OPSC is difficult to detect loop closure. Therefore, the trajectory error is relatively large, as shown in Table VII, and the local map overlap effects is poor, as shown in regions A and B in Fig. 15(b). Then, due to the fact that OF-ICP performs loop closure detection based on radius, it is difficult to accurately detect loop closure when the odometry error is large. Therefore, the trajectory error is very large in Table VII, and the leaving and returned frames cannot match well, as shown in Fig. 15(c). Our method uses FA-RANSAC to remove dynamic objects, so there are only static points in the local map as shown in Fig. 15(a), and loop closure features of the leaving and returned frames are similar to accurately achieve loop closure detection. Therefore, after global optimization, the trajectory error obtained by our method is smallest in Table VII, and the local map overlap effects is better, as shown in regions A and B in Fig. 15(a).

After that, for bumpy road Scene 4, there is a relatively steep slope and deceleration zone on the road. When the vehicle turns quickly and passes through Scene 4, the pose change of the LiDAR coordinate will cause the height of point cloud in the same scene to change accordingly. At this time, OF-OPSC uses maximum height as features, making it difficult to accurately detect loop closure. In addition, due to pedestrian interference, the trajectory error is relatively large, as shown in Table VII, and the local map overlap effects are poor as shown in regions A–C in Fig. 16(b). However, our method improves the ScanContext algorithm by using height difference as features to solve the problem of height changes caused by bumpy road and can accurately detect loop closure.

TABLE VIII
RUNTIMES ON THE ALL DATASETS IN DIFFERENT ALGORITHMS

	Datasets	A-LOAM	F-LOAM	LeGO-LOAM (with ICP)	Optimized-SC-F-LOAM	PWCLO-Net	ML-RANSAC	Ours
Point Processing (ms)	KITTI	21.6	**3.5**	6.8	**3.5**	10.1	3.6	15.6
	Urbanloco	16.4	2.8	5.4	**2.6**	7.4	2.9	14.1
	Extracted	22.4	3.9	7.6	**3.8**	10.3	4.1	21.6
Odometry Mapping (ms)	KITTI	122.9	62.1	53.8	51.7	52.6	58.4	**47.4**
	Urbanloco	98.3	50.9	46.2	43.8	44.5	48.9	**32.2**
	Extracted	125.2	63.4	56.8	53.4	54.4	61.2	**40.3**
Loop Global Optimization (ms)	KITTI	/	/	163.3	79.3	/	/	**60.2**
	Urbanloco	/	/	158.6	75.6	/	/	**58.1**
	Extracted	/	/	212.3	94.2	/	/	**72.6**

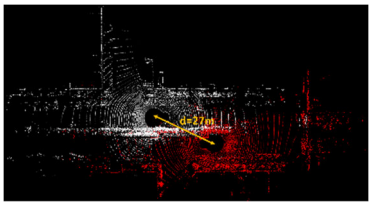


Fig. 17. Raw data of similar roads Scene 5.

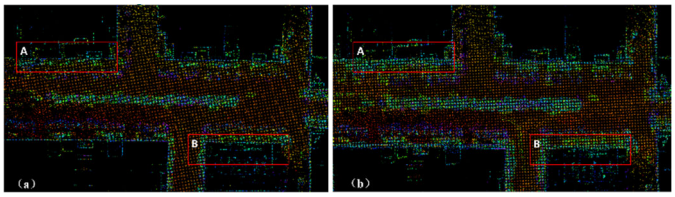


Fig. 18. Comparison of the mapping results in Scene 5. (a) Local mapping by RF-LOAM. (b) Local mapping by OF-OPSC.

Therefore, the trajectory error of our method can be well corrected by global optimization as shown in Table VII, and the local map overlap effects are better as shown in regions A–C in Fig. 16(a).

Finally, for similar roads in Scene 5, there are two frame point clouds that are far away but have similar scene structures as shown in red and white points in Fig. 17. Therefore, the feature matrixes of two frame point clouds are similar. At this time, OF-OPSC cannot filter out the false loop frame that is far away using adaptive distance threshold that increases with the number of keyframes. Also, considering that the two frame scenes are similar and only a small part of the point clouds coincide, it is difficult to obtain accurate loop closure constraint for global optimization. Thus, the trajectory error is large as shown in Table VII, and the local map overlap effects are poor as shown in regions A and B in Fig. 18(b). Our method adaptively increases the distance threshold from the minimum value after each global optimization, which makes the distance threshold and trajectory error change synchronously. Thus, our method can accurately remove false loop closure detection of similar scenes that are far away and avoid global optimization errors. Therefore, the trajectory error is smaller as shown in Table VII, and the local map overlap effects are better as shown in regions A and B in Fig. 18(a).

D. Runtime

In practical applications, real-time performance is a crucial indicator to evaluate SLAM systems. We test the runtime of different methods on all datasets, and the results are shown in Table VIII. It can be seen that Our-Front is the fastest among these odometry methods without loop closure detection, and our method is also the fastest among these SLAM methods with loop closure detection. Then, Our-Front can quickly obtain odometry pose through initial pose provided by FA-RANSAC. Therefore, the runtime of Our-Front is less than that of F-LOAM with one-time match, end-to-end PWCLO-Net, and lightweight ML-RANSAC. A-LOAM takes the longer time because the iteration number of mapping is large. LeGO-LOAM and optimized-SC-F-LOAM take more time than A-LOAM because they both have loop closure detection. In our method, when the height difference of fan-shaped grids is used for object segmentation, these grids are also used for loop closure feature extraction that can reduce runtime. Then, the search efficiency is improved by deleting historical matching frames and the matching computation is simplified by improving feature matching. Therefore, although our method adds the steps of segmentation and registration that increases runtime for point cloud processing, our method reduces the runtime by 12% compared with Optimized-SC-F-LOAM, as shown in Table VIII.

V. CONCLUSION

In this article, the RF-LOAM LiDAR SLAM algorithm is proposed by adding lightweight segmentation, registration, and improved ScanContext algorithm based on F-LOAM. The innovations of our method are mainly reflected in the following three points.

1) Skipping traditional process of object detection and tracking, a new FA-RANSAC algorithm is proposed based on object segmentation to quickly and accurately remove dynamic objects as outliers.
2) Based on the ScanContext algorithm, the height difference and adaptive distance threshold are used to improve the loop closure detection accuracy, and the loop closure detection efficiency is improved by deleting the loop closure historical matching frames and simplifying the feature matching.
3) The FA-RANSAC and the improved ScanContext algorithm are unified into the F-LOAM framework with

the minimum cost, and the LiDAR SLAM is implemented quickly and accurately in the urban dynamic environment.

Under KITTI datasets, Urbanloco datasets, and Extracted urban dataset, our method is validated. The experimental results show that our method can effectively overcome the interference of dynamic objects and accurately realize loop closure detection. The average trajectory error of our method is smaller than those in other compared algorithms in all datasets. Considering that there are more dynamic objects in Urbanloco datasets and Extracted urban dataset, our method has more obvious advantages and stronger robustness in these datasets. For the runtime, our method improves the candidate frame search and feature matching of ScanContext algorithm so that its runtime is shortest among these SLAM methods with loop closure detection, which meets the real-time requirement. Therefore, our method can provide both robust and fast SLAM for UGV and is competitive among state-of-the-art algorithms.

However, there are still some shortcomings in this method. For example, when UGV travels to scenes with single features, the accuracy of odometry and loop closure detection is poor. Next, we will try to introduce the intensity of point clouds or fuse the RGB of images to build map for better SLMA performance.

REFERENCES

[1] C. Debeunne and D. Vivet, "A review of visual-LiDAR fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, pp. 2068–2088, Apr. 2020.

[2] A. Gostar et al., "State transition for statistical SLAM using planar features in 3D point clouds," *Sensors*, vol. 19, no. 7, p. 1614, Apr. 2019.

[3] J. Konecny, M. Prauzek, P. Kromer, and P. Musilek, "Novel point-to-point scan matching algorithm based on cross-correlation," *Mobile Inf. Syst.*, vol. 2016, pp. 1–11, Apr. 2016.

[4] R. W. Wolcott and R. M. Eustice, "Fast LiDAR localization using multiresolution Gaussian mixture maps," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2814–2821.

[5] S. Milz, G. Arbeiter, C. Witt, B. Abdallah, and S. Yogamani, "Visual SLAM for automated driving: Exploring the applications of deep learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 360–36010.

[6] A. Shahmansoori, G. E. Garcia, G. Destino, G. Seco-Granados, and H. Wymeersch, "Position and orientation estimation through millimeter-wave MIMO in 5G systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1822–1835, Mar. 2018.

[7] H. Wymeersch, G. Seco-Granados, G. Destino, D. Dardari, and F. Tufvesson, "5G mmWave positioning for vehicular networks," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 80–86, Dec. 2017.

[8] B. Li, Y. Wang, and Y. Zhang, "GP-SLAM: Laser-based SLAM approach based on regionalized Gaussian process map reconstruction," *Auto. Robots*, vol. 44, no. 6, pp. 80–101, Jun. 2020.

[9] J. S. Levinson, *Automatic Laser Calibration, Mapping, and Localization for Autonomous Vehicles*. Stanford, CA, USA: Stanford Univ., 2011.

[10] M. Aldibaja, N. Suganuma, and K. Yoneda, "Robust intensity-based localization method for autonomous driving on snow-wet road surface," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2369–2378, Oct. 2017.

[11] H. Zhang and J. Peng, "Visual SLAM location methods based on complex scenes: A review," in *Proc. Int. Conf. Artif. Intell. Secur.*, 2020, pp. 487–498.

[12] M. Li and F. Rottensteiner, "Vision-based indoor localization via a visual SLAM approach," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 42, pp. 827–833, Jun. 2019.

[13] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[14] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LiDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1271–1278.

[15] J. Zhang and S. Singh, "Low-drift and real-time LiDAR odometry and mapping," *Auto. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[16] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-LOAM: Fast LiDAR odometry and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 4390–4396.

[17] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[18] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Digit. Imag. Model.*, 2001, pp. 145–152.

[19] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Proc. Robot., Sci. Syst.*, vol. 2, no. 4, p. 435, Jun. 2009.

[20] D. Qiu, S. May, and A. Nuchter, "GPU-accelerated nearest neighbor search for 3D registration," in *Proc. Int. Conf. Comput. Vis. Syst.*, Oct. 2009, pp. 194–203.

[21] 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. (IROS)*, Oct. 2003, pp. 2743–2748.

[22] E. Einhorn and H.-M. Gross, "Generic 2D/3D SLAM with NDT maps for lifelong application," in *Proc. Eur. Conf. Mobile Robots*, Sep. 2013, pp. 240–247.

[23] A. Schabert, "Integrating the use of prior information into Graph-SLAM with NDT registration for loop detection," M.S. thesis, Dept. Elect. Eng., Orebro Univ., Orebro, Sweden, 2017.

[24] 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.

[25] T. Li et al., "P^3-LOAM: PPP/LiDAR loosely coupled SLAM with accurate covariance estimation and robust RAIM in urban canyon environment," *IEEE Sensors J.*, vol. 21, no. 5, pp. 6660–6671, Mar. 2021.

[26] L. Liao, C. Fu, B. Feng, and T. Su, "Optimized SC-F-LOAM: Optimized fast LiDAR odometry and mapping using scan context," in *Proc. 6th CAA Int. Conf. Veh. Control Intell. (CVCI)*, Oct. 2022, pp. 1–6.

[27] M. Velas, M. Spanel, M. Hradis, and A. Herout, "CNN for IMU assisted odometry estimation using velodyne LiDAR," in *Proc. IEEE Int. Conf. Auto. Robot Syst. Competitions (ICARSC)*, Apr. 2018, pp. 71–77.

[28] Z. Li and N. Wang, "DMLO: Deep matching LiDAR odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 6010–6017.

[29] G. Wang, X. Wu, S. Jiang, Z. Liu, and H. Wang, "Efficient 3D deep LiDAR odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5749–5765, May 2023.

[30] W. Kesai, Y. Xifan, and H. Yu, "Review of visual SLAM in dynamic environment," *Robot*, vol. 43, no. 6, pp. 715–732, Dec. 2021.

[31] A. Schlichting and C. Brenner, "Vehicle localization by LiDAR point correlation improved by change detection," *ISPRS-Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 41, pp. 703–710, Jun. 2016.

[32] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robot. Res.*, vol. 26, no. 9, pp. 889–916, Sep. 2007.

[33] G. Kim and A. Kim, "Remove, then revert: Static point cloud map construction using multiresolution range images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10758–10765.

[34] H. Lim, S. Hwang, and H. Myung, "ERASOR: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3D point cloud map building," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2272–2279, Apr. 2021.

[35] S.-W. Yang and C.-C. Wang, "Multiple-model RANSAC for ego-motion estimation in highly dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3531–3538.

[36] R. Wang et al., "A robust registration method for autonomous driving pose estimation in urban dynamic environment using LiDAR," *Electronics*, vol. 8, no. 1, p. 43, Jan. 2019.

[37] B. Zou, K. Liu, and K. Wang, "Dynamic obstacle detection and tracking method based on 3D LiDAR," *Automobile Technol.*, vol. 4, no. 8, pp. 19–25, 2017.

[38] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–10, Aug. 2008.

[39] 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.

[40] 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, Sep. 2013.
[41] W. Wen et al., "UrbanLoco: A full sensor suite dataset for mapping and localization in urban scenes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2310–2316.
[42] J. Li, Y. Zhang, X. Liu, X. Zhang, and R. Bai, "Obstacle detection and tracking algorithm based on multi-LiDAR fusion in urban environment," *IET Intell. Transp. Syst.*, vol. 15, no. 11, pp. 1372–1387, Aug. 2021.
[43] C. Qian, Z. Xiang, Z. Wu, and H. Sun, "RF-LIO: Removal-first tightly-coupled LiDAR inertial odometry in high dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 4421–4428.

Jiong Li received the master's degree in mechanical engineering from the Army Military Transportation Academy, Tianjin, China, in 2019.

He is currently an Assistant Engineer with the 95848 Army of PLA, Xiaogan, China. His research interests are LiDAR SLAM and environment perception of unmanned systems.

Xudong Zhang (Member, IEEE) received the Ph.D. degree in mechanical engineering from the Technical University of Berlin, Berlin, Germany, in 2017.

He is currently an Associate Professor with the Beijing Institute of Technology. His main research interests include unmanned system localization, environment perception, and vehicle dynamics control.

Yu Zhang is currently a Teaching Assistant with the Department of Vehicle Engineering, Army Academy of Armored Forces, Beijing, China. His main research interests are unmanned system motion planning and localization.

Yunfei Chang received the master's degree in software engineering from Beihang University, Beijing, China, in 2018.

He is currently working at Beijing North Vehicle Group Company Ltd., Beijing, engaged in vehicle design. His main research interests are unmanned system motion planning and localization.

Kai Zhao received the master's degree in mechanical engineering from the Army Military Transportation Academy, Tianjin, China, in 2018.

He is currently an Assistant Engineer with the Institute of Military Transportation, Army Military Transportation Academy. His main research interests are localization and mapping using multisensor fusion for intelligent vehicle.